# Inteligencia Artificial y Big Data en Salud.
# M4. Adquisición, filtrado y seguridad de datos

Joan Bartrina, Bernat Gastón, Ramon Martí
Joan Oliver, Marta Prim, Mercè Villanueva

Curso de Especialización
Institut Universitari Parc Taulí - UAB

# Contents

# Chapter 1

# Introducción a la adquisición de datos

*Joan Oliver*

La adquisición de datos trata de la toma de señales del mundo físico real y de su conversión en valores numéricos digitales que pueden ser manipulados por un computador u otro sistemas digital. Para ello, los sistemas de adquisición de datos realizan la conversión de magnitudes físicas del mundo real en valores digitales (datos) para su almacenamiento, procesado y monitorización.

Esta unidad trata de las señales que se obtienen del mundo físico real y de su procesado. Se realizará un breve repaso a la representación de datos en el computador. Se introducirán los componentes y las arquitecturas básicas utilizadas en la conversión de las señales en datos digitales.

## 1.1 Introducción

### 1.1.1 Dispositivo de adquisición de datos

Un dispositivo de adquisición de datos es un sistema electrónico que captura un conjunto de señales físicas (del mundo real), las convierte en tensiones eléctricas y las digitaliza, de forma que puedan ser procesadas en un ordenador.

La composición de un dispositivo de adquisición de datos es:

- Sensores. Dispositivos que transforman una señal de una magnitud física determinada capturada a una señal electrónica, continuas en el tiempo y amplitud, con salida en voltaje o corriente. Es lo que se llama normalmente señal analógica.

- Elementos acondicionamiento de señal. Adecuan la señal *cruda* o *raw* que sale del sensor a una señal sin ruido y con rango suficiente para poder ser convertida a señal digital con la resolución que se requiere. Suele constar de las dos etapas:

  - Filtraje. De la señal de salida del sensor sólo se dejan pasar aquellas frecuencias que resultan de interés a la salida del sensor. También se elimina ruido que puede introducirse en la etapa de medida.

  - Amplificación. La señal se adapta al rango de voltaje o corriente para que el conversor analógico/digital trabaje a la máxima resolución.

- Conversor analógico/digital o A/D. La señal analógica resultante de la etapa de amplificación es convertida a señal digital o dato utilizando conversores analógicos/digitales.

- Procesador. Es el encargado del procesado de la señal y del almacenaje.

La figura 1.1 esquematiza los elementos de que consta un dispositivo de adquisición de datos.


Figure 1.1: Elementos de que se compone un dispositivo de adquisición de datos.

### 1.1.2 Sistemas de adquisición de datos basados en procesador.

Los sistemas de adquisición de datos son elementos genéricos basados en procesador que realizan toda la tarea de capturar magnitudes físicas del mundo real y convertirlas a datos. Incorporan como elemento de proceso a un microcontrolador o a un procesador, en caso de sistemas más grandes.

Los sistemas de adquisición de datos, según sus prestaciones, se suelen clasificar como tarjetas de adquisición de datos o como sistemas integrados.

Las tarjetas de adquisición de datos son sistemas genéricos creados para usarse en diversas aplicaciones. Son dispositivos que contienen un conjunto de entradas y salidas que están preparados para la adquisición de señales en general provenientes de sensores diversos. Aunque sus prestaciones suelen ser inferiores a las de los sistemas de adquisición de datos a medida, son suficientes para muchas aplicaciones en la adquisición de señales. Por ello, generalmente, presentan un coste menor al de un sistema a medida.

Éstas tarjetas se conectan directamente al bus del ordenador. Son las encargadas de realizar las conversiones de señales desde analógicas a digitales (mediante el uso de dispositivos ADC) así como de establecer las comunicaciones con el ordenador. La figura 1.2 muestra un ejemplo de este tipo de tarjetas.


Figure 1.2: Tarjeta de adquisición de datos

Las características principales de las tarjetas de adquisición de datos son:

- Número de canales analógicos. Para la conectividad con los sensores las tarjetas disponen de un número determinado de canales, o puertos, analógicos. Cada canal está dedicado a la señal de un sensor.

- Número de canales digitales. Algunas tarjetas también disponen de canales de entrada para sensores tienen salida ya digital.

- Resolución. Es la cantidad mínima de entrada discernible por el canal. Suele ser uno de los parámetros críticos en muchas aplicaciones. Depende del número de bits del conversor analógico/digital que se utiliza para representar cada muestra. A mayor número de bits la tarjeta podrá detectar cantidades más pequeñas en la señal.

- Capacidad de temporización. Algunas tarjetas tienen salidas temporizadas para la sincronización de las medidas.

- Forma de comunicarse con el ordenador. Existen múltiples protocolos de comunicación usados en los ordenadores y con los que las tarjetas pueden comunicarse. Cuando las tarjetas son externas (no conectadas a ningún bus interno del ordenador) protocolos usados son el paralelo y el serie (como protocolos más antiguos) y el USB (más usado actualmente). Algunas tarjetas de instrumentación también incorporan el bus de instrumentación específico GPIB.

- Velocidad de muestreo: Cuando la señal analógica llega al canal se debe discretizar en el tiempo, lo que se llama muestrear. La velocidad de muestreo indica cuán rápidamente pueden tomarse las muestras. Cuanto mayor sea la frecuencia de muestreo mejor será la representación que se obtendrá de la señal analógica. Como se introducirá, la frecuencia de muestreo debe ser mayor al doble de la frecuencia de la señal a muestrear (teorema de Shannon-Nyquist)

- Rango de entrada. Es el rango de valores de la señal de entrada permitidos para no saturar el canal de entrada de la tarjeta.

Los sistemas de adquisición de datos basados en sensores pueden ser creados a medida, fabricados específicamente para la conexión con sensores determinados. Aunque también pueden pueden ser sistemas integrados a partir de tarjetas de adquisición de datos.

Como ejemplo, la plataforma de adquisición de datos *MySignals* (1.3), desarrollada por *Libelium*, es un sistema completo de *eHealth* que permite la medida y monitorización de distintos parámetros biométricos del cuerpo (presión, pulso, glucosa, pulso, oxígeno en sangre, ECG, etc.) con tan sólo cambiar el sensor correspondiente con conexión IoT para el guardado de datos en la nube.

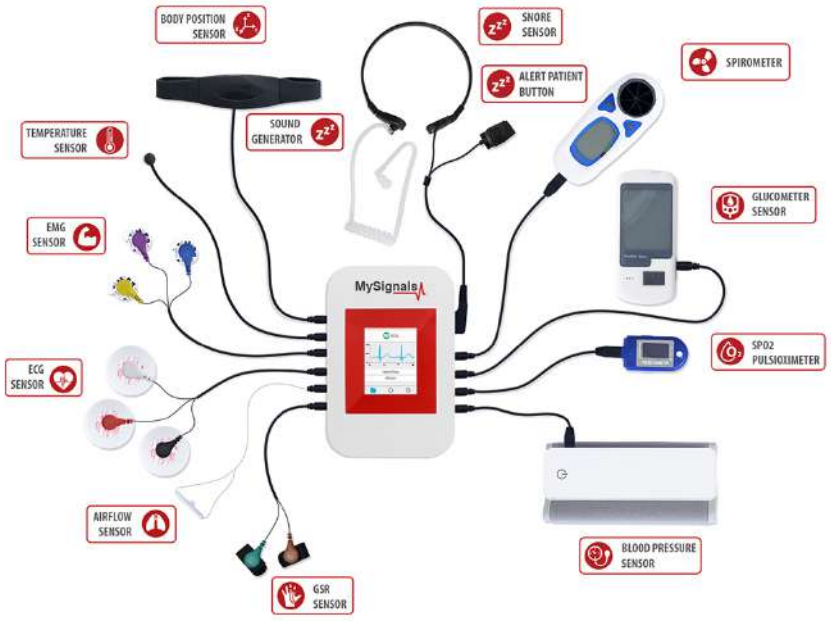



Figure 1.3: Plataforma *MySignals* (*Libelium*) de *eHealth*.

### 1.1.3 Arquitectura de un sistema de adquisición computarizado de datos

Un sistema de adquisición de datos es responsable de la conversión de magnitudes físicas de entrada a señales electrónicas que se procesan y/o almacenan en el computador. En sistemas autónomos, el sistema de adquisición también puede ser responsable del control de mecanismos externos o actuadores. La figura 1.4 muestra el proceso completo que se lleva a cabo. En este proceso se contemplan las siguientes acciones:

- Transducción de señales. El **transductor** es el dispositivo responsable de convertir magnitudes físicas a otras magnitudes físicas. Cuando la magnitud resultante es electrónica se emplea el término **sensor**. De hecho, es el término comúnmente usado puesto que todo el tratamiento posterior es mediante circuitos electrónicos.

- Acondicionamiento de la señal. La señal electrónica del sensor a menudo debe ser acondicionada para obtener un rango de entrada suficiente para el conversor analógico/digital que cumpla con los requerimientos de precisión, resolución y con el rango de frecuencias esperado. Para ello suelen ser necesarios los procesos de calibración, filtraje y amplificación de la señal.

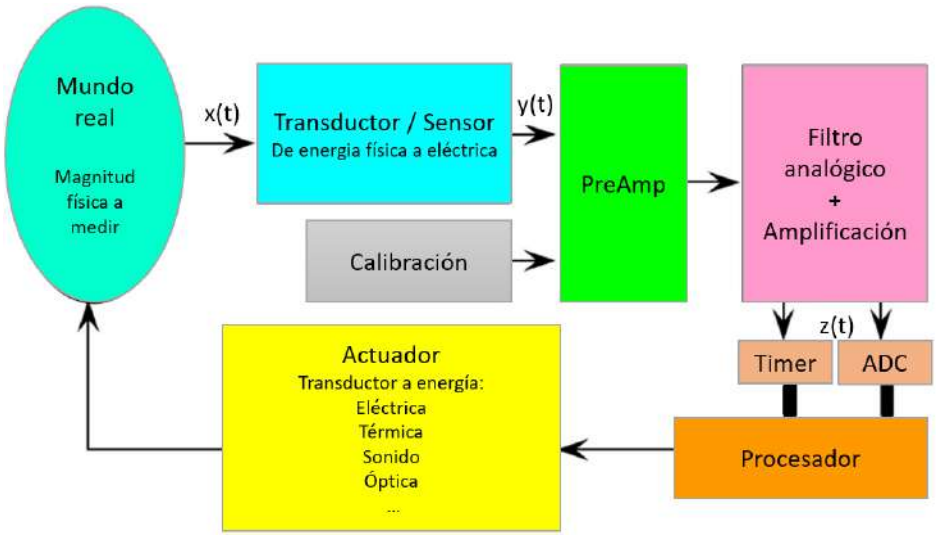



Figure 1.4: Acciones en un sistema de adquisición de datos y de control.

- El conversor analógico/digital o ADC es el responsable de convertir la señal electrónica analógica en digital. Es decir, se discretiza la señal.

- La señal está lista para ser procesada y/o almacenada en el computador.
  También la señal se tiene que presentar de forma comprensible al operador, o se debe elaborar para que pueda ser interpretada por un sistema supervisado.
  Tradicionalmente la señal se presentaba analógicamente en una aguja o en registradores de papel, Actualmente los terminales gráficos suelen ser el método más utilizado.

- En sistemas de control, el procesador puede enviar órdenes a mecanismos externos para que actúen de acuerdo a los datos recibidos. Estos mecanismos externos reciben el nombre de actuadores.
  Es en este paso que se cierran todas las acciones necesarias para la adquisición de datos, procesado y control en actuadores en sistemas autónomos.

La figura 1.2 muestra que en un sistema de procesamiento a menudo no sólo se llevan a cabo acciones de almacenaje de datos sino también de control. Para estos casos se ha acuñado el término de *Big-Loop*, en el sentido que un sistema de adquisición de datos puede llevar todas las acciones necesarias para el auto-control de sistemas, desde la adquisición de datos hasta el control.

## 1.2 Representación de datos en el computador.

### 1.2.1 Introducción.

En el mundo de los sensores, los datos son fundamentalmente señales analógicas. Los circuitos de lectura electrónicos transforman las señales en datos que se pueden almacenar en la computadora.

Los datos que se guardan y procesan en la computadora son de índoles muy diferentes. Y para cada tipo de dato se establece un mecanismo concreto de codificación.

Basándose en la tipología de los datos, se puede establecer una primera clasificación de acuerdo al mecanismo que posteriormente se utiliza para ser procesados:

- Texto. Son los símbolos como letras, números, caracteres especiales, etc. Por lo general, el texto se guarda en la computadora enumerando, o codificando, cada símbolo de forma única.

- Números. Pueden introducirse como texto o caracteres. Sin embargo, si se tienen que realizar operaciones matemáticas con ellos se codifican de manera que cada símbolo tiene un valor acorde a su peso y posición que ocupa.

- Sonido. El sonido es una onda que se propaga. El sonido es capturado por un micrófono que realiza la transducción de onda sonora (onda de presión) a señal electrónica. Posteriormente, esta señal se digitaliza y procesa en formato binario.

- Imágenes. Las imágenes se componen de muchos valores que requieren de un uso intensivo de la computadora. Las imágenes pueden ser guardadas en formato geométrico (de poco peso) o con todo el detalle de una foto (con mucho peso). Las imágenes pueden ser estáticas o dinámicas (vídeo). Para reducir el tamaño que ocupa una imagen se utilizan técnicas de compresión de imágenes.

### 1.2.2 Texto.

Los caracteres son representaciones de datos que pueden tener distinto significado: alfabéticos, numéricos, especiales, gráficos, caracteres de control, etc.

Desde el comienzo de los ordenadores se han sucedido distintas codificaciones con el propósito de poder representar a todos los caracteres posibles. De entre estas codificaciones hay dos que son las más utilizadas actualmente:

- La codificación **ASCII** o ***American Standard Code for Information Interchange***. Utiliza una codificación de 8 bits, lo que permite representar hasta 256 caracteres. Los primeros 128 códigos son comunes a todas las variantes del código ASCII y se utilizan para representar a todas las letras mayúsculas y minúsculas, números y algunos signos de puntuación. Con las 128 codificaciones restantes se representan caracteres personalizadas de regiones. Por ejemplo, el estándar ASCII personalizado *ISO 8859-Latin-1* (tabla 1.1) es la norma que recoge los símbolos del área latina.

| ISO-8859-1 | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | x0 | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | xA | xB | xC | xD | xE | xF |
| 0x | NUL | SOH | STX | ETX | EOT | ENQ | ACK | BEL | BS | TAB | LF | VT | FF | CR | SO | SI |
| 1x | DLE | DC1 | DC2 | DC3 | DC4 | NAK | SYN | ETB | CAN | EM | SUB | ESC | FS | GS | RS | US |
| 2x | SP | ! | " | # | $ | % | & | ' | ( | ) | * | + | , | - | . | / |
| 3x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 4x | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5x | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ^ | _ |
| 6x | ` | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7x | p | q | r | s | t | u | v | w | x | y | z | { | \| | } | ~ | DEL |
| 8x | PAD | HOP | BPH | NBH | IND | NEL | SSA | ESA | HTS | HTJ | VTS | PLD | PLU | RI | SS2 | SS3 |
| 9x | DCS | PU1 | PU2 | STS | CCH | MW | SPA | EPA | SOS | SGCI | SCI | CSI | ST | OSC | PM | APC |
| Ax | NBSP | ¡ | ¢ | £ | ¤ | ¥ | ¦ | § | ¨ | © | ª | « | ¬ | SHY | ® | ¯ |
| Bx | ° | ± | ² | ³ | ´ | µ | ¶ | · | ¸ | ¹ | º | » | ¼ | ½ | ¾ | ¿ |
| Cx | À | Á | Â | Ã | Ä | Å | Æ | Ç | È | É | Ê | Ë | Ì | Í | Î | Ï |
| Dx | Ð | Ñ | Ò | Ó | Ô | Õ | Ö | × | Ø | Ù | Ú | Û | Ü | Ý | Þ | ß |
| Ex | à | á | â | ã | ä | å | æ | ç | è | é | ê | ë | ì | í | î | ï |
| Fx | ð | ñ | ò | ó | ô | õ | ö | ÷ | ø | ù | ú | û | ü | ý | þ | ÿ |

Tabla 1.1: Estándar ISO-8859-1 de caracteres ASCII (Latin-1)

En la tabla cada carácter se le asocia un código binario de 8 bits (o dos caracteres hexadecimales). Por ejemplo, la letra $A$ se encuentra en la celda $0x41$ ($0x$ significa código hexadecimal, el 4 viene de la fila $4x$, y el 1 de la columna $x1$). $0x41$ corresponde al código binario $b0100\_0001$.

- Con el auge de los ordenadores surgió la norma **UNICODE** con el objetivo de poder representar a todos los caracteres de forma global.

Con una codificación de 16 bits por símbolo, es el conjunto de caracteres más completo, y se ha convertido en la opción elegida por entornos plurilingües. Contiene símbolos (matemáticos, lógicos, musicales, ...), de ornamentación, i sistemas de escritura de la antigüedad, como runas, o diacríticos del griego clásico.

### 1.2.3 Números.

Los números se pueden representar en el ordenador de varias formas: como texto, como números, codificados, ...

#### Representación como texto

Cuando el número viene descrito como texto suele tomar la codificación de código ASCII. Por ejemplo, el número 2021, descrito como texto utilizando el código ASCII (tabla 1.1) se almacenaría en el ordenador como $0x32\_30\_32\_31$.

#### Representación como número

La representación de números en la computadora como números sigue las reglas posicionales de la notación numérica.

**Símbolos.**

Para representar los números suele usarse cualquiera de los tres conjuntos de símbolos, o bases, siguientes:

- Base binaria: $B_2 = \{0, 1\}$
- Base decimal: $B_{10} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$
- Base hexadecimal: $B_{16} = B_{10} + \{A, B, C, D, E, F\}$

Puede observarse que la base binaria representa la codificación lógica en el ordenador.

Sin embargo, para los humanos es difícil trabajar con ella porque cada número se representa con un conjunto ingente de dígitos (bits) que difícilmente recordamos.

Por ello, lo normal es utilizar la representación hexadecimal, en la que cada cuatro dígitos binarios se convierten en un dígito hexadecimal. La representación del número es más corta (menos dígitos) y más fácil de recordar.

La base decimal es un subconjunto de la base hexadecimal. Normalmente no se utiliza internamente en el ordenador ya que no es fácil operar directamente con ella.

**Notación posicional.**

La norma que se utiliza en la representación es la posicional, que es la utilizada en el sistema decimal. Esto es, el valor de un número es función del símbolo que representa y de la posición que ocupa. Por ejemplo:

$$1101_{(2} = D_{(16} = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 13_{(10}$$

El número de bits necesarios para representar un número depende de la representación elegida. Por ejemplo, el valor del número 3567,

> ...representado como texto (código ASCII) es 0x33, 0x35, 0x36, 0x37. Tomado un byte para la representación de cada dígito) requerirá 32 bits.
>
> ...tomado como un número, solo necesita 12 bits: $3567_{(10} = 1101\_1110\_1111_{(2}$

El mayor número que se puede representar depende del número de bits utilizados. Por ejemplo, utilizando números enteros en formato de 32 bits, los números van desde $-2^{-31} - 1$ hasta $2^{31} - 1$ tomándolos como signo y magnitud.

### 1.2.4 Codificación del sonido.

Las aplicaciones multimedia procesan texto, imagen y sonido. El sonido es una onda de presión que se transmite en un medio y es capturada por un micrófono. El micrófono proporciona una señal analógica continua a lo largo del tiempo, como se muestra en la figura 1.5.

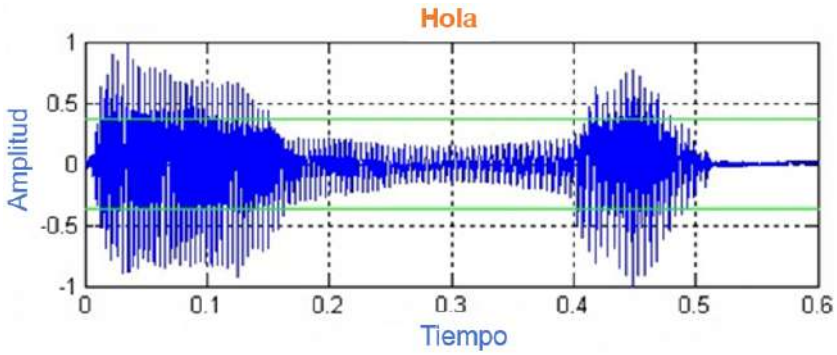



Figure 1.5: Registro de la palabra *hola*.

Si se amplía la señal anterior se puede observar que es una señal con forma sinusoidal. Para poder procesarla, Alec Reeves, en 1937, introdujo la **modulación de código de pulso o PCM** (figura 1.6) para transformar una señal analógica en una secuencia de bits (señal digital).

Básicamente, los pasos que se siguen para digitalizar la señal son:

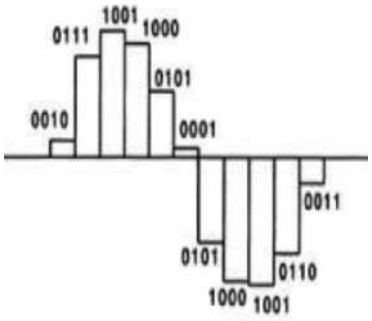



Figure 1.6: Codificación PCM de una señal.

- El sonido se muestrea a una frecuencia específica $f_s$, tomando una muestra de señal cada determinado tiempo $T_s = 1/f_s$. En la figura 1.5, si el sonido dura 0.6 segundos y se muestrea a 23.225 KHz, se registran un total de 13935 puntos.
- Los puntos se codifican analógicamente (figura 1.5). Un ADC convierte cada punto en un valor binario. Si se utiliza una codificación binaria de un byte/muestra, un segundo de sonido grabado requiere una capacidad de memoria de $22.225kB$.

  *Nota: Estos conceptos se detallarán en los próximos apartados.*

La calidad de la codificación PCM se basa en dos parámetros:

- La frecuencia de muestreo, responsable de la calidad.
- El número de bits por código, que determina la precisión.

Actualmente existen diferentes sistemas de digitalización de sonido, de acuerdo con la tabla 1.2. Su uso depende de la calidad que se necesite en la digitalización de la señal. Además, se utilizan diversos **CODEC** (codificadores/decodificadores) para reducir el tamaño de los archivos de sonido, como son WAV, DPCM, MPEG, MP3, ...

### 1.2.5 Representación de imágenes en el computador.

La representación de imágenes en el ordenador tiene dos métodos de codificación fundamentales:

- Formato vectorial.
- Formato bitmap.

La necesidad de usar archivos en formato vectorial o bitmap dependerá de las necesidades del proyecto.

| | Bits/muestra | Fs (kMuestras) | Ts ($\mu s$) |
|---|---|---|---|
| Telefonía - PCM | 8 | 8 | 125 |
| Telefonía de calidad | 8 | 11.025 | 90.7 |
| Radio | 8 | 22.05 | 45.4 |
| CD (por canal) | 16 | 44.1 | 22.7 |
| Audio profesional (R-DAT) | 20 | 96 | 10.4 |
| DVD-Audio (x4) | 24 | 192 | 5.2 |

Tabla 1.2: Codificaciones en sonido

### Formato vectorial.

Una imagen en formato **vectorial** es una imagen formada por objetos geométricos como segmentos, polígonos, arcos, etc. Cada uno de éstos objetos se guarda mediante los atributos matemáticos de forma, de posición, etc. del objeto.

Por ejemplo el círculo de la figura 1.7 queda definido por la posición de su centro, su radio, el grosor de línea y su color. Cada uno de los símbolos geométricos de la figura se define de forma similar.

Figure 1.7: Imagen representada en formato vectorial.

Las ventajas principales del formato vectorial son:

- La definición de la imagen está en la precisión del display, puesto que, matemáticamente se puede aproximar totalmente la precisión de la forma en cualquiera de sus puntos.
- La simplicidad de guardar imágenes y el poco espacio que ocupan en memoria.

Bajo el formato vectorial existen distintos estándares de empresas para guardar las imágenes. Cabe destacar los formatos DXF (*Drawing Exchange Format*, usado por AutoCAD), EPS (*Encapsulated Postcrip*), AI (Adobe Illustrator), etc.

**Formato bitmap.**

Se utiliza para almacenar imágenes digitales en forma de mapa de bits, siendo independientemente del dispositivo de visualización. Es capaz de almacenar imágenes digitales bidimensionales tanto monocromáticas como en color, en varias profundidades de color y, opcionalmente, con compresión de datos, canales alfa y perfiles de color.

El formato bitmap divide la imagen en una matriz de unidades denominadas **píxel**. Cada píxel almacena información sobre el color que contiene.

Por tanto, los parámetros que definen una imagen en bitmap son:

- La profundidad del color. Viene definido por la paleta que se utiliza. Por ejemplo:
  - Una imagen en blanco y negro sólo necesita 1 bit para codificar el color. 1 el color blanco, y 0 el negro.
  - Cuando se codifica en niveles de gris, la resolución suele ser de 1 byte. Entonces se pueden codificar 256 niveles de gris.
  - Para el color existen múltiples paletas. Por ejemplo, la paleta RGB (*Red*, *Green*, *Blue*) codifica utilizando 3 bytes, un byte para cada color. El rango de colores que puede entonces representarse es de alrededor de $2^{24}$ colores (algunos colores pueden tener codificación repetida).
  - Existen otras paletas de colores, como la CMYK, YUV, etc. La elección de la paleta de color a usar depende de la aplicación.
- La resolución de la imagen, o número de píxeles que contiene. Se mide en número de píxeles horizontales por número de píxeles verticales. Por ejemplo, una imagen capturada con una cámara de fotos de 6000x4000 píxeles, el tamaño de la imagen es de 24 Mpíxeles.

La figura 1.8 muestra una imagen y sucesivas amplificaciones de detalles de la misma. A la izquierda hay un detalle de una fotografía. La imagen del centro es una amplificación de la parte terminal del número 2. En ella ya se ve detalle del píxel. Se puede comprobar que la imagen tiene 16x16 píxeles. La imagen de la derecha se adentra más en esta amplificación y muestra un detalle de 4x4 píxeles.

Puede observarse que al amplificar la imagen se va perdiendo precisión, a diferencia de lo que pasa con las imágenes vectoriales.

Al igual que con los formatos de imagen vectoriales, también existen múltiples estándares de formatos para guardar imágenes en formato bitmap. Sólo por enumerar algunos de ellos:

Figure 1.8: Formato Bitmap.

- **BMP** (*Bitmapped File Format*). Formato gráfico bitmap desarrollado con la aparición de Windows 3 adaptado a este sistema operativo. Tiene resoluciones de 1, 4, 8 y 24 bits. Guarda las imágenes descomprimidas, lo que incremetna la velocidad de carga y el espacio requerido. Con la información de la imagen se guarda información sobre el tamaño, el número de colores y la paleta de colores utilizadas.

- **GIF** (*Graphic Interchange Format*). Creado por Compuserve (1987) es uno de los formatos más utilizados. Utiliza una paleta de entre 2 y 256 colores y tienen una rutina de compresión que reduce eficazmente el tamaño de los archivos a costa de poca demora en la carga. Por ello es muy usado en Internet.

- **TIFF** (*Tagged Image File Format*). Creado por Aldus Corporation es un formato de archivo popular en la industria que permite guardar imágenes raster utilizando un algoritmo de compresión no destructiva. Con la imagen original también se guardan miniaturas de la imagen, denominadas *thumbnails* que permiten previsualizar rápidamente la imagen. Debido al algoritmo de compresión no destructivo los ficheros de imágenes TIFF suelen ser de tamaño grande.

- **JPG** (JPEG, *Joint Photographic Experts Group*) Formato muy utilizado por la capacidad de compresión a costa de poder conservar alta calidad en la imagen. Trabaja con una paleta de 16 millones de colores **(true color)**. Puede realizar compresiones muy grandes de la imagen (superiores a 1:20), pero entonces la calidad de la imagen disminuye. Por ello se descarta su uso en aplicaciones en las que se desea mantener la calidad de la imagen.

- Algunas cámaras fotográficas también pueden guardar las imágenes en los llamados formatos personalizados RAW. Son formatos que guardan toda la información de la imagen capturada sin compresión. Ello permite poder realizar *a posteriori* un procesado de la imagen con los datos originales de captura de la imagen.

### 1.2.6 Proceso y procesado en la toma de imágenes con cámara digital

El proceso que se realiza en la toma de imágenes por una cámara digital es laborioso. La figura 1.9 muestra, *a grosso modo*, los pasos necesarios.

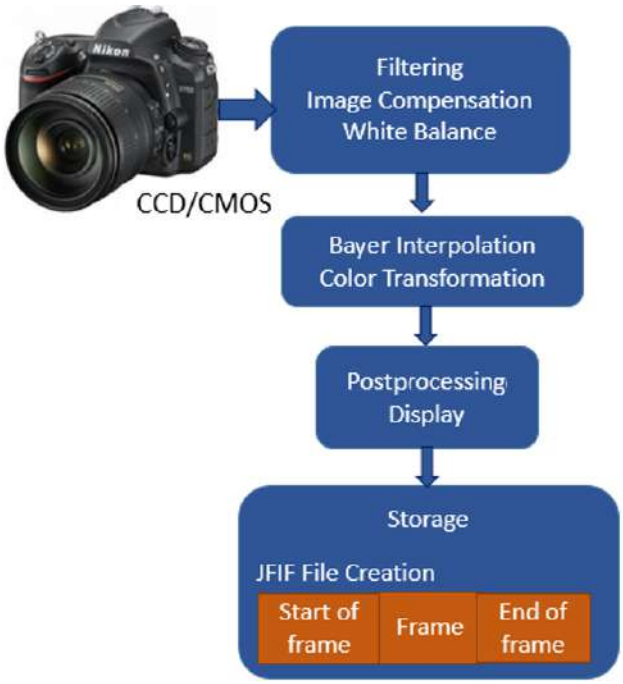



Figure 1.9: Proceso de captura y guarda de imagen.

- La captura de la imagen se produce en el momento de apretar el botón de toma de fotografía. La luz incidente (reflejada del objeto al que disparamos) sensibiliza el sensor de la camera. Atualmente los sensores son de tipo CCD (*Charge Coupled Device*) o CMOS (*Complementary Metal Oxyde Semiconductor*). El sensor realiza la transformación de fotones a electrones.

- Con los parámetros indicados al tomar la foto, la cámara realiza un primer filtrado de los datos y el balance de blancos.

- En la descarga de la imagen se realiza el filtrado del color (interpolación de Bayer) en cada píxel y se realiza una transformación del color de paleta RGB (que es la que utiliza el sensor) a paleta YUV. La paleta YUV aumenta la sensibilidad de la imagen a la luminosidad respecto a los colores de forma similar al tratamiento de la luz que se realiza

en nuestros ojos, además de ser la paleta de color utilizada en la compresión a JPEG.

- Seguidamente se realiza el procesado y compresión de la imagen acorde con el formato de salida especificado para la imagen. Por ejemplo, si la imagen se guarda en formato JPEG, debe realizarse todo el procesado, compresión y codificaciones que se requieren para guardar la imagen en JPEG.
- Se realiza una miniaturización de la imagen para mostrarla por el display.
- Finalmente se guarda la imagen. La figura 1.10 muestra la información que se guarda en caso que el formato de salida sea JPG:

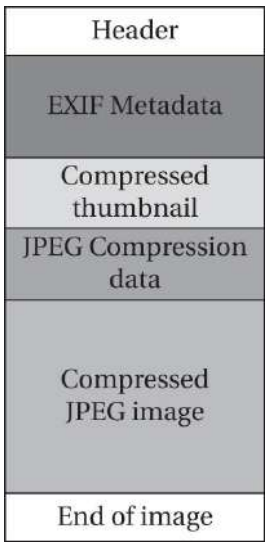



Figure 1.10: Información en un fichero de imagen JPG.

  - Se introduce un encabezado de información del protocolo de guarda.
  - Se introducen *metadatos* de acuerdo con la especificación EXIF. Esto es, entre la información guardada está del autor de la foto, el día y hora, la cámara, objetivo y parámetros de la toma de la foto.
  - Se guarda una copia de baja resolución de la imagen para poder visualizarla de forma rápida.
  - Se introduce la imagen en formato JPEG.
  - Finalmente se introduce información de final de fichero.

### 1.2.7 Protocolo DICOM

El protocolo **DICOM** (*Digital Imaging and Communication in Medicine*) es un estándar internacional para manipular, almacenar y transmitir los datos de imágenes médicas. Se especifica el formato que debe utilizarse en el archivo DICOM y los protocolos de comunicación que se utilizan para intercambiar información en la red.

Los ficheros DICOM tienen una cabecera con campos estandarizados y de forma libre, y un cuerpo con datos de la imagen. Un objeto DICOM simple puede contener sólo una imagen, pero esta imagen puede tener múltiples fotogramas, lo que permite el almacenamiento de bloques de cine o cualquier otro tipo de datos con varios fotogramas. Los datos de la imagen pueden estar comprimidos utilizando una gran variedad de estándares, entre los que se incluyen: JPEG, JPEG sin perdida, LZW y RLE (*Run-lenght Encoding*).

DICOM permite integrar y comunicar (figura 1.11) distintos aparatos médicos como radiógrafos, escáneres, estaciones de trabajo, etc. de distintos proveedores.

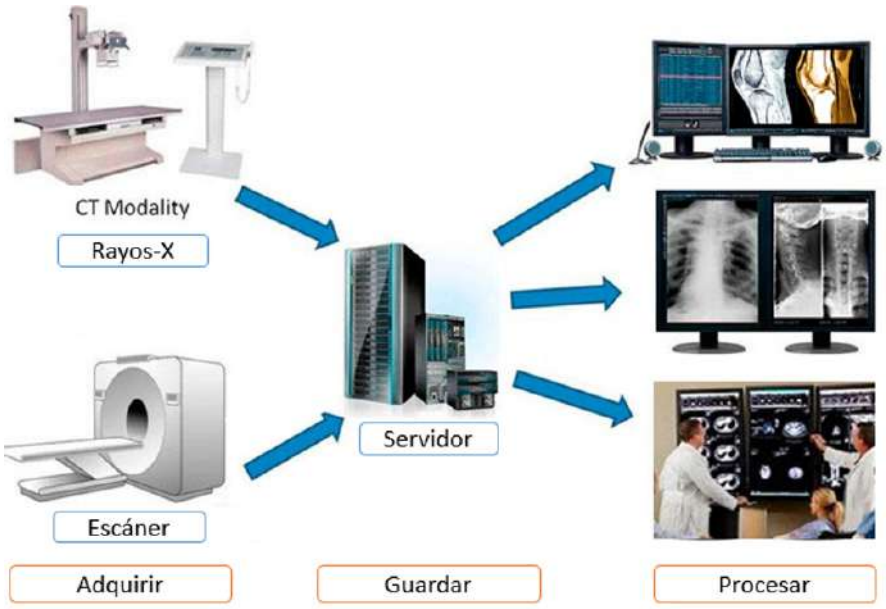



Figure 1.11: Protocolo DICOM.

A través de un visor DICOM se puede visualizar información relacionada con el paciente: identificador, ubicación geográfica, médico del paciente, información de pruebas de diagnosis del paciente (tomografías, radiografías, ecografías, etc.). Como la base es única y la información está integrada, no puede haber errores de intercambio de información entre pacientes.

Por lo tanto, esta integración ayuda a almacenar la información del paciente, a agilizar todos los procesos de documentación, y a diagnosticar las pruebas inter-médico de los pacientes.

## 1.3 Señales y procesado de señales

### 1.3.1 Señales y sistemas.

#### Señales y procesos

Toda magnitud física que varía en el tiempo es transformada por el sensor en una señal electrónica, que también varía en el tiempo.

Toda señal electrónica que varía en el tiempo es, o bien señal (si lleva información que nos interesa), o bien ruido (si la información no nos interesa).

A nivel físico, una **señal** se define como una función de una o más variables que transporta información sobre un cierto fenómeno físico.

Un sistema es una entidad capaz de manipular señales para extraer de ellas cierta cantidad de información. Un sistema (figura 1.12) debe ser capaz de:

- Adquirir datos
- Procesar estos datos
- Y realizar acciones en respuesta a este proceso.


Figure 1.12: Adquisición de datos, proceso y actuación.

La figura 1.13 muestra un ejemplo directo de proceso de datos de un electrocardiograma (ECG). La señal original es un conjunto de señales provenientes de diferentes electrodos. Estas señales eléctricas son enviadas al procesador que realiza las derivaciones de las conexiones entre cada par de electrodos. La salida puede a su vez estar compuesta por un conjunto de señales.

Como se introducirá en el siguiente capítulo, las señales pueden provenir de múltiples tipos de sensores (figura 1.14). De forma común a todas ellas, el sensor transforma las señales físicas en señales eléctricas. A su vez, el proceso


Figure 1.13: Sistema de registro, visualización y extracción en un electrocardiograma.

es el encargado de modificar estas señales para enviarlas a un transductor de salida que transforma la señal eléctrica procesada en otra magnitud física que actúa en el medio físico. En un termómetro digital, por ejemplo, el sensor térmico (termistor, termopar o RTD, normalmente) transforma una señal térmica en eléctrica. Una vez realizado el procesado, la señal eléctrica se envía a un display para la visualización. El display transforma la señal eléctrica en radiante.


Figure 1.14: Proceso de transducción en un termómetro digital.

### Tipos de señales.

Las señales pueden ser deterministas o aleatorias.

Las señales deterministas:

- Se pueden aproximar por una función matemática. Por tanto, se permite la realización de operaciones entre ellas.

- Pueden ser a su vez periódicas, cuando su valor se repite cada cierto tiempo T (periodo), o no periódicas.

- Las señales pueden ser analógicas (continuas) o discretas. Son analógicas cuando pueden tomar cualquier valor dentro de un rango. Son discretas si sólo pueden tomar un conjunto finito de valores posibles.
- Pueden medirse continuamente en el tiempo o en momentos determinados. Al proceso de discretizar la señal se denomina muestreo.

Las señales aleatorias no forman parte del proceso de medida y siempre suelen estar presentes en las mediciones. Normalmente se trata de lo que se denomina ruido que se introduce en cualquier proceso de medida.

### Señal analógica.

Una señal analógica es continua en el tiempo, como, por ejemplo, el registro de la toma de presión arterial sistólica de la figura 1.15.

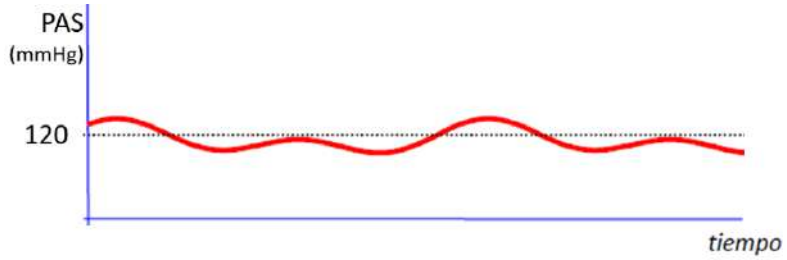



Figure 1.15: Señal continua.

### Señal discreta en el tiempo.

Una señal discreta sólo toma valores en tiempos determinados. Sería el caso de una toma de temperatura casa dos horas, como muestra la figura 1.16.

### Señal discreta en amplitud.

La gráfica 1.17 muestra una toma de frecuencia cardíaca. Como se observa sólo toma valores discretos en amplitud.

### Representación de señales analógicas.

En el registro del electrocardiograma de la figura 1.13, en la gráfica de salida, se observan un conjunto de señales de forma diversa. Desde el punto de vista electrónico, a este tipo de señal continua en tiempo y en valor la denominaremos señal analógica. Es el tipo de señal que tratan los dispositivos

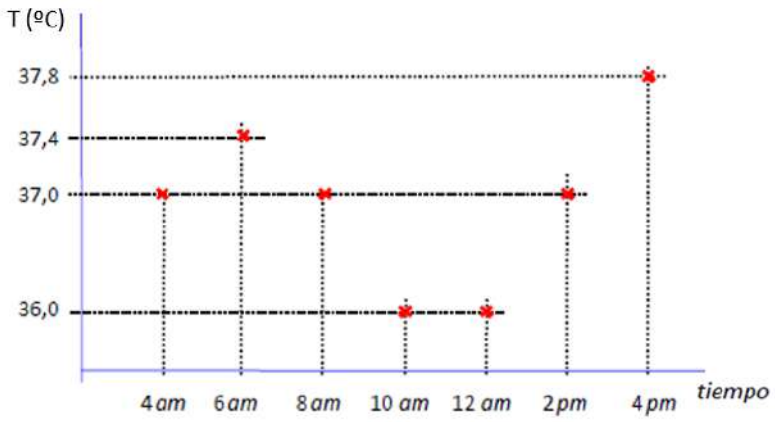



Figure 1.16: Señal discreta.

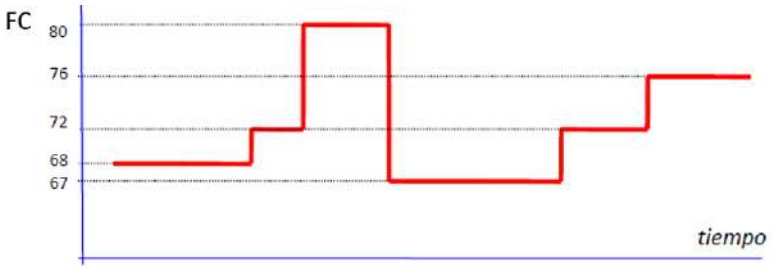



Figure 1.17: Señal discreta, en amplitud.

electrónicos para obtener, por proceso, los datos. Matemáticamente, se parte de la función sinusoidal (1.18) como función base para la generación de todo este tipo de ondas.

Las características principales de una función sinusoidal son:

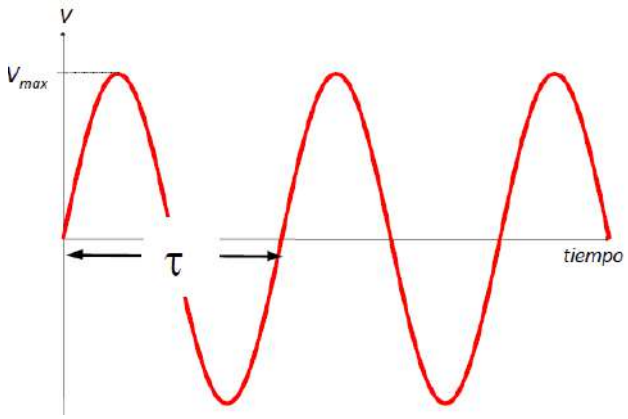



Figure 1.18: Función sinusoidal.

- La forma de onda se repite de forma periódica con un periodo $\tau$.
- O, lo que es lo mismo, la señal sinusoidal se repite con frecuencia f = $1/\tau$.
- La amplitud de la señal es $V_{max}$.
- La función sinusoidal también puede tener un desfase inicial $\varphi$ que se llama fase. En este caso, como no existe, vale 0.

De forma matemática, esta función sinusoidal se describe mediante la ecuación (1.1)

$$V_t = V_{max} \sin(2\pi f t + \varphi) = V_{max} \sin(\omega t + \varphi) \tag{1.1}$$

Donde $V_t$ representa el valor de la función a lo largo del tiempo.

Normalmente la función seno se expresa en función de la frecuencia angular o pulsación $\omega$

$$\omega = 2\pi f \tag{1.2}$$

Como se introduce más adelante, la función seno es la base de construcción de toda forma de onda.

Cuando una señal entra en un proceso, como en la figura 1.19, se observa que el proceso transforma la señal entrante en una nueva forma de onda analógica. Aunque las señales son distintas, ambas señales pueden tratarse considerándolas combinaciones de formas de onda seno con distinta amplitud y frecuencia. La relación entre ambas señales se establece a partir de la *función de transferencia* del proceso.


Figure 1.19: La función de transferencia del proceso establece la relación entre las señales entrante y saliente.

### 1.3.2 Dualidad tiempo-frecuencia.

Todas la señales analógicas pueden analizarse desde el punto de vista frecuencial. De acuerdo con la ecuación (1.1), se observa que la frecuencia determina la rapidez con la que la señal sinusoidal se repite. Utilizando las *series de Fourier* (Apéndice A), toda señal analógica puede expresarse como combinación de ondas sinusoidales. Además, mediante el *análisis de Fourier* también se puede realizar el proceso inverso, que es encontrar las frecuencias de las señales sinusoidales que componen la señal.

Pongamos un ejemplo. La ecuación (1.3) muestra una onda compuesta por dos señales sinusoidales puras de frecuencias 5 Hz y 10 Hz, cuya representación gráfica se muestra en la figura 1.20. La gráfica superior muestra las dos sinusoidales puras que, sumadas, forman la onda de la gráfica inferior. Las frecuencias que componen una señal se les denomina **armónicos**.

$$y1 = 2sin(5 \cdot 2\pi t) + 3cos(10 \cdot 2\pi t) \tag{1.3}$$

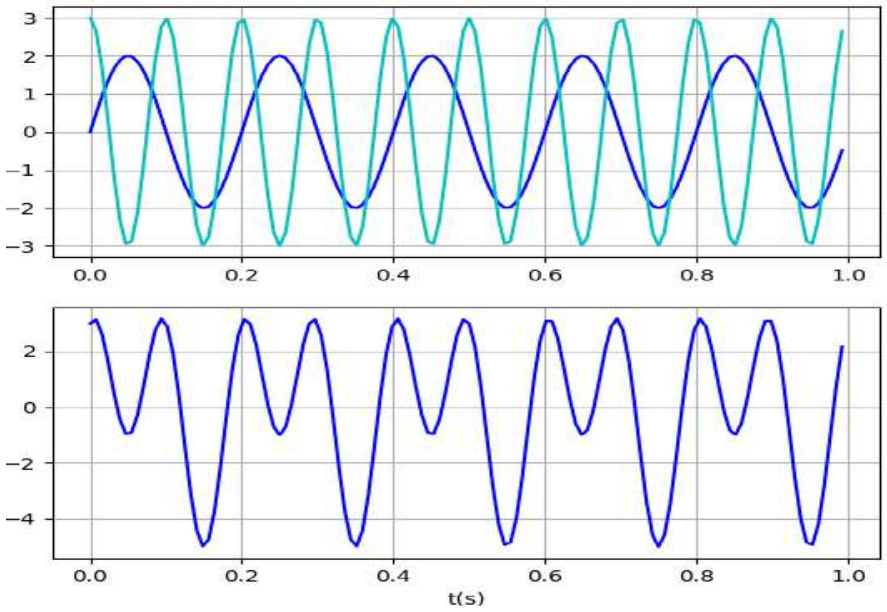



Figure 1.20: Representación de la onda sinusoidal dela ecuación (1.3), compuesta por dos armónicos de frecuencias 5 Hz y 10 Hz.

Esta misma ecuación puede también representarse en el dominio frecuencial. El resultado es la gráfica de la figura 1.21. En este caso, el eje de las abscisas representa a la frecuencia. Puede observarse que la gráfica consiste en dos picos justamente en las frecuencias 5 Hz y 10 Hz, que son las frecuencias de las señales sinusoidales que forman la onda. Y la amplitud de estos

picos también se corresponde proporcionalmente a la amplitud relativa entre las dos ondas.

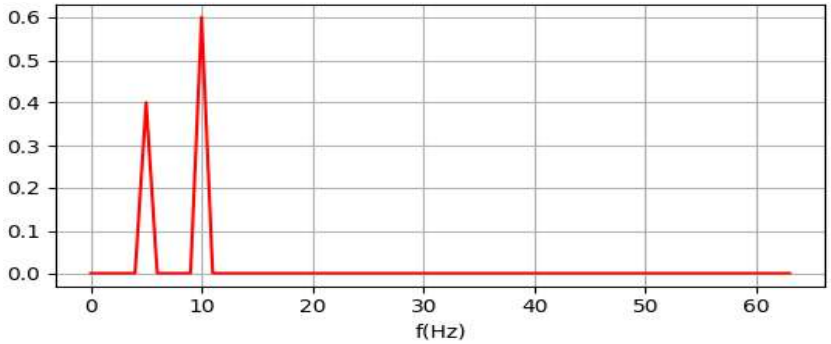



Figure 1.21: Espectro frecuencial de la señal de la ecuación (1.3).

El estudio frecuencial puede realizarse sobre cualquier señal. Así, por ejemplo, la figura 1.22 muestra la señal obtenida de un registro bioeléctrico.

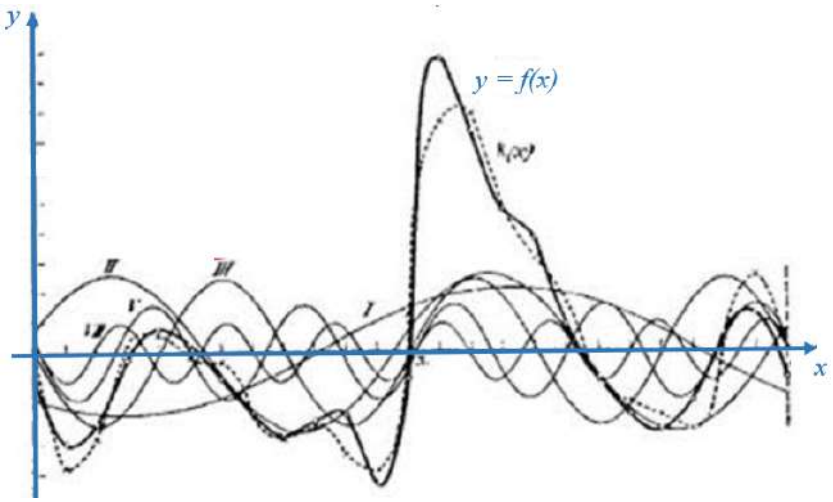



Figure 1.22: Registro bioeléctrico.

La aplicación del análisis de Fourier permite obtener el conjunto de armónicos que forman la señal (figura 1.23).

La figura 1.24 muestra de forma muy gráfica como el dominio temporal se complementa con el dominio frecuencial. Existe, así, una dualidad tiempo-frecuencia en las señales:

- El dominio temporal muestra la evolución de una señal en el tiempo.

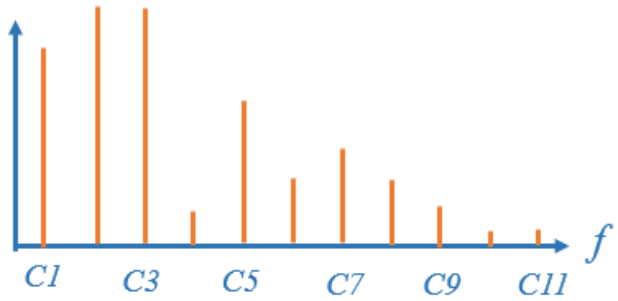



Figure 1.23: Espectro del registro bioeléctrico de la figura 1.22.

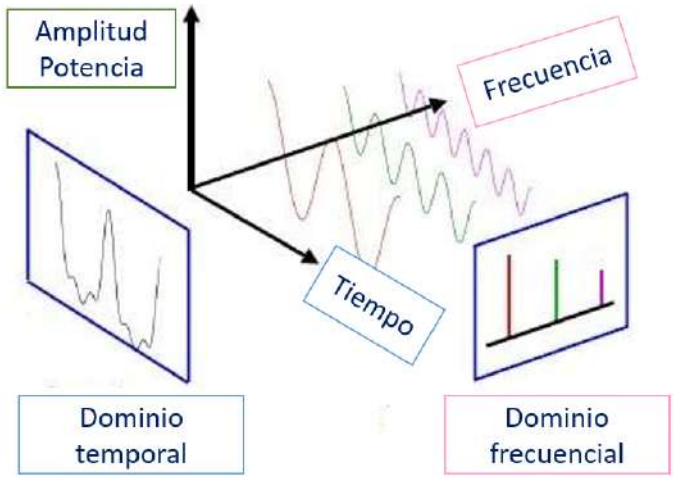



Figure 1.24: Dualidad tiempo-frecuencia.

- El dominio frecuencial muestra las componentes de la señal de acuerdo con la frecuencia con que oscilan dentro de un rango determinado.
- Por brevedad no se introduce en este espacio, pero debe puntualizarse que las señales frecuenciales también están sujetas al desplazamiento de fase de la señal que se aplicará cuando se quiera volver a recuperar la señal original.
- Fourier muestra cómo realizar la conversión entre dominios

El análisis frecuencial de señales y datos es fundamental en múltiples áreas del conocimiento, de entre las que se encuentra la biomedicina. Sólo por citar algunas aplicaciones, se puede encontrar en:

- Aplicación al análisis de ECG, EMG, EECs y, en general, todo tipo de señales continuas bioeléctricas.

- En el tratamiento digital de imágenes el análisis de Fourier y sus variantes (como la transformada discreta del coseno) tienen múltiples aplicaciones.

En una señal bioeléctrica, el análisis frecuencial da información sobre las frecuencias o armónicos que componen dicha señal. Evidentemente, cada señal tiene su propio espectro y, por ende, tendrá señales dentro de un rango acotado de frecuencias. A este rango de frecuencias se le da el nombre de **ancho de banda**. La tabla 1.3 muestra algunos de los anchos de banda típicos que se encuentran en el registro de biopotenciales eléctricos.

| Señal | Ancho de banda (Hz) | Amplitudes típicas (mV) | Comentarios |
|---|---|---|---|
| ECG | 0.05–100 | 1 | |
| EMG | 20–500 | 1 | Medido en músculo |
| | 300–3000 | 1 | Según músculo y técnica de registro |
| EEG | 0.5–50 | 0.2 | Medido en cuero cabelludo |

Tabla 1.3: Anchos de banda de señales bioeléctricas: electrocardiograma (ECG), electromiograma (EMG), electroencefalograma (EEC)

El análisis frecuencial es una herramienta indispensable en el tratamiento de señales y datos, pero al mismo tiempo un tema complejo a nivel matemático. Aunque para este curso es suficiente con entender que las señales se pueden mostrar tanto en el dominio temporal como en el frecuencial, se ha extendido su explicación en el Apéndice A para profundizar un poco más en el tema.

### 1.3.3 Muestreo. Teorema de Shannon-Nyquist.

Las señales que provienen de los sensores son señales continuas. Y las señales continuas no pueden ser guardadas en el ordenador como tales por la cantidad de espacio que requerirían.

Es por ello que la señal continua del sensor se adquieren de una forma discreta en el tiempo. Un temporizador, usualmente denominado *reloj*, se encarga de adquirir a espacios de tiempo muy precisos los datos. A este proceso se le denomina **muestreo**. Por ejemplo, de la señal continua en el tiempo de la figura 1.25 se toman datos, o *muestrea*, en momentos regularmente espaciados en el tiempo.

El muestreo debe realizarse de forma que, una vez adquirida la señal, ésta se pueda recomponer más adelante. Por ello, al muestrear una señal debe considerarse:

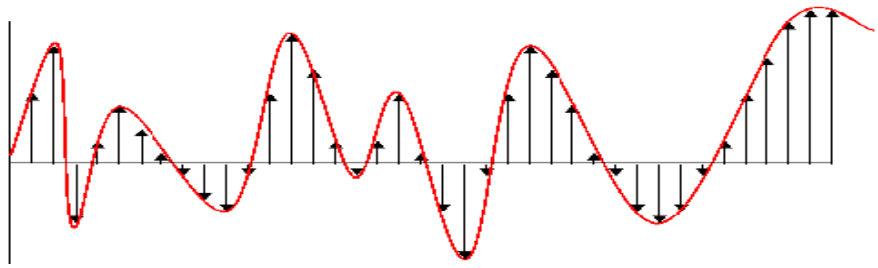


Figure 1.25: Una señal muestrea toma muestras de la señal a tiempos igualmente espaciados.

- El tiempo durante el cual debe adquirirse y muestrear una señal. Depende totalmente del tipo de señal. En señales aleatorias, la señal recompuesta a partir de un muestreo será diferente para cada muestreo debido a que la composición frecuencial será distinta. Sin embargo, en señales que de antemano se sabe que son periódicas, de realizarse bien el muestreo, con sólo un período de adquisición de datos, es suficiente para recomponer la señal.

- El espaciado entre muestras que permita recomponer la señal. La frecuencia con que se deben adquirir las muestreas viene determinada por el **criterio de Shannon-Nyquist**, normalmente conocido como **teorema del muestreo de Nyquist**.

El **teorema del muestreo de Nyquist** establece que:

*La frecuencia de muestreo debe ser, como mínimo, dos veces la frecuencia máxima de la señal que se quiera recomponer.*

No cumplir con el teorema de Nyquist, es decir, muestrar una señal a una frecuencia inferior de la máxima frecuencia que se quiera adquirir, puede producir efectos de *aliasing* en la recomposición de la señal.

El **aliasing** es el efecto que causa que señales continuas distintas sean indistinguibles cuando se muestrean. Por ejemplo, en la figura 1.26, una señal de 1 Hz se muestrea a 0.85 Hz, frecuencia de muestreo que no cumple con el teorma de Nyquist. La reconstrucción de la señal generará una señal de 0.16 Hz.

## 1.4 Sistema de adquisición de datos

En las secciones anteriores se han introducido los conceptos básicos de la adquisición de datos. Esta sección repasa los dispositivos electrónicos y

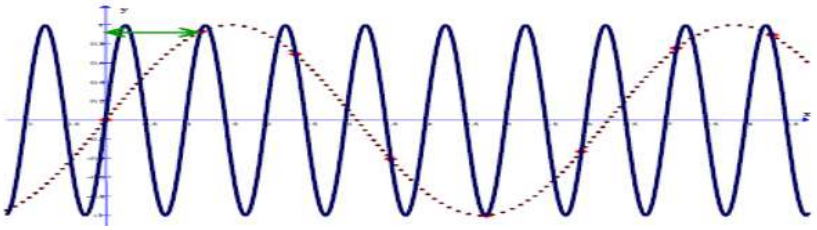


Figure 1.26: Una frecuencia de muestreo demasiado baja da lugar al aliasing.

técnicas más utilizadas en la adquisición y acondicionamiento de los datos. Son los circuitos responsables de realizar el muestreo, la discretización y la digitalización de las señales de salida de los sensores.

En primer lugar se introduce el canal de acondicionamiento en la adquisición de señales con sensores y se presentan, de forma breve, circuitos electrónicos fundamentales que se utilizan en el canal de adquisición. Aunque es un apartado un poco técnico, en términos electrónicos, se debe tener presente que estos circuitos están en la base de todo canal de acondicionamiento de señal de sensores.

### 1.4.1 Introducción.

El diagrama de bloques de la figura 1.27 repasa los elementos básicos que interviene en el registro de señales.


Figure 1.27: Adquisición (registro) de señales.

Recordemos que para adquirir y procesar una señal es necesario:

- Mediante sensores realizar el sensado de magnitudes físicas del entorno. El sensor es el responsable de realizar la transducción de una magnitud física a electrónica, paso necesario para que los sistemas electrónicos puedan procesar la señal.

- La señal eléctrica del sensor es acondicionada y amplificada. En el acondicionamiento suele filtrarse la señal para dejar eliminar ruidos o

señales indeseados y/o pasar las frecuencias de interés. En la amplificación la señal de adecua a los niveles de voltaje con que trabaja el procesador.

- La señal se muestrea y digitaliza. El muestreo discretiza la señal en tiempo. La digitalización discretiza en amplitud y convierte en valores digitales. El conversor analógico/digital o *ADC* es el dispositivo fundamental de esta etapa.

- Ya en el procesador, los algoritmos correspondientes procesan la señal.

- Finalmente la señal se almacena y/o se visualiza.

### 1.4.2 Interfase de adquisición de señales con sensores.

La figura 1.28, detalla la interfase o canal de adquisición típico de un sistema electrónico responsable de la adquisición de señales basado en sensores.

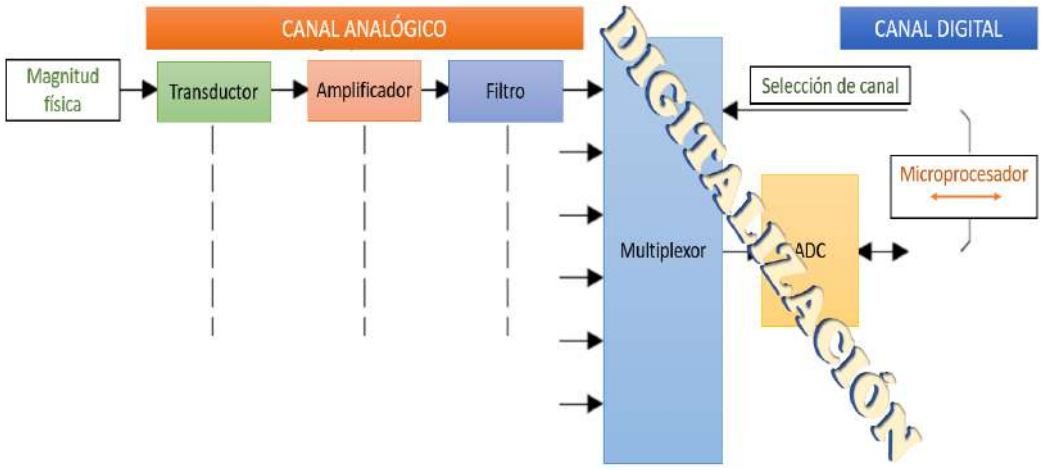



Figure 1.28: Interfase de adquisición de señales con sensores.

Los procesos ejecutados en el canal analógico se encargan de la adaptación de la señal adquirida de manera que llegue en condiciones de máxima resolución y mínimo ruido al ADC. Una vez convertida a señal digital entonces es almacenada y/o procesada en el ordenador.

Consiguientemente, en el acondicionamiento hay tres pasos esenciales:

- Amplificación. Se amplifica el nivel de salida de los sensores de manera que en la digitalización haya suficiente discriminación entre niveles.

- Filtraje. La señal de salida del sensor debe adecuarse a la ventana frecuencial esperada de las señales.

- Conversión A/D. La señal continua en voltaje se se discretiza y es convertida a códigos binarios.

El acondicionamiento de señal es particular dependiendo del tipo de señal de entrada o del sensor, tal como muestra la figura 1.29. Así, el termopar, sensor de temperatura con rango de salida muy pequeño, necesita de amplificación; sensores de temperatura como los termistores requieren linealizar la señal; sensores de presión diferenciales, como las galgas extensiométricas, suelen montarse sobre un puente de Wheatstone para fortalecer su sensibilidad; los electrodos suelen requerir aislamiento de señales para mejorar su inmunidad al ruido. Son sólo unos ejemplos de que cada tipo de sensor necesita de su canal de acondicionamiento.


Figure 1.29: Ejemplos de acondicionamiento en sensores.

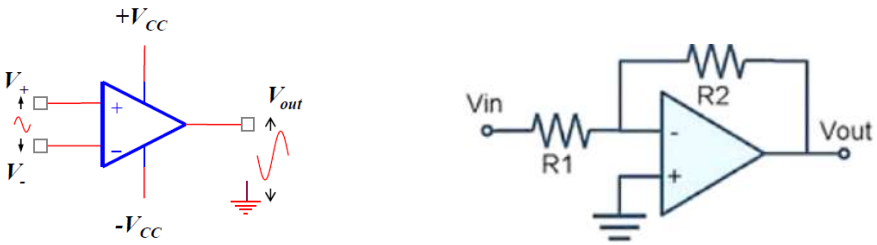
### 1.4.3 Amplificación.

En términos electrónicos un amplificador es un dispositivo con entrada y salida capaz de aumentar la magnitud de la señal de la entrada, que puede llegar como voltaje o corriente.

Aunque la amplificación puede llevarse a cabo utilizando distintos dispositivos electrónicos, seguramente el dispositivo actualmente más empleado para ello es el amplificador operacional.

Un **amplificador operacional** o **OpAmp** (figura 1.30a) es un módulo electrónico, normalmente formado por transistores, que tiene una entrada no inversora (+), una inversora (-) y una salida. La tensión de salida es

la diferencia entre las entradas + y - multiplicada por la ganancia G del dispositivo en lazo abierto, ecuación 1.4.




(a) Símbolo (b) Modelo interno

Figure 1.30: Amplificador operacional.

$$V_{out} = G \cdot (V_+ - V_-) \tag{1.4}$$

La ganancia en tensión se calcula de acuerdo a la ecuación 1.5:

$$\Delta V = \frac{V_{out}}{V_{in}} \tag{1.5}$$

El OpAmp puede funcionar en lazo abierto o lazo cerrado. Se entiende **lazo abierto** cuando no hay realimentación de la salida a la entrada. Es decir, que no hay ninguna conexión mediante dispositivo (resistencia, capacidad, etc.) entre la salida y la entrada, como en la figura 1.30a. En **lazo cerrado**, figura 1.30b, se conecta la salida con la entrada del OpAmp por medio de algún dispositivo electrónico.

Debido a este poder de amplificación que tienen los OpAmp normalmente se utilizan en lazo cerrado, lo que permite controlar la ganancia del amplificador.

Además, como el OpAmp dispone de dos entradas puede funcionar tanto en modo diferencial como con referencia a tierra.

### 1.4.4 Filtros.

En el apartado 3.2 se ha introducido que las señales se pueden descomponer en sumas de señales frecuenciales simples. Toda señal que procede de sensores puede ser vista como sumas de señales frecuenciales. Es más, el mismo proceso que se lleva a cabo durante el proceso de adquisición de señales puede incorporar señales adicionales con frecuencias espúreas. Sea como sea, es frecuente que la señal que se desea procesar esté compuesta por señales de

frecuencias deseadas y otras que no interesan en el proceso. El proceso de filtraje se encarga de dejar pasar sólo las frecuencias deseadas.

Los elementos que se encargan del filtraje de señales se denominan **filtros**. Los filtros *dejan pasar* o *bloquean* el paso de señales dependiendo de su frecuencia.

Según el rango de frecuencias que dejan pasar o bloquean los filtros se clasifican en:

- Filtros **pasa-alto**: Dejan pasar las frecuencias altas.
- Filtros **pasa-bajo**: Dejan pasar las frecuencias bajas
- Filtros **pasa-banda**: Dejan pasar un intervalo concreto de frecuencias
- Filtros **trampa** o ***notch***: Suprimen un intervalo concreto de frecuencias

La figura 1.31 muestra el diagrama de Bode de estos filtros. Es fácil distinguir el comportamiento de cada uno de ellos.

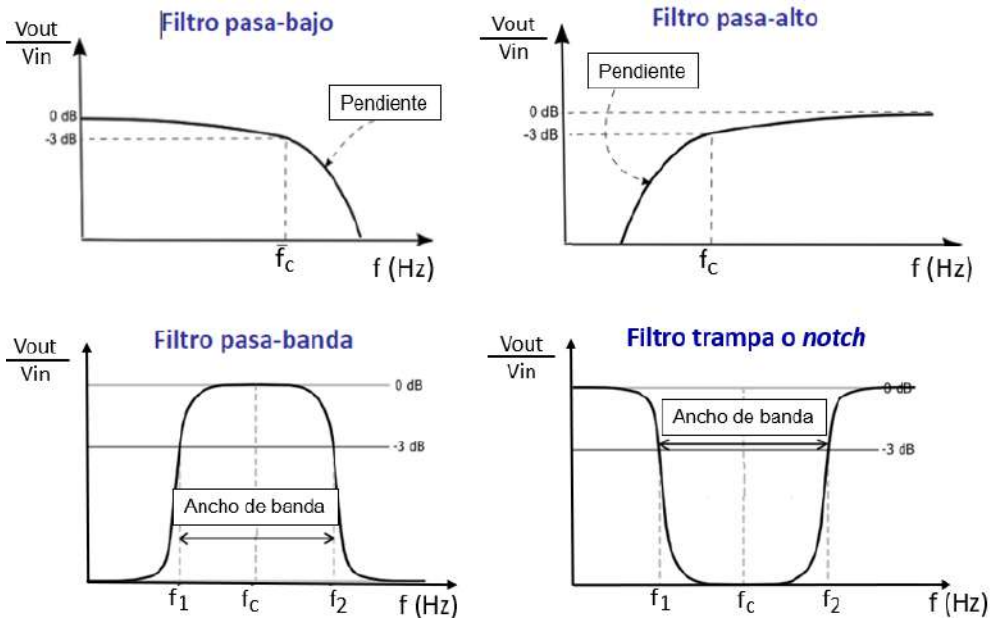



Figure 1.31: Tipos de filtros según su comportamiento.

No cabe decir que el uso de filtros es indispensable en el tratamiento de señales procedentes de sensores para dejar pasar sólo las frecuencias de interés. Por citar un ejemplo, en un electrocardiograma se registran señales electrocardiográficas y electromiográficas. Las frecuencias registradas de

señales EMG suelen estar en un rango superior a los 500 Hz (figura 1.32), mientras que en señales ECG estarían por debajo de los 100 Hz. Filtros pasa-bajos ayudan a filtrar las señales ECG.

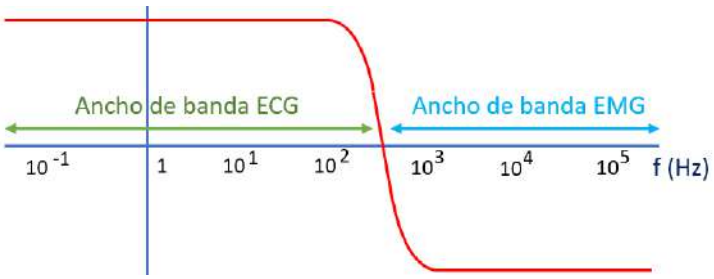



Figure 1.32: Clasificación de señales ECG i EMG.

### 1.4.5 Conversión A/D y D/A.

En el acondicionamiento la señal se ha filtrado, se ha reducido el ruido y se ha adecuado su rango para maximizar la resolución en la conversión analógica/digital o A/D. La conversión A/D realiza el proceso de digitalización de la señal. El **conversor analógico/digital o ADC** es el dispositivo encargado de ello.

El ADC va siempre acompañado del **conversor digital/analógico o DAC**, responsable de la conversión digital/analógica o D/A, proceso inverso que transforma una señal digitalizada en continua.

En la conversión A/D existe pérdida de información. Ello es consecuencia de la digitalización de la señal analógica, que transforma una señal continua en el tiempo y en potencia a un conjunto de valores muestreados y discretos en potencia. Por consiguiente, se va a producir un **error de cuantificación** en la conversión A/D.

#### Conversión ADC y DAC.

El conversor ADC convierte una señal analógica en digital mediante los procesos de discretización y codificación con un determinado número de bits. La amplitud en número de bits de la codificación determina el número total de valores discretos posibles. La precisión de la conversión depende del número de bits de la codificación.

La figura 1.33 muestra como se realizan las conversiones A/D y D/A. En la figura:

- Los valores de la abscisa corresponden al rango de valores analógicos posibles. El valor máximo suele estar acotado por el valor de la tensión de alimentación del circuito, 3.3 V en este caso.

- El eje de ordenadas contiene el conjunto de valores digitales posibles. En este caso 4096 valores, que van desde el valor 0 hasta el valor 4095. Cabe recordar que $4096 = 2^{12}$. Por tanto, el 4095 es el mayor número binario posible con 12 bits, $b1111\_1111\_1111$.

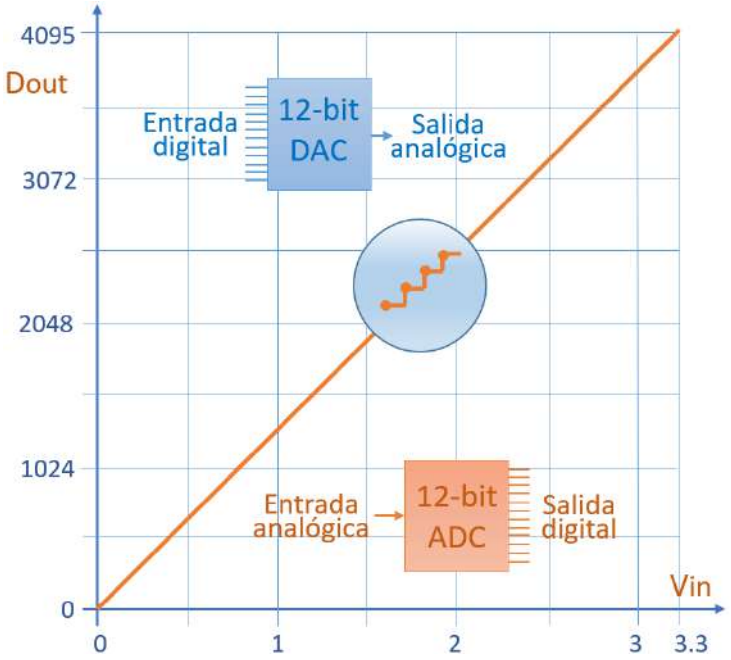



Figure 1.33: Conversión A/D y D/A.

Entonces sólo queda por realizar la conexión entre cada valor analógico con su correspondiente valor digital. Puesto que los valores analógicos van de 0 a 3.3, y existen 4096 valores digitales, la relación entre valor analógico y valor digital viene dado por la ecuación 1.6.

$$D_{out} = V_{in} \cdot \frac{4096 pasos}{3.3V} = entero(1.241 \cdot V_{in}) \quad pasos \tag{1.6}$$

El valor inverso de la fracción en la ecuación 1.6 es la resolución del conversor, o paso mínimo. A este valor se le denomina el ***Least Significant Bit*** o **LSB** del conversor, ecuación 1.7.

$$\frac{3.3V}{4096} = 0.806mV \tag{1.7}$$

En resumen,

- Una conversión A/D toma un valor analógico de entrada y lo convierte a un valor digital. Para ello, se divide el valor analógico por el LSB (ecuación 1.6). El valor resultante es el valor digitalizado del valor analógico. ¡Cuidado que se trunca la división!, por lo que no hay decimales.

  Por ejemplo, en un ADC de 12 bits, el valor digital correspondiente al valor analógico 2.0 V es

$$D_{out} = 2.0V \cdot \frac{4096pasos}{3.3V} = 2482pasos = 1001\_1011\_0010$$

- En la conversión D/A se realiza el proceso contrario. Se toma el valor digital y se multiplica por el LSB. Entonces se obtiene el valor analógico correspondiente. Fíjese que en este caso a un valor digital le corresponden muchos valores analógicos.

  Por ejemplo, en un ADC de 12 bits, el calor analógico correspondiente al valor digital 2000 es

$$2000pasos \cdot \frac{3.3V}{4096} = 1.61V$$

**Errores de conversión.**

La figura 1.34 resume los errores que se introducen en una conversión A/D o D/A. Para simplificar la nomenclatura, la gráfica corresponde a un conversor de 3 bits. Por ello los valores digitales posibles van de 0 a 7. Codificando en binario, corresponde a los valores 000 a 111.

La imagen de la izquierda muestra una conversión D/A (en el eje de abscisas está el valor digital y en el de ordenadas el valor analógico). La imagen de la derecha es una conversión A/D (valores analógicos en abscisas, y valores digitales en ordenadas).

Cuando se realiza una conversión D/A sólo se obtienen unos pocos valores analógicos de todo el rango analógico. Por ejemplo, al convertir el valor digital $b100$, el valor analógico obtenido es el valor $b100 \cdot FS/8 = 4 \cdot FS/8$, siendo FS el valor de fondo de escala o valor máximo. El siguiente valor de conversión es el correspondiente a $b101$ que es $5 \cdot FS/8$. Y todos los valores analógicos entre $4 \cdot FS/8$ y $5 \cdot FS/8$ no tienen conversión digital equivalente. Lo inverso ocurre en la conversión A/D. Muchos valores analógicos son convertidos a un único valor digital. Por ejemplo, todos los valores entre $[4 \cdot FS/8, 5 \cdot FS/8)$ son convertidos al valor digital $b100$.

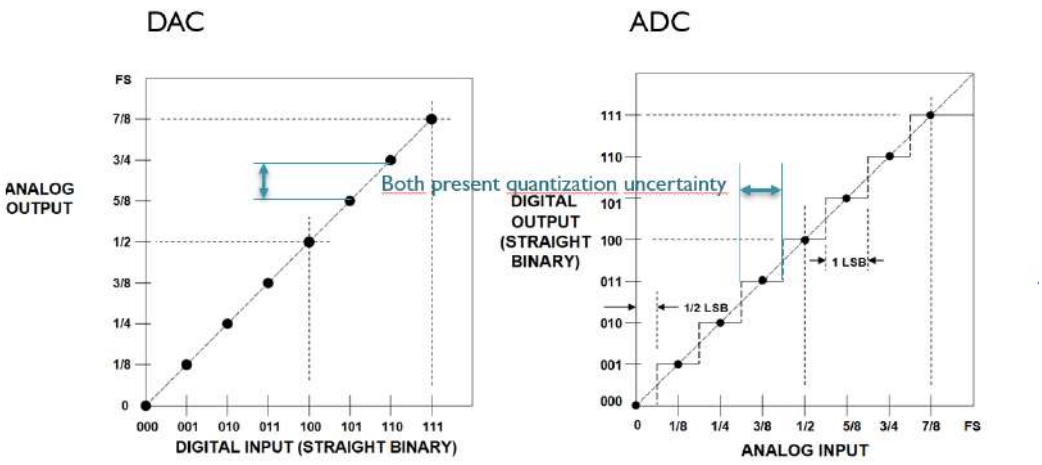



Figure 1.34: Errores de cuantificación en la conversión A/D y D/A.

# Bibliografía

1. M. Singh. *Introduction to Biomedical Instrumentation*. Phi learning, 2021.

2. S.W. Smith. *The Scientist and Engineer Guide to Digital Signal Processing*. California Technical Publishing, San Diego. 1999.

3. Analog.ppt. *Lecture 10: Digital-to-Analog Conversion*. ELITC.

4. Stuart Ball. *Analog Interfacing to Embedded Microprocessors. Real World Design*. Edit Newnes. 2004.

5. D.G. Bailey. *Design for Embedded Image Processing on FPGAs*. John Wiley & Sons. 2011.

6. N.K. Jog. *Electronics in Medicine and Biomedical Instrumentation*. PHI Learning. 2013.

# 1.A Análisis de Fourier y dominio frecuencial.

Las series de Fourier fueron introducidas por Joseph Fourier con el objetivo de solucionar una ecuación del calor en un plato metálico. Desde entonces el análisis de Fourier se ha convertido en la herramienta esencial para analizar la composición frecuencial de cualquier señal.

Utilizando las series de Fourier, las señales analógicas pueden expresarse mediante combinaciones de señales sinusoidales. Mediante el análisis de Fourier se pueden encontrar las frecuencias de las señales sinusoidales que componen la señal.

### Requisitos para el análisis de Fourier.

El análisis de Fourier establece que toda señal analógica se puede aproximar sumando funciones sinusoidales que cumplan:

- Las frecuencias de todas las funciones son múltiples de una primera función.
- Las amplitudes de las funciones son submúltiplos de la amplitud de la primera función.
- Todas ellas están en fase.

El resultado es una señal compuesta por un conjunto de señales sinusoidales de diferentes frecuencias:

- La frecuencia más baja es la frecuencia fundamental y se la denomina primer armónico.
- La segunda frecuencia, correspondiente al doble de la primera, se la llama segundo armónico.
- Y así sucesivamente: tercer armónico, etc.

### Ejemplo 1:

La figura 1.35 muestra cómo se puede formar una señal pulsante a partir de suma de señales sinusoidales.

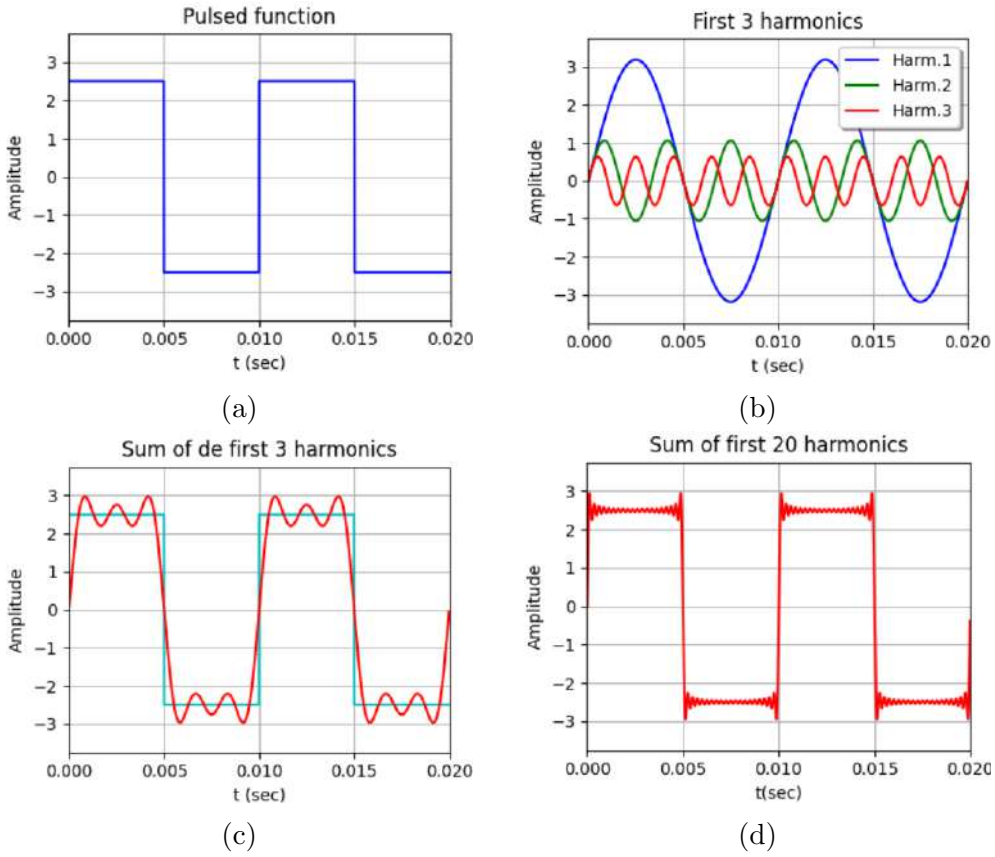



Figure 1.35: Señal pulsante a 100 Hz generada a partir de la serie de Fourier de la ecuación (1.8)

- La figura 1.35a muestra la señal pulsante. De acuerdo con el análisis de Fourier esta señal se puede expresar como suma de señales sinusoidales de acuerdo con la ecuación (1.8).

$$f(t) = \frac{4}{1 \cdot \pi} \sin(1 \cdot 2\pi f t) + \frac{4}{3 \cdot \pi} \sin(3 \cdot 2\pi f t) + \frac{4}{5 \cdot \pi} \sin(5 \cdot 2\pi f t) + \ldots \quad (1.8)$$

- En la figura 1.35b se muestran los primeros tres términos de la serie. Cada término se denomina **armónico**. En este caso se muestran los armónicos primero, segundo y tercero. El primer armónico, de frecuencia más baja, se denomina **fundamental**. Todos los armónicos superiores corresponden a frecuencias múltiples del armónico fundamental, de acuerdo a como se ha expresado en la serie (ecuación (1.8)).

- La figura 1.35c muestra la suma de éstos tres primeros armónicos. Se puede apreciar como ya se intuye la señal pulsante que se construye.

- Conforme aumenta el número de armónicos sumados se va formando mejor la señal analógica. La figura 1.35d corresponde a la suma de los 20 primeros términos, mostrando ya una forma bastante definitiva de la señal pulsante.

**Ejemplo 2:**

Cualquier forma de onda puede expresarse de forma parecida en términos de sumas de senos y cosenos. Por ejemplo, la ecuación (1.9) corresponde a la serie de Fourier que genera la señal triangular de la figura 1.36.

$$f(t) = 1 + 2\sin(t) - 2\sin(2t) + \frac{2}{3}\sin(3t) - \frac{1}{2}\sin(4t) + \ldots \qquad (1.9)$$

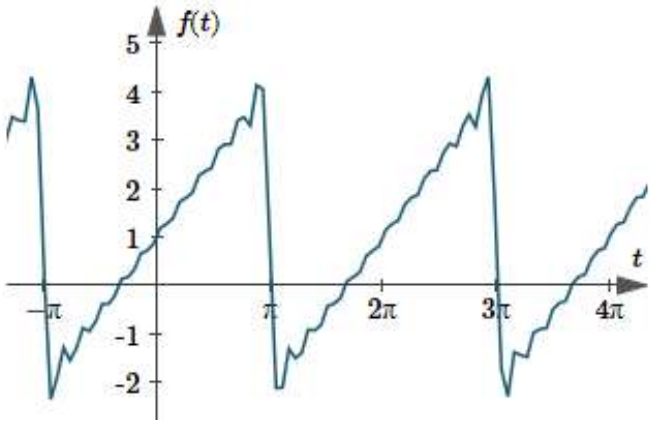


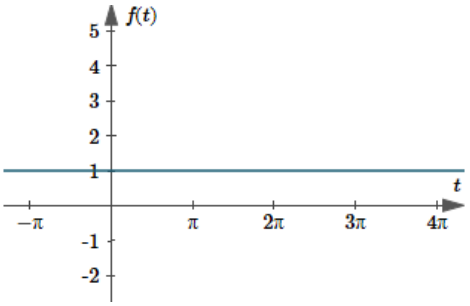

Figure 1.36: Señal triangular generada a partir de la serie de Fourier de la ecuación (1.9).

En este caso se observa que toda la señal se encuentra ligeramente desplazada hacia la parte positiva del eje de ordenadas. Esto se traduce en una constante que se suma a toda la onda. Esta constante toma diferentes nombres, como **valor DC**, o **término promedio**. Éste término constante no altera la constitución frecuencial de la serie.

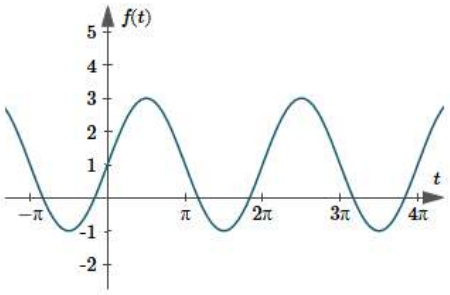
La figura 1.37 muestra los primeros armónicos de la señal triangular de la figura 1.36.

## 1.A.1 Forma canónica de la serie de Fourier.

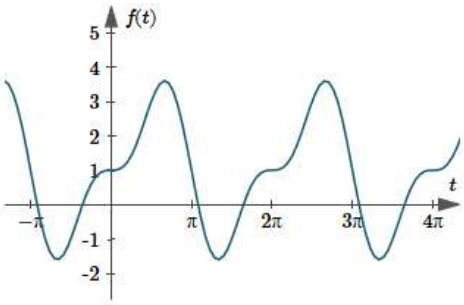
De forma genérica, toda función que cumpla con los requisitos para el análisis de Fourier se puede expresar como serie de Fourier aplicando el siguiente conjunto de ecuaciones.



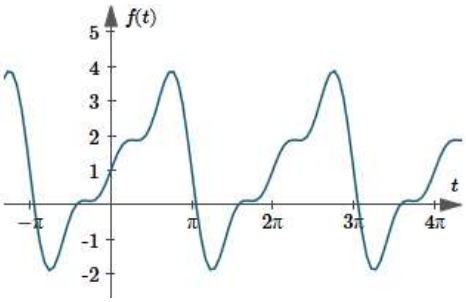

(a) $f(t) = 1$




(b) $f(t) = 1 + 2\sin(t)$




(c) $f(t) = 1 + 2\sin(t) - 2\sin(2t)$




(d) $f(t) = 1 + 2\sin(t) - 2\sin(2t) + \frac{2}{3}\sin(3t)$

Figure 1.37: Primeros términos de la serie de Fourier de la ecuación (1.9) que construyen la señal triangular.

Para ello se parte de la forma genérica del desarrollo en series de Fourier que viene dada por la ecuación (1.10)

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(nt) + \sum_{n=1}^{\infty} b_n \sin(nt) \tag{1.10}$$

donde $a_n$ y $b_n$ son los coeficientes de Fourier y $a_0/2$ representa el término constante o valor medio. Si el periodo va de $-\pi$ a $+\pi$ los coeficientes ($n = 1, 2, \dots$) vienen dados por las ecuaciones (1.11) a (1.13).

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\mathrm{d}t \tag{1.11}$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos(nt)\mathrm{d}t \tag{1.12}$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\sin(nt)\mathrm{d}t \tag{1.13}$$

Separando en los términos $n$ de la ecuación (1.10) se obtiene la serie de Fourier.

$$f(t) = \frac{a_0}{2} + (a_1 \cos(t) + b_1 \sin(t)) + (a_2 \cos(2t) + b_2 \sin(2t)) + (a_3 \cos(3t) + b_3 \sin(3t)) + \ldots \tag{1.14}$$

En la que quedan determinados los armónicos de la serie:

$$\text{Fundamental:} \quad (a_1 \cos(t) + b_1 \sin(t)) \tag{1.15}$$

$$\text{Segundo armónico:} \quad (a_2 \cos(2t) + b_2 \sin(2t)) \tag{1.16}$$

$$\text{Tercer armónico:} \quad (a_3 \cos(3t) + b_3 \sin(3t)) \tag{1.17}$$

## 1.A.2 Dominio frecuencial frente a dominio temporal.

Así como el dominio temporal muestra la evolución de una señal en el tiempo, el dominio frecuencial permite conocer la composición en términos de frecuencias que forma una señal. De ahí la importancia del análisis de Fourier que permite determinar el espectro frecuencial de una señal analógica cualquiera.

Aunque no so se introduce por simplicidad, debe comentarse que una representación frecuencial también incluye información sobre el desplazamiento de fase de la señal. Este desplazamiento de fase debe ser aplicado a cada frecuencia si se quiere recuperar de nuevo la señal original.

### Ejemplo 3: Señal suma de armónicos.

Supongamos que tenemos la señal que es suma, en amplitud unitaria, de las formas de onda sinusoidales de la figura 1.38, cuyas frecuencias angulares son 1, 2, 3 y 4 y todas de amplitud 1.

Puesto que las cuatro formas de onda son señales sinusoidales y se conoce su amplitud y su período, la suma puede expresarse de acuerdo a la ecuación (1.18).

$$f(t) = 1 \cdot cos(1 \cdot 2\pi t) + 1 \cdot cos(2 \cdot 2\pi t) + 1 \cdot cos(3 \cdot 2\pi t) + 1 \cdot cos(4 \cdot 2\pi t) \tag{1.18}$$

Si se expresa en una gráfica amplitud-frecuencia la composición frecuencial de esta señal, se obtiene la gráfica de la figura 1.39. A esta gráfica se la denomina **espectro frecuencial**.

### Ejemplo 4: Espectro frecuencial de la señal pulsante del ejemplo 1

En términos generales, encontrar la composición frecuencial de una señal analógica requiere la aplicación de la **transformada de Fourier**. Puede entenderse como una generalización del método introducido aplicando la forma canónica de la serie

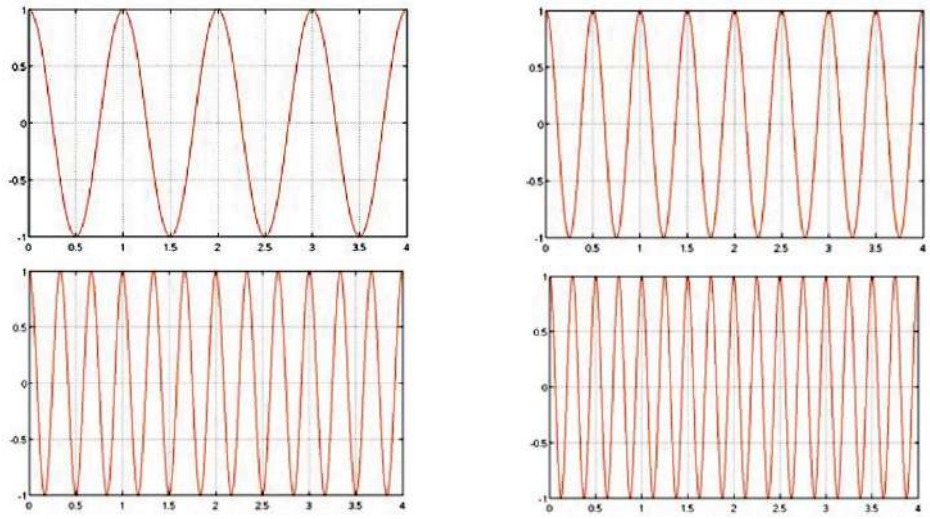


Figure 1.38: Señales sinusoidales

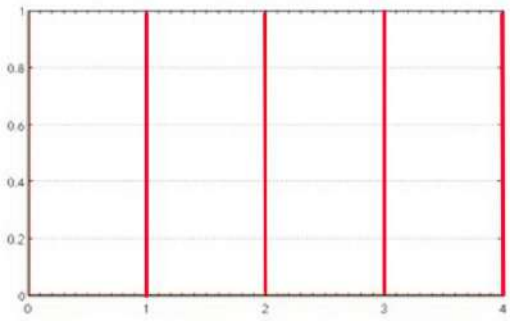


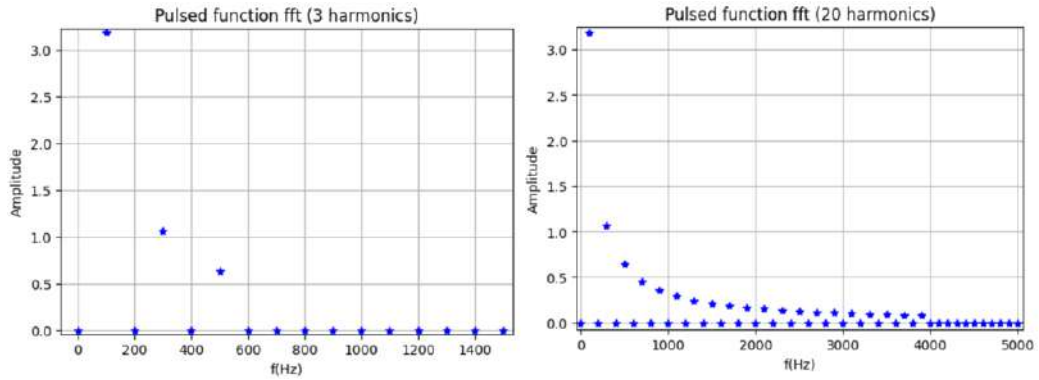
Figure 1.39: Espectro frecuencial del ejemplo 3.

de Fourier. Su aplicación permite determinar las amplitudes, frecuencias y fases de los armónicos que componen la señal.

Así, por ejemplo, aplicando la transformada de Fourier a la señal pulsante del ejemplo 1, se obtiene el espectro frecuencial que se muestra en la figura 1.40.

- La figura 1.35a muestra el espectro que se obtendría de la señal formada con sólo tres términos de la serie (figura 1.35c). Puede comprobarse como la amplitud de los armónicos se corresponde con la amplitud de los armónicos de la señal, y como las frecuencias con amplitud corresponden a las frecuencias de los términos significativos.

- De manera similar puede entenderse la figura 1.40b, que se corresponde a la suma de los 20 primeros armónicos de la serie (cuya gráfica es la figura 1.35d).




(a) (b)

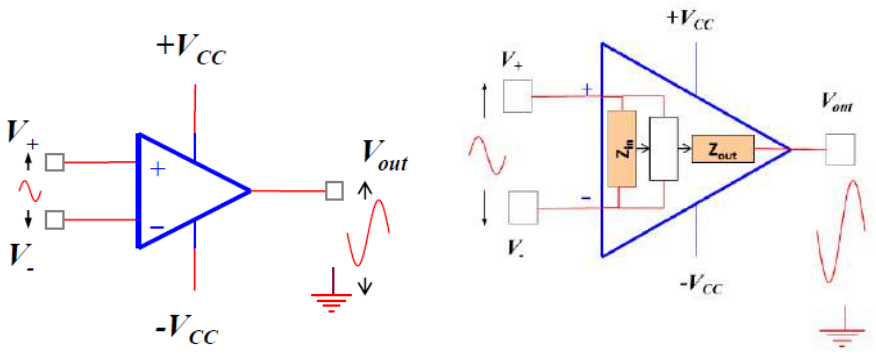
Figure 1.40: Espectro frecuencial de la señal pulsante del ejemplo 1.

*(*) El programa Python utilizado para generar la forma de onda, calcular la serie de Fourier y encontrar el espectro frecuencial se han añadido como apéndice al final del capítulo.*

## 1.B El amplificador operacional.

Un **amplificador operacional** o **OpAmp** (figura 1.41) es un módulo electrónico, normalmente formado por transistores, que tiene una entrada no inversora (+), una inversora (-) y una salida. La tensión de salida es la diferencia entre las entradas + y - multiplicada por la ganancia G del dispositivo en lazo abierto, ecuación 1.19.

$$V_{out} = G \cdot (V_+ - V_-) \tag{1.19}$$




(a) Símbolo (b) Modelo interno

Figure 1.41: Amplificador operacional.

La amplificación es la relación entre la señal de salida respecto a la de entrada, y viene dada por la ecuación 1.20.

$$\Delta V = \frac{V_{out}}{V_{in}} = \frac{V_{out}}{V_+ - V_-} \tag{1.20}$$

El OpAmp puede funcionar en lazo abierto o lazo cerrado. En **lazo abierto** funciona cuando no hay ninguna conexión mediante dispositivo (resistencia, capacidad, etc.) entre la salida y la entrada. En **lazo cerrado** se conecta la salida con la entrada del OpAmp por medio de algún dispositivo electrónico. Como dispositivo activo, el OpAmp debe conectarse a la tensión de alimentación. Puede funcionar con tensiones bipolares (+Vcc y –Vcc), o sólo con tensiones positivas (+Vcc y –Vcc = 0 V).

### Propiedades del OpAmp.

El OpAmp es un dispositivo muy utilizado debido a las características que presenta.

1. Tiene muy alta ganancia $G$. La ganancia en lazo abierto suele ser superior a 100000 llegando, fácilmente a 1000000, lo que le permite amplificar señales muy pequeñas.

2. Idealmente se supone muy alta resistencia de entrada $Z_{in} = \infty$, y muy baja resistencia de salida $Z_{out} = 0$ (ver el modelo presentado en la figura 1.41b). Ello permite la conexión de dispositivos en cascada sin pérdidas de señal en las interfaces.

3. Funciona como dispositivo bipolar cuando se alimenta con tensiones positivas y negativas; y como unipolar en caso de conectar el terminal negativo a 0 V (siendo normalmente el terminal *tierra*).

4. Tiene respuesta de salida lineal entre Vss, -Vss (denominado *rail-to-rail*).

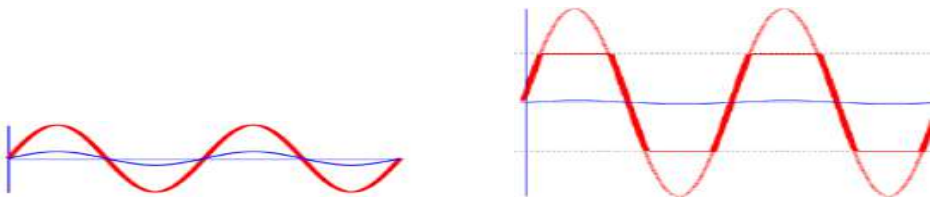
5. En el modelo ideal, $V_+ = V_-$. Ello permite realizar circuitos electrónicos utilizando OpAmps de forma muy fácil.

6. Debido a la alta ganancia del OpAmp se satura fácilmente en lazo abierto.

   $V_{out} = V_{cc}$ cuando $V_+ > V_-$

   $V_{out} = -V_{cc}$ cuando $V_+ < V_-$

   Entonces, para señales no muy pequeñas actúa como comparador de voltaje (figura 1.42).



(a) Amplificación x100 (b) Amplificación x1000

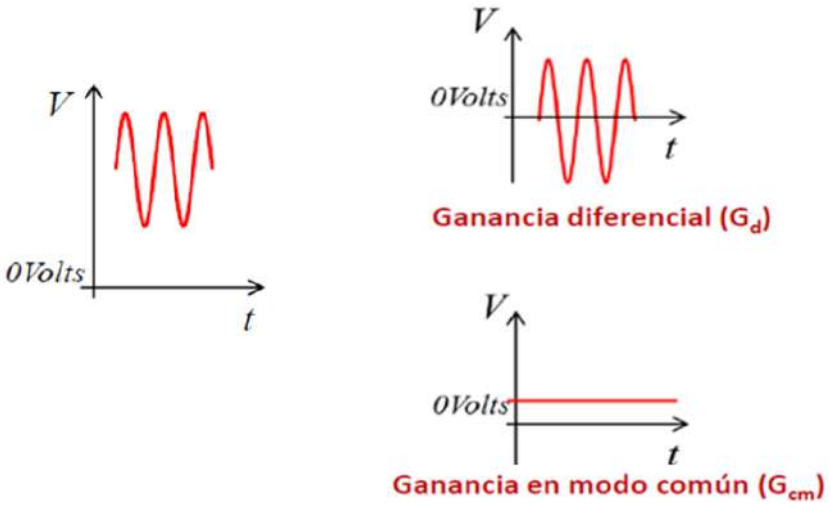
Figure 1.42: Amplificación en un OpAmp. En azul la señal de entrada. En rojo la señal de salida.

7. Una propiedad muy importante del OpAmp es el denominado **CMRR** ó ***Common-Mode Rejection Ratio*** ó **Razón de Rechazo en Modo Común**. Esto significa cuán grande es la capacidad de separar la ganancia diferencial del dispositivo frente a la ganancia en modo común.

El CMRR se calcula de acuerdo a la ecuación 1.21, en la que $V_d$ y $V_c$ son las ganancias diferencial y en modo común, respectivamente, y $V_{cm}$ es la tensión media de las dos entradas.

$$\left.\begin{aligned} CMRR &= 20log\frac{A_d}{A_c} \\ A_d &= \frac{V_{out}}{V_+ - V_-} \\ A_c &= \frac{V_{out}}{V_{cm}} \end{aligned}\right\} \tag{1.21}$$

La figura 1.43 ayuda a entender el significado del CMRR. En la figura 1.43a se muestra una señal que llega al amplificador que se compone de una señal DC (estática o valor medio) más una señal AC (variable sobre el valor medio) (figura 1.43b). Normalmente, la señal AC es la que lleva la información interesante de la medida y la que interesa amplificar, rechazando así la señal DC. Un OpAmp con buen CMRR será capaz de amplificar la señal AC, denominando a esta amplificación **ganancia diferencial**, y de no amplificar la señal DC, denominándolo **ganancia en modo común**.




(a) (b)

Figure 1.43: a) Señal de entrada al amplificador. b) Capacidad de separar la señal diferencial de la señal en modo común.

Se considera que el CMRR comienza a ser aceptable a partir de 70 dB

(es decir 10000), y bueno cuando es superior a los 100 dB (es decir 100000).

Un buen CMRR es fundamental en instrumentación, puesto que significa que el amplificador es capaz de amplificar correctamente señales pequeñas.

8. El ancho de banda de un amplificador es un indicador del rango de frecuencias en al que amplificador funciona como tal. Suele representarse en un **diagrama de Bode**.
En el diagrama de Bode de la figura 1.44 el ancho de banda del filtro es de 1 kHz.

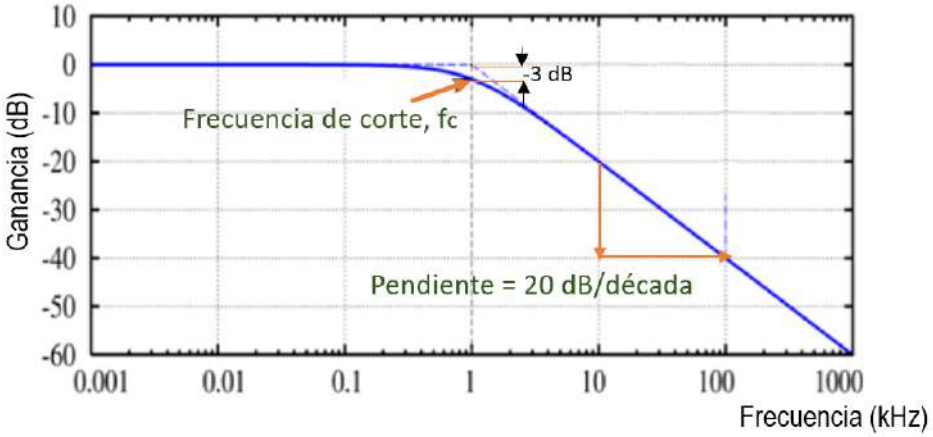



Figure 1.44: Representación en diagrama de Bode.

### Diagrama de Bode

El diagrama de Bode es una representación ganancia-frecuencias en escala logarítmica que informa de forma muy gráfica del comportamiento del circuito con tan sólo prestar atención a tres parámetros:

- El rango de frecuencias de pendiente plana. Indica que en todo este rango de frecuencias se mantendrá estable la ganancia del circuito.

- La frecuencia a la que empieza a decaer la ganancia. Es la **frecuencia de corte** $f_c$ y corresponde a la frecuencia a la que el circuito atenúa en un factor mitad a la señal de entrada. En la escala en dB se corresponde con una caída en -3 dB.

- La pendiente de la rampa. Es el factor de caída, o cuán rápidamente se pierde amplificación conforme aumenta la frecuencia. Puesto que se representa en escala logarítmica, la recta de caída corresponde s una caída exponencial de la amplificación. Una caída en 20 dB/década indica que la potencia decae en un factor 10; en una caída en 40 dB/década, la potencia decae en un factor 100; etc.

Considerando estas tres características, se comprueba que el diagrama de Bode de la figura 1.44 representa el comportamiento de un filtro pasa-bajas (deja pasar las frecuencias bajas):

- La pendiente plana corresponde a una ganancia de 0 dB. Es decir, ganancia igual a 1. El circuito ni amplifica ni reduce la señal de entrada.
- La frecuencia de corte se encuentra en 1 kHz. A partir de la frecuencia 1 kHz, el circuito reducirá la señal de entrada. Es decir, la filtrará.
- La velocidad de caída es de 20 dB/década.

En la gráfica, 0 dB corresponde a una ganancia igual a 1. Es decir, es el límite entre que un circuito amplifique o filtre (figura 1.45). Valores superiores a 1 dB indicarán amplificación. Valores inferiores indicarán reducción o filtraje.

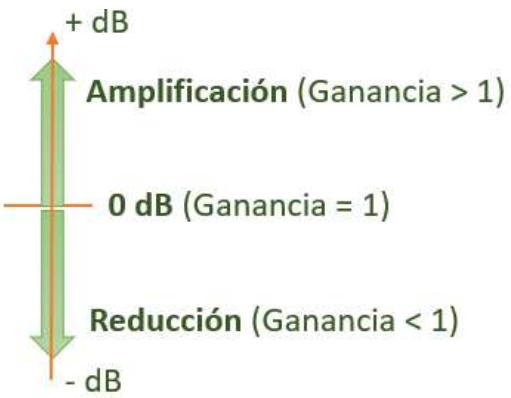



Figure 1.45: En una escala en dB, 0 dB es el límite entre amplificar o filtrar.

### Concepto de decibelio.

En electrónica suele ser común utilizar el concepto de **decibelio** o **dB** cuando hay gran variación en las magnitudes, como ocurre normalmente cuando se

intenta encontrar la el comportamiento (ganancias, respuesta frecuencial, CMRR, etc.) de un circuito electrónico.

El decibelio se calcula tomando el logaritmo de la división de dos magnitudes y multiplicando por 20. Por ejemplo, el cálculo de la ganancia de un OpAmp se realiza de acuerdo a la ecuación 1.22:

$$dB = 20log\frac{V_{out}}{V_{in}} \qquad \frac{V_{out}}{V_{in}} = 10^{dB/20} \tag{1.22}$$

donde $V_{in} = V_+ - V_-$.

Así, aplicando la ecuación 1.22, es fácil pasar de dB a ganancias normales y viceversa:

$$\begin{aligned} &0 \quad dB \implies V_{out} = 10^0 \cdot V_{in} = V_{in} \\ &20 \quad dB \implies V_{out} = 10^1 \cdot V_{in} = 10 \cdot V_{in} \\ &40 \quad dB \implies V_{out} = 10^2 \cdot V_{in} = 100 \cdot V_{in} \\ &Etc. \end{aligned}$$

Por ejemplo, cuando se habla de sonido se dice que el nivel legal de confort acústico son 55 dB, tomando como referencia el nivel mínimo audible.

### Circuitos en lazo abierto.

Cuando se utiliza un OpAmp en lazo abierto, y no como comparador, debe considerarse el rango de entrada de la señal a amplificar. También se debe recordar que el rango de salida estará acotado por las tensiones de alimentación.

Así, por ejemplo, considerando un OpAmp con ganancia de 120 dB a 1 kHz, y que se alimente entre +5 V y -5 V, aplicando la ecuación 1.19, se tiene que el rango de la señal de entrada para no saturar el OpAmp es de:

Considerando que 100 dB equivale a una ganancia de G = 1000000,

$$Rango(V_{in}) = \frac{Rango(V_{out})}{G} = \frac{10V}{1000000} = 10\mu V$$

Realmente hay pocos dispositivos con rango tan pequeño, y que no sufran de ruido o señales espúreas, que puedan ser amplificados con esta ganancia sin saturar el amplificador. Lo usual es utilizar el amplificador en lazo cerrado.

**Circuitos en lazo cerrado.**

Debido a este poder de amplificación que tienen los OpAmp normalmente se utilizan en lazo cerrado, lo que permite controlar la ganancia del amplificador.

Existen múltiples configuraciones en lazo cerrado con OpAmps. Las dos más habituales son como *inversor* (tabla 1.4a), en la que se realimenta la salida a la entrada mediante una resistencia, y como circuito *diferencial* (tabla 1.4b), que es usado en la amplificación de señales diferenciales.

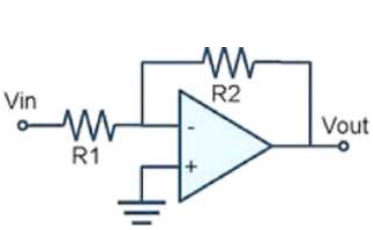
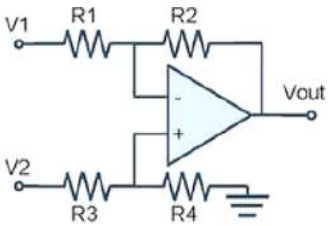
|  |  |
|---|---|
| $\left.\begin{array}{r}\frac{V_{in}-V_-}{R_1}=\frac{V_--V_{out}}{R_2}\\ V_-=V_+=0\end{array}\right\}$ | $\left.\begin{array}{r}\frac{V_{in}-V_-}{R_1}=\frac{V_--V_{out}}{R_2}\\ V_-=V_+=0\\ \frac{V_2-V_+}{R_3}=\frac{V_+-0}{R_4}\\ SiR_1=R_3 \quad y \quad R_2=R_4\end{array}\right\}$ |
| $V_{out}=-V_{in}\cdot\frac{R_2}{R_1}$ | $V_{out}=(V_2-V_1)\cdot\frac{R_2}{R_1}$ |
| a) Circuito inversor | b) Circuito diferencial |

Tabla 1.4: Circuitos con OpAmp en Lazo cerrado

# Chapter 2

# Sensores y datos

*Joan Oliver*

Un sistema de adquisición de datos toma señales del mundo real para generar datos digitales que puedan ser manipulados por dispositivos computacionales. La toma de señales físicas y transducción al dominio eléctrico se realiza a través de sensores. El sensor es el *front-end* o punto de entrada de la información que se digitaliza y es guardada, monitorizada y manipulada por los computadores.

La cantidad de sensores que existen huy en día es enorme para cada dominio energético y para cada aplicación, Por ello, este capítulo sólo es una introducción al campo de la sensórica. Se presentan las propiedades fundamentales que se exigen a un sensor y se muestran áreas de aplicación, centrándose fundamentalmente en el campo biomédico. Para cada sensor y cada aplicación es interesante vislumbrar el ingenio que se ha utilizado para desarrollar la tecnología así como las técnicas de procesado que llevan a cabo el proceso de transducción de magnitud física a dato.

## 2.1 Introducción a los sensores

Los sistemas de adquisición de señales se basan en dispositivos transductores que son capaces de adquirir información del mundo externo y transportarlo a energía eléctrica que puede ser procesada y almacenada como datos por los actuales circuitos electrónicos.

Un **transductor** es un dispositivo que transforma un tipo de magnitud física en otro tipo de magnitud física. Cuando la transformación es a magni-

tud eléctrica se denomina **sensor**. La información del sensor normalmente es acondicionada y enviada a un procesador que procesa los datos y los guarda.

Hay muchos tipos de sensores: de temperatura, de fuerza y de presión, optoelectrónicos, de medida inercial, electrodos, magnéticos, químicos, etc.

En general, un transductor o sensor se compone de:

- Elemento excitador
- Elemento transductor
- Encapsulado

Por ejemplo, en un pulsioxímetro (figura 2.1) el elemento excitador son los diodos emisores de luz, el elemento transductor son foto-receptores y el encapsulado es el mecanismo que acopla a todo el sistema.

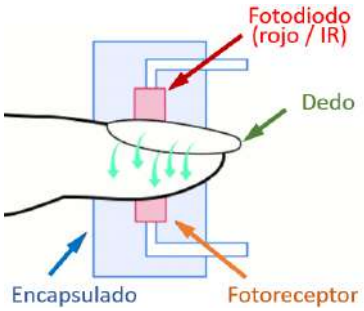



Figure 2.1: Elementos de un pulsioxímetro.

### Clasificación.

Dependiendo de los requerimientos existen distintas clasificaciones de sensores. Por mencionar algunas:

- Según el dominio energético del sensor.
  Hay una clasificación clásica que divide los sensores en seis dominios de energía:
  - Mecánico. Agrupa las magnitudes de fuerza, velocidad, aceleración, posición, presión.
  - Térmico. Son magnitudes de temperatura, calor, entropía, flujo de calor.
  - Químico, que contiene magnitudes de concentración, entropía, calor.

  - Radiante. Magnitudes de intensidad electromagnética, fase, polarización, reflexión/refracción, rayos (X, gamma, ...), radioisótopos
  - Magnético. Integra magnitudes como intensidad de campo, flujo magnético, permeabilidad, momento magnético.
  - Eléctrico. Las magnitudes utilizadas son voltaje, corriente, carga, resistencia, capacidad, polarización.

  Es evidente que algunos sensores pueden ser agrupados dentro de varios dominios.

- Según el tipo de señal de salida.

  - Analógicos. La salida abarca un rango continuo en voltaje o corriente. El acondicionamiento de la señal suele necesitar de amplificación y digitalización. Ejemplo: sensores de temperatura como termistor, sensores optoelectrónicos (como diodos fotoemisores o transistores fotoreceptores), etc.
  - Digitales. Son sensores de salida discreta, ya codificada en bits. Por ejemplo, contadores de señal, codificadores.

- Según la aportación de energía.

  - Moduladores o pasivos. Cuando se necesita de aportación de energía externa, como es el caso de los termistores (sensores de temperatura), en los que cambios de temperatura provocan cambios de resistencia. Entonces se necesita un circuito de polarización.
  - Autogeneradores, o sensores activos, Son capaces de generar energía. Por ejemplo, los termopares (sensores de temperatura) son uniones metálicas en la que la recombinación de los electrones en la unión genera una diferencial de potencial en sus extremos.

### 2.1.1 Propiedades de los sensores

Las propiedades de los sensores se agrupan en dos grupos fundamentales: dinámicas y estáticas.

Las propiedades dinámicas analizan el comportamiento del sensor cuando es excitado por una entrada, normalmente de tipo pulsante o escalón. En este caso se determina su velocidad de respuesta, tiempo de arranque y comportamiento dinámico en general.

Las propiedades estáticas indican como se comporta el dispositivo y cuáles son sus características cuando es polarizado y a entrada constante. De entre ellas, cabe destacar las siguientes:

- Sensibilidad.
- Linealidad.
- Precisión.
- Exactitud.
- Fondo de escala.
- *Offset*.
- Deriva.

### Sensibilidad

A nivel de interfaz, un sensor es un dispositivo que se compone de entrada $x$, salida $y$ y que internamente se excita ante cambios en la entrada.

Entonces, la sensibilidad $S$ se define como la dependencia de la salida $y$ respecto a un parámetro $a$ de las magnitudes dependientes de la entrada $x$. Formalmente se expresa con la ecuación 2.1.

$$S(x_a) = \left.\frac{dy}{dx}\right|_{x=x_a} \tag{2.1}$$

La sensibilidad es la pendiente de la gráfica en la función de transferencia (que relaciona el cambio de la salida respecto a la entrada). La figura 2.2 muestra la función de transferencia de un termistor. La resistencia del sensor varía con la temperatura, $R = R(T)$.

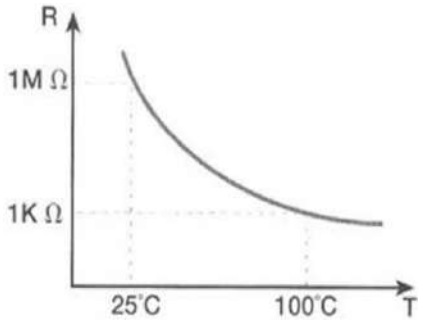



Figure 2.2: Función de transferencia R = R(T) típica de un termistor.

**Linealidad.**

Indica si la sensibilidad se mantiene constante en todo el rango. Se corresponde con una curva lineal en la gráfica de sensibilidad.

Claramente, el termistor en un sensor no lineal. A temperaturas bajas la resistencia es mucho más sensible a los cambios de temperatura que a temperaturas altas.

**Precisión (*Precision*).**

Es la capacidad de medir el mismo valor cuando se realiza la medida varias veces en las mismas condiciones. Es una medida de la variabilidad estadística.

**Exactitud (*Accuracy*).**

Es el grado de proximidad de las medidas al valor verdadero. El valor verdadero es aquel obtenido en una medida realizada siguiendo un método ejemplar.

Precisión y exactitud van de la mano. Como muestra la figura 2.3, puede pasar cualquiera de los cuatro casos: poco preciso y poco exacto; poco preciso y exacto; preciso y poco exacto; y preciso y exacto.

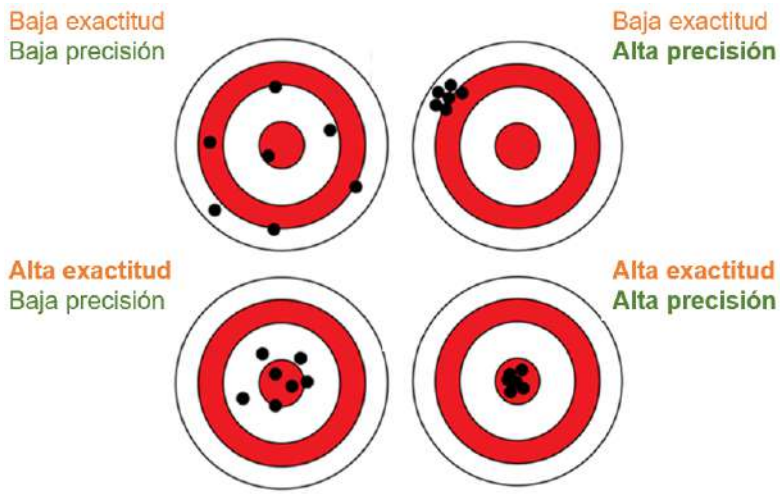



Figure 2.3: Exactitud y precisión..

Los errores aleatorios y de ruido y los errores sistemáticos afectan a la precisión y exactitud (figura 2.4). Los primeros producen variaciones respecto a la media de las medidas (precisión), mientras que los segundos afectan cuan alejados estamos del valor verdadero de la medida (exactitud).

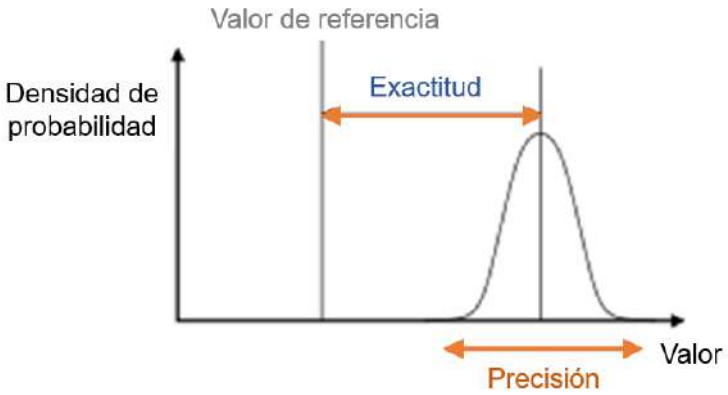



Figure 2.4: Exactitud y precisión.

**Rango o fondo de escala (*Full Scale Range*).**

Son los valores mínimo y máximo que puede tomar la medida. En la figura 2.5 correspondería a los valores máximo y mínimo de la salida del sensor.

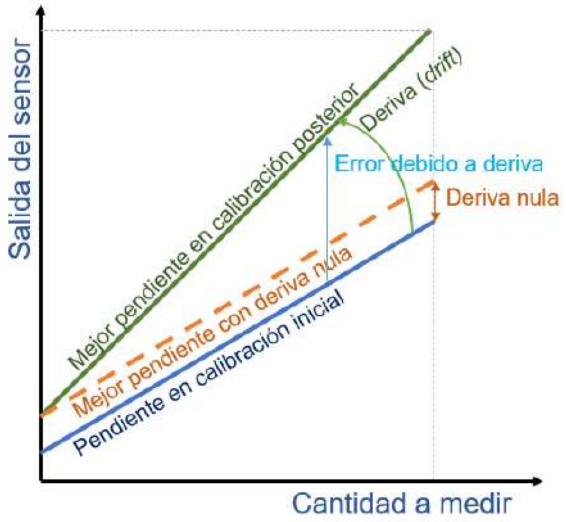



Figure 2.5: Función de transferencia. Calibración y deriva.

***Offset*.**

Es la salida que da el sensor ante una entrada nula. En la figura 2.5 la línea continua que parte sobre el eje de ordenadas representa la salida exacta del sensor. El offset es la línea discontinua paralela que presenta un error de inicio respecto a la salida exacta

**Deriva (*Drift*).**

Es la desviación que va mostrando la salida del sensor conforme aumenta el valor de la entrada.

## 2.2 Sensor de temperatura.

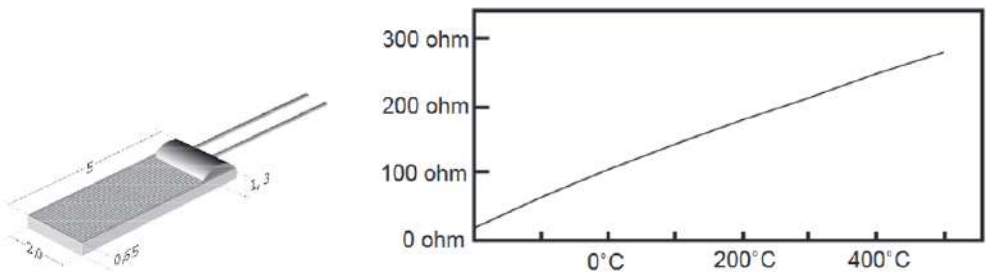
El sensor de temperatura es uno de los sensores más fáciles de usar. Se puede construir utilizando múltiples tecnologías, incluso usando la dependencia de la conductividad respecto a la temperatura en un dispositivo semiconductor.

Como dispositivo base existen tres tipos de sensores muy extendidos en su uso:

- RTDs o *Resistance Temperature Detector.*
- Termistor
- Termopar (*Thermocouple*).

### 2.2.1 RTD.

Es un dispositivo construido con bobina de metal o película metálica que tiene una dependencia lineal con la temperatura. La figura 2.6 muestra la dependencia de la resistencia con la temperatura de un RTD PT100. El número 100 indica que a la temperatura de referencia ($0^oC$ en este caso) la resistencia vale $100\Omega$. $\alpha$ es el coeficiente de temperatura del RTD, que depende del material.




(a) $\alpha(Pt) = 0.003927/^oC$ (b) Función de transferencia PT100

Figure 2.6: RTD típico y función de transferencia.

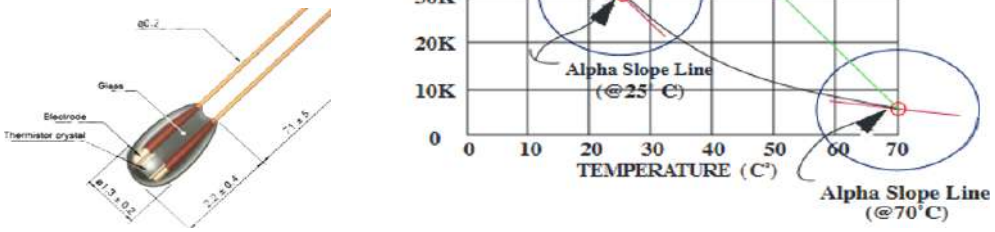
La ecuación 2.2 describe el modelo típicamente utilizado de un RTD

PT100. Se observa como la resistencia varía con la temperatura de manera muy lineal.

$$R = 100 \cdot (1 + \alpha \cdot (T - T_0)) \tag{2.2}$$

### 2.2.2 Termistor.

Es un dispositivo resistivo hecho de óxidos metálicos soldados en una bola de epoxi o vidrio como en la figura 2.7. Pueden ser PTC (*Positive Temperature Coeficient*) o NTC (*Negative Temperature Coeficient*) siendo estos últimos los más empleados.




(a) Encapsulado termistor (b) Función de transferencia PT100

Figure 2.7: Curva de transferencia.

El principio de funcionamiento es una variación de la resistencia no lineal con la temperatura. La figura 2.7b muestra la curva característica de un termistor NTC. Se toma como resistencia base la que tiene el termistor a 25 ºC. Así, la gráfica corresponde a un termistor NTC de 30 $k\Omega$.

Para variaciones de temperatura muy pequeñas se puede utilizar un modelo lineal del termistor calculando el **coeficiente de temperatura** $\alpha$ (2.3) en el punto de temperatura T. El coeficiente de temperatura es la pendiente de la curva en el punto de temperatura especificado.

$$\alpha = \frac{1}{R_T} \cdot \frac{dR}{dT} \cdot 100 \quad (\%/^oC) \tag{2.3}$$

Cuando hay variaciones más grandes, es muy extendido el uso del modelo Steinhart-Hart (2.4) para calcular la temperatura. A, B y C son tres constantes que se calculan formando un sistema de tres ecuaciones, y tomando de la tabla de especificaciones del termistor, temperatura y su resistencia de tres puntos distintos.

$$\frac{1}{T} = A + B \cdot (logR) + C \cdot (logR)^3 \tag{2.4}$$

### 2.2.3 Termopar.

El termopar es un dispositivo formado por dos cables de metales diferentes en unión (figura 2.8). En el otro extremo de los cables hay otra unión (unión de referencia) compensada electrónicamente para la temperatura ambiente. El acondicionamiento requiere de amplificación.

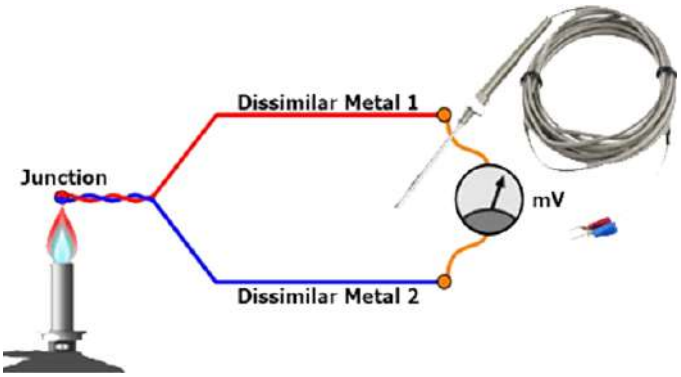



Figure 2.8: Termopar

El principio de funcionamiento se basa en efecto Seeback: dos cables unidos formando un bucle con uniones a diferente temperatura, provocan una corriente que fluye por el bucle. Así, cuando se calienta la unión del sensor se genera un potencial termoeléctrico de pocos $\mu V$ proporcional a la diferencia de temperatura entre las dos uniones.

### 2.2.4 Comparación de tecnologías.

El uso de cada tipo de sensor depende totalmente de la aplicación. La tabla 2.1 resume las principales propiedades de cada dispositivo.

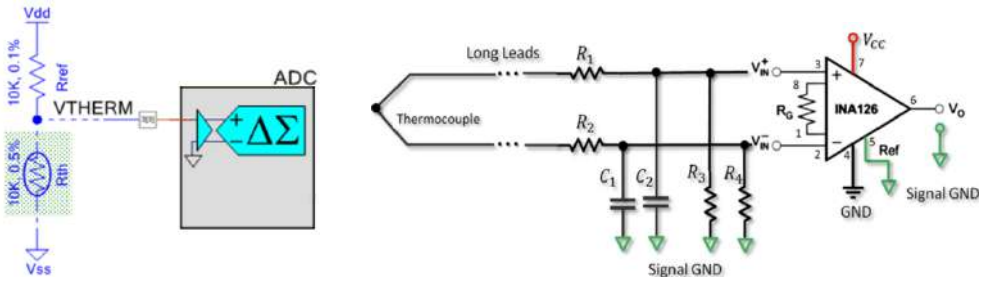
Así, en una aplicación donde sea necesaria alta sensibilidad, y atendiendo a la facilidad de aplicar post-proceso hoy en día, los termistores son buenos candidatos como sensores de temperatura.

| | RTD | Termistor | Termopar |
|---|---|---|---|
| Rango de temperatura | $-260^oC \dots 850^oC$ | $-80^oC \dots 260^oC$ | $-270^oC \dots 1800^oC$ |
| Sensibilidad | Media | Alta | Baja |
| Linealidad | Buena | Mala | Mediana |
| Coste dispositivo | Medio-Alto | Bajo | Muy bajo |
| Coste sistema medida | Medio-Bajo | Medio-Bajo | Alto (Amplificación) |

Tabla 2.1: Características de los sensores de temperatura

### 2.2.5 Arquitectura de adquisición típica.

El circuito de adquisición de señal es personalizado para cada tipo de sensor. La figura 2.9b muestra un posible circuito para un termistor y un termopar.




(a) En termistor. (b) En termopar.

Figure 2.9: Circuitos de acondicionamiento.

El termistor es un elemento pasivo por lo que requiere de circuito de polarización. Puede ser tan simple como un divisor de tensión (figura 2.9a). El termistor funciona como resistencia variable con la temperatura, provocando variaciones de tensión en el nodo VTHERM. Entonces la tensión del nodo es digitalizada a través del ADC y enviada al procesador para su tratamiento. Para conocer la temperatura, el procesador realiza el tratamiento inverso. Conociendo el valor digital de llegada, calcula el correspondiente valor analógico que entra en el ADC. Es decir, encuentra el valor VTHERM. Y por consiguiente, a través del divisor de tensión encuentra la resistencia del termistor $R_{th}$. Aplicando la ecuación de Steinhart-Hart 2.4 se calcula la temperatura que se está midiendo.

En cambio el termopar es un elemento auto-generador, pero de señal muy pequeña. De acuerdo con la figura 2.9b, no hace falta polarizar el termopar, pero sí amplificar. La señal amplificada será enviada a un ADC para ser digitalizada y después al procesador.

## 2.3 Sensor de presión.

La presión se define como una fuerza ejercida por unidad de superficie. Se puede medir de diferentes maneras según se quiera presión absoluta o relativa. La medida de la presión puede realizarse de valor absoluto, o valor diferencial.

El método clásico de medir la presión es mediante el manómetro de columna líquida (figura 2.10a. Si se ejerce una presión diferente a cada lado del manómetro $P_a, P_b$, la presión ejercida es función de la diferencia de altura a ambos lados del manómetro $h$, la gravedad $g$ y de la densidad del líquido $\rho$, ecuación 2.5

$$h = \frac{P_a - P_b}{g\rho} \tag{2.5}$$

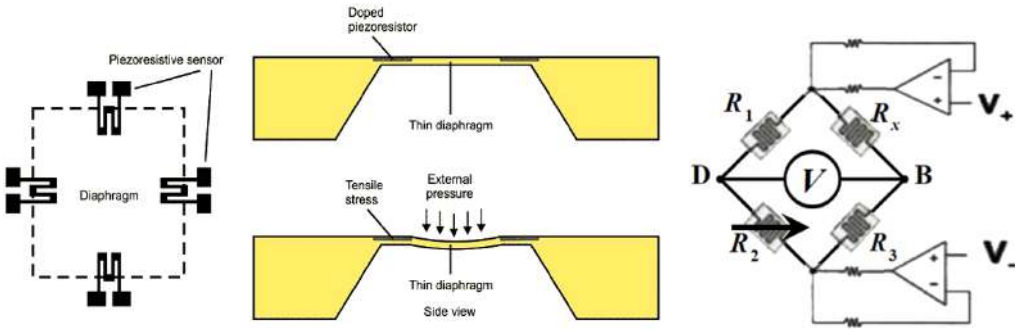

(a) Manómetro de columna líquida. (b) Sensores de presión integrados.

Figure 2.10: Sensores de presión.

La figura 2.10b muestra sensores de presión actuales integrados. En el sensor miniaturizado se incluye una membrana sobre el circuito integrado sensible a la presión.

El sensor traduce cambio de presión a cambio eléctrico. El principio que se utiliza para medir la presión es similar al mecanismo que se utiliza en el manómetro. Utilizando técnicas de micro-mecanización se crea, conjuntamente, la circuitería electrónica de lectura y una membrana sensible al cambio de presión. Los sujetadores de la membrana están unidos a puentes resistivos o capacitivos capaces de cambiar su magnitud a cualquier cambio de fuerza que se ejerza en sus extremos 2.11a.

La figura 2.11b muestra el circuito en Puente de Wheatstone que es capaz de medir cualquier cambio de magnitud en cualquiera de sus resistencias. En el circuito $R_1$, $R_2$ y $R_3$ son resistencias de valores conocidos, además la resistencia $R_2$ es ajustable. Si la relación de las dos resistencias del brazo conocido $R_1/R_2$ es igual a la relación de las dos del brazo desconocido $R_x/R_3$, el voltaje entre los dos puntos medios será nulo y por tanto no circulará




(a) Micromecanizado para sensor de presión.

(b) Circuito de lectura en Puente de Wheatstone.

Figure 2.11: Circuito de lectura en un sensor de presión.

corriente alguno entre estos dos puntos C y B. Entonces, pequeños cambios en el valor de $R_x$ romperán el equilibrio y serán claramente detectados por la indicación del galvanómetro. Cuando el puente está construido de manera que $R_3$ es igual a $R_2$, $R_x$ es igual a $R_1$ en condición de equilibrio. Y en condición de equilibrio siempre se cumple que (ecuación 2.6):

$$R_x = \frac{R_1 R_3}{R_2} \tag{2.6}$$

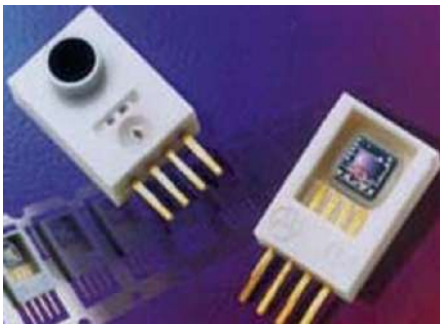
**Disposables.**

Las tecnologías MEMS (*MicroElectronic MicroSystems*) permiten la creación de la circuitería de lectura cerca del sensor con lo que se miniaturiza todo el sistema (figura 2.12a), y abaratan su coste. Con ello se reemplazan los sensores externos de presión arterial que cuestan más, y requieren de esterilización y tienen que ser recalibrados antes de su uso.

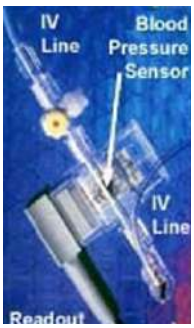
La figura 2.12b muestra un ejemplo de sensor de presión miniatura de un solo uso (**disposable**) utilizado para controlar la presión arterial en los hospitales. Estos sensores se conectan a una línea intravenosa de pacientes y monitorizan la presión arterial.

## 2.4 Sensores optoelectrónicos.

Los sensores sensibles a la radiación son dispositivos que operan en frecuencias determinadas de todo el espectro electromagnético. Tal como muestra la figura 2.13 abarcan sensores sensibles al espectro electromagnético usado por



(a) Sensores de presión miniaturizados.




(b) Sensor de presión miniatura.

Figure 2.12: MEMS lectores de presión.

las comunicaciones hasta los rayos $\gamma$. Más o menos en el medio del espectro (en la figura) queda la radiación visible.

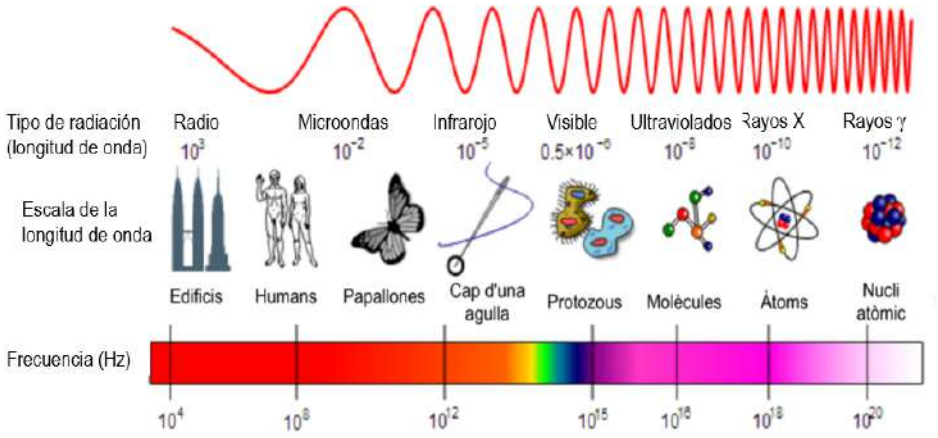



Figure 2.13: Espectro electromagnético.

Los sensores optoelectrónicos son sensores sensibles al rango óptico o próximo, como son los rangos ultraviolado e infrarrojo. Convierten la energía lumínica en energía eléctrica. Son o bien materiales sensibles a la luz o semiconductores que generan electrones a partir de los fotones que inciden en el semiconductor.

Los tecnologías habituales de sensores ópticos son:

- Dispositivos fotoconductores, como las resistencias dependientes de la luz, (*LDR*, *Ligth Dependent Resistance*), Materiales como el sulfuro de cadmio funcionan como resistencias variables en función de la luz incidente..

- Los dispositivos fotovoltaicos (células solares) y los dispositivos *CCD* o *Charge Coupled Device* generan una corriente de electrones a partir de los fotones incidentes.

- Los fotodiodos y fototransistores son materiales semiconductores que convierten luz incidente en corriente de salida. Debido a su potencial amplificador, los fototransistores suelen actuar como dispositivos receptores.

La figura 2.14 muestra una de la arquitecturas típicas de emisor-receptor optoelectrónico. Se trata de un emisor, normalmente un fotodiodo, que emite luz en una frecuencia determinada (el infrarrojo es muy utilizado) y un fototransistor que es capaz de amplificar la corriente generada en función de la luz recibida.

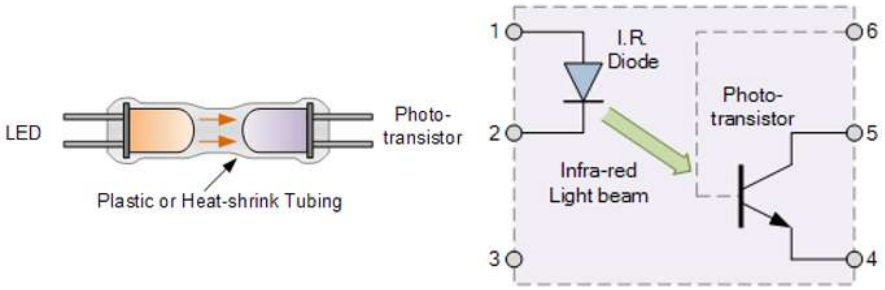



Figure 2.14: Dispositivos fotoeléctricos: fotodiodo (emisor) y fototransistor (receptor).

Existen múltiples aplicaciones que utilizan este par emisor-receptor. Por ejemplo, interruptores, detectores de distancia, circuitos opto-acopladores, aisladores en circuitos eléctricos, eliminadores de ruido en instrumentos MIDI, etc. O pulsioxímetros, por ejemplo, en medicina.

### 2.4.1 Pulsioxímetro.

#### Absorción de la luz por la hemoglobina.

El color de una sustancia disuelta que no emite luz depende de su capacidad especifica de absorber la luz. Esta absorción de la luz tiene lugar en rangos de longitudes de onda muy específicos. La parte proporcional absorbida de la luz en diversas longitudes de onda determina el color de la sustancia.
La hemoglobina (*haemoglobin*, *Hb*, *HGB*) es la proteína que transporta el oxígeno y contiene hierro. Se encuentra en los glóbulos rojos de los vertebrados y los tejidos de algunos invertebrados. El oxígeno es transportado de

los pulmones a los tejidos combinándose con la hemoglobina, formando la oxihemoglobina. La hemoglobina tiene la propiedad de ligarse al oxígeno o de cederlo donde se necesita (ecuación 2.7).

$$Hb \quad + \quad O_2 \quad \rightleftharpoons \quad HbO_2 \tag{2.7}$$

En promedio, la hemoglobina de 100 ml de sangre puede combinarse con un total de 10 ml de oxigeno, cuando la saturación es del 100%. El color rojo de una solución de hemoglobina se debe a que la radiación de longitud de onda corta (luz azul) es absorbida bastante bien, mientras que la radiación de longitud de onda más alta (luz roja) tiene mejor transmisión.

Si se analiza con un espectroscopio la luz que pasa por la hemoglobina oxigenada se encuentran dos bandas de absorción, en la radiación de longitudes de onda $lambda = 541nm$ y $lambda = 577nm$. En cambio, la hemoglobina reducida absorbe la luz en medio de ambas longitudes de onda, mostrando una banda de absorción más ancha con pico en $lambda = 555nm$ (figura 2.15).

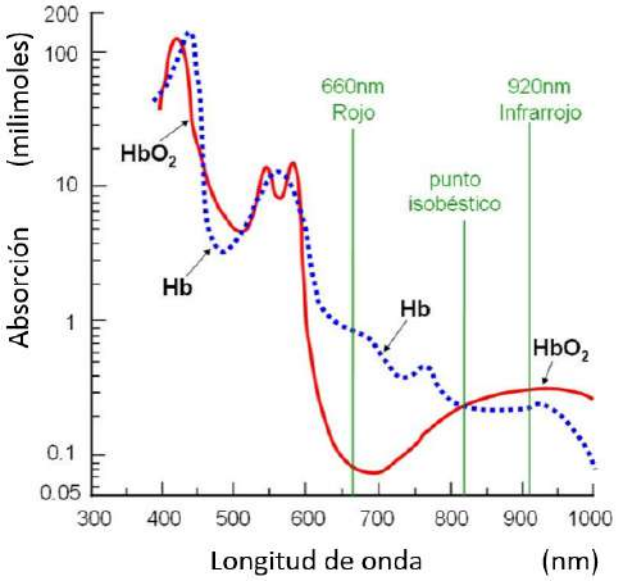



Figure 2.15: Curvas de absorción de la hemoglobina

**Oximetría.**

La **oximetría** se encarga de la medida de la saturación de oxígeno en la sangre, y se basa en las diferencias que hay en la absorción espectral entre la

hemoglobina reducida $Hb$ y la hemoglobina oxigenada $HbO_2$. Es interesante para ello observar la gráfica de la figura 2.15 más allá de las longitudes de onda de absorción:

- En general, a medida que la longitud de onda de la luz incidente aumenta, la absorción de la luz por las $Hb$ y $HbO_2$ disminuye siguiendo un modelo distinto en cada una de ellas.
- Entre los $600nm$ y los $800nm$ la oxihemoglobina absorbe menos luz que la hemoglobina reducida, lo que significa que menos luz roja es absorbida por la $HbO_2$. De ahí el color rojo intenso de la sangre arterial.
- Entre los $800nm$ y los $1000nm$ (región infrarroja del espectro) hay el punto *isobéstico*, en los $805nm$, punto de longitud de onda en el que las dos hemoglobinas muestran la misma absorción de la luz.

**Oxímetro de pulso ó pulsioxímetro.**

La pulsioxometría se basa en la asunción que los cambios en la absorción de la luz están totalmente causados por la sangre arterial. Por tanto, los cambios en la intensidad de la luz transmitida en un vaso sanguíneo pulsante pueden ser utilizados para obtener una medida precisa no invasiva de la saturación de oxígeno en la sangre.

Si un vaso arterial pulsante se sitúa entre un emisor de luz y un sensor de luz, la absorción de la luz variará con los cambios de grosor debido a la pulsación arterial. Como la forma de onda pulsante detectada sólo está producida por la sangre arterial, se puede medir la oxigenación de la sangre sin tener que considerar atenuaciones debidas al grosor de piel y tejido, pigmento de la piel o flujo venoso.

La pulsioxometría es una técnica fotopletismográfica ya que mide en partes del organismo cambios de volumen resultantes de las pulsaciones de la sangre utilizando métodos fotoeléctricos.

Existe una fórmula empírica para calcular la saturación de oxígeno $SaO_2$ a partir de la relación entre las intensidades de luz roja e infrarroja transmitidas, 2.8:

$$SaO_2 = A - B\frac{R(roja)}{R(infrarroja)} \tag{2.8}$$

donde $A$ y $B$ son constantes empíricas determinadas estadísticamente en una calibración *in vivo* en la que se comparan resultados obtenidos por el pulsioxímetro con análisis sanguíneos. También son función de los emisores y receptores de luz.

Las intensidades de luz roja $R(roja)$ e infrarroja $R(infrarroja)$ se obtienen a partir de un proceso de normalización por la componente pulsante (AC) de la intensidad de luz dividida por la componente no pulsante (DC), ecuación 2.9.

$$R(roja) = \frac{AC(roja)}{DC(roja)}, \quad R(infrarroja) = \frac{AC(infrarroja)}{DC(infrarroja)} \tag{2.9}$$

A diferencia de otros oxímetros, este tipo de medida hace el pulsioxímetro independiente del grosor del tejido, del pigmento de la piel, o o de la cantidad de luz incidente.

Los pulsioxímetros suelen utilizar la luz roja a $660nm$ y luz infrarroja a $940nm$.

### Pulsioxímetros de reflexión y de transmisión

Ambos tienen un funcionamiento similar. La diferencia está en el tipo de sensor óptico utilizado. Ambos consisten de fotodiodos emisores de luz roja e infrarroja, i de fototransistor como receptor. Pero los reflexivos utilizan la reflexión de la luz en el tejido, mientras que los transmisores la luz es transmitida a través del tejido.

La principal limitación de los pulsioxímetros de reflexión está en el relativamente bajo nivel fotopletismográfico que se recibe de la piel.

La aplicación de los pulsioxímetros de transmisión está limitada a áreas del cuerpo humano como mano, dedos, o lóbulos de las orejas. La figura 2.16 muestra un ejemplo de montaje del sensor.

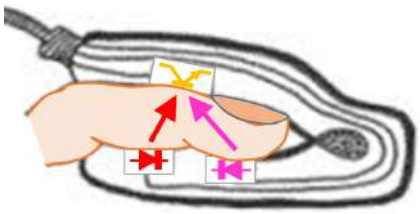


Figure 2.16: Sensor pulsioxímetro.

El sensor del pulsioxímetro sólo es la parte frontal de todo el sistema. La figura 2.17 muestra un posible circuito acondicionador. Está compuesto por los siguientes bloques:

- El sensor, compuesto de los fotodiodos de radiación roja e infrarroja, y el fotoreceptor.

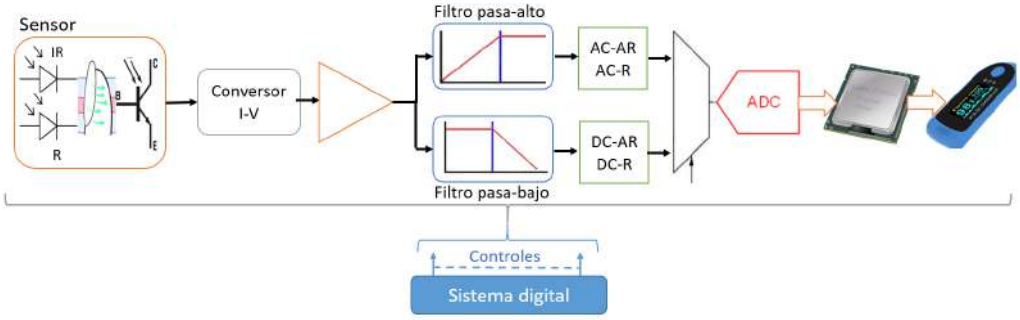



Figure 2.17: Circuito pulsioxímetro.

  - La entrada de los fotodiodes es una señal pulsante en corriente de frecuencia de 370 Hz. Esta frecuencia es suficiente para que todo el sistema sea capaz de detectar las componentes pulsantes de las señales pletismográficas, que se encuentran alrededor de los 5 Hz. Los pulsos conmutan alternativamente entre led rojo y led infrarrojo del sensor.
  - La salida del fotoreceptor es una corriente que depende de la radiación recibida.

- Un conversor corriente-voltaje transforma la salida del fotoreceptor en voltaje de entrada del amplificador.

- La salida del amplificador contiene las componentes DC y AC de los pulsos IR y R. Mediante los respectivos filtros pasa-bajos y pasa.altos se separan estas componentes.

- La conversión ADC de los componentes envía los datos digitales al procesador y display para ser procesados, registrados y mostrados.

- Los correspondientes circuitos digitales se encargan de generar los pulsos de los fotodiodos y de sincronizar correctamente todo el proceso.

## 2.5 Electrodos y medida de biopotenciales.

Un **electrodo** es un conductor eléctrico que se utiliza para establecer un circuito cerrado y hacer posible el paso de una corriente eléctrica a través de un medio (que separa dos electrodos).

Visto como sensor en biomedicina, el **electrodo** es el dispositivo que realiza la conversión de potencial iónico a potencial eléctrico.

### 2.5.1 Potencial de acción

Las células musculares y nerviosas están envueltas por una membrana semipermeable que permite, así como también limita, el paso de soluciones líquidas conductoras formada por iones, principalmente de sodio $Na^+$, potasio $K^+$ y cloro $Cl^-$.

**Estado de reposo.**

La membrana permite el paso hacia el interior de la célula de iones $K^+$ y $Cl^-$, pero limita el paso de iones $Na^+$. En el estado de reposo, la permeabilidad de $K^+$ es de 50 a 100 veces mayor que la del $Na^+$. Esta semi-permeabilidad crea un desequilibrio de carga eléctrica y de concentración entre el interior y el exterior de la célula. El exterior de la célula tenderá a ser positivo respecto al interior por la mayor concentración de $Na^+$ del exterior, al mismo tiempo que la concentración de $K^+$ será mayor en el interior de la célula para compensar esta diferencia de concentración.

El equilibrio de toda esta dinámica se consigue creando una diferencia de potencial entre el interior y el exterior de la célula (figura 2.18a). Es el que se denomina **Potencial de reposo**. Se dice entonces que la célula está **polarizada**. Este potencial de membrana vale entre $-60mV$ i $-100mV$. Se toma como referencia el exterior de la célula.

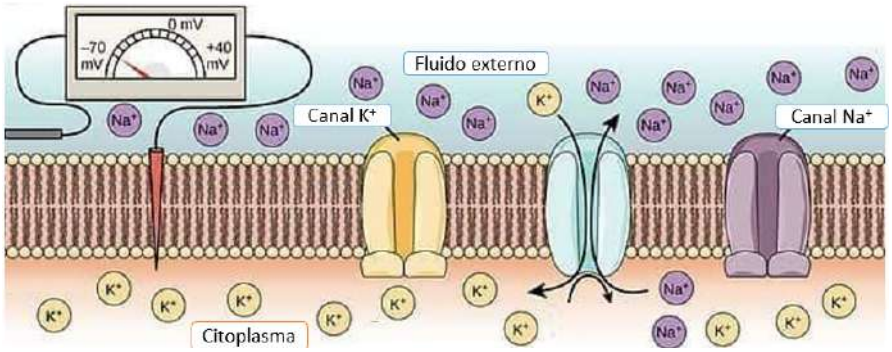
La excitación de esta membrana por cualquier evento externo permite la entrada de $Na^+$, provocando un flujo de corriente iónica que reduce la permeabilidad de la membrana a este ion, y permitiendo su entrada en masa. Los iones $Na^+$ penetran al interior de la célula intentando establecer su equilibrio de concentración iónica con el exterior. Al mismo tiempo los iones $K^+$ salen hacia el exterior también para establecer su equilibrio de concentración iónica (figura 2.18b).

Sin embargo, los iones $K^+$ tienen menor movilidad que los iones $Na^+$, lo que provoca que la célula quede momentáneamente **despolarizada**, con potencial positivo. Es el **potencial de acción**. El potencial de acción tiene un valor aproximado de $+20mV$.

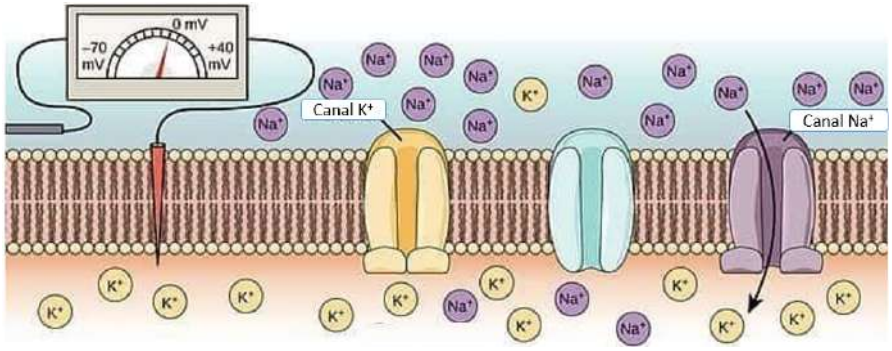
Cuando este efecto alud se desvanece, la membrana semipermeable restablece sus características de permeabilidad selectiva, bloqueando de nuevo la entrada de $Na^+$ hacia el interior y expulsando su exceso del interior hacia el exterior. Es el proceso conocido como **bomba de sodio** y constituye la etapa de **repolarización** de la célula.

La figura 2.19 resume todo el proceso que se lleva a cabo durante la ejecución de un potencial de acción en una célula.

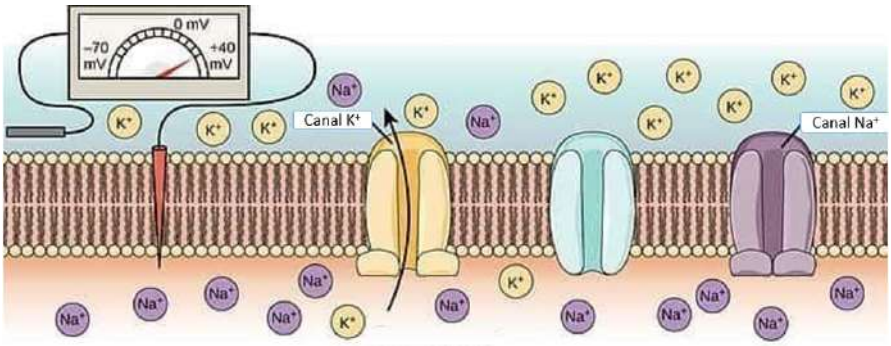
Esta excitación que se produce en una célula, a su vez, sirve de estímulo




(a) Célula polarizada (en reposo).




(b) Célula despolarizada (potencial de acción).




(c) Repolarización.

Figure 2.18: Transporte de iones en la ejecución de un potencial de acción.

a las células vecinas, provocando la generación de potenciales de acción en éstas, y así sucesivamente. Como efecto resultante se tiene la propagación del potencial de acción.

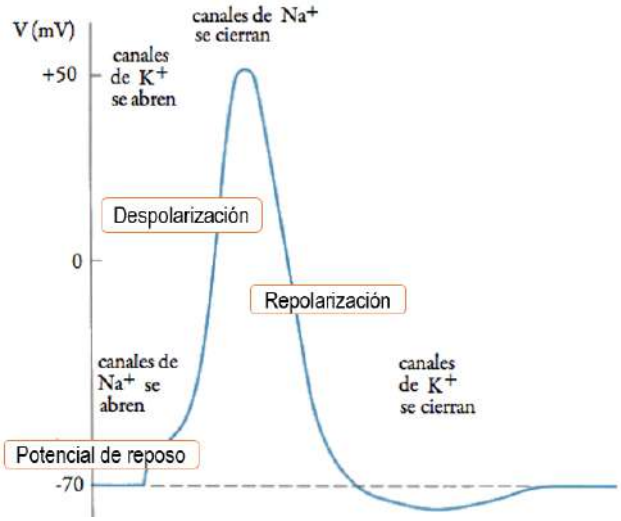



Figure 2.19: Potencial de acción.

### Características del potencial de acción.

La duración del potencial de acción y el tiempo de repolarización depende del tipo de tejido. Por ejemplo, en las células musculares y nerviosas la duración del potencial de acción es de alrededor de 1 ms, siendo el tiempo de repolarización muy pequeño. En el musculo cardíaco el tiempo de repolarización está entre 150 y 300 ms.

La amplitud y la dirección del potencial de acción son fijos. No dependen de la intensidad o duración del estímulo.

El tiempo entre excitaciones de una célula se denomina **tiempo refractario**. Es de alrededor de 1 ms en las células nerviosas.

La **velocidad de propagación** es la velocidad con la que se mueve el frente del potencial de acción de una superficie celular. Esta velocidad depende del tipo, y características física y geométrica de la célula.

- En los nervios la velocidad de propagación está entre los 20 ms y los 150 ms.
- En el músculo cardiaco la velocidad de propagación es de 0.2 a 0.4 m/s.

### Medida del potencial bioeléctrico.

Los potenciales de acción que se producen en la célula se deben a corrientes iónicas. Y para poder medir estas corrientes iónicas se necesita un transductor que sea capaz de transportar a corrientes eléctricas. Este potencial de acción se puede medir de forma diferencial utilizando diversos **electrodos**.

Es difícil medir, sin embargo, un potencial de acción único, ya que sólo al poner el electrodo se alterarían las condiciones de medida. Es más fácil medir la contribución de muchas células en la generación y propagación del potencial de acción. Por tanto, el registro se va a corresponder a la actividad global mostrada por el tipo de célula. Por ejemplo, en la figura 2.21 se muestra el registro de un potencial de acción generado por el músculo cardíaco. La suma de los potenciales de acción generados por los distintos músculos y células que componen el corazón va a formar un **latido** del corazón.

El hecho de que el cuerpo humano sea un medio líquido rico en electrolitos permite que sea capaz de transmitir la actividad iónica generada por los grupos de células excitables hacia la superficie corporal, que es donde se colocarán los electrodos para registrar esta actividad. Así, la medida de un potencial bioeléctrico cualquiera va a ser el resultado de medir en la superficie del cuerpo la suma en el tiempo de la contribución de los potenciales de acción de un grupo de células de un determinado tipo.

Los potenciales bioeléctricos más significativos que se pueden registrar son:

- **Electrocardiograma** o **ECG**. Es el registro de los biopotenciales generados por los músculos cardíacos. Se realiza en la superficie del cuerpo.

- **Electromiograma** o **EMG**. Es el registro de los biopotenciales de los músculos. Cuando se actúa en la superficie del cuerpo se realiza cerca del músculo de interés.

- **Electroencefalograma** o **EEG**. Es un registro de la actividad neuronal del cerebro. El encefalograma presenta unas formas de onda complejas y difíciles de reconocer. Depende mucho de la situación de los electrodos. Por ello se trabaja con unas posiciones normalizadas.

- Existen otros registros no tan conocidos como el **electrorretinograma** o **ERG**, **Electrooculograma** o **EOG**, etc.

Las características eléctricas más importantes de los biopotenciales son:

- **Rango dinámico de amplitud**. Es el rango que toman los valores del biopotencial.

- **Rango dinámico de frecuencia**. Son los valores de frecuencias de los biopotenciales.

La tabla 2.2 resume valores usuales para los biopotenciales mencionados.

| Biopotencial | Frecuencial (Hz) | Amplitud (mV) |
|---|---|---|
| ECG | $0.05 - 100$ | $0.1 - 4$ |
| EMG | $10 - 2000$ | $0.01 - 15$ |
| EEG | $0.5 - 30$ | $< 0.3$ |

Tabla 2.2: Características eléctricas de los biopotenciales

### 2.5.2 Electrodos.

Los **electrodos**, aparte de ser los elementos que realizan la función de punto de contacto entre el cuerpo y el dispositivo electrónico, su misión es la de realizar la transducción de potencial iónico a potencial eléctrico. Asimismo, la función principal de la pasta electrolítica que se coloca entre piel y electrodo, no es la de disminuir la impedancia de la piel para reducir la impedancia de entrada total del sistema, sino que es la transmisora del potencial iónico al electrodo.

El electrodo, que hace de interfase entre un material metálico y la disolución iónica, da lugar a un potencial eléctrico en la interfase denominado **potencial de electrodo**, y es el resultado de la diferencia de ritmos de difusión de los iones hacia dentro y hacia fuera del electrodo. Se llega al equilibrio con la formación de una carga superficial en la interfase, con polaridades inversas en la capa próxima al metal respecto de la capa próxima a la disolución.

La figura 2.20a muestra un modelo eléctrico de la interfase de un electrodo.


(a) Electrodo. (b) Electrodo diferencial.

Figure 2.20: Modelización del electrodo.

- $R_g$ es la resistencia del gel o pasta conductora situada entre el electrodo y la piel.
- $R_j$ representa la resistencia en la interfase electrodo-electrolito.

- $C_j$ es la capacidad generada por la doble capa de carga en la interfase electrodo-electrolito.

- $V_e$ es el potencial de electrodo de la interfase.

Debido a que es imposible determinar exactamente el potencial de superficie de un electrodo, las medidas con electrodos se realizan de forma diferencial (2.20b). La medida es la diferencia de potencial instantánea entre los dos electrodos. Si los electrodos son del mismo tipo, la diferencia de potencial es muy pequeña y depende sólo de la diferencia de potencial iónico entre los dos puntos del cuerpo donde se ejecuta la medida.

Aún en el caso de trabajar con electrodos idénticos, en la práctica debe considerarse también que:

- Suele aparecer una pequeña tensión continua de offset.

- Puede haber una pequeña actividad química en el electrodo que genere variaciones en el potencial $V_e$.

Un electrodo utilizado como electrodo de referencia es el de plata - cloruro de plata $Ag/AgCl$ por tener una excelente estabilidad. Este tipo de electrodo se recubre electrolíticamente de un trozo de plata pura con cloruro de plata. Los iones de plata se combinan con los iones cloruro para producir moléculas neutras de cloruro de plata.

En el modelo del la figura 2.20 se observa que en la impedancia del electrodo se incluye un efecto capacitivo. A causa de este efecto, la respuesta del electrodo dependerá de la frecuencia. Además, tanto el potencial del electrodo como la impedancia dependen de la *polarización*. Es decir, del paso de corriente continua en la interfase metal-electrolito. Por ello es recomendable utilizar una impedancia de entrada muy grande en la etapa de entrada (acondicionamiento) al circuito.

**Tipos de electrodos.**

Existen múltiples tipos de electrodos en función de la aplicación para la que se requieren. Sólo por citar alguno podemos nombrar:

- **Electrodos superficiales**.
  Se utilizan para obtener la medida de potenciales bioeléctricos en la superficie corporal. Presentan diámetros de alrededor de 0.4 cm, pudiendo llegar a 1 cm. La impedancia de la piel, vista por el electrodo, suele variar entre $0.5k\Omega$ en piel sudorosa hasta $20k\Omega$ en piel seca.
  En el diseño del circuito cabe considerar al electrodo como una fuente

de voltaje de alta impedancia, lo que influye en el diseño del circuito. Existen múltiples tipos. Entre los más usados, electrodo de plata, de succión, flexibles, flotantes.
En la monitorización de ECG, EEG, EMG es común el uso de botones metálicos con contacto de plata que se mantienen en su posición mediante un hule de espuma con adhesivo.

- **Microelectrodos**.
  Utilizados en encefalografía para medir EEG y que consisten de pequeñas agujas subdèrmicas que penetran en la piel.
  Su uso es encefalografía es delicado y suelen implantarse quirúrgicamente.

- **Electrodos superficiales**.
  Son empleados para medir la diferencia de potencial que se establece entre la parte interna y externa de células. Permite analizar el comportamiento de la membrana celular en aplicación de diferentes potenciales externos. Sus dimensiones¡es han de ser lo suficientemente pequeñas para no dañar la membrana celular.

### 2.5.3 Electrocardiograma.

El **electrocardiograma, ECG**, es un registro gráfico obtenido por los electrodos que muestra la evolución temporal de la actividad cardíaca. Es decir, muestra la evolución temporal del latido del corazón. Y el latido del corazón se forma por composición de todos los potenciales de acción que se generan en los músculos y células del corazón (figura 2.21).

El resultado, como muestra la figura 2.22, es la forma de onda típica que se repite en un ECG. Se compone de una serie de lóbulos nombrados alfabéticamente:

- La onda P representa el proceso de despolarización de los músculos auriculares. Es la propagación del impulso que se ha generado en el nódulo sinoauricular o SA del corazón.

- Las ondas Q, R y S forman el complejo QRS. Es la despolarización de los músculos ventriculares y la repolarización de los músculos auriculares, que se produce cuasi-simultáneamente. El tiempo de conducción es de $0.06s - 0.08s$.

- La onda T representa la repolarización de los músculos ventriculares.

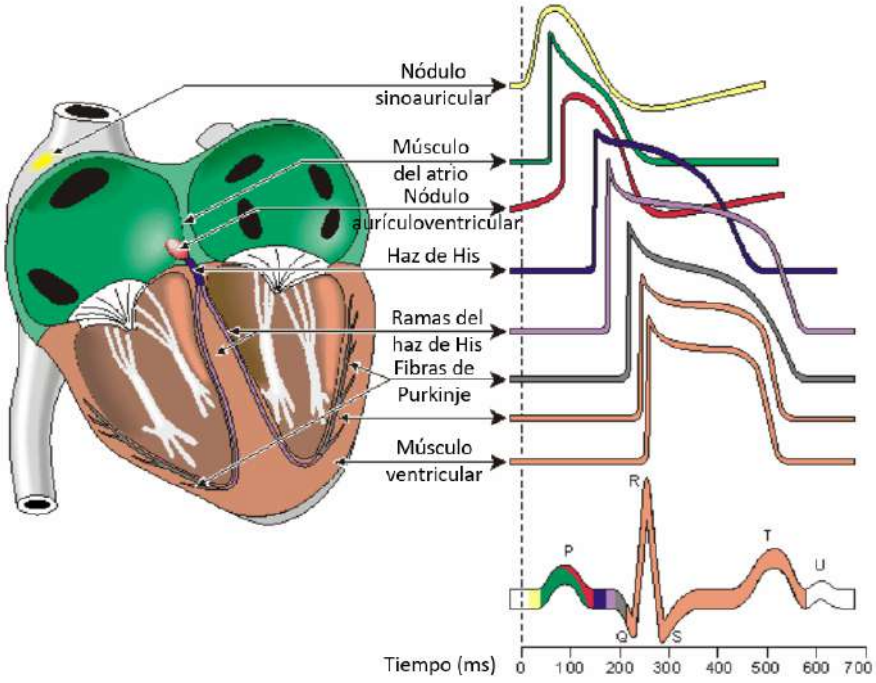



Figure 2.21: El latido del corazón es la superposición de los potenciales de acción de las células y músculos del corazón.

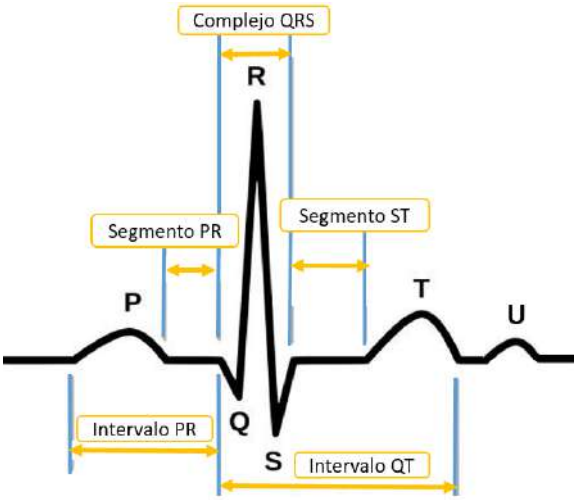



Figure 2.22: Forma de onda típica de un latido.

- Ocasionalmente aparece una onda U, mas pequeña, que sigue a la onda T y es del mismo signo. Cuando aparece de signo invertido es síntoma de alguna anomalía.

- El intervalo PQ indica el tiempo que ha pasado desde que se genera el estímulo en el nódulo SA hasta que llega al nódulo aurículo-ventricular o AV. El tiempo de propagación está entre $0.12s - 0.2s$.

- El intervalo QT es el tiempo transcurrido desde el inicio de la estimulación de los ventrículos hasta el final de la repolarización ventricular. Oscila entre $0.3s - 0.4s$.

La forma y polaridad de los lóbulos varía según la posición de los electrodos y según las derivaciones que se utilice con los electrodos.

La tabla 2.3 da una relación de las amplitudes y tiempos más importantes en un ECG de una persona adulta normal.

| | | |
|---|---|---|
| Amplitud (mV) | Onda P | 0.1 - 0.25 |
| | Onda Q | 25% onda R |
| | Onda R | 1 - 5 |
| | Onda T | 0.1 - 0.5 |
| Tiempo (s) | Onda P | 0.11 |
| | Intervalo PQ | 0.12 - 0.20 |
| | Intervalo QT | 0.35 - 0.44 |
| | Intervalo QRS | 0.05 - 0.09 |
| | Segmento ST | 0.05 - 0.15 |

Tabla 2.3: Amplitud y tiempo en un ECG

En la práctica la forma, amplitudes y tiempos del latido varían, puesto que los electrodos pueden ponerse en muchos puntos del cuerpo para registrar los potenciales generados en el corazón.

### 2.5.4 Derivaciones electrocardiográficas.

El cuerpo humano, por la gran cantidad de electrolitos que contiene, *actúa como volumen conductor*. Es decir, que *la actividad eléctrica del corazón, puede registrarse en cualquier otro punto del cuerpo por lejos que esté del punto origen*. Sin embargo, la posición de los electrodos en cada una de las derivaciones va a determinar el valor que el circuito registra.

Cuando se registra un latido suele ser normal utilizar varios o muchos electrodos colocados en el cuerpo. La señal obtenida al registrar el ECG depende de la localización de los electrodos y se encuentra normalizada. A cada par de electrodos o combinación entre ellos se le llama **derivación electrocardiográfica**. Hay diversas morfologías, y cada una de ellas da

una visión global de los fenómenos eléctricos que ocurren en los músculos del cuerpo. Y además es dependiente del ángulo en el que el electrodo enfoca al corazón.

Según la posición de los electrodos en el cuerpo se contemplan 12 derivaciones posibles en la práctica:

- 3 derivaciones bipolares.
- 3 derivaciones unipolar aumentada.
- 6 derivaciones unipolar precordiales

Para entender mejor las conexiones de los electrodos en el circuito, a continuación se introducen brevemente las conexiones de los electrodos en estas derivaciones.

### Derivaciones bipolares de Einthoven.

Son las derivaciones clásicas establecidas por W. Einthoven en el 1912, y se conocen por las siglas DI, DII y DIII. Las conexiones son diferenciales y se establecen acorde con la figura 2.23 y tabla 2.4.

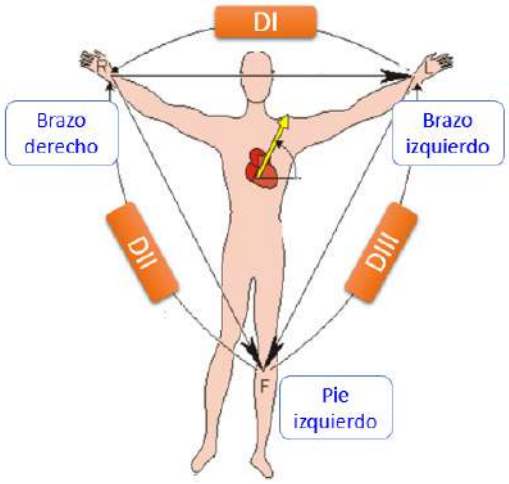



Figure 2.23: Derivaciones clásicas de W. Einthoven.

W. Einthoven postuló que:

- En un instante cualquiera de tiempo del ciclo cardíaco, la representación en el plano frontal del campo eléctrico del corazón es un vector de dos dimensiones.

| Derivación | Electrodo positivo | Electrodo negativo |
|---|---|---|
| I | Brazo izquierdo (LA) | Brazo derecho (RA) |
| II | Pierna izquierda (LL) | Brazo derecho (RA) |
| III | Pierna izquierda (LL) | Brazo izquierdo (LA) |

Tabla 2.4: Derivaciones clásicas de W. Einthoven

- Cada una de las derivaciones registra a una de las componentes unidimensionales de este vector.
- Los potenciales en los puntos del eje de un punto de referencia es muy parecido al tomado en este punto. Por ejemplo, que el potencial de la pierna difiere muy poco del potencial del pie.
- En un instante cualquiera del ciclo cardíaco, la medida en tensión de una de las derivaciones es aproximadamente igual a la suma algebraica de las otras dos dimensiones. Es decir, que (ecuación 2.10)

$$D.II = D.I + D.III \tag{2.10}$$

La figura 2.24 muestra un ECG utilizando la derivación bipolar de Einthoven. Se puede comprobar que se cumple la relación 2.10.

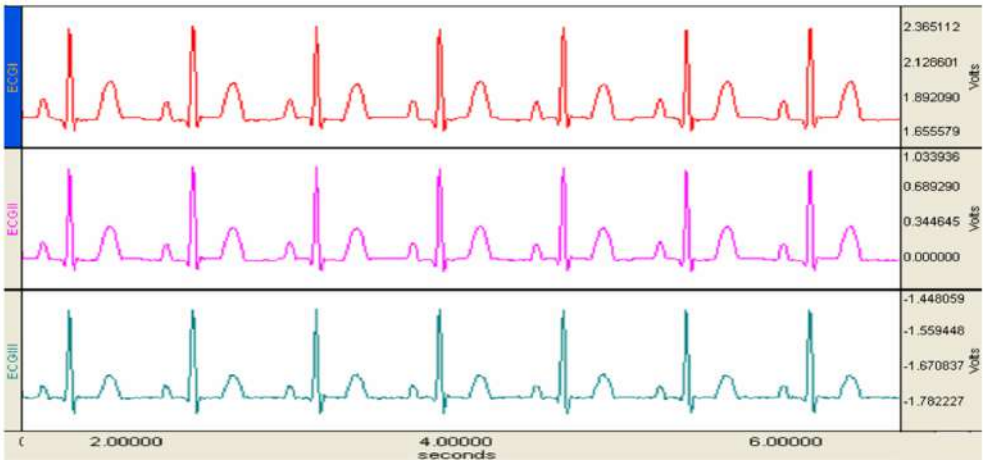


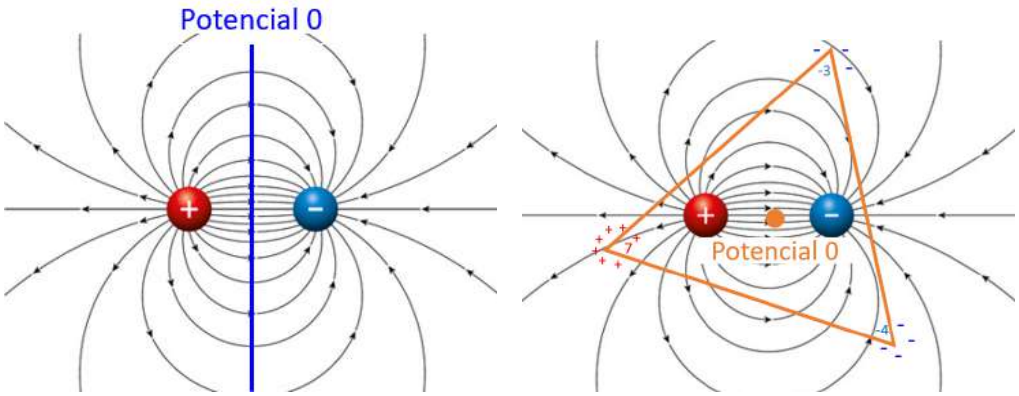

Figure 2.24: ECG con derivación clásica.

**Electrodo de referencia o terminal indiferente.**

En las derivaciones bipolares se conocen las posiciones anatómicas del electrodo explorador, que es el que registra el potencial. En las derivaciones

unipolares el electrodo explorador necesita de otro electrodo a potencial 0. Y en el cuerpo no existe un punto de potencial 0. Ello es posible resolverse mediante la teoría del triángulo.

Se sabe que en un dipolo (figura 2.25a), el potencial nulo se encuentra en las posiciones equidistantes de las cargas.




(a) Dipolo. (b) Teoría del triangulo.

Figure 2.25: Obtención del electrodo indiferente (potencial 0).

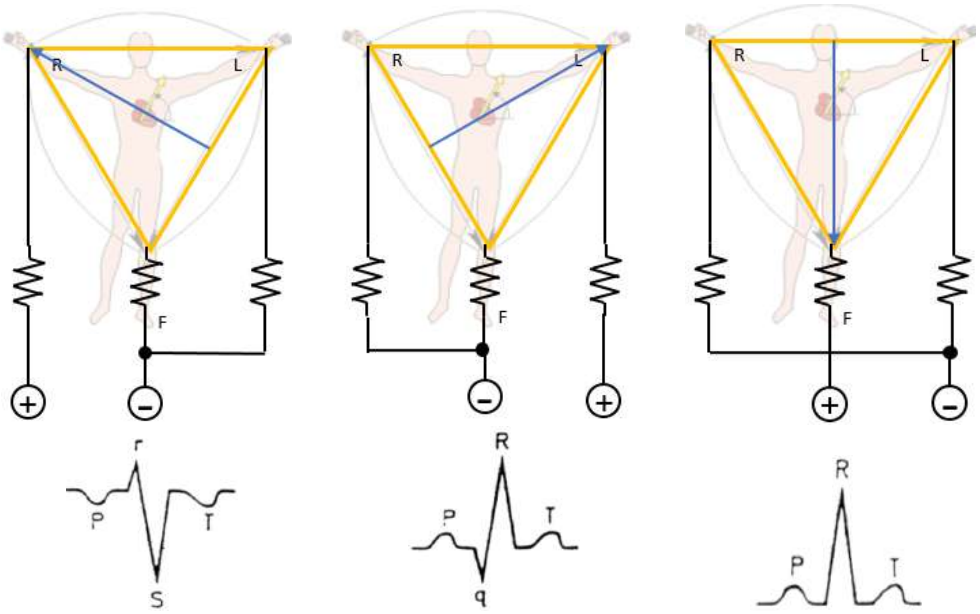
Cuando se sitúan tres electrodos en un conductor volumétrico formando un triángulo equilátero con su centro geométrico coincidente con el centro del dipolo (figura 2.25b) la suma de los potenciales de los electrodos será siempre nula. Si se rota el triángulo, los vértices irán pasando por distintas superficies equipotenciales, y la suma de sus potenciales continuará siendo 0. Este punto central es el **Terminal central de Wilson** en las derivaciones precordiales.

En la práctica, para obtener este punto central indiferente se va a situar una red de resistencias de manera que realice la suma algebraica de los tres vértices, que representan puntos anatómicos del cuerpo humano.

### Derivaciones unipolar aumentadas.

Las derivaciones unipolar aumentadas fueron introducidas por F. Wilson en 1932. Son tres derivaciones unipolar en las que los electrodos se sitúan en puntos distales formando un triángulo equilátero, en los antebrazos y en la pierna izquierda. Comparan el potencial del punto en que se coloca el electrodo explorador contra la suma de los potenciales de los tres miembros activos o *Terminal Central* (pierna izquierda y brazos derecho e izquierdo) que da como resultado 0 (figura 2.26). Debido al efecto de carga de la red de

resistencias para formar el terminal indiferente, los potenciales registrados están atenuados.




(a) aVR (b) aVL (c) aVF

Figure 2.26: Derivaciones unipolar aumentadas y deflexiones morfológicas

La denominación de las tres derivaciones toman el nombre del vértice del triángulo:

- aVR: voltaje aumentado en el vértice de la mano derecha.
- aVL: voltaje aumentado en el vértice de la mano izquierda.
- aVF: voltaje aumentado en el vértice de la pierna.

La morfología de las ondas en cada punto es función del vector de propagación de la onda que se genera. Así, por ejemplo, la onda R, que se positiva por aproximarse en la derivación aVL, es la onda R negativa en aVR, por alejarse del vértice.

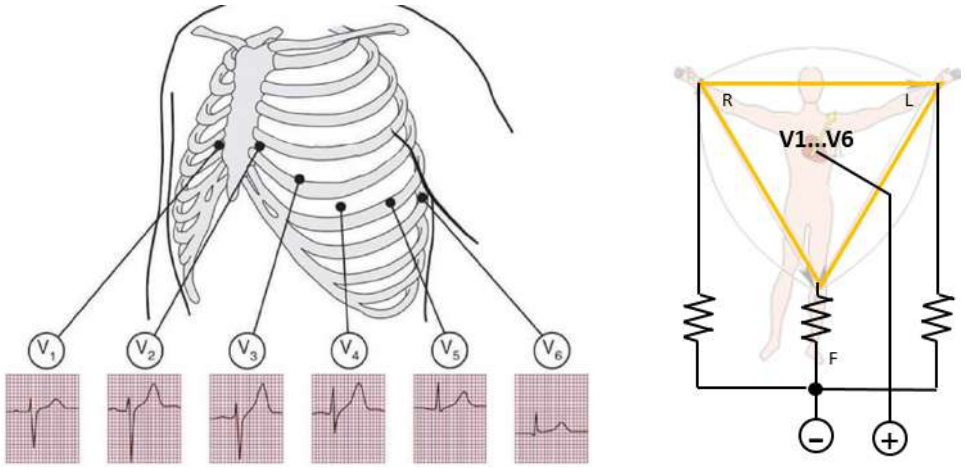
A nivel morfológico, las derivaciones aumentadas realizan un registro de los potenciales del corazón en el plano paralelo a él. Las derivaciones precordiales, que a continuación se comentan, van a realizar un registro de los potenciales en un plano perpendicular a él.

Existe una relación entre las derivaciones unipolar aumentadas y las derivaciones bipolares clásicas de Einthoven (ecuación 2.11):

$$\left.\begin{aligned} DI &= aVL - aVR \\ DII &= aVF - aVR \\ DIII &= aVF - aVL \end{aligned}\right\} \tag{2.11}$$

**Derivaciones unipolar precordiales.**

Son derivaciones unipolares que proporciona una visión perpendicular completa de los fenómenos que ocurren en el corazón. Para ello se han establecido internacionalmente unos puntos concretos para situar los electrodos (figura 2.27a). Se designan con la letra V y un número entre 1 i 6 que corresponde a cada posición anatómica.




(a) Puntos de las derivaciones precordiales. (b) Interfase electrodos.

Figure 2.27: Derivaciones precordiales.

- V1 y V2 registran los potenciales de la pared anterior y la pared interior del ventrículo derecho.
- V3 y V4 registran los potenciales de la región punta del corazón y la pared interventricular.
- V5 y V6 registran los potenciales presentes en la pared del ventrículo izquierdo.

Por lo tanto, el electrodo explorador va a registrar el potencial de acuerdo con el punto precordial. El electrodo indiferente se va a situar en el *Terminal central de Wilson* (figura 2.27b).

### 2.5.5 Interpretación del ECG.

El ECG es un método gráfico de comprobar el funcionamiento del corazón. Un método sistemático de interpretarlo consiste en determinar cuatro parámetros:

- La frecuencia.
  La **frecuencia auricular** se inicia con la actividad sinoauricular. Es decir, es el inicio de la onda P. El tiempo que pasa entre dos ondas P y P' consecutivas es el período. Entonces, la frecuencia auricular por minuto viene dada por (ecuación 2.12):

$$f_{auricular}/min = \frac{60}{distancia(P - P')(en\ s.)} \tag{2.12}$$

  Aunque normalmente al hablar de frecuencia cardíaca se suele asociar con la frecuencia ventricular que se detecta con el pulso. La **frecuencia ventricular** se determina siguiendo el mismo procedimiento pero con el complejo QRS. Está relacionada con la despolarización y repolarización ventricular.

- El ritmo.
  El **ritmo** indica la constancia en la pulsación. Como con la frecuencia, se puede calcular el ritmo auricular a partir de las ondas P o el ritmo ventricular a parir del complejo QRS. Entonces el ritmo se mida tomando sucesivamente los tiempos auriculares y/o ventriculares.

- La conducción.
  La **conducción** mide el tiempo que tarda el impulso desde que se inicia en el nódulo sinoauricular, pasando por la aurícula y el nódulo ventrículo-auricular, hasta que llega el ventrículo. Se obtiene midiendo el segmento PR y la duración del complejo QRS.

- La configuración y la localización.
  La interpretación de un trazo ECG se realiza siguiendo la generación del trazo y estudiando la forma y la localización de cada una de las ondas que se generan en al ECG: onda P, complejo QRS, segmento ST y onda T.
  En un humano normal, la simetría en la forma y localización de las ondas es la norma.

### 2.5.6 Diseño de un ECGafo: requerimientos y especificaciones.

En todo diseño de un sistema de instrumentación es requisito imprescindible prestar a atención a los ruidos e interferencias que puedan afectar la calidad de la señal que se mide. Esto es especialmente importante en un electrocardiógrafo donde se miden señales que pueden estar afectadas a muchas interferencias, aparte de las propias interferencias que la instrumentación que se acopla directamente al cuerpo humano pueda introducir.

#### Consideraciones al diseño, artefactos.

En medicina se utiliza el término *artefacto* para definir lo que en ingeniería denominamos normalmente *interferencia*. El **artefacto** es cualquier componente de una señal extraña a lo que representa la señal.

Una relación de los artefactos que podemos encontrar en un ECGafo es la siguiente.

- Artefacto debido al movimiento.
  Se ha comentado que la utilización de geles electrolíticos ayuda a la conducción de los iones hacia el electrodo. Según el tipo de ECG que se realice hay que impedir el desplazamiento de los electrodos de manera que exista contacto directo electrodo-piel. De producirse, se generaría un artefacto puesto que se producirían cambios en el potencial e impedancias del electrodo. Este artefacto puede ser muy importante en el caso de deportistas donde se analiza la actividad del corazón con el cuerpo en movimiento.

- Artefacto debido a señales EMG.
  Es un artefacto inevitable que siempre está presente en el ECG. De acuerdo con las especificaciones de frecuencia y amplitud presentadas en la tabla 2.2 se puede comprobar que este artefacto es inevitable en un ECG. Para limitar su influencia es imprescindible el uso de filtros.

- Artefacto debido al potencial de contacto.
  El electrodo, el gel electrolítico y la piel generan un potencial denominado **potencial de contacto**. Para los electrodos $Ag/AgCl$ es del orden de $0.22V$. Se produciría, entonces, un artefacto, en el caso que los dos potenciales de contacto no fuesen exactamente igual. Dado que la entrada al sistema electrónico se produce mediante un amplificador de instrumentación diferencial, al existir un diferencial en la entrada produciría una corriente continua que podría llegar a saturar el amplificador.

Para evitar el artefacto se puede realizar un acoplamiento en alterna del sistema en la etapa de entrada.

- Artefacto debido a la impedancia del electrodo.
  Los artefactos debido a cambios de impedancia en la interfase electrodo-gel-piel se mitigan con un diseño adecuado de las impedancias de la etapa de entrada, normalmente un amplificador en modo diferencial. Se requieren impedancias de entrada grandes, tanto del modo diferencial, como del modo común.

- Artefacto de inducción electrostática debida a la red de distribución eléctrica.

  Los problemas de interferencia con la red eléctrica son comunes en todos los equipos electrónicos.

  Por una parte el cuerpo humano se comporta como un conductor volumétrico con acoplamiento capacitivo con los conductores eléctricos, modelado con las capacidades $C1$ y $C2$ en la figura 2.28. Este acoplamiento induce una corriente en el paciente a través de las capacidades, creando una tensión en modo común, que puede ser de alrededor de $0.3V$, entre electrodos y masa del amplificador. Esta tensión se transforma en tensión diferencial si existe desequilibrio de impedancias entre los electrodos.

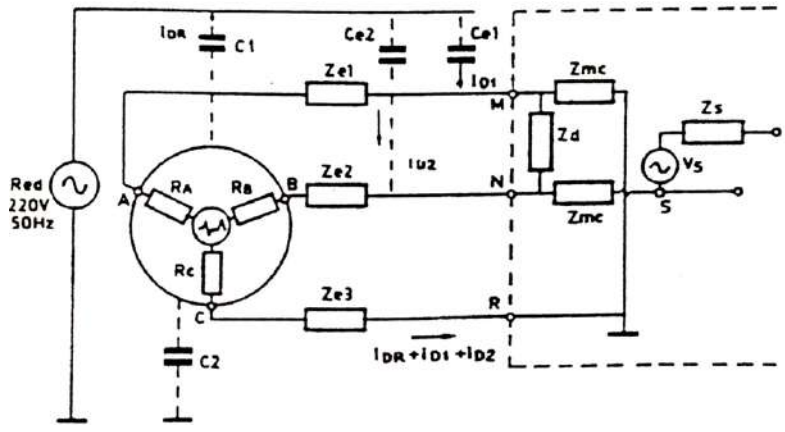



Figure 2.28: Acoplamiento de la red de tensión con el cuerpo humano.

Por otra parte, el cuerpo humano también es sensible a la frecuencia de 50 Hz con la que se distribuye la red eléctrica. Su amplitud puede llegar a ser más importante que la propia señal que se mide. Para evitar esta tensión, a los primeros pacientes con registro ECG se les conectaba

el pie izquierdo al terminal tierra con una impedancia muy pequeña. Con ello se minimizaban todas los voltajes creados por inducción electromagnética. Sin embargo, se corría el riesgo de poner en peligro al paciente por que, con la impedancia de la piel, los voltajes continuos creados podían provocar en grandes corrientes en el cuerpo.

Estos artefactos se pueden evitar con un diseño preciso de la etapa de entrada, por ejemplo, conectando el tercer electrodo en realimentación activa, reduciendo la interferencia de la señal en modo común y ampliando el margen de ruido.

Finalmente, el ruido de la red también puede llegar a través de los cables de alimentación. Se pueden inmunizar apantallando correctamente los cables.

- Artefacto de inducción electromagnética debida a la red de distribución eléctrica.
  La inducción electromagnética se produce siempre que exista un circuito cerrado por donde pasa corriente. Por consiguiente, se creará también con los cables de los electrodos. Pueden llegar a crear tensiones superiores a los $10\mu V$.

  Para eliminar su efecto es recomendable trenzar los cables de los electrodos hasta las proximidades del paciente.

- Artefacto debido al ruido electrónico del circuito de acondicionamiento. Todo dispositivo electrónico en un sistema de instrumentación tiene problemas de offset y ruido en todo el ancho de banda.

  Por ello es recomendable utilizar dispositivos de bajo ruido y filtrar las señales en el ancho de banda de interés.

**Requerimientos del ECGafo.**

Los requerimientos de un ECGafo son:

- La entrada se acopla al cuerpo humano. Por consiguiente, debe tener una impedancia de entrada en modo diferencial muy elevada.

- Se deben evitar los artefactos debidos a los electrodos. Se consigue con impedancias en modo común también grandes.

- Los biopotenciales son señales pequeñas, de pocos mV. Se debe conseguir la amplificación necesaria acorde con el margen dinámico de entrada del ADC.

- Respecto al ancho de banda:
  - La frecuencia de corte del filtro pasa-alta tiene que evitar el artefacto producido por el potencial de contacto en la interfase electrodo-piel.
  - La frecuencia de corte pasa-baja es necesaria para reducir el ruido total del sistema.
- Para evitar los acoplamientos capacitivos al sistema es conveniente que el amplificador tenga un CMRR alto.
- Finalmente, todos los elementos del sistema deben reducir al máximo el ruido electrónico.

Una cuantificación de todas estas recomendaciones se encuentra en la normativa de la *American Heart Association*.

**Especificaciones del ECGafo.**

De acuerdo con los requerimientos detallados en el apartado precedente, se detallan las siguientes especificaciones (tabla 2.5) para el ECGafo.

| # | Especificación | Rango |
|---|---|---|
| 1 | Ganancia | $60dB$ |
| 2 | CMRR | $100dB\ a\ 50Hz$ |
| 3 | Ancho de banda | $0.02 - 500Hz$ |
| 4 | Impedancia en modo diferencial | $> 20M\Omega\ a\ 50Hz$ |
| 5 | Impedancia en modo común | $> 100M\Omega\ a\ 50Hz$ |
| 6 | Nivel de ruido a la entrada | $< 1\mu V$ |

Tabla 2.5: Especificaciones del ECGafo.

### 2.5.7 Diseño de un ECGafo: arquitectura.

*Nota: Puesto que se trata de un apartado más técnico a nivel electrónico, se puede saltar sin perder conexión en la lectura del tema.*

A continuación se describen los aspectos técnicos del diseño del circuito electrónico de registro de un canal ECG. Para múltiples canales puede repetirse la arquitectura descrita. Asimismo, se recomienda la lectura del capítulo 7 de F. X, Villasevil [1] para los detalles técnicos de este diseño.

Los requerimientos que se imponen es este sistema son más exigentes de lo que se esperaría en un ECGafo normal. Ello es debido a que este sistema está pensado para el análisis del ritmo cardíaco en deportistas, donde el movimiento introduce artefactos que no se producen en pacientes estáticos.

La circuitería que compone el canal corresponde a la etapa de acondicionamiento de la señal ECG. Un ADC en la etapa de salida se encarga de digitalizar la señal ECG para posteriormente ser monitorizada y/o tratada en el procesador.

La figura 2.29 muestra el diagrama de bloques del diseño del canal ECG. Se compone de los siguientes elementos:

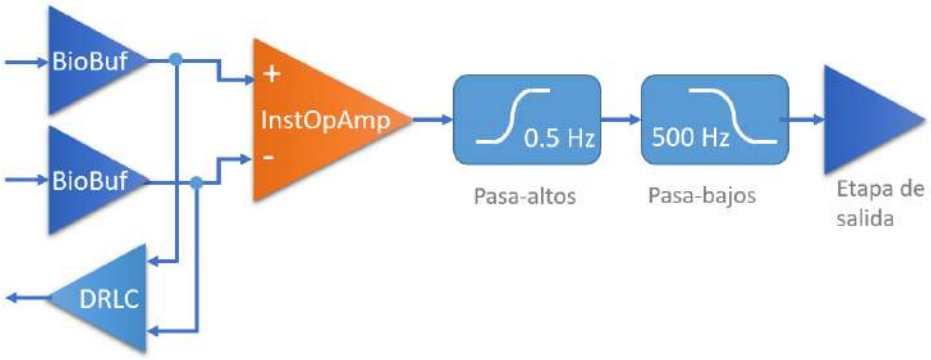



Figure 2.29: Arquitectura de un canal ECG.

**Etapa de entrada, denominada *BioBuffer*.**

Realiza las funciones de:

- Acoplamiento en AC para evitar saturación debida al potencial de contacto en la interfase electrodo-piel. Con ello se aprovecha para imponer la especificación pasa-alto (3) de ancho de banda.
- Debe tener la impedancia de entrada muy elevada en modo común, para evitar la variabilidad del electrodo (especificación (5)).
- También debe cumplir con la especificación (6) para evitar amplificaciones de ruido en etapas posteriores.

La figura 2.30 muestra un diseño del BioBuffer que cumple con las tres especificaciones de diseño:

- Acoplamiento en AC: Frecuencia de corte inferior a $-3dB\ a\ 0.02Hz$.

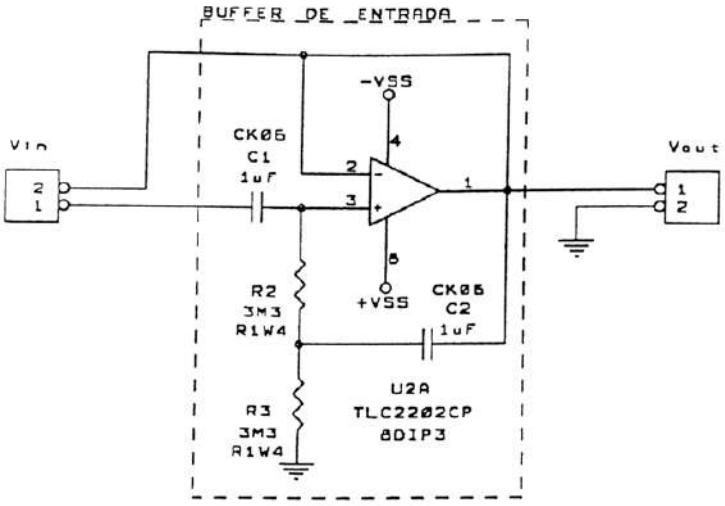



Figure 2.30: BioBuffer.

- Impedancia de entrada muy elevada: $100M\Omega\, a\, 0.02Hz$.
- Nivel de ruido inferior a $1\mu V$.

**Circuito de realimentación activa.**

El circuito DLRC o *Driven-Rigth-Leg Circuit* es un sistema de realimentación activa para reducir la tensión en modo común que puede adquirir el cuerpo por la presencia de la red de distribución eléctrica que, como se ha comentado, puede introducir una tensión diferencial en el sistema. Esta reducción de la tensión en modo común permitirá al sistema tener un buen CMRR (especificación 2). El circuito limita la corriente inyectada sobre el paciente a valores seguros y proporcionales al valor de la señal en modo común presente en el paciente.

La figura 2.31 muestra todos los componentes que integran el módulo DLRC. Puede observarse que:

- Las resistencias $R_a$ proporcionan la tensión en modo común presentes en el cuerpo. A través de un tercer electrodo esta corriente es realimentada hacia el cuerpo.
- El sistema utiliza dos tierras. Esto permite que el sistema pueda ser alimentado independientemente de la masa de la red. En la figura, las capacidades $C1$ y $C2$ modelan las capacidades parásitas de la tensión de alimentación.

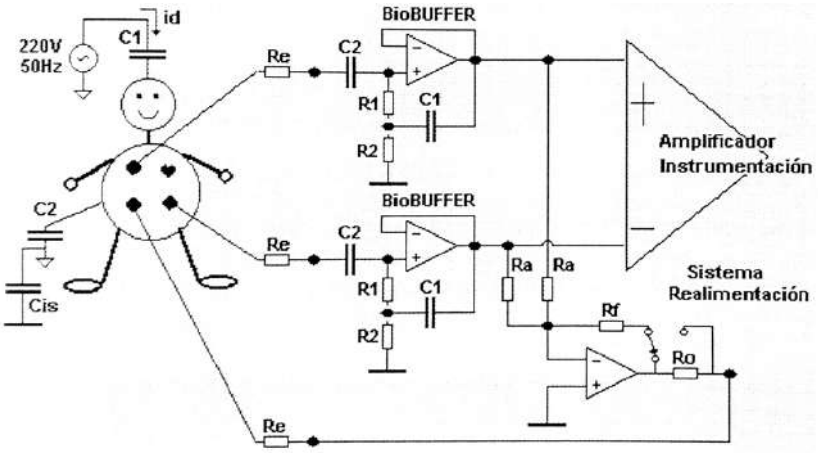



Figure 2.31: Circuito DLRC.

**Amplificador de instrumentación.**

El amplificador de instrumentación 2.32 es el encargado de realizar la amplificación de la señal ECG que llega al circuito. Tiene que cumplir con la especificación (1).

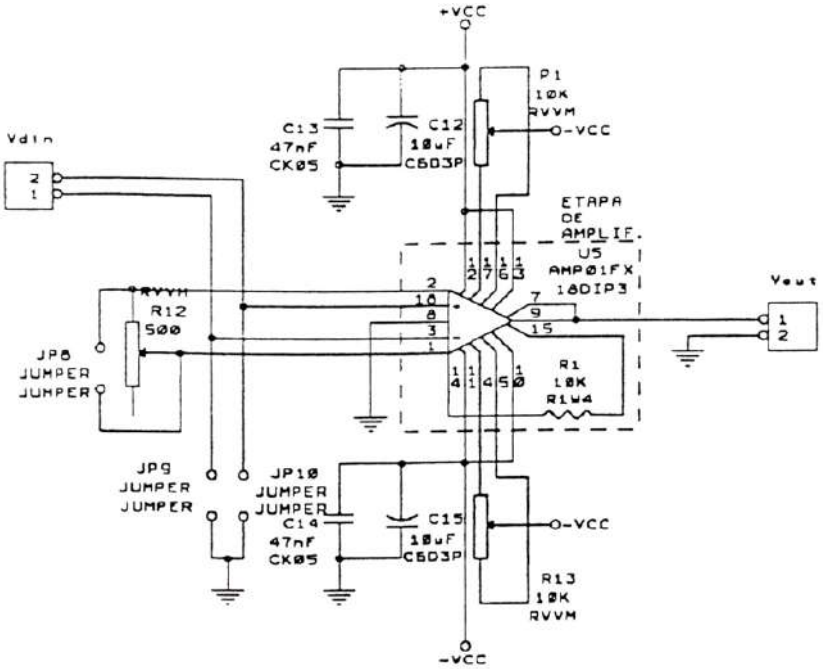
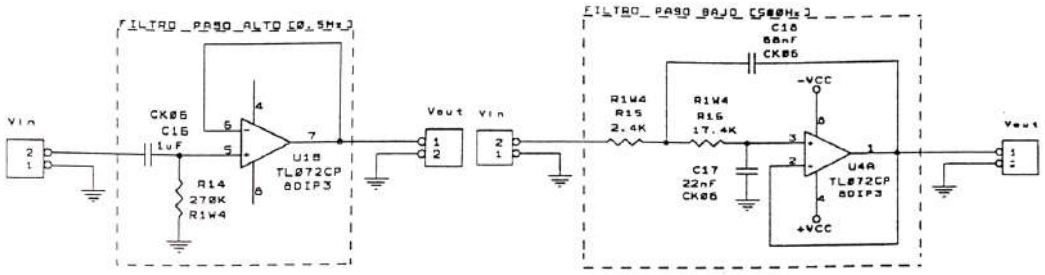



Figure 2.32: Amplificador de instrumentación.

**Filtro pasa-altas.**

En algunos pacientes (y en especial en pruebas de esfuerzo en deportistas) la respiración puede producir un artefacto con señales de frecuencia inferiores a 0.5 Hz. Para minimizar esta señal se introduce un filtro pasa-altos de 0.5 Hz (figura 2.33a).




(a) Filtro pasa-altas. (b) Filtro pasa-bajas.

Figure 2.33: Filtros de señal.

**Filtro pasa-bajas.**

Se introduce de acuerdo con la especificación (2) de limitar las señales por encima de 500 Hz para reducir el ruido total del sistema. La figura 2.33b muestra el circuito pasa-bajas.

**Etapa de salida.**

Es para adaptar la señal ECG al rango de entrada del ADC. Por tanto, en esta etapa sólo se impone que el señal ECG no sea modificado y que permita el ajuste de un offset de salida de forma regulable. El circuito de la figura 2.34 cumple con esta función.

**Resultados.**

La figura 2.35 muestra un ECGafo de la derivación II tomado con el circuito descrito en esta sección. En la forma de onda se observan de modo muy nítido las ondas del estímulo.

Se aprovecha la electrónica del canal realizada para realizar automáticamente también el conteo del pulso.

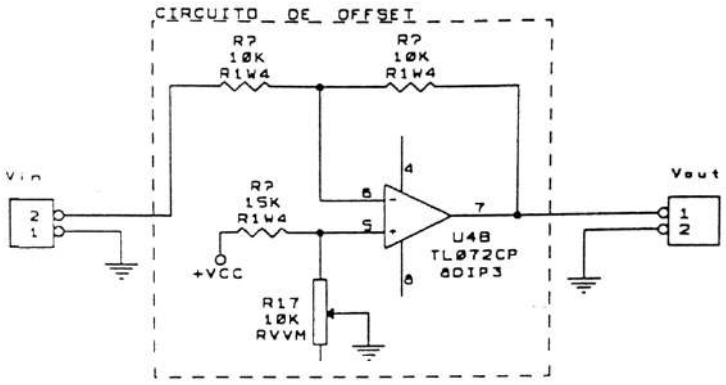



Figure 2.34: Circuito DLRC.

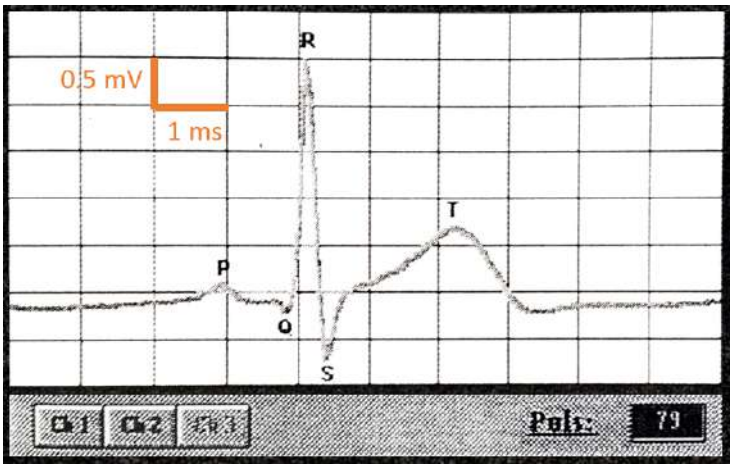



Figure 2.35: ECGafo derivación II.

## 2.6 Imagen médica

Cuando se habla de imagen médica se entiende un conjunto de técnicas y procesos usados para crear imágenes del interior del cuerpo o técnicas funcionales, con propósitos clínicos o para la ciencia médica. Muchas de ellas toman el nombre genérico de radiología por utilizar técnicas radiantes para excitar el mecanismo que permite sensibilizar la imagen.

El conjunto de técnicas existente es enorme. Sólo por citar algunas:

- Técnicas estructurales o morfológicas:
  - Radiografía. Es una técnica radiológica de diagnóstico por imagen basada en rayos X que trata de auxiliar en el diagnóstico y

pronóstico del estado de salud y la enfermedad a través del uso de tecnologías de análisis de la imagen.

- Tomografía computarizada (TC). La tomografía, inicialmente denominada TAC, es la obtención de una imagen de corte o seccion del cuerpo. La posibilidad de obtener imágenes de cortes tomográficos reconstruidas en planes no transversales, ha hecho que en la actualidad se prefiera llamar a esta técnica *tomografía computarizada* o TC.
- Resonancia magnética (RM). La resonancia magnética utiliza la radiación absorbida y emitida por los núcleos atómicos para obtener información sobre sus propiedades magnéticas. En medicina se utiliza como herramienta espectroscópica para obtener datos físicos y químicos de la composición del cuerpo, lo que hace que sea una herramienta de análisis y diagnóstico altamente utilizada para conocer de la salud de la persona.

- Técnicas funcionales o moleculares:
  - Ecografía. La ecografía es un procedimiento para obtener imágenes a partir de los ecos generados en una emisión de ultrasonidos dirigida sobre un cuerpo. Es una fuente de datos con las que se forma una imagen de los órganos o masas internas con fines de diagnóstico de la salud. Es un procedimiento que no utiliza radiacion, por lo que es inocuo,.
  - SPECT o Single Photon Emission Computed Tomography. Es un tipo de medicina nuclear, radiotrazadora, capaz de medir la actividad metabólica del cuerpo. Es una técnica similar a una radiografía, pero la fuente de radiación son rayos gamma (radiofármaco) provenientes de la desintegración de un elemento radioactivo (tecnecio 99) introducido en el cuerpo por vía endovenosa, y no una fuente exterior.
  - Positron Emission Tomography o PET. Técnica radiotrazadora que mide la actividad metabólica del cuerpo. Es medicina nuclear similar a la SPECT, pero el radiofármaco genera positrones que se aniquilan al chocar con un electrón, produciendo dos rayos gamma en direcciones opuestas que son los que se detectan. La detección de los dos rayos gamma permite una mayor precisión que el SPECT. Los radiofármacos tienen una vida ultracorta y, por tanto, son más difíciles de manipular. En cuanto a sus usos, por ejemplo, permite medir el consumo de glucosa.

Desde el el punto de vista de información para la creación de la imagen médica, todas estas técnicas crean la imagen a partir de una o múltiples tomas, por radiación, que aportan datos del interior del cuerpo. En una radiografía la técnica de tratamiento de la imagen, se puede decir, que es inmediata, puesto que la técnica crea una única imagen utilizando la radiación de rayos X. Sin embargo, en técnicas como la TA, la RM, o la PET, por citar algunas, las imágenes se obtienen a partir de múltiples tomas por radiación que aportan información del cuerpo visto desde distintos ángulos. Entonces, el procesado de datos puede llegar a ser bastante laborioso.

En este capítulo, por brevedad, se van a introducir la radiografía como técnica clásica del tratamiento de datos en imagen médica y la tomografía computarizada como ejemplo del tratamiento de datos a partir de múltiples tomas que se requiere para la obtención de la imagen.

### 2.6.1 Rayos X.

La radiación electromagnética es la emisión de energía en forma de ondas (o partículas) que puede propagarse tanto en medios materiales como en el vacío. Como se muestra en la figura 2.36, es la superposición de los campos eléctrico y magnético, comportándose como ondas sinusoidales, perpendiculares y en fase. Sus propiedades son:

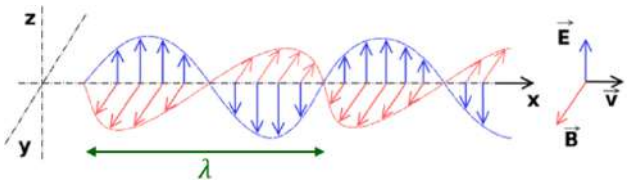



Figure 2.36: La radiación electromagnética es superposición de los campos eléctrico y magnético.

- Tiene longitud de onda $\lambda$, en $m$.
- La frecuencia de la onda es $\nu = c/\lambda$, en $Hz$.
- La energía de la onda, denominada *radiante*, es $E = \hbar \cdot \nu$, en $J$, y donde $\hbar$ es la constante de Planck, cuyo valor es $\hbar = 6.63 \cdot 10^{-34}\ J \cdot s$.

Se clasifica en función del valor de su frecuencia. La figura 2.37 muestra el espectro de la radiación electromagnética. La pequeñísima parte marcada con colores es la única parte visible para el ojo humano. Conforme nos movemos de derecha a izquierda en la figura, la frecuencia aumenta y, por ende, la

radiación es más energética. A mayor energía de radicación, mayor capacidad de inducir mutaciones en la célula. La *radiación ionizante* (radiación de alta energía capaz de desplazar a un electrón de su átomo y provocar así su ionización) comienza en en la zona del ultravioleta (UV).

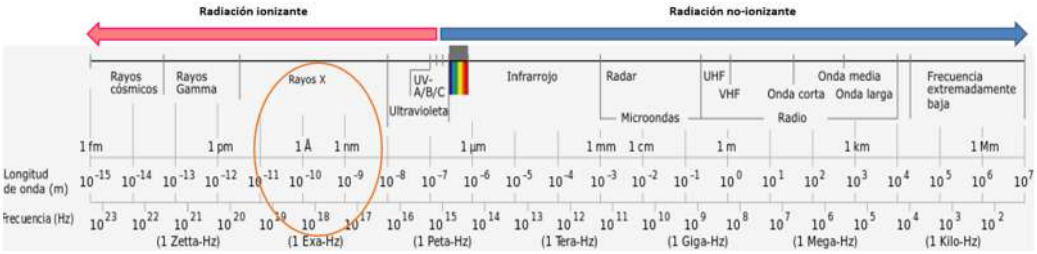



Figure 2.37: Espectro electromagnético de la zona de rayos X.

Los rayos X forman parte del conjunto de técnicas que utilizan la radiación electromagnética para generar una imagen interna del cuerpo. Fueron aplicados por primera vez por W.G. Röntgen en 1895, ingeniero alemán, mecánico y físico. Creó el instrumento capaz de generar y detectar una radiación electromagnética de longitudes de onda entre los 10 a 0.01 nm., lo que corresponde a frecuencias en el rango de los 30 a 30.000 PHz ($1\ peta = 10^{15}$).

La imagen de la figura 2.38 es la primera radiografía que tomó Röntgen. Corresponde a la mano izquierda de su esposa, en el que se ve claramente el anillo que llevaba en el dedo anular.

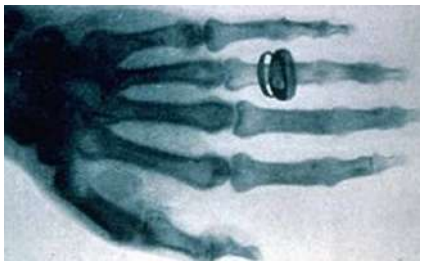


Figure 2.38: Primera radiografía tomada con rayos X.

La aplicabilidad de los rayos X en medicina radica en que son muy penetrantes. Pueden atravesar el cuerpo humano. Pero al mismo tiempo, al atravesar un material, son parcialmente absorbidos por éste dependiendo de sus propiedades. Por tanto, mediante rayos X se puede conocer el estado de la materia que lo ha absorbido. Y por ello es también una técnica peligrosa en elevadas dosis de radiación.

La tasa de absorción de los rayos X al atravesar un material depende de:

- El material en sí (número atómico). A mayor número atómico (más masivo), mayor absorción..
- La densidad (peso/volumen) del material. A mayor densidad, mayor absorción.
- El espesor del material. A mayor espesor, mayor absorción.
- La energía de los rayos X. A mayor energía mayor absorción.

Así, por ejemplo, de mayor a menor absorción tenemos los materiales: metal, calcio (hueso), tejidos blandos, agua, grasa, aire.

**Física de los rayos X.**

Los rayos X se generan utilizando un campo eléctrico entre dos polos entre los que crea una corriente continua. Los polos están formados por:

- Un filamento, que forma el cátodo, y que emite electrones. Debido a la temperatura a la que trabaja los filamentos suelen ser de molibdeno o wolframio.
- Una placa metálica sometida a gran temperatura que forma el ánodo, y que es bombardeado por los electrones a gran velocidad.

Al chocar contra los átomos del ánodo los electrones se frenan drásticamente y pierden parte de su energía.

- Una parte de esta energía se libera en forma de calor. Por esto el ánodo debe girar continuamente para que el metal no se funda.
- De otra parte se libera energía en forma de radiación electromagnética en la zona de los rayos X.

Para evitar que los electrones choquen con otros átomos antes de llegar al cátodo, todos los componentes se encierran en cámaras al vacío.

**Formación de la imagen.**

La figura 2.40a esquematiza el funcionamiento general de un aparato de rayos X. Está compuesto por el emisor de rayos X y una pantalla receptora. La persona se pone entre ambas. La emisión de rayos X atraviesa la persona en diferentes grados según el tejido. La pantalla captura los rayos X que no son absorbidos por los tejidos. Se detectan o bien sobre una placa fotográfica

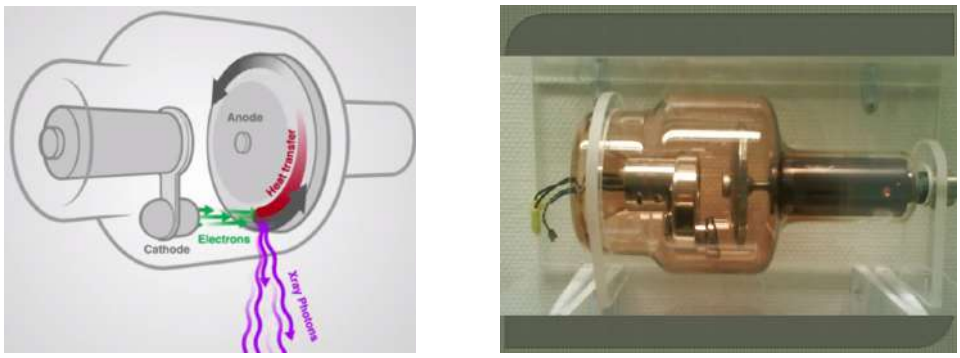
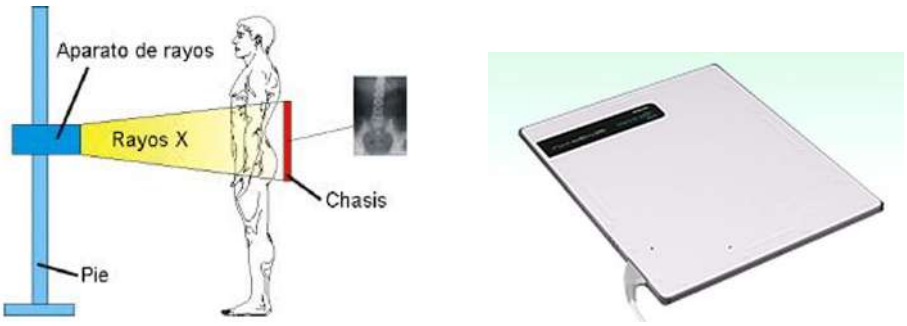



Figure 2.39: Generación de los rayos X. La emisión de electrones se efectúa en el cátodo. El ánodo lo forma una placa metálica giratoria.




(a) Pantalla de rayos X clásica (b) Panel sensor plano en radiología digital

Figure 2.40: Formación de la imagen en rayos X.

(fluorescencia) o bien, actualmente, en un detector digital (sensores CCD o CMOS) (figura 2.40b).

La **fluorescencia** es un tipo particular de *luminiscencia* (emisión de luz por parte de un cuerpo) que caracteriza a las sustancias que son capaces de absorber energía de radiación electromagnética y luego emitir parte de esa energía en radiación electromagnética con longitud de onda diferente. En este sentido y en otros campos, la espectroscopia de fluorescencia de rayos X es un método común que utiliza la fluorescencia de rayos X para el análisis de materiales.

En la película de rayos X suelen utilizarse halogenuros de plata. Son cristales de bromuro de plata en un 95% y el restante de yoduro de plata. Son compuestos que tienen un número atómico elevado que es lo que hace que los rayos X más los fotones de luz procedentes de las pantallas reaccionen con ellos y den lugar a la formación de la imagen. Estas sustancias fluorescentes

forman una capa uniforme sobre un soporte polimérico (similar a las películas fotográficas de antaño) . La energía cedida por la radiación se traduce en una imagen latente, que requiera de un proceso de revelado.

Actualmente se utilizan paneles con sensores capaces de detectar la radiación electromagnética que reciben. En un único paso se realiza tanto la exposición a los rayos X como la lectura de la radiación recibida, de modo que la imagen se puede hacer visible inmediatamente en una pantalla (monitor). Los sensores son muy eficientes (detectan pequeñas cantidades de rayos X), lo que permite disminuir la dosis de radiación que recibe el paciente. En radiología digital la imagen acaba siendo un fichero de tamaño entre los 5-9 Mpixeles, y creciendo a medida que se utilizan sensores con más resolución.

La figura 2.41 muestra tres ejemplos de radiografías. La imagen 2.41a es una radiografía de la zona pulmonar. Por comparación, puede observarse en 2.41b como queda expuesta la radiografía en caso de padecer edema pulmonar. La imagen 2.41c muestra el caso de un *tumor pardo* en un dedo, un tumor óseo que se produce como consecuencia de un exceso de destrucción ósea. En todas las radiografías se observa claramente como el hueso (calcio) absorbe mejor la radiación (se ve más claro) respecto a los tejidos más blandos (que son más oscuros).

(a) Zona pulmonar (b) Edema pulmonar (c) Tumor pardo

Figure 2.41: Ejemplos de radiografías.

### 2.6.2 Tomografía computarizada.

**Escáneres.**

La **tomografía computarizada (TC)** es una técnica de imagen médica que utiliza rayos X para obtener cortes o secciones de objetos anatómicos con fines diagnósticos. Sus bases matemáticas fueron planteadas en 1917 por Johann Radon.

En lugar de obtener una imagen de proyección, como las radiografías convencionales, la TC obtiene múltiples imágenes al efectuar la fuente de

rayos X y los detectores de radiación movimientos de rotación alrededor del cuerpo. La representación final de la imagen tomográfica se obtiene mediante la captura de las señales por los detectores y su posterior proceso mediante algoritmos de reconstrucción.

La figura 2.42 muestra el tomógrafo computarizado *Aquilion One* de Toshiba, uno de los más avanzados. El anillo de la máquina es una fuente de rayos X con escáneres que tienen alrededor de 64 filas de detectores. Cada detector registra una imagen y, a continuación, el anillo gira para obtener una visión del paciente desde otro punto de vista (figura 2.43a). El escáner avanza de forma helicoidal (figura 2.43b). Las imágenes obtenidas (centenares) se procesan computacionalmente para dar una visión 3D (en distintas cortes o *slices*) del sujeto. El Aquilion ONE realiza 320 cortes. Los detectores de rayos X actuales están basados en semiconductores.

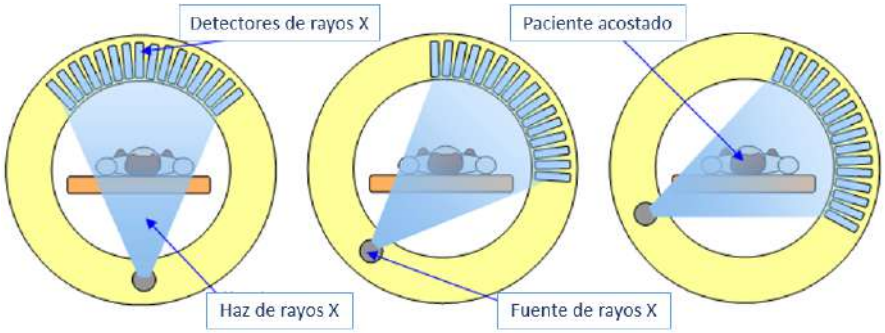
Figure 2.42: Tomógrafo computado Aquilion One (Toshiba).

**Reconstrucción de la imagen.**

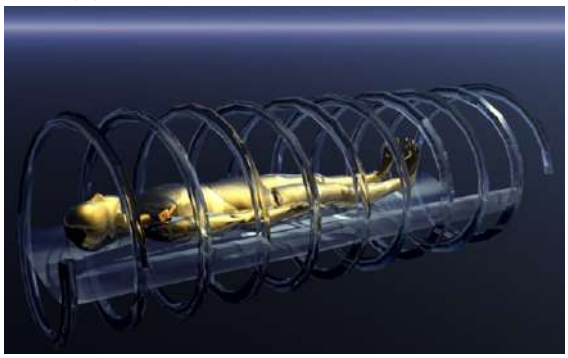
Existen muchas técnicas de reconstrucción de la imagen. Por simplicidad aquí se van a presentar el método *directo*, que es poco eficiente, y el de *back-propagation*.

**Método directo.**

Los diferentes tejidos absorben de forma distinta los rayos X. Los detectores miden la *cantidad* de radiación que consigue atravesar la sección bajo análisis y la convierten en una señal eléctrica que llega a un ordenador. Se ha introducido en el capítulo 1 que la imagen se dividía en píxeles. En el




(a) Movimiento circular del tomógrafo.



(b) Avance helicoidal del tomógrafo.

Figure 2.43: Funcionamiento del tomógrafo computarizado.

tomógrafo, cada píxel representa un trozo de imagen tridimensional, por lo que se le va a denominar **vóxel** (figura 2.43b), proveniente del inglés *volumetric pixel*. Entonces a cada vóxel se le va a asignar un nivel de gris acorde con la radiación recibida.

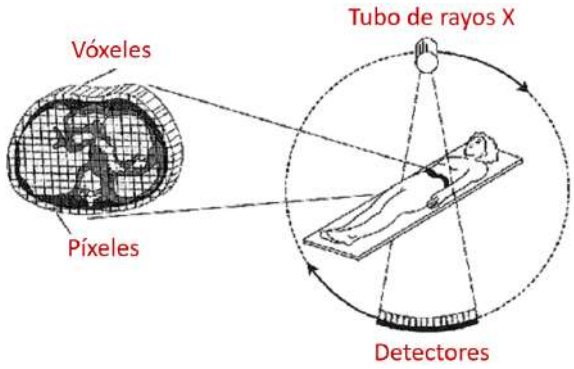


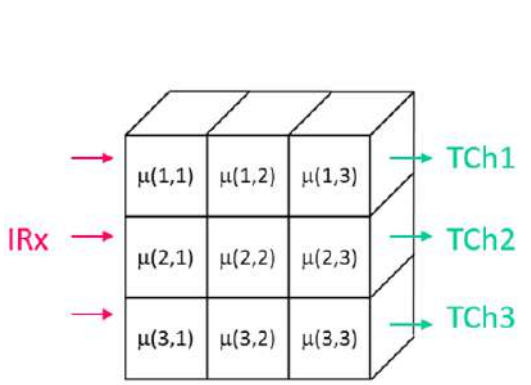

Figure 2.44: Discretización del cuerpo en *vóxeles*.

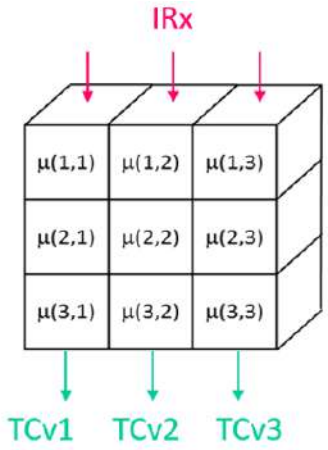
Para entenderlo, imaginemos la simplificación de un cuerpo de 9 vóxeles,

como el mostrado en la tabla 2.6, cada uno de ellos con un coeficiente de atenuación $\mu_{i,j}$. Se supone por ahora que la radicación en rayos X incidente $IRx$ es horizontal. Esto quiere decir que el vóxel (1,1) deja pasar una parte 1-$\mu_{1,1}$ de la radiación que incide sobre él, es decir, $(1-\mu_{1,1})\cdot IRx$. El vóxel (1,2) deja pasar una parte $(1-\mu_{1,2})$ de la radiación que incide sobre él, es decir, $(1-\mu_{1,2})\cdot(1-\mu_{1,1})\cdot IRx$. Lo mismo se aplica con el vóxel (1,3), con lo que, finalmente, la radiación que habrá logrado traspasar los 3 vóxeles situados en la fila 1 vendrá dada por la ecuación 2.13.

$$(1-\mu_{1,1})\cdot(1-\mu_{1,2})\cdot(1-\mu_{1,3}) = \frac{TCh1}{IRx} \tag{2.13}$$

Repitiendo el mismo proceso para los tres vóxeles de la segunda línea y luego para los de la tercera línea, se obtiene un sistema el sistema de 3 ecuaciones con 9 incógnitas, de la columna izquierda de la tabla 2.6.




| Radiación horizontal incidente: | Radiación vertical incidente: |
|---|---|
| $(1-\mu_{1,1})\cdot(1-\mu_{1,2})\cdot(1-\mu_{1,3}) = TCh1/IRx$ | $(1-\mu_{1,1})\cdot(1-\mu_{2,1})\cdot(1-\mu_{3,1}) = TCv1/IRx$ |
| $(1-\mu_{2,1})\cdot(1-\mu_{2,2})\cdot(1-\mu_{2,3}) = TCh2/IRx$ | $(1-\mu_{1,2})\cdot(1-\mu_{2,2})\cdot(1-\mu_{3,2}) = TCv2/IRx$ |
| $(1-\mu_{3,1})\cdot(1-\mu_{3,2})\cdot(1-\mu_{3,3}) = TCh3/IRx$ | $(1-\mu_{1,3})\cdot(1-\mu_{2,3})\cdot(1-\mu_{3,3}) = TCv3/IRx$ |

Tabla 2.6: Cálculo de la radiación incidente por vóxel.

Repitiendo el mismo proceso para la radiación vertical se obtienen las ecuaciones de la columna derecha de la tabla 2.6.

Finalmente, repitiendo también el proceso para la radiación en diagonal se obtiene un sistema de 9 ecuaciones con 9 incógnitas de fácil resolución.

Resolviendo el sistema de ecuaciones se identifican los coeficientes de atenuación de cada vóxel que, a su vez, se correlacionan con el tipo de material

(hueso, partes blandas, agua, aire,...) del que está formado el material. Basta ahora con *pintar* cada vóxel del color que se haya asignado al coeficiente y se obtiene una imagen de la estructura interna del cuerpo (9 vóxeles en este ejemplo) bajo estudio.

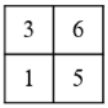
Siguiendo con el proceso, también se debería sumar la incidencia de los rayos a 45º, 135º etc.

Se observa que el método que se sigue cuando se aumenta el número de direcciones del que se hace la tomografía aumenta considerablemente. Dado el elevado número de vóxeles que se deberían analizar para una imagen precisa de sección del cuerpo humano es un método poco eficiente.

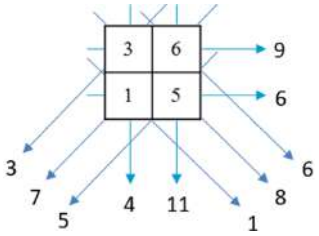
**Método de *Backpropagation*.**

Para mejorar la eficiencia del algoritmo de reconstrucción existen métodos computacionalmente más eficientes como el de *Backprojection* o *propagación retrógrada*.

Para entenderlo de forma simple se va a suponer que se tiene una matriz de 2x2 vóxeles con los valores de atenuación calculados a partir de la tomografía de la figura 2.45a. Estos son los valores de atenuación de los vóxeles que se han calculado a partir de los valores tomados de las distintas proyecciones (valores que se calculan a partir de las medidas de los detectores) que se muestran en la figura 2.45b..




(a) Atenuación en los vóxeles.




(b) Atenuación de las proyecciones.

Figure 2.45: Modelo de 2x2 vóxeles.

Entonces, partiendo de un factor de atenuación nulo por vóxel, se aplica el algoritmo simple a nivel computacional de la tabla 2.7 que devuelve los valores de atenuación de cada vóxel.

Una vez obtenidos los coeficientes de atenuación de cada vóxel, la imagen de cada sección se forma asignando informáticamente niveles de grises a cada vóxel. Se hace una asignación de nivel de gris ateniendo al coeficiente de atenuación del tejido medido en **unidades Hounsfield o HU**,

El coeficiente de atenuación en unidades Hounsfield se obtiene de aplicar a la ecuación 2.14 el coeficiente de atenuación de la radiación obtenida del

| | |
|---|---|
| Se parte de un valor de atenuación 0 en cada vóxel. | 0 0<br>0 0 |
| A cada vóxel de la proyección 0º se le suma la atenuación de dicha proyección | 9 9<br>6 6 |
| Ídem con la proyección -45º | 17 15<br>7 14 |
| Ídem con la proyección -90º | 21 26<br>11 25 |
| Ídem con la proyección -135º | 24 33<br>18 30 |
| A continuación, se resta la suma de las atenuaciones de la proyección 0º (que vale 15) | 9 18<br>3 15 |
| Y finalmente se divide por el número de proyecciones menos 1 (que es 4-1=3) | 3 6<br>1 5 |

Tabla 2.7: Especificaciones del ECGafo.

tejido, y en la que $\mu_{agua}$ es el coeficiente de atenuación del agua, y $\mu_{aire}$ es el del aire que es prácticamente 0.

$$HU = 1000 \cdot \frac{\mu_{tejido} - \mu_{agua}}{\mu_{agua}} \tag{2.14}$$

La tabla 2.8 muestra valores típicos de asignación de unidades HU a colores en los tejidos del cuerpo.

En resumen, tanto los rayos X como la tomografía computarizada se basan en la capacidad de penetración de los rayos X y en la distinta atenuación que sufren éstos al pasar por los tejidos. Pero a diferencia de la imagen plana de la radiografía, la tomografía computarizada visualiza cortes del cuerpo humano o del segmento analizado. Los tomógrafos computarizados suelen trabajar con alrededor de 40 cortes de 5-8 cm de grosor, y con una matriz de 512x512 pixeles.

La cantidad de memoria necesaria para almacenar un TC medio es de alrededor de 14,5 Mbytes. Depende de la resolución del tomógrafo y del

| | Unidades HU | Color |
|---|---|---|
| Hueso | 400 a 1000 | Blanco |
| Órganos no-huecos | 40 a 80 | Niveles de grises |
| Sangre | 30 a 45 | |
| Agua | 0 | |
| Grasa | -60 a -100 | |
| Tejidos blandos | -100 a -300 | |
| Pulmones | -400 a -600 | |
| Aire | Nivel de ruido a la entrada | Negro |

Tabla 2.8: Unidades Hounsfield de diferentes tejidos.

número de bits para representar los niveles de grises. Y para la obtención de la imagen se necesita la aplicación de algoritmos de reconstrucción.

# Bibliografía

1. A.G. Webb. *Principles of Biomedical Instrumentation*. Cambridge University Press. 2018.

2. X. Villasevil. *Biomats, sistema d'adquisició, tractament i transmissió de senyals biológics i mecànics*. Memòria de Projecte Fi de Carrera en Enginyeria Electrònica. Universitat Autònoma de Barcelona. 1996.

3. E. Valderrama. *Apuntes de imagen médica*. Asignatura de Fundamentos físicos para la adquisición de datos. Titulación de Ingeniería de Datos. Escola d'Enginyeria. Universidad Autònoma de Barcelona. 2018.

4. J.E. Quintero Muñoz. *Electrocardiografía básica*. Asociación Colombiana de Bioingeniería y Electrónica Médica.

5. J. R. Zaragoza. *Física e instrumentación médicas*. Barcelona: Ediciones científicas y técnicas S.A. Masson-Salvat medicina. 1992.

6. *American Heart Association Journals*. https://www.ahajournals.org/

7. W. Herring. *Radiología Básica. Aspectos fundamentales*. Elsevier 2012.

8. *https://www.feandalucia.ccoo.es/docu/p5sd6452.pdf*

# Chapter 3

# Fiabilidad en la Transmisión y Almacenaje de los Datos

*Mercè Villanueva*

## 3.1 Introducción

Hacia el año 1948, Claude E. Shannon formuló en sus trabajos el problema de la transmisión de información en términos estadísticos, utilizando modelos probabilísticos para las fuentes de información y los canales de comunicación [16]. Con estos trabajos nació un nuevo campo llamado teoría de la información [1]. Dentro de este campo se encuentra la teoría matemática de la comunicación digital que estudia la transmisión de la información entre un emisor y un receptor a través de un canal. En muchos casos, los canales que se utilizan son inseguros y con ruido, como por ejemplo el canal de comunicación entre dos teléfonos móviles vía satélite donde la información viaja por el espacio a través de ondas.

Así, en general, la información puede ser interceptada y manipulada por terceros. Además, se pueden producir errores e interferencias que alteran la señal transmitida, haciendo que el receptor no reciba exactamente la misma información que había enviado el emisor. Otros ejemplos de comunicación digital se encuentran en la televisión digital, el almacenamiento de datos en diferentes dispositivos (memorias, CD, Blue-Ray, etc), Internet, IoT, etc.

En la transmisión de información se aplican diferentes mecanismos, de manera que se pueda garantizar que la comunicación entre el emisor y el receptor sea eficiente, segura y exacta. La Figura 3.1 muestra un esquema general de un sistema de comunicación digital a través de un canal inseguro y con ruido. Para que la transmisión sea más eficiente, se aplica un compresor que permite reducir el tamaño de los datos que queremos enviar o almacenar,

aprovechando la redundancia que estos presentan. También se puede realizar un proceso de cifrado (también conocido como encriptación) para evitar que un tercero pueda acceder a los datos y garantizar así su confidencialidad y/o autenticidad. Finalmente, para asegurar una comunicación fiable y exacta, se codifican utilizando códigos correctores de errores.

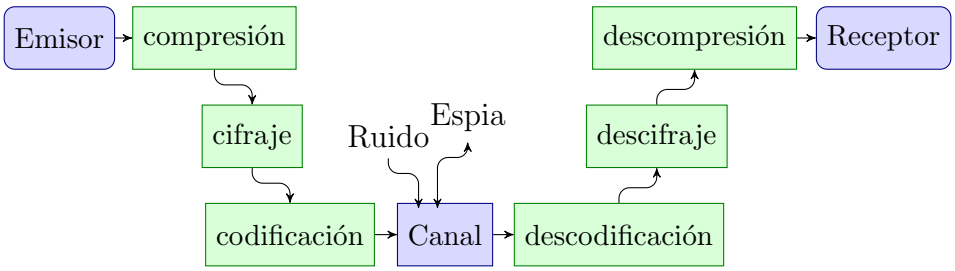



Figure 3.1: Sistema de comunicación digital.

Durante la transmisión de datos digitales (normalmente en forma de secuencia de bits) a través de canales con ruido, se pueden producir errores y, por tanto, no recibir a la salida del canal exactamente los mismos datos que se habían enviado. Por ejemplo, puede haber ruido térmico generado por el equipamiento electrónico. De la misma forma, también podemos considerar que se han producido errores en la información almacenada cuando algún disco duro falla (la probabilidad de fallo está aproximadamente entre el 2% y 4% al año).

Los códigos correctores de errores tienen como objetivo detectar estos errores y corregirlos. Este proceso lo que permite es, por ejemplo, recibir buenas imágenes desde los satélites, o disfrutar de la música almacenada en un CD aunque el disco esté rayado, o recuperar la información almacenada en un disco duro si este falla. El precio que hay que pagar para corregir los errores es añadir una cierta redundancia a los datos que se quieren transmitir o almacenar. En la Sección 3.3 introduciremos algunos conceptos básicos de la teoría clásica de los códigos correctores de errores.

No únicamente en las transmisiones de información, donde el canal es un cable, el aire o el espacio, es necesario disponer de mecanismos que puedan permitir la detección y, si es posible, la corrección de errores. Si consideramos, por ejemplo, que el canal es el papel o un disco duro, nos podemos plantear si en escribir o almacenar algún número largo, queremos poder detectar posibles errores. En estos canales, los errores más frecuentes son equivocarse introduciendo algún dígito o bien intercambiando dos dígitos. Para poder detectar estos tipos de errores, usualmente se añade redundancia incorporando un nuevo símbolo al número. Así es como se han diseñado códigos tan cotidianos como el código del Documento Nacional de Identidad o DNI, el

*International Standard Book Number* o ISBN de los libros, el *International Bank Account Number* o IBAN y el código de Cuenta Corriente o CCC de las cuentas bancarias, y el *European Article Number* o EAN que son los conocidos códigos de barras. A parte de estos más conocidos, hay muchos más como los que aparecen en las tarjetas de crédito, los billetes de avión, o servicios de mensajería, entre otros. En la Sección 3.2 describiremos dos de ellos, el código del DNI y el EAN.

Otro campo importante donde también es necesario disponer de mecanismos que garanticen la fiabilidad de los datos es en el almacenaje, y especialmente en el almacenaje distribuido para volúmenes grandes de información. Teniendo en cuenta el gran incremento de datos almacenados hoy en día, es importante disponer de mecanismos que garanticen su fiabilidad y disponibilidad, de forma que si algunos se pierden debido a posibles fallos en los discos duros utilizados, estos puedan ser recuperados. En la sección 3.4 veremos algunos de los mecanismos utilizados, problemas y posibles soluciones.

## 3.2 Códigos detectores de errores

En esta sección, describiremos algunos de los códigos detectores de errores más conocidos, ya mencionados en la introducción: el código asociado al DNI, y el EAN o código de barras. Los dos son bastante cotidianos y nos los encontramos a menudo en el día a día. Además, están basados en la aritmética modular y detecten pero no permiten corregir los errores.

### 3.2.1 Aritmética modular

El concepto de congruencia o aritmética modular ya aparece en el día a día ya que hay muchos eventos que són cíclicos. Por ejemplo cuando contamos los días de la semana o los meses de un año, ya que cada semana o cada año empieza de nuevo. Las congruencias y la aritmética modular formalizaran el proceso de "empezar de nuevo" o "dar la vuelta". Otra situación donde sin saberlo estamos usando aritmética modular es en las horas del dia. Si por ejemplo utilizamos las horas del 0 al 12, y son las $10h$, pero queremos saber que hora será dentro de 4 horas, vemos que $10 + 4 = 12 + 2$ y decimos que serán las $2h$, ya que a partir de las $12h$ empezamos de nuevo a contar. De la misma forma, si son las $10h$ y queremos saber que hora será al cabo de 49 horas, como $10 + 49 = 4 \cdot 12 + 11$ y al cabo de 48 horas volveran a ser las $10h$, en 49 horas serán las $11h$.

Para presentar de forma más rigurosa este concepto necesitamos recordar algunos conceptos básicos sobre los números enteros que denotaremos por

$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. Dados dos enteros $a, b \in \mathbb{Z}$ decimos que $a$ es múltiplo de $b$ o que $b$ es divisor de $a$ si $a = b \cdot q$ para algún $q \in \mathbb{Z}$, o lo que es lo mismo si el residuo de la división de $a$ entre $b$ es 0. En general, dados dos enteros $a$ y $b \neq 0$, siempre existen dos números enteros $q$ y $r$ únicos tales que $a = b \cdot q + r$, donde $0 \leq r < |b|$ y $|b|$ es el valor absoluto de $b$. Los enteros $q$ y $r$ se llaman cociente y residuo de la división de $a$ entre $b$.

**Ejemplo 3.2.1** *Si dividimos* 88 *entre* 11 *obtenemos residuo* 0*, por lo tanto decimos que* 88 *es múltiplo de* 11 *o que* 11 *es divisor de* 88*. También podemos decir que* $-88$ *es múltiplo de* 11*, o que* 11 *es divisor de* $-88$*.*

*Si dividimos* 59 *entre* 12 *obtenemos que* $59 = 12 \cdot 4 + 11$*, donde* 4 *es el cociente y* 11 *el residuo ya que* $11 < 12$*. En cambio, si dividimos* $-59$ *entre* 12 *obtenemos que* $-59 = 12 \cdot (-5) + 1$*, o sea el cociente es* $-5$ *y el residuo* 1*.*

A continuación, ya podemos definir el concepto de congruencia que permitirá definir la aritmética modular. Sea $n \in \mathbb{N} = \{1, 2, \ldots\}$. Una vez fijado el valor $n$, decimos que dos números enteros $a$ y $b$ son congruentes módulo $n$, si y solo si $b - a$ es múltiplo de $n$. En este caso, escribiremos $a \equiv b \mod n$. También se puede ver que $a \equiv b \mod n$ si y solo si $a$ y $b$ tienen el mismo residuo al dividir por $n$ [3]. Así, dado un número entero $a$, siempre podemos escribir que $a \equiv r \mod n$, donde $r$ es el residuo de dividir $a$ entre $n$.

**Ejemplo 3.2.2** *Siguiendo el ejemplo de las horas, podemos escribir que* $14 \equiv 2 \mod 12$*, o* $59 \equiv 11 \mod 12$*, ya que* $14 - 2 = 12$ *y* $59 - 11 = 48$ *son múltiplos de* 12*. También es fácil comprobar que el residuo de dividir* 14 *entre* 12 *es* 2*, ya que* $14 = 12 \cdot 1 + 2$*; o que el residuo de dividir* 59 *entre* 12 *es* 11*, ya que* $59 = 12 \cdot 4 + 11$*.*

*De la misma forma, tenemos por ejemplo que* $123 \equiv 3 \mod 10$*, o que* $-5 \equiv 17 \mod 11$*. También es facil comprobar que el residuo de dividir* 123 *entre* 10 *es* 3*, o que* $-5$ *y* 17 *tienen el mismo residuo* 6 *al dividir por* 11 *ya que* $-5 = 11 \cdot (-1) + 6$ *y* $17 = 11 \cdot 1 + 6$*. En este último caso también podemos escribir que* $-5 \equiv 17 \equiv 6 \mod 11$*.*

Fijado un valor $n \in \mathbb{N}$, podemos decir si dos números enteros están o no relacionados módulo $n$. Esta relación se dice que es de equivalencia. No entraremos en este concepto, pero es importante destacar que gracias a esta propiedad podemos clasificar todos los números enteros en $n$ subconjuntos disjuntos: $n\mathbb{Z} = \{k \in \mathbb{Z} : k \equiv 0 \mod n\} = \{nk : k \in \mathbb{Z}\}$ el subconjunto formado por los múltiplos de $n$, o equivalentemente los que tienen residuo 0 al dividir por $n$; $n\mathbb{Z} + 1 = \{k \in \mathbb{Z} : k \equiv 1 \mod n\} = \{nk + 1 : k \in \mathbb{Z}\}$ el subconjunto de los múltiplos de $n$ más 1, o dicho de otra forma, los que

tienen residuo 1 al dividir por $n$; y así sucesivamente hasta el subconjunto $n\mathbb{Z}+n-1 = \{k \in \mathbb{Z} : k \equiv n-1\} = \{nk+n-1 : k \in \mathbb{Z}\}$ formado por los enteros que tienen residuo $n-1$ al dividir por $n$. Fijaros que al dividir por $n$, los únicos residuos posibles son $0, 1, 2, \ldots, n-1$. Dado un número entero $a \in \mathbb{Z}$, éste pertenece a uno de estos conjuntos, exactamente $a \in n\mathbb{Z}+r$, donde $r$ es el residuo de dividir $a$ entre $n$. Así tenemos que

$$\mathbb{Z} = n\mathbb{Z} \cup (n\mathbb{Z}+1) \cup \cdots \cup (n\mathbb{Z}+n-1).$$

Además, dado que al dividir el residuo es único, estos subconjuntos no pueden compartir ningún entero y por tanto son disjuntos.

**Ejemplo 3.2.3** *Si por ejemplo $n = 2$, tenemos dos conjuntos, el conjunto $2\mathbb{Z} = \{\ldots, -4, -2, 0, 2, 4, \ldots\}$ de los números pares y el conjunto $2\mathbb{Z}+1$ de los impares, o sea el conjunto de números que tienen residuo 0 al dividir por 2 y el que de los que tienen residuo 1 al dividir por 2. Claramente, estos dos conjuntos son disjuntos y con la unión de los dos se obtiene el conjunto $\mathbb{Z}$.*

*Si $n = 3$, tenemos tres conjuntos, el $3\mathbb{Z} = \{\ldots, -3, 0, 3, 6, \ldots\}$, el $3\mathbb{Z}+1 = \{\ldots, -2, 1, 4, 7, \ldots\}$ y el $3\mathbb{Z}+2 = \{\ldots, -1, 2, 5, 8, \ldots\}$ que coinciden con el conjunto de enteros que tienen residuo 0, 1 y 2 al dividir por 3, respectivamente. Estos conjuntos son claramente disjuntos y si los unimos obtenemos cualquier entero, o sea el conjunto $\mathbb{Z}$.*

*Si $n = 10$, tenemos 10 conjuntos: $10\mathbb{Z}$, $10\mathbb{Z}+1$, $10\mathbb{Z}+2$, $10\mathbb{Z}+3$, $10\mathbb{Z}+4$, $10\mathbb{Z}+5$, $10\mathbb{Z}+6$, $10\mathbb{Z}+7$, $10\mathbb{Z}+8$, $10\mathbb{Z}+9$. También podemos ver que su unión es $\mathbb{Z}$, ya que al dividir por 10 los residuos posibles son $0, 1, \ldots, 9$. Dado un entero cualquiera $a$, éste está exactamente en uno de estos conjuntos. Por ejemplo, $2345 \in 10\mathbb{Z}+5$ ya que $2345 \equiv 5 \mod 10$, o $3457679 \in 10\mathbb{Z}+9$, ya que $3457679 \equiv 9 \mod 10$, o equivalentemente el residuo de dividir 2345 entre 10 es 5, y 9 es el residuo de dividir 3457679 entre 10.*

Finalmente, vemos que podemos realizar la aritmética habitual de sumar y multiplicar, pero en lugar de con números enteros, con los conjuntos $n\mathbb{Z}+r$, $r \in \{0, \ldots, n-1\}$. Para ello, utilizaremos las siguientes propiedades [3]:

$$(a \mod n) + (b \mod n) \equiv (a+b) \mod n \tag{3.1}$$

$$(a \mod n) \cdot (b \mod n) \equiv (a \cdot b) \mod n. \tag{3.2}$$

Estas dos nuevas operaciones están bien definidas [3], ya que si escogemos elementos diferentes de un mismo conjunto $n\mathbb{Z}+r$, el resultado que se obtiene es exactamente el mismo. Dicho de otra forma, si $a \equiv a' \mod n$ y $b \equiv b' \mod n$, entonces $a+b \equiv a'+b' \mod n$ y $a \cdot b \equiv a' \cdot b' \mod n$. Normalmente, al sumar o multiplicar buscamos siempre el número entero más pequeño que pertenece al conjunto $n\mathbb{Z}+r$, o sea el valor $r \in \{0, \ldots, n-1\}$.

**Ejemplo 3.2.4** *Sea* $n = 10$*. Por ejemplo, tenemos que* $3 + 5 \equiv 8 \mod 10$*, pero también que* $7+9 \equiv 6 \mod 10$*. En este último caso, al sumar obtenemos* $7 + 9 = 16 \in 10\mathbb{Z} + 6$*, ya que el residuo de dividir* 16 *entre* 10 *es* 6*.*

*De la misma forma podemos multiplicar. Por ejemplo,* $3 \cdot 5 \equiv 5$ *ya que* $15 \in 10\mathbb{Z} + 5$*, o también* $7 \cdot 9 \equiv 6$ *ya que* $56 \in 10\mathbb{Z} + 6$*.*

*En general, al sumar o multiplicar buscamos el elemento más pequeño del conjunto* $10\mathbb{Z} + r$ *que es el residuo* $r$ *al dividir entre* 10*. Esto se puede hacer antes y después de operar. Por ejemplo, para realizar la suma* $14 + 27 \mod 10$*, primero podemos considerar que* $14 \equiv 4 \mod 10$ *y* $27 \equiv 7 \mod 10$*, y en lugar de* $14+27$ *calcular* $4+7 \equiv 1 \mod 10$*. El resultado es el mismo que si realizamos la suma antes del módulo* 10*, o sea* $14 + 27 = 41 \equiv 1 \mod 10$*.*

### 3.2.2 Código DNI

El código del Documento Nacional de Identidad (DNI) consiste en un número de 8 cifras decimales seguido de una letra. Esta letra, de hecho, representa la redundancia que permite detectar los errores más frecuentes en escribir el número. La letra se asigna según el valor que resulta de calcular el número del DNI módulo 23, o lo que es lo mismo, según el residuo que se obtiene en dividirlo por 23, de acuerdo con las equivalencias que se muestran en la Tabla 3.1.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| T | R | W | A | G | M | Y | F | P | D | X | B |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | |
| N | J | Z | S | Q | V | H | L | C | K | E | |

Tabla 3.1: Assignación de letras para el DNI.

El cálculo se realiza módulo 23 con 23 un número primo para que sea un cuerpo finito [3]. Además, se excluyen las letras I, O y U, porque éstas se pueden confundir más fácilmente con el 1, el 0 y la letra V, respectivamente. Podemos simplificar los cálculos si tenemos precalculadas las potencias de 10 módulo 23. Así, como $10^2 \equiv 8$, $10^3 \equiv 11$, $10^4 \equiv 18$, $10^5 \equiv 19$, $10^6 \equiv 6$ y $10^7 \equiv 14$ módulo 23, calcular el número del DNI módulo 23 equivale a calcular $x_0 + 10x_1 + 8x_2 + 11x_3 + 18x_4 + 19x_5 + 6x_6 + 14x_7$ módulo 23, donde $x_7x_6x_5x_4x_3x_2x_1x_0 = \sum_{i=0}^{7} 10^i x_i$ representa el número del DNI.

**Ejemplo 3.2.5** *La letra del DNI correspondiente al número 34149351 es D, ya que* $34149351 \equiv 9 \mod 23$*, o equivalentemente,* $14 \cdot 3 + 6 \cdot 4 + 19 \cdot 1 + 18 \cdot 4 + 11 \cdot 9 + 8 \cdot 3 + 10 \cdot 5 + 1 \equiv 9 \mod 23$*, y la letra correspondiente al valor* 9 *és D de acuerdo con la Tabla 3.1.*

Este código permite detectar si ha habido un error en uno de los dígitos del DNI, o bien si ha habido una transposición entre dos dígitos [4]. En cambio, si hay dos o más, no siempre se pueden detectar. También permite recuperar uno de los dígitos si éste no se visualiza correctamente, simplemente resolviendo una ecuación lineal módulo 23.

**Ejemplo 3.2.6** *Continuando con el DNI del Ejemplo 3.2.5, ¿qué pasa si al escribir el DNI nos equivocamos en un dígito y escribimos, por ejemplo,* 34**2**49351$D$*? Como* $34249351 \equiv 5 \mod 23$ *y la letra correspondiente al* 5 *es la M, podemos detectar que ha habido un error, y repasar así la escritura de este DNI hasta obtener la letra correcta.*

*¿Y si nos equivocamos en dos dígitos que se han intercambiado de posición, y escribimos, por ejemplo,* 34**41**9351$D$*? En este caso, de nuevo, al calcular* $34419351 \equiv 12 \mod 23$*, podemos detectar el error ya que* 12 *corresponde a la letra N.*

*Finalmente, ¿que podemos hacer si uno de los dígitos está borroso, por ejemplo si tenemos* 341493$\square$1$D$ *donde el séptimo dígito no se reconoce? En este caso, podemos plantear la siguiente ecuación lineal módulo* 23, $14 \cdot 3 + 6 \cdot 4 + 19 \cdot 1 + 18 \cdot 4 + 11 \cdot 9 + 8 \cdot 3 + 10 \cdot x_1 + 1 \equiv 9 \mod 23$*, o sea* $10x_1 \equiv 4 \mod 23$*. Como* $10 \cdot 7 \equiv 1 \mod 10$*, multiplicando por* 7 *los dos términos de la ecuación módulo* 23*, tenemos que* $x_1 = 4 \cdot 7 = 28 \equiv 5 \mod 23$*. Por lo tanto, el dígito de la posición séptima es* 5 *y obtenemos el DNI* 34149351$D$*.*

### 3.2.3 Código EAN

La mayoría de los productos comerciales llevan asociado un número de 13 cifras decimales, llamado *European Article Number* (EAN) o más recientemente *International Article Number*. Este aparece justo debajo de un código de barras que sirve para facilitar su lectura. Los dos o tres primeros dígitos identifican el estado o asociación a la que está registrado el fabricante, los siguientes de 5 a 8 dígitos identifican la empresa, los siguientes hasta el 12º el producto, y finalmente el último es el dígito de control [18, 9].

El dígito de control se calcula a partir de los anteriores. Si el EAN lo denotamos por $x_1x_2\cdots x_{13}$, entonces $x_{13}$ se calcula de la siguiente forma:

$$x_{13} \equiv -\sum_{i=1}^{6}(x_{2i-1} + 3x_{2i}) \mod 10.$$

o utilizando la siguiente expresión, ya que $-1 \equiv 9$ y $-3 \equiv 7 \mod 10$:

$$\begin{aligned} x_{13} \equiv\ & 9x_1 + 7x_2 + 9x_3 + 7x_4 + 9x_5 + 7x_6 + 9x_7 \\ & + 7x_8 + 9x_9 + 7x_{10} + 9x_{11} + 7x_{12} \mod 10. \end{aligned}$$

Al igual que el código detector que hemos visto entes, el código EAN permite detectar si se ha producido un error en uno de los dos dígitos, pero a diferencia del anterior, no permite detectar siempre la transposición de dos dígitos. Concretamente, si se intercambian dos dígitos diferentes, $x_i$ y $x_j$, tales que $i$ y $j$ tienen la misma paridad, o bien $x_i - x_j \equiv 5 \mod 10$, la transposición no se detecta [4, 9]. También es fácil ver que podemos recuperar un dígito ilegible, resolviendo una ecuación, ya que el 1 y el 3 (o equivalentemente, sus opuestos, el $9 \equiv -1$ y el $7 \equiv -3$) son invertibles módulo 10 [3].

**Ejemplo 3.2.7** *Si la empresa utiliza el código EAN de la Asociación Española de Codificación Comercial (AECOC), los dos primeros dígitos son 84. En cambio, en otros países como Alemania los primeros dígitos son números del 400 al 440.*

*Si tenemos el siguiente número* 1512345678909 *de 13 cifras, podemos comprobar fácilmente que es un código EAN válido. Si se produce un error en uno de los dígitos, y obtenemos por ejemplo* 151**8**345678909*, podemos detectar el error ya que el dígito de control debería ser* 1 *en lugar de* 9*.*

*Si se intercambian dos dígitos no siempre se detecta. Por ejemplo, si obtenemos* 151234**0**6789**5**9 *debido a un intercambio entre las posiciones* $7^{\underline{a}}$ *y* $12^{\underline{a}}$*, podemos comprobar que el dígito de control es correcto y por tanto el error no se detecta. En cambio, si el intercambio se produce entre las posiciones* $1^{\underline{a}}$ *y* $6^{\underline{a}}$*, se detectaría, ya que* **4**5123**1**5678909 *debería tener el* 5 *como dígito de control en lugar del* 9*.*

## 3.3 Códigos correctores de errores

En esta sección, veremos los conceptos básicos de la teoría de los códigos detectores y correctores de errores. Describiremos los tres parámetros más importantes que definen un código y diremos cuando éste es óptimo. Para profundizar en esta teoría clásica de los códigos correctores de errores, se puede consultar, por ejemplo, [11, 12, 14].

Como hemos dicho, si queremos detectar y/o corregir posibles errores que se producen en las transmisiones a través de canales con ruido, o recuperar cierta información almacenada, tenemos que añadir cierta redundancia a la información, expandiéndola de forma que haya cierta "distancia" entre dos informaciones diferentes que será la que nos permitirá corregir errores y conseguir así una transmisión lo más fiable y exacta posible.

**Ejemplo 3.3.1** *Supongamos que queremos transmitir una secuencia de bits a través de un canal binario. Para poder corregir errores, en vez de enviar*

*un* 0 *o un* 1*, enviamos* 000 *o* 111*, respectivamente. Por ejemplo, si queremos transmitir* 1101*, enviamos* 111111000111 *a través del canal.*

*Supongamos que a la salida del canal recibimos* 101100001111*. A continuación, descodificamos según el número de ceros y unos que hay en cada bloque de* 3 *bits. Por ejemplo, si recibimos* 101*, como hay más unos que ceros, descodificamos por* 111*. Así, obtenemos* 111000000111*, o sea* 1001*. Observamos que algunos errores se han corregido y otros no.*

*El precio que pagamos es que hemos triplicado el número de bits que se envían por el canal y por lo tanto se hace más lenta la transmisión.*

Sea $\mathbb{F}_q$ un alfabeto con $q$ elementos. En los ejemplos de esta sección, únicamente consideraremos el caso más habitual en que el alfabeto contiene dos elementos, $q = 2$, y por tanto las operaciones son módulo 2; y el caso en que $q = 3$ y por tanto las operaciones son módulo 3. Un *código $q$-ario* $C$ es un conjunto finito de secuencias o vectores sobre $\mathbb{F}_q$, llamadas *palabras código*. Si todas las secuencias son de la misma longitud $n$, entonces decimos que $C$ es un *código bloque* de longitud $n$. En este caso, $C \subseteq \mathbb{F}_q^n$, donde $\mathbb{F}_q^n$ es el conjunto de vectores de longitud $n$ sobre $\mathbb{F}_q$. A partir de ahora, nos centraremos en los códigos bloque y supondremos que la información a transmitir esta representada por una secuencia de elementos del alfabeto $\mathbb{F}_q$.

**Ejemplo 3.3.2** *1.* $C_1 = \{000, 111\}$ *es el código bloque de longitud* $n = 3$ *sobre* $\mathbb{F}_2 = \{0, 1\}$ *con* $q = 2$ *elementos que aparece en el Ejemplo 3.3.1.*

*2.* $C_2 = \{000000, 111111, 000111, 111000\}$ *es un código de longitud* $n = 6$ *sobre el alfabeto* $\mathbb{F}_2$*.*

*3.* $C_3 = \{0000000, 1111111, 1110010, 1100101, 1101000, 1010001, 0100011, 0101110, 0010111, 1011100, 0011010, 0001101, 1001011, 0111001, 1000110, 0110100\}$ *tiene longitud* $n = 7$ *sobre* $\mathbb{F}_2$*, y se llama código de Hamming.*

*4.* $C_4 = \{00000, 21020, 12010, 21112, 12221, 12102, 21201, 00122, 00211\}$ *es un código de longitud* $n = 5$ *sobre* $\mathbb{F}_3$*.*

*5.* $C_5 = \{0000, 1212, 2121, 1122, 2211\}$ *es de longitud* 4 *sobre* $\mathbb{F}_3$*.*

En una transmisión utilizando códigos bloque, normalmente la información se divide en bloques de $k$ símbolos de $\mathbb{F}_q$. A cada bloque de $k$ símbolos se le asigna una palabra código diferente de un código $q$-ario $C$ de longitud $n$ formado por un mínimo de $q^k$ palabras código. Este proceso de asignación se llama *codificación.*

**Ejemplo 3.3.3** *1. Como* $|C_1| = 2^1$, $k = 1$, *para codificar debemos asignar a cada bit* 0 *o* 1 *una palabra código diferente. Por ejemplo,* $0 \mapsto 000$ *y* $1 \mapsto 111$ *como en el Ejemplo 3.3.1.*

*2. Como* $|C_2| = 2^2$, $k = 2$, *para codificar asignamos a cada dos bits, una palabra código diferente. Por ejemplo,* $00 \mapsto 000000$, $11 \mapsto 111111$, $01 \mapsto 000111$, *y* $10 \mapsto 111000$. *Así, la secuencia de información* 101101 *codificada utilizando este código seria* 111000 111111 000111.

*3. Como* $|C_3| = 2^4$, $k = 4$, *por ejemplo podemos codificar utilizando la siguiente asignación:* $0000 \mapsto 0000000$, $1111 \mapsto 1111111$, $1110 \mapsto 1110010$, $1100 \mapsto 1100101$, $1101 \mapsto 1101000$, $1010 \mapsto 1010001$, $0100 \mapsto 0100011$, $0101 \mapsto 0101110$, $0010 \mapsto 0010111$, $1011 \mapsto 1011100$, $0011 \mapsto 0011010$, $0001 \mapsto 0001101$, $1001 \mapsto 1001011$, $0111 \mapsto 0111001$, $1000 \mapsto 1000110$, $0110 \mapsto 0110100$. *Fijaros que los* 4 *bits de información coinciden con los primeros* 4 *bits de la palabra código correspondiente.*

*4. Como* $|C_4| = 3^2$, *asignamos a cada* $k = 2$ *elementos de* $\mathbb{F}_3$ *una palabra código. Por ejemplo:* $00 \mapsto 00000$, $10 \mapsto 21020$, $20 \mapsto 12010$, $11 \mapsto 21112$, $22 \mapsto 12221$, $21 \mapsto 12102$, $12 \mapsto 21201$, $01 \mapsto 00122$, $02 \mapsto 00211$. *Los* 2 *bits de información coinciden con el segundo y el tercer bit de la palabra código correspondiente. Utilizando este código, la información* 12110002 *codificada seria* 21201 21112 00000 00211.

*5. En este caso* $|C_5| \neq 3^k$. *Normalmente, para facilitar la codificación se consideran códigos tales que* $|C| = q^k$ *para algún valor entero* $k$.

Es fácil ver que cuanto más separadas estén estas palabras código entre ellas, más errores permitirá corregir el código. Así, se define la *distancia (de Hamming)* entre dos vectores $v = (v_1, \ldots, v_n)$ y $w = (w_1, \ldots, w_n)$, que denotaremos por $d_H(v, w)$, como el número de coordenadas que tienen diferentes. El *peso (de Hamming)* de un vector $v = (v_1, \ldots, v_n)$, denotado por $\mathrm{wt}_H(v)$, es el número de coordenadas distintas de cero y, por tanto, $\mathrm{wt}_H(v) = d_H(v, \mathbf{0})$, donde $\mathbf{0}$ es el vector con todas las coordenadas iguales a cero.

**Ejemplo 3.3.4** *1. La distancia entre los vectores* $v = 1011100$ *y* $w = 0111101$ *es* 3, *ya que las coordenadas en las posiciones* 1, 2 *y* 7 *son diferentes. Podemos escribir* $d_H(1011100, 0111101) = 3$.

*2. De forma similar, para vectores sobre* $\mathbb{F}_3$, $d_H(012012, 011121) = 4$.

*3. El peso de* $v = (1011100)$ *is* 4, *ya que tiene* 4 *coordenadas diferentes de cero. Podemos escribir* $\mathrm{wt}_H(1011100) = 4$.

*4. El peso de* $w = (011121)$ *is* 5*, o sea* $\mathrm{wt}_H(0111101) = 5$*.*

La descodificación por mínima distancia escoge la palabra código más cercana al vector recibido, de acuerdo con la distancia de Hamming. Este método también minimiza la descodificación por máxima verosimilitud [14, p. 128], suponiendo que cada símbolo del alfabeto tiene la misma probabilidad de error $p$ y la misma probabilidad de aparecer. Un canal con estas dos propiedades se llama *canal $q$-ario simétrico*. Cuando $q = 2$, se denomina canal binario simétrico o BSC($p$).

**Ejemplo 3.3.5** *En una transmisión de información, suponemos que usamos el código $C_4$ del Ejemplo 3.3.2 junto con la codificación descrita en el Ejemplo 3.3.3. Después de enviar dos palabras código a través de un canal con ruido, recibimos los vectores $w_1 = (22010)$ y $w_2 = (22211)$. Luego, calculamos la distancia de $w_1$ y $w_2$ a cada una de las palabras código de $C_4$:*

| $c \in C_4$ | $d_H(c, w_1)$ | $d_H(c, w_2)$ |
|---|---|---|
| 00000 | 3 | 5 |
| 21020 | 2 | 4 |
| 12010 | **1** | 3 |
| 21112 | 3 | 3 |
| 12221 | 4 | **2** |
| 12102 | 4 | 4 |
| 21201 | 4 | **2** |
| 00122 | 5 | 5 |
| 00211 | 4 | **2** |

*Hay una palabra código* 12010 *a una distancia mínima* 1 *de $w_1$, por lo que elegimos esta palabra código en el proceso de decodificación. La información correspondiente a esta palabra código es* 20*, ya que* $20 \mapsto 12010$*.*

*Para $w_2$, hay tres palabras de código a una distancia mínima* 2*:* 12221*,* 21201 *y* 00211*, por lo que podemos elegir cualquiera de ellas en el proceso de decodificación. Por ejemplo, si elegimos la palabra código* 12221*, obtenemos que la información enviada es* 22*, ya que* $00 \mapsto 12221$*.*

La *distancia mínima* de un código $q$-ario $C$ es $d(C) = \min\{d_H(v, w) : v, w \in C, v \neq w\}$. Si un código $q$-ario $C$ de longitud $n$ sobre $\mathbb{F}_q$ contiene $M$ palabras código y tiene distancia mínima $d = d(C)$, diremos que es un código $q$-ario $(n, M, d)$. La *tasa de transmisión* de $C$ es $R = \log_q(M)/n$ y mide la proporción de información que contiene cada palabra código.

**Ejemplo 3.3.6** *Para los códigos dados en el Ejemplo 3.3.2, calculamos la distancia mínima $d$, parámetros $(n, M, d)$ y tasa de transmisión $R$.*

1. *El código $C_1$ tiene una distancia mínima 3, ya que $d_H(000, 111) = 3$, entonces es un código $(3, 2, 3)_2$ con $R_1 = \log_2(2)/3 = 1/3 \approx 0.33$.*

2. *Para el código $C_2$, $d = 3$, ya que las distancias entre palabras código son 3 o 6. Es un código $(6, 4, 3)_2$ con $R_2 = \log_2(4)/6 = 2/6 = 1/3 \approx 0.33$.*

3. *El código $C_3$ también tienen distancia mínima 3, por lo tanto es un código $(7, 16, 3)_2$ con $R_3 = \log_2(16)/7 = 4/7 \approx 0.57$.*

4. *Se puede calcular que la distancia mínima es $d(C_4) = 3$, por lo tanto $C_4$ es un código $(5, 9, 3)_3$ con $R_4 = \log_3(9)/5 = 2/5 = 0.4$.*

5. *Para el código $C_5$, tenemos que $d_H(0000, 1212) = d_H(0000, 2121) = d_H(0000, 1122) = d_H(0000, 2211) = d_H(1212, 2121) = 4$, así como $d_H(1212, 1122) = d_H(1212, 2211) = d_H(2121, 1122) = d_H(2121, 2211) = 2$ y $d_H(1122, 2211) = 4$. Como el valor mínimo es 2, $d(C_5) = 2$ y $C_5$ es un código $(4, 5, 2)_3$ con $R_5 = \log_3(5)/4 \approx 0.36$.*

Encontrar algoritmos de descodificación eficientes es una de las tareas de investigación más importantes en la teoría de códigos dadas sus aplicaciones prácticas. Si enviamos una palabra código $c \in C$ y recibimos el vector $w \in \mathbb{F}_q^n$, decimos que $e = w - c$ es el *vector de error*. La descodificación consiste en encontrar el vector de error $e$ a partir de $w$, que equivale a encontrar la palabra código $c$ enviada. Entre todos los posibles vectores de errores, elegimos el de peso menor, que equivale a elegir la palabra código a menor distancia de $w$.

Diremos que un código $C$ *detecta* el vector de error $e$ si y solo si $c+e \notin C$ para cualquier $c \in C$, y diremos que *corrige* el vector de error $e$ si y solo si para todo $c \in C$, $c + e$ está más cercano a $c$ que a cualquier otra palabra código. Si un código corrige todos los vectores de error de peso como mucho $t$, y no puede corregir como mínimo uno de peso $t + 1$, diremos que el código es *$t$-corrector* o que $t$ es la *capacidad correctora* de $C$. De forma similar se define la *capacidad detectora* de $C$.

**Ejemplo 3.3.7** *1. Consideramos el código $C_1 = \{000, 111\}$ del Ejemplo 3.3.2. El vector de error $e = 100$ puede ser detectado ya que $\{c + 100 : c \in C_1\} = \{100, 011\}$ no contiene ninguna palabra código. El vector de error $e = 110$ también puede ser detectado ya que $\{c + 110 : c \in C_1\} = \{110, 001\}$ no contiene palabras código. En general, es fácil comprobar que cualquier vector de error de peso uno o dos puede ser detectado. En cambio, si $e = 111$, $\{c + 111 : c \in C_1\} = \{\mathbf{111}, \mathbf{000}\}$ contiene palabras código. Por lo tanto, $C_1$ és un código 2-detector.*

2. *Consideramos el código $C_5$ del Ejemplo 3.3.2. El vector de error $e =$ 2000 (y de la misma forma cualquier vector de peso uno) puede ser detectado, ya que $\{c+2000\} = \{2000, 0212, 1121, 0122, 1211\}$ no contiene ninguna palabra código. En cambio, el vector $e =$ 2001 no puede ser detectado, ya que $\{c + 2001\} = \{2001, 0210, \mathbf{1122}, 0120, 1212\}$ sí que contiene una palabra código. Por lo tanto, $C_5$ es un código 1-detector.*

**Ejemplo 3.3.8** 1. *Consideramos el código $C_1 = \{000, 111\}$ del Ejemplo 3.3.2. El vector de error $e = 100$ puede ser corregido ya que $e+c$ está a distancia mínima 1 de una única palabra código, para todo $c \in C_1$. En general, es fácil comprobar que cualquier vector de error de peso uno puede ser corregido. En cambio, si $e = 110$, vemos que para $c = 000$, el vector $e + c = 110 + 000 = 110$ está más próximo a 111 que a 000. Por lo tanto, $C_1$ és un código 1-corrector.*

2. *Consideramos el código $C_5$ del Ejemplo 3.3.2. El vector de error $e =$ 2000 no puede ser corregido, ya que $1212 + 1000$ está a distancia 1 de dos palabras código, 1212 y 2211. Por lo tanto, $C_5$ es 0-corrector.*

El vector recibido $w \in \mathbb{F}_q^n$, aparte de errores, puede contener borrones. Diremos que hay un *borrón* en una coordenada de $w$ si no se ha podido determinar qué símbolo de $\mathbb{F}_q$ se había enviado. La posición de un error dentro del vector $w$ es desconocida, mientras que en un borrón sí es conocida.

**Ejemplo 3.3.9** *Si consideramos el código $C_1 = \{000, 111\}$ del Ejemplo 3.3.2, y recibimos el vector $w =$??1 con dos borrones en las dos primeras posiciones, podemos suponer que la palabra código enviada era 111. O si recibimos $w =$?0? que la palabra era 000.*

*El código $C_1$ se dice que es un código de repetición. Este es el que usamos cuando por ejemplo realizamos dos copias de seguridad de un disco duro. Así, en caso de que falle uno o dos de los discos duros, aún podemos recuperar la información, reemplazar los dos discos defectuosos por unos nuevos, y recuperar de nuevo el sistema con el mismo grado de fiabilidad.*

A partir de la distancia mínima de un código, veremos que podemos calcular su capacidad detectora y correctora, tanto de errores como de borrones. Para examinar los vectores que están más cercanos a una palabra código, introducimos el concepto de bola alrededor de una palabra código. Una bola de radio $r$ centrada en el vector $c \in \mathbb{F}_q^n$, denotada por $S_r(c)$, contiene todos los vectores que están a distancia menor o igual que $r$ de $c$, o sea, $S_r(c) = \{v \in \mathbb{F}_q^n : d_H(v, c) \leq r\}$. Para corregir hasta $t$ errores, es necesario que las bolas de radio $t$ alrededor de cada palabra código sean disjuntas dos a dos, tal como se muestra en la Figura 3.2.

**Teorema 3.3.1** *[14, p. 117] Un código q-ario $C$ puede detectar hasta $d(C)-1$ errores. El código permite corregir hasta $t$ errores y $s$ borrones si y solo si $2t+s<d(C)$. Además, la capacidad correctora es $t=\lfloor (d(C)-1)/2 \rfloor$.*

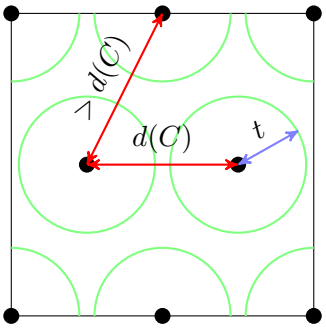



Figure 3.2: Bolas de radio $t$ centradas en las palabras código de $C$.

**Ejemplo 3.3.10** *Los códigos $C_1$, $C_2$ y $C_3$ del Ejemplo 3.3.2 pueden detectar hasta dos errores o corregir un error, ya que la distancia mínima es 3. Los tres son códigos sobre $\mathbb{F}_2$ con la misma capacidad detectora y correctora, pero $C_3$ tiene una mejor tasa de transmisión ya que $R_3 = 0.57 > 0.33 = R_1 = R_2$.*

*El código $C_4$ tiene la misma capacidad detectora y correctora, ya que $d(C_4)=3$. En cambio el código $C_5$ permite detectar un error pero no puede corregir ninguno. Comparando $C_4$ y $C_5$, ambos sobre $\mathbb{F}_3$, vemos que $C_4$ es mejor tanto en capacidad detectora y correctora como en tasa de transmisión.*

Por el Teorema 3.3.1, a fin de corregir el máximo número de errores necesitamos que el valor de la distancia mínima $d = d(C)$ sea grande. Al mismo tiempo, pero, queremos que la longitud $n$ sea pequeña para que la transmisión sea rápida, y que el número de palabras código $M$ también sea grande para codificar el mayor número de mensajes. Estas condiciones no se pueden alcanzar todas a la vez dado las relaciones que existen entre estos parámetros. Por ejemplo, tenemos que $M \leq q^n$. También se cumple la llamada *cota de Singleton*, $M \leq q^{n-d+1}$; y la *cota de Hamming*,

$$M \cdot \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^n, \tag{3.3}$$

donde $t = \lfloor (d-1)/2 \rfloor$ es la capacidad correctora del código, [12, p. 20].

Se dice que un código es *Máxima Distancia Separable* (MDS) si se cumple la igualdad en la cota de Singleton, o sea cuando $M = q^{n-d+1}$. Se dice que un

código es *óptimo* si tiene los mejores parámetros posibles $n, M, d, q$, es decir, al mismo tiempo no se puede incrementar $M$ para valores de $n, d, q$ fijados; y no se puede incrementar $d$ para valores de $n, M, q$ fijados.

**Ejemplo 3.3.11** *1. El código $C_1$ del Ejemplo 3.3.2 es MDS, ya que $q = 2$, $M = q^k = 2^1$, $n = 3$, $d = 3$, y por tanto $2 = 2^{3-3+1}$. También cumple la cota de Hamming, ya que $2^3 = 2(\binom{3}{0} + \binom{3}{1}) = 2(1 + 3) = 2^3$.*

*2. El código $C_3$ del Ejemplo 3.3.2 no es MDS, ya que $q = 2$, $M = q^4 = 16$, $n = 7$, $d = 3$, y por tanto $16 \neq 2^{7-3+1}$. Este código sí que cumple la cota de Hamming, ya que la capacidad correctora es $t = 1$ y tenemos que $2^7 = 16 \cdot \sum_{i=0}^{1} \binom{7}{i}(2-1)^i = 16(\binom{7}{0} + \binom{7}{1}) = 16(1 + 7) = 2^7$.*

*3. Sea $D = \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\}$. Es fácil comprobar que el último bit de cada palabra código se calcula sumando los tres primeros bits, o sea si $x_1x_2x_3x_4$ es una palabra código, $x_4 = x_1 + x_2 + x_3 \mod 2$. En este caso, se dice que es un código de paridad. Como $q = 2$, $M = 8$ $n = 4$ y $d = 2$, se puede comprobar que es MDS.*

## 3.4 Códigos para el almacenaje distribuido

Los códigos correctores de errores o borrones que hemos introducido en la sección anterior se aplican en multitud de situaciones, de hecho, aparecen en cualquier transmisión de datos digitales, ya sea por aire o por cable. También se utilizan en el almacenaje de datos, concretamente en los sistemas RAID, y en otros sistemas más recientes de almacenaje distribuido para grandes cantidades de datos. Empezaremos explicando los sistemas RAID, a continuación, mostraremos sus limitaciones, y finalmente algunas de las soluciones propuestas más recientes.

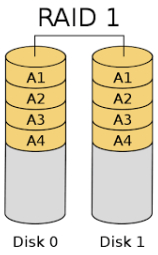
### 3.4.1 Sistemas RAID

RAID es el acrónimo de *Redundant Array of Independent Disks* (o sea matriz redundante de discos independientes) [6]. Fueron creados por David A. Patterson, Garth Gibson y Randy H. Katz de la Universidad de California, Berkeley. Su finalidad es garantizar la disponibilidad de los datos almacenados en caso de que un disco duro falle. La idea es crear un único volumen con varios discos duros que funcionen como un único disco, y conseguir tolerancia a fallos en el caso de que uno falle, o mayor velocidad de lectura y escritura. En definitiva, RAID permite aumentar el rendimiento de acceso y escritura de datos, al tiempo que mejora la seguridad de la información.

Existen diferentes configuraciones o niveles. Inicialmente se crearon 5, RAID 0, RAID 1, hasta RAID 5. IBM tiene los derechos de propiedad sobre el RAID 5. En algunos RAIDs se utiliza la duplicación de los datos en más de un disco. También en algunos de ellos se dividen los datos distribuyéndolos en más de un disco, y se añaden datos redundantes utilizando códigos correctores de errores para poder recuperar la información y el sistema inicial, cuando alguno de los discos duros falla. Los diferentes niveles de RAID utilizan una o más de estas técnicas, según los requisitos del sistema. A continuación, vamos a definir algunos de los más comunes. En el siguiente enlace, se puede ver la simulación del funcionamiento para cada uno de los niveles `https://primearraystorage.com/raid.php`.

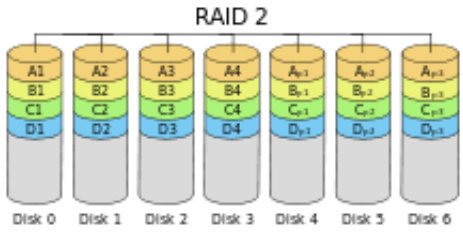
- RAID 0. Este tipo de RAID contiene la idea general de proporcionar mayor velocidad al sistema. La información se va escribiendo en dos discos de manera alterna, es decir, un bit en uno, y otro bit en otro, de manera que el ancho de banda es literalmente el doble y por eso se mejora notablemente el rendimiento. La parte negativa es que si falla alguno de los dos discos duros, toda la información se pierde ya que está repartida entre los dos discos.

- RAID 1. Este tipo contiene la idea general de redundancia. Los datos se escriben en los dos discos de manera simultánea, siendo el uno una copia exacta del otro, motivo por el que se conoce como *mirroring*. En este caso, si se estropeara uno de los dos discos no pasaría nada porque los datos estarían todavía en el otro, y bastaría con reemplazar el disco estropeado por uno nuevo para volver a restablecer el sistema. El inconveniente es que no se mejora el rendimiento, sino al contrario ya que los datos deben escribirse dos veces.



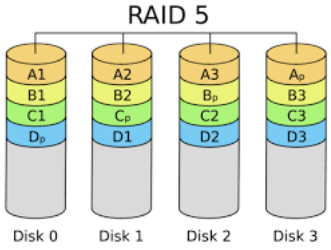

  En este caso, de acuerdo con la notación utilizada en la sección anterior, se estaría utilizando un código $C = \{00, 11\}$ de repetición, donde cada dato simplemente se repite una vez. La distancia mínima es $d = 2$, por lo tanto, permite corregir hasta $d-1$ borrones, o sea que un disco falle. En cambio, la redundancia es muy alta, ya que $R = 1/2 = 0.5$. Es fácil comprobar que $C$ es un código MDS, ya que $2 = M = 2^{n-d+1} = 2^{2-2+1}$.

- RAID 2. En este caso la información se distribuye en 4 discos, y se añaden 3 más de redundancia calculada utilizando el código de Hamming, o sea el código $C_3$ de los Ejemplos 3.3.2 y 3.3.3. Como hemos visto, este código tiene distancia mínima $d = 3$ y por lo tanto permite corregir hasta 2 borrones, o sea recuperar hasta la pérdida de 2 de cada 7 discos. La redundancia es mejor, ya que $R = 4/7 \approx 0.57$. El código no es MDS, tal como se muestra en el Ejemplo 3.3.11.



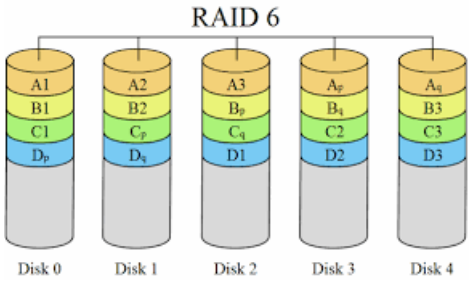

- RAID 5. Este es uno de los modos más utilizados. La información se divide en tres partes y se añade una de redundancia que consiste en un bit de paridad. Concretamente, el código utilizado es el código $D$ del Ejemplo 3.3.11 que es MDS. Como se puede ver en el siguiente dibujo, los datos de paridad, denotados por $A_p, B_p, C_p, D_p$, no se escriben en un disco duro fijo, sino que se distribuyen en todos ellos.

  En este caso, como el código tiene $d = 2$, el sistema soporta la tolerancia a fallos de $d-1 = 1$ disco: si falla un disco, no se pierde la información. La lectura de datos es muy rápida, mientras que la escritura es más lenta debido al cálculo de la paridad. La parte negativa es que si fallaran dos discos, sí que tendríamos pérdida de datos. También podríamos perder toda la información, si durante el proceso de reconstrucción (que puede llegar a durar un día o más si es de 4TB) otro disco falla.




- RAID 6. Este es el que se utiliza más, aunque solo en entornos empresariales. Es una variante del RAID 5 pero que emplea dos discos como redundancia en lugar de uno. También se utiliza un código MDS, pero con distancia mínima $d = 3$, por lo tanto, si dos discos fallan, todavía

se tiene acceso a todos los datos. Así tenemos que es más seguro que el RAID 5. De nuevo la lectura es rápida, pero la escritura es lenta debido al cálculo de la redundancia.




### 3.4.2 Sistemas basados en códigos MDS

Claramente, en los últimos años se han ido incrementando las necesidades de almacenaje de datos. Además, es importante que gran cantidad de estos datos estén disponibles fácilmente des de cualquier lugar y para siempre. Por todo ello, se han ido popularizando los sistemas de almacenaje distribuido, conocidos como NDSS (*Network Distributed Storage Systems*). Estos están basados en almacenar los datos en discos duros, pero a la vez estos están conectados a través de una red y distribuidos en diferentes lugares.

Hay dos grandes familias de NDSS. Por un lado, se encuentran los sistemas *peer-to-peer* donde un usuario tiene un fichero que comparte con otros utilizando aplicaciones P2P. Otros usuarios pueden descargar el fichero, almacenarlo y compartirlo con otros. Por otro lado, se encuentran los centros de datos o *data centers*, que son edificios que guardan grandes cantidades de discos duros, normalmente organizados en *racks*. Cada centro de datos contiene miles de racks y cada rack está formado por docenas de discos duros.

La tolerancia a fallos o *fault tolerance* de un NDSS es el número de fallos de discos duros que el sistema puede tolerar sin perder información. El gasto adicional de almacenaje o *storage overhead* es la ratio entre la cantidad de información y la cantidad de datos almacenados con la redundancia.

El almacenaje de datos es muy caro tanto en términos económicos, de hardware, software, mantenimiento y energía consumida; como en términos de espacio necesario. La pérdida de datos se puede producir debido a fallos en los discos duros, o también a problemas de seguridad. Respecto al primero, es fácil que un disco falle o de forma equivalente que no esté disponible debido a acciones de mantenimiento temporales. El objetivo es encontrar sistemas que por un lado garanticen la fiabilidad contra la pérdida de datos, y al mismo tiempo minimicen el storage overhead. Además, se busca que la

recuperación frente al fallo de un disco sea eficiente, o sea que se realize lo más rápido posible para poder reestablecer el sistema.

Como hemos visto en los RAIDs, para asegurar la disponibilidad de los datos y por tanto su fiabilidad, podemos utilizar sistemas basados en la réplica, o también conocidos como *backups*, o bien sistemas que incorporan códigos correctores de errores o borrones, más óptimos. Éstos últimos permiten reducir la redundancia, y por tanto minimizar el storage overhead, intentando mantener el mismo grado de fault tolerance. El problema es que incorporan el conocido problema de la reparación, ya que para reparar un disco duro se requiere aumentar el tráfico de datos y por lo tanto también el ancho de banda, haciendo que este sea más lento. En sistemas basados en la réplica, la reparación de un disco duro se realiza mucho más rápido ya que únicamente es necesario descargar los datos de uno de los discos duros correctos.

En general, en los sistemas RAID se utilizan códigos $(n, 2^k, d)$ MDS (Máxima Distancia Separable), o sea tales que cumplen la cota de Singleton $q^k = q^{n-d+1}$ sobre un alfabeto con $q$ elementos. De esta forma, por cada $k$ bloques de datos se añaden $n-k$ bloques hasta obtener $n$. El uso de estos códigos óptimos permite minimizar la redundancia manteniendo la misma fiabilidad. Concretamente, la tolerancia a fallos es $n-k$, ya que la distancia mínima es $d = n-k+1$ y el código permite corregir hasta $d-1 = n-k$ borrones. El problema principal de estos sistemas es que si un bloque (o hasta $n-k$ bloques) se estropea y hay que recuperar los datos almacenados, hay que descargar información de los $k$ bloques correctos para volver a tener el mismo sistema. Esto hace que no siempre sean la mejor solución. En general, la recomendación es utilizar sistemas de réplica para datos primarios o secundarios, y sistemas con otros códigos correctores para datos archivados donde el rendimiento no es un problema.

Normalmente, como códigos MDS, se utilizan los conocidos códigos Reed-Solomon, que fueron descritos ya en los años 1960's [11, 12, 14]. Por ejemplo, el sistema Hadoop Distributed File System (HDFS-EC) utiliza un código Reed-Solomon de longitud $n = 9$ con $n-k = 3$ de redundancia; el sistema de almacenaje de Facebook f4 BLOB utiliza un código de longitud $n = 14$ con una redundancia de 4; y el almacenaje Baidu's Atlas cloud uno de longitud $n = 12$ con 4 de redundancia [15].

**Ejemplo 3.4.1** *El sistema basado en la réplica y utilizado normalmente en el sistema de ficheros de Hadoop (HDFS) consiste en triplicar la información. Por tanto, soporta hasta el fallo de* 2 *discos (fault tolerance) y tiene una eficiencia de* $1/3$*, o sea del* $33\%$*, o un gasto adicional (storage overhead) del* $200\%$*. En cambio, cuando se utiliza en HDFS un código Reed-Solomon*

*de longitud* $n = 9$ *con* $n - k = 3$ *de redundancia (introducido a partir de la versión 3.x), este soporta hasta el fallo de* 3 *y su eficiencia sería de* $k/n = 6/9$ *(o sea del* 67%*) con un* 50% *de storage overhead. Si tenemos* 6 *bloques de información, con el primero necesitamos* $6 \cdot 3 = 18$ *bloques, y con el segundo únicamente* $6+3 = 9$*. Normalmente los bloques son de* $128MB$*. Por ejemplo, un fichero de* $700MB$ *se dividiría en* 6 *bloques, cinco de* $128MB$ *y uno de* $60MB$*. Con la réplica, se consumen* 18 *bloques o* $2100MB$*.*

| Código | $(n, k, d)$ | Fault tolerance | Storage efficiency | Storage overhead |
|---|---|---|---|---|
| 3-replica | $(3, 1, 3)$ | 2 | 33% | 200% |
| RS(6,3) | $(9, 6, 3)$ | 3 | 67% | 50% |
| RS(10,4) | $(14, 10, 4)$ | 4 | 71% | 40% |

Tabla 3.2: Comparación entre distintos sistemas.

### 3.4.3 Alternativas más recientes

A continuación, detallaremos algunas de las soluciones o sistemas híbridos que intentan solucionar el problema de la reparación, o sea el problema del incremento del ancho de banda para la reparación o substitución de un disco duro o bloque de dentro del sistema.

#### Códigos regenerativos

Supongamos que se utiliza un código de longitud 4 con 2 de redundancia. Si cada disco duro o nodo es de 1MB, cuando cualquiera de los 4 nodos se estropea, para obtener de nuevo los datos almacenados en él, hay que descargar los datos de dos cualesquiera de los tres nodos restantes, por lo tanto, habría que descargar 2MB de datos.

Las soluciones basadas en los llamados "códigos regenerativos" (RGC) [7, 13], intentan disminuir el ancho de banda. En el ejemplo dado, en lugar de descargar 2MB sería suficiente descargar 1.5MB. La Figura 3.3 muestra un esquema de la situación, y la Figura 3.4 como descargar 1.5MB en lugar de 2MB, dividiendo la información de cada nodo o disco duro. Esta solución permite disminuir el ancho de banda y recuperar de nuevo el sistema con la misma *fault tolerance*, aunque el nodo recuperado no contenga necesariamente los mismos datos que el nodo original que falló.

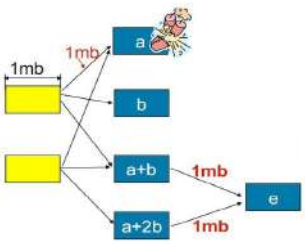



Figure 3.3: Problema de ancho de banda en la reparación

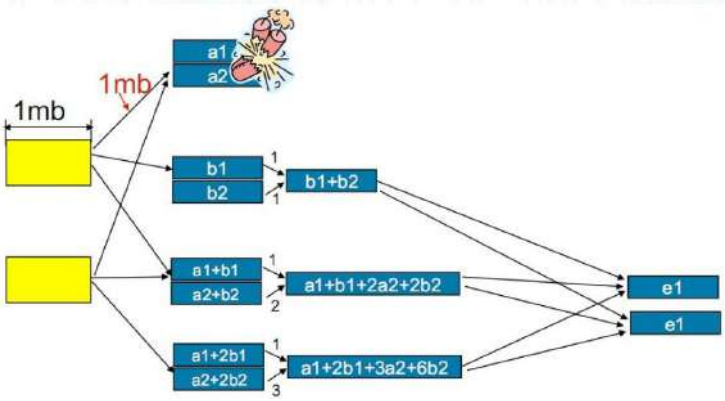



Figure 3.4: Solución al problema de ancho de banda en la reparación

### Códigos localmente reparables

Otras soluciones se basan en el uso de los llamados "códigos localmente reparables" (LRC). Estos códigos en lugar de reducir el ancho de banda sin reducir el número de nodos de los que hay que descargar datos, intentan reducir el número de nodos a los que hay que conectarse para recuperar el nodo dañado. De esta forma, a la vez, se reduce el ancho de banda y por lo tanto aumenta la velocidad de recuperación del sistema. Sin embargo, los códigos tienen un mayor incremento en el storage overhead comparado con los códigos MDS.

Un ejemplo de esta familia de códigos sería el utilizado en el Windows Azure Storage (WAS) [5], la solución de almacenamiento en la nube de Microsoft. La Figura 3.5 muestra un esquema del código utilizado, que fue presentado en [10] y que explicamos a continuación. En este ejemplo, la información se distribuye en 6 nodos, denotados por $x_0, x_1, x_2$ y $y_0, y_1, y_2$. La redundancia estaría formada por 4 nodos más, denotados por $p_0, p_1, p_x, p_y$. En los sistemas anteriores, utilizando un código MDS, los 4 nodos de redundancia se calcularían a partir de los 6 nodos de información. En cambio, en

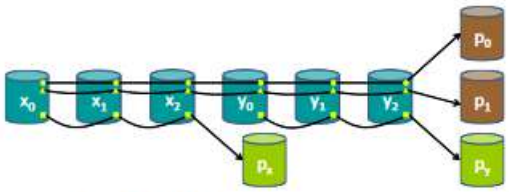



Figure 1: **A (6, 2, 2) LRC Example.** ($k = 6$ data fragments, $l = 2$ local parities and $r = 2$ global parities.)

Figure 3.5: Ejemplo LRC utilizado en el Windows Azure Storage [10]

este nuevo sistema, únicamente dos de ellos $p_0$ y $p_1$ se calcularían a partir de todos los nodos de información, y por tanto son llamados nodos de paridad global. El nodo de paridad $p_x$ se calcularía únicamente a partir de la información $x_0, x_1, x_2$; y $p_y$ a partir de la información $y_0, y_1, y_2$. Estos dos últimos nodos de redundancia $p_x$ y $p_y$, se llaman nodos de paridad local.

En general, en un $(k, l, r)$ LRC los datos se dividen en $k$ fragmentos, estos $k$ fragmentos se dividen en $l$ grupos con $k/l$ fragmentos en cada grupo. Se calcula una paridad local para cada grupo, y $r$ paridades globales a partir de todos los $k$ fragmentos de datos. El número total de fragmentos es $n = k + l + r$. El storage overhead sería $(l + r)/k$ y la eficiencia en el almacenaje $k/n$. De acuerdo con el ejemplo, 66% y 60%, respectivamente.

Continuando con el ejemplo anterior, si el nodo $x_0$ falla, este se puede reconstruir únicamente con los datos de los nodos $x_1, x_2, p_x$, o sea en lugar de necesitar 6 nodos con 3 es suficiente. Si dos nodos fallan y estos están ubicados en bloques diferentes, cada nodo se puede reconstruir con los datos de 3 nodos. En cambio, si los dos nodos están en el mismo bloque, por ejemplo $x_0, x_1$ fallan, hay que utilizar los nodos de paridad global y los bloques de información restantes, o sea $p_0, p_1, x_3, y_0, y_1, y_2$. De forma similar si tres nodos fallan, se pueden reconstruir. Si fallan cuatro nodos, en función de donde estén ubicados podrían ser corregidos o no.

### Replicated erasure codes

Finalmente, también se han propuesto soluciones híbridas conocidas con el nombre de "Replicated erasure codes" (REC) o sea códigos correctores de borrones replicados [8]. Éstos combinan la eficiencia del espacio de almacenaje que presentan el uso de los códigos clásicos MDS, con la eficiencia del tráfico en la reparación que tienen los códigos basados en la replicación. A continuación, presentamos también un ejemplo mostrado en [8].

Supongamos que un fichero es dividido en $k$ bloques, y transformados

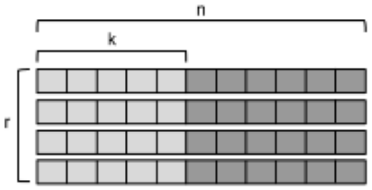



Figure 3.6: Ejemplo de sistema REC

en $n$ bloques con algún código MDS, añadiendo por tanto $n - k$ bloques de redundancia. A continuación, cada bloque es replicado $r$ veces, tal como se muestra en la Figura 3.6. Si $n = k$, el sistema equivale a la réplica, y si $r = 1$ equivale a utilizar un código corrector de borrones MDS. Para recuperar el fichero original, es necesario descargar $k$ bloques de diferentes conjuntos replicados o sea de diferentes columnas.

Supongamos que algunos bloques se estropean. Dependiendo de si todos, ninguno o algunos fallan en un conjunto replicado, se dice que el conjunto está completo, parcial o borrado. En el proceso de recuperación, primero se completan todos los conjuntos replicados parciales. Si en una columna hay bloques que han fallado se pueden recuperar a partir de algún bloque en esa misma columna, ya que son iguales. A continuación, los conjuntos replicados totalmente borrados se reestablecen descargando $k$ bloques que pertenezcan a diferentes conjuntos replicados, y aplicando el código MDS utilizado.

# Bibliography


[1] R. B. Ash, *Information theory*. New York: John Wiley and Sons Inc, 1965.

[2] S. B. Balaji, M. N. Krishnan, M. Vajha, V. Ramkumar, B. Sasidharan, P. V. Kumar, "Erasure coding for distributed storage: an overview", Science China, Information Sciences, vol 61(10), 2018.

[3] J.M. Basart, J. Rifà, M. Villanueva, *Fonaments de matemàtica discreta. Elements de combinatòria i d'aritmètica*. Col·lecció Materials de la UAB, n. 36, 1997.

[4] J. M. Brunat Blay, E. Ventura Capell, *Informació i codis*. Barcelona: Edicions UPC, 2001.

[5] B. Calder et al., "Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency," ACM SOSP, 2011.

[6] Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy H. Katz, David A. Patterson, "RAID: high-performance, reliable secondary storage," *ACM Computing Surveys*, 26(2), 1994, pp. 145–185.

[7] A. Dimakis, et alt. "Network coding for distributed Storage Systems", IEEE Trans. Inf. Theory, vol. 56(9), 2010.

[8] R. Friedman, Y. Kantor and A. Kantor. "Combining erasure-code and replication redundancy schemes for increase storage and repair efficiency in P2P storage systems. Technical report CS-2013-03, 2013.

[9] L. Hernández, A. Martín, "Codificación de información mediante códigos de barras", Bol. Soc. Esp. Mat. Apl., 27, (2004), 29–48.

[10] C. Huang, H. Simitci, Y Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, S. Yekhanin, Microsoft corporation. "Erasure coding in Windows Azure Storage", in proc. of 2012 USENIX Annual Technical Conference, 2012.

[11] W. C. Huffman, V. Pless, *Fundamentals of error-correcting codes*. Cambridge: Cambridge University Press, 2003.

[12] F. J. MacWilliams, N. J. A. Sloane, *The theory of error-correcting codes*. New York: North-Holland Publishing Company, 1977.

[13] V. Ramkumar, M. Vajha, S. B. Balaji, N. K. M. Krishnan, B. Sasidharan, P. V. Kumar, "Codes for Distributed Storage", to appear as a chapter in "A Concise Encyclopeia of Coding Theory", CRC Press. arxiv:2010.01344v1

[14] J. Rifà, Ll. Huguet, *Comunicación digital*. Barcelona: Masson, 1991.

[15] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, D. Borthakur, "XORing Elephants: Novel Erasure Codes for Big Data", In: Very large data bases; 39th international conference (VLDB 2013: proceedings of the VLDB endowment, volume 6, no. 1-14, pp. 325-336, 2014.

[16] C. Shannon, C. A mathematical theory of communication. *The Bell System Technical Journal*, 27, (1948), 379–423.

[17] A. Wang, Z. Zhang, K. Zheng, U. Maheshwara and V. Kumar, "Introduction to HDFS Erasure Coding in Apache Hadoop", 2015, Cloudera.

[18] ISO/IEC 15420:2009(en) *Information technology - Automatic identification and data capture techniques - EAN/UPC bar code symbology specification*. International Organization for Standardization. Switzerland, segona edició, 12-2009.

# Chapter 4

# Pre-procesado de Datos

*Bernat Gastón*

## 4.1 Introducción

Una buena calidad de los datos es la base de un análisis de datos exitoso. Sin embargo, a menudo la información contiene errores, sobretodo debido a la interacción de los humanos en el proceso de entrada de los mismos. Si pensamos en como se entran datos en un sistema nos daremos cuenta de lo fácil que es cometer errores. Los sistemas más habituales de entrada de datos son:

- Formularios: Se usan habitualmente como interfaz a una base de datos. Contienen campos limitados que el usuario tiene que rellenar (figura 4.1). A veces pueden contener filtros que se usan para limitar las opciones del usuario en la entrada de datos (p.e. formato de correo electrónico, calendario para la fecha de nacimiento, etc.)

- Hojas de Cálculo: Son sistemas de almacenamiento de datos mas rudimentarios, organizados por páginas que contienen matrices de dos dimensiones. Habitualmente los campos están en las columnas, y cada fila es una entrada.

### 4.1.1 Bases de datos

La mayoría de organizaciones no almacenan la información en hojas de cálculo, sino que lo hacen en bases de datos más o menos complejas. Las bases de datos pueden ser de dos tipos:

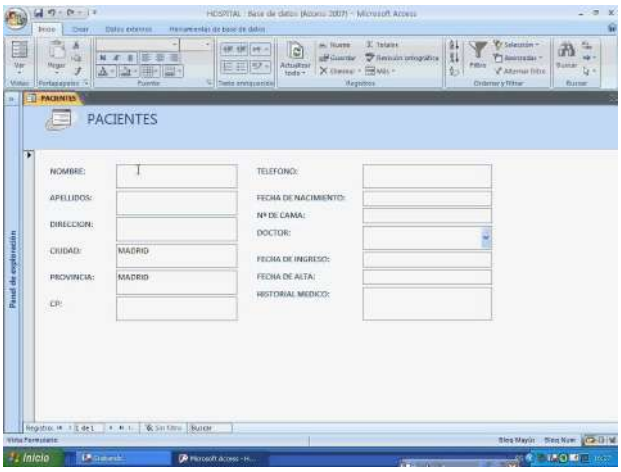



Figure 4.1: Ejemplo de formulario en base de datos.

- Entidad-Relación (o simplemente Relacionales): Están organizadas en tablas que representan entidades lógicas p.e. Paciente, Prueba (médica), Medicamentos, etc. Cada tabla tiene campos (también llamados atributos), por ejemplo la tabla Pacientes podría contener:
  - Identificador Paciente
  - Nombre
  - Edad
  - Dirección
  - ...

  Las tablas se relacionan entre ellas mediante campos compartidos p.e. la tabla Paciente y la tabla Prueba pueden contener (ambas) el campo Identificador paciente, de manera que las pruebas médicas de cada paciente se pueden enlazar al paciente en cuestión. Las bases de datos relacionales son un invento de 1970 por E.F. Codd, un investigador de IBM. El objetivo era mantener información cruzada (siguiendo el ejemplo, pacientes con pruebas, diagnósticos con pacientes y con medicamentos administrados, etc.) de una manera ordenada y **CONGRUENTE**.

  Esto último se refiere al hecho que toda la información debe respetar la estructura de la base de datos, sin excepción. Por ejemplo, si definimos una relación entre Paciente y Prueba, quiere decir que **NO** se puede añadir una Prueba sin que esté dado de alta el paciente en la tabla Paciente. De la misma manera, si se borra un paciente, podemos

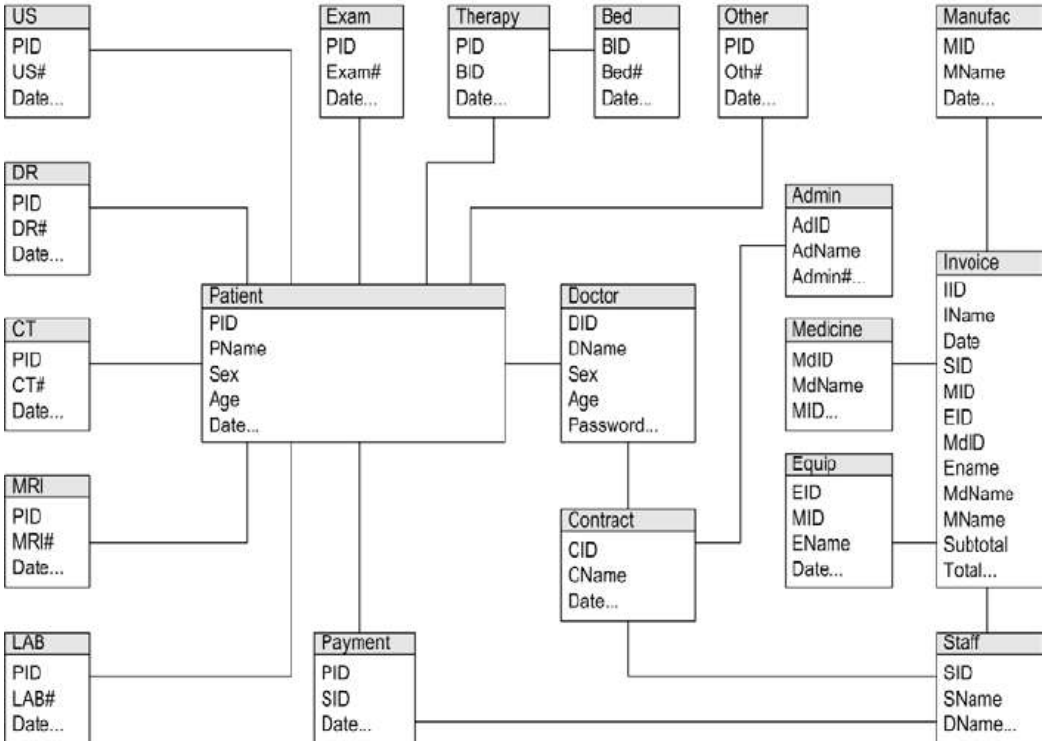



Figure 4.2: Ejemplo de estructura de una base de datos relacional.

definir que se deben borrar todas sus pruebas médicas. Encontraréis un ejemplo de una estructura de base de datos en la figura 4.2.

- No Relacionales: Son bases de datos que mantienen una estructura (o modelo) de relaciones débiles. Eso quiere decir, siguiendo nuestro ejemplo, que podríamos tener pruebas sin tener un paciente asignado. Las bases de datos no relacionales nacen por la necesidad de flexibilizar las estructuras de datos y permitir más velocidad a la hora de buscar información. Habitualmente las encontraremos en entornos que requieran GRAN VELOCIDAD a la hora de responder a peticiones a la base de datos, por ejemplo en páginas web.

### 4.1.2 Data Warehouses

Hasta ahora hemos visto, muy por encima (veréis más sobre bases de datos en módulos siguientes), los tipos de sistemas para almacenar datos. En organizaciones grandes, p.e. hospitales, habitualmente encontraremos bases de datos relacionales (aunque se están empezando a introducir las no rela-

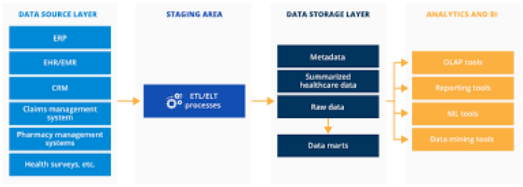



Figure 4.3: Arquitectura de un Data Warehouse.

cionales para almacenar algunos tipos muy específicos de datos). Cuando una base de datos crece y sus tablas (entidades) son accedidas desde diferentes sitios con necesidades distintas (p.e. unidades distintas), necesitamos modificar el modelo **rígido** de una base de datos relacional para adaptarnos a esas necesidades sin perder la **congruencia** que nos aporta el modelo de base de datos relacional. Para eso nacen los Data Warehouses (DW), figura 4.3 [1].

Los Data Warehouse son sistemas que contienen:

- Una, o más bases de datos
- Sistemas para mejorar el rendimiento de la base de datos (caches, metadata, etc.).
- Un conjunto de procesos que modelan la entrada de datos a la base de datos y que están adaptados a cada usuario que debe entrar datos al sistema. Estos procesos reciben el nombre en inglés de Extract Transform and Load (ETL) y se refieren a los datos de entrada.
- Un conjunto de interfaces, adaptadas a cada usuario, para poder realizar consultas a la base de datos (datos de salida).

El problema que nos encontramos con los Data Warehouses es que son estructuras complejas, que no se adaptan bien a los cambios y que requieren conocimiento muy especializado. Por ejemplo, añadir un simple campo a una tabla de la base de datos es una tarea compleja y crítica: hay que actualizar todas las entradas de la tabla que existían antes de ese campo, hay que adaptar todas las ETL que entraban datos a esa tabla y hay que cambiar las interfaces de salida para que se refleje el cambio.

El resultado es que a menudo se evita realizar cambios en el DW si no son absolutamente necesarios, creando consecuentemente una necesidad no cubierta.

Así pues nos encontramos multitud de situaciones en las que las entradas de datos a la base de datos se realizan erróneamente:

- campos sin valor,
- valores erróneos,
- campos usados de forma incorrecta,
- incongruencias entre valores de campos distintos.

Estos errores son habituales y penalizan los resultados del análisis de datos. Antes pues de hacer un análisis, hace falta comprobar y reparar nuestros datos, los procesos que componen este objetivo los llamamos preparación de datos (Data Preparation). Dividimos la preparación de datos en 5 subcategorías, dependiendo de cual es nuestro objetivo: limpieza de datos (Data Cleansing), enriquecimiento de datos (Data Enrichment), Integración de datos (Data Integration), conservación de datos (Data Curation) y Anonimización de datos (Data Anonymization). Este último caso, al tener entidad propia y ser diferente del resto, le dedicaremos el siguiente capítulo.

## 4.2 Limpieza de datos (Data Cleansing)

La limpieza de datos define todos esos procesos el objetivo de los cuales sea eliminar los valores incorrectos [2]. La limpieza de datos se divide entre las técnicas cuantitativas y las técnicas cualitativas.

### 4.2.1 Técnicas cuantitativas

Las técnicas cuantitativas se basan en el análisis matemático de los datos. Asimismo hay dos tipos principales de técnicas cuantitativas:

- Técnicas basadas en estadística: Son técnicas que usan la estadística tradicional para encontrar datos erróneos, normalmente focalizados en un único campo:
  - Análisis de máximo y mínimo: Se establecen valores máximos y mínimos para una variable y se considera que cualquier valor que los exceda es erróneo o necesita revisión. Por ejemplo, podríamos definir un mínimo y máximo para el campo edad: $0 - 120$ años.
  - Análisis de varianza: Se establece un valor de varianza máximo. Los valores que excedan este nivel se consideran erróneos o que necesitan revisión. Por ejemplo para el campo Tiempo de espera de usuarios con un mismo nivel de prioridad en Urgencias de un hospital, podríamos definir un filtro de varianza y detectar todas esas varianzas mayores que 3 veces la media.

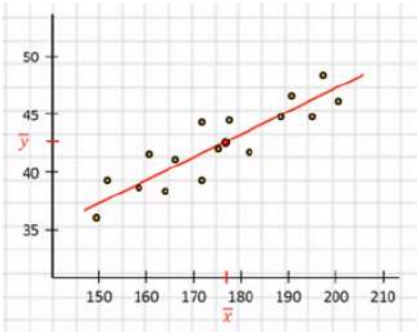



Figure 4.4: Recta de regresión de dos variables

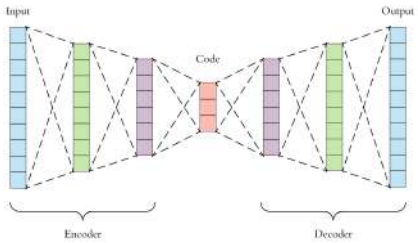



Figure 4.5: Red neuronal de tipo autoencoder

  - Análisis de distribución de probabilidad: Se escoge una distribución de probabilidad conocida y se analiza los valores del campo en función de esa distribución específica. Los valores con poca probabilidad se consideran errores o que necesitan revisión. Por ejemplo podríamos asumir una distribución normal para el valor de una prueba determinada. Aquellos valores que estén en el 5% de menor probabilidad, pueden ser erróneos o necesitan revisión.

- Técnicas basadas en aprendizaje: Son técnicas matemáticas, lineales o no, que usan algoritmos de aprendizaje sobre los datos para detectar desviaciones sobre lo esperado. Hay tres tipos de técnicas de aprendizaje:

  - Filtros: Se trata de filtros matemáticos que usan los datos ya procesados para predecir los siguientes valores. Mediante la estimación del error, se pueden detectar valores erróneos o que necesiten revisión. Es habitual encontrarlo en datos secuenciales. Un filtro muy habitual es el filtro de Kalman.
  - Técnicas de aprendizaje supervisado: Requieren un entrenamiento previo con datos correctos. Aprenden de ese conjunto y son capaces de predecir el resultado ante una nueva entrada. Con esa predicción y el resultado real, se puede establecer un error. Ese error será el parámetro que usaremos como límite para detectar entradas erróneas o que requieran revisión. Por ejemplo, la recta de regresión (lineal, figura 4.4) o la red neuronal de tipo autoencoder (no-lineal, figura 4.5) son algoritmos de aprendizaje supervisado que se usan para la limpieza de datos.

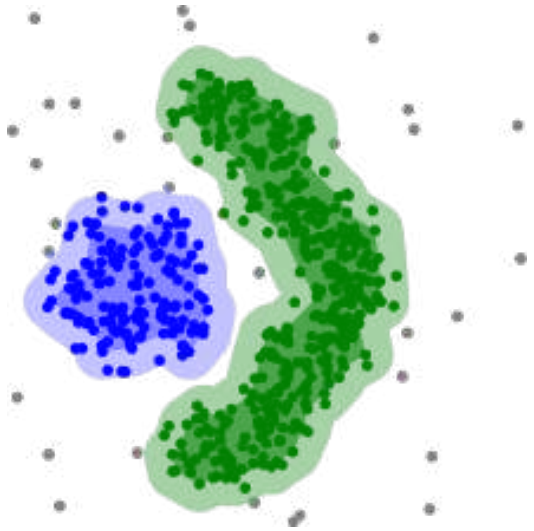


Figure 4.6: Algoritmo de tipo no supervisado. Los grupos están marcados en color (azul y verde), los posibles errores son los valores que se encuentran fuera de esos grupos.

– Técnicas de aprendizaje no supervisado: Son técnicas, habitualmente no lineales, que no requieren un conjunto de entrenamiento. En este caso se crean grupos (clusters) de valores cercanos y se consideran los valores que quedan fuera de esos grupos como erróneos o que requieren revisión (ver figura 4.6).

### 4.2.2 Técnicas cualitativas

Son técnicas basadas en relaciones entre cualidades de los atributos o patrones. Es decir, se relacionan diferentes atributos para una misma entrada buscando patrones poco habituales o erróneos. Existen dos tipos de técnicas cualitativas:

- Técnicas basadas en reglas: Se definen en forma de regla, por ejemplo "Si el valor del campo $x$ es $a$, el valor del campo $y$ tiene que ser $b$ o $c$". Se usan cuando se sabe del cierto la relación entre los campos implicados. Por ejemplo, para un atributo de diagnóstico que contenga el colesterol alto, el atributo del análisis de sangre de colesterol debe estar por encima de 220.

- Técnicas basadas en patrones: Hay distintas manera de encontrar patrones, eso es, valores de distintos atributos que se dan juntos. Estos

algoritmos determinan como de habitual es que dos o mas valores de distintos atributos se den de forma conjunta. Mediante esta probabilidad, podemos determinar que esos eventos con baja probabilidad son erróneos o requieren revisión. Por ejemplo, al principio de la pandemia de la COVID, podríamos definir un patrón entre haber viajado a países con más afectación (Italia, China, Irán, etc.) y sintomatología de gripe. Notase que este caso no se puede definir con una regla, pues no es siempre cierto, pero si que es un evento con alta probabilidad.

## 4.3 Enriquecimiento de datos (Data Enrichment)

El enriquecimiento de datos incluye todos los procesos destinados a **refinar, mejorar, o añadir valor** a un conjunto de datos. Hay dos técnicas principales de enriquecimiento de datos:

- Atributos precalculados: A veces, nos interesa precalcular algunos valores que nos facilitaran el análisis posterior. Es habitual precalcular aspectos estadísticos (p.e. varianza) o probabilísticos (p.e. probabilidad condicionada) de los datos en un nuevo campo.

- Atributos añadidos: Se trata de añadir información que no está disponible directamente en el conjunto de datos pero que se pueda obtener de otro conjunto de datos o de una simulación.

Podemos ver un claro ejemplo de enriquecimiento en los datos de evolución de la pandemia. A las altas hospitalarias, casos detectados, etc. a menudo se les han añadido datos provenientes de simulaciones estadísticas y/o probabilísticas. Otro caso habitual de enriquecimiento de datos es añadir información epigenética [3], climatológica, etc.

## 4.4 Integración de datos (Data Integration)

La integración de datos comprende los procesos destinados a **unir** conjunto de datos diferentes en un único conjuntos de datos [4].

La integración de datos es muy habitual sobretodo en el uso de bases de datos relacionales. El hecho que los datos estén almacenados en diferentes tablas (entidades), que se relacionan entre ellas, implica que a menudo necesitamos la información contenida en mas de una tabla a la hora de preparar nuestros datos para el posterior análisis.

| ID paciente | Nombre | Apellidos | Dirección | Edad |
|---|---|---|---|---|
| 66530 | Juan | García Clavo | C/ San José 3, 08035 | 55 |
| ... | ... | ... | ... | ... |

Tabla 4.1: Tabla Pacientes

| ID paciente | Prueba | Unidad | Médico | Observaciones |
|---|---|---|---|---|
| 66530 | R.Magnética | Radiología | Dra. Pérez | Mano derecha |
| ... | ... | ... | ... | ... |

Tabla 4.2: Tabla Pruebas

Para integrar los datos requerimos que haya como mínimo un campo compartido, que habitualmente recibe el nombre de *clave*. Siguiendo el ejemplo de la introducción, imaginemos que tenemos una tabla Pacientes con información básica sobre los pacientes 4.1 y tenemos una tabla Pruebas con las pruebas realizadas a los pacientes 4.2. Si queremos crear un único conjunto de datos para analizar, debemos ser capaces de unir esas dos tablas para poder trabajar con los datos de ambas a la vez. En este caso lo haremos por el campo ID paciente, que ambas tablas comparten.

En bases de datos relacionales esta unión es conocida como *join* y tiene diversas modalidades que se corresponden con diferentes operaciones sobre conjuntos. Por ejemplo, si queremos unir la tabla Pacientes y la tabla Pruebas nos podríamos hacer varias preguntas: ¿Quiero la información de pacientes que no tengan ninguna prueba? ¿Es posible, y en caso que lo sea, quiero disponer de la información de las pruebas que no tengan un paciente asignado?

Imaginemos que tenemos un conjunto $A$ que contiene todos los pacientes y un conjunto $B$ que contiene todos los pacientes que se han hecho pruebas. Si sólo me interesa la información de pacientes que se hayan hecho pruebas entonces lo que tengo es una intersección sobre los conjuntos A y B (figura 4.7).

En cambio, si quiero la información de todos los pacientes, se hayan o no hecho una prueba entonces necesito todo $A$, incluyendo la intersección con $B$. Si lo que quiero es la información de las pruebas que no tienen asignado ningún paciente (por ejemplo para detectar errores) entonces lo que necesitaré es $B$ menos la intersección de $A$ y $B$.

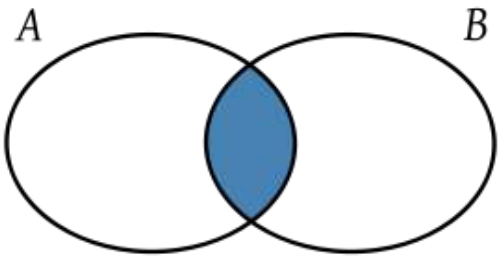



Figure 4.7: Intersección de los conjuntos $A$ y $B$.

## 4.5 Conservación de datos (Data Curation)

La conservación de datos comprende los procesos destinados a facilitar la clasificación, persistencia y accesibilidad de los datos [5]. Hay dos técnicas principales para la conservación de los datos:

- Metadatos: Se trata de datos escondidos que se añaden al conjunto de datos visibles. Estos datos pueden ser leídos por sistemas de tratamiento de datos, ya sean de gestión (como las bases de datos) o de análisis de los mismos. Un ejemplo clásico de metadatos los encontramos en las páginas web. En estas, se puede encontrar información escondida que ayuda a los buscadores (p.e. Google) a determinar los contenidos más relevantes para una búsqueda concreta. Otro ejemplo lo encontramos en los índices creados en las bases de datos para facilitar y acceder a la información. En el caso de bases de datos relacionales, es habitual crear campos únicos para cada fila (key) para evitar entradas repetidas. En el caso de bases de datos no relacionales, estos índices le indican al motor de la base de datos como ordenar estos datos en memoria para que sean de más fácil acceso. Otro ejemplo habitual de metadatos son las etiquetas o *tags* con las que se clasifican contenidos según la temática correspondiente.

- Esquemas: Se trata de relaciones predefinidas entre variables o conjuntos de datos. Por ejemplo, una base de datos relacional es un esquema, pues define las relaciones existentes entre las diferentes entidades lógicas. En el momento que el diseño de la base de datos relaciona que un paciente puede tener $n$ pruebas asociadas, en el fondo está definiendo un esquema. Estos esquemas nos ayudan a entender la

lógica que siguen los datos, y por lo tanto, a ser capaces de recuperarlos de forma más fácil y eficiente.

# Bibliografía


[1] AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of data integration*. Elsevier, 2012.

[2] Matthias Jarke, Maurizio Lenzerini, Yannis Vassiliou, and Panos Vassiliadis. *Fundamentals of Data Warehouses*. Springer-Verlag, Berlin, Heidelberg, 2nd edition, 2001.

[3] Jonathan I Maletic and Andrian Marcus. Data cleansing: Beyond integrity analysis. In *Iq*, pages 200–209. Citeseer, 2000.

[4] Merinda McLure, Allison V Level, Catherine L Cranston, Beth Oehlerts, and Mike Culbertson. Data curation: a study of researcher practices and needs. *portal: Libraries and the Academy*, 14(2):139–164, 2014.

[5] Aaron L Statham, Dario Strbenac, Marcel W Coolen, Clare Stirzaker, Susan J Clark, and Mark D Robinson. Repitools: an r package for the analysis of enrichment-based epigenomic data. *Bioinformatics*, 26(13):1662–1663, 2010.

# Chapter 5

# Anonimización

*Bernat Gastón*

## 5.1 Introducción

El acceso a los datos, y en especial a los datos de carácter médico (por su especial sensibilidad) están regidos, en el marco de la Unión Europea, por la Ley General de Protección de Datos (GDPR por sus siglas en inglés). Esta ley regula la obtención, mantenimiento y eliminación de los datos por parte de la entidad responsable, así como los derechos por parte del mismo sujeto de los datos. Además prohíbe de manera explícita la compartición de datos personales entre entidades siempre y cuando no se haya aceptado por parte del usuario de manera previa. Finalmente, este permiso debe ser detallado y explícito, estableciendo a con qué entidades se comparten los datos, qué datos y de qué manera. Así pues, es muy difícil dar a los datos personales un tratamiento que no se hubiera previsto en el momento de su obtención.

En resumen, esta ley permite analizar datos personales siempre y cuando se haya obtenido el permiso para hacerlo, pero no es sencillo darles un uso diferente en el futuro. A la práctica, esto crea silos de datos: conjuntos de datos aislados, parcialmente repetidos y con restricciones de uso. Por ejemplo, si desde este máster quisiéramos acceder a datos reales con fines docentes, seria muy difícil acceder a aquellos ya existentes, y en todo caso habría que diseñar un proceso y protocolo de obtención, mantenimiento y eliminación nuevo, pedir aprobación a los entes regulatorios y finalmente a los sujetos de los datos.

Es por ese motivo que existe la posibilidad de trabajar con datos anonimizados, esto es, datos que continúan teniendo valor analítico pero han perdido la potencialidad de personalizarlos. En este sentido la Directiva 95/46/CE [2] constituye el texto de referencia, a escala europea, en mate-

ria de protección de datos personales. Crea un marco regulador destinado a establecer un equilibrio entre un nivel elevado de protección de la vida privada de las personas y la libre circulación de datos personales dentro de la Unión Europa (UE). Con ese objeto, la Directiva fija límites estrictos para la recogida y utilización de los datos personales y solicita la creación, en cada Estado miembro, de un organismo nacional independiente encargado de la supervisión de cualquier actividad relacionada con el tratamiento de los datos personales.

En este capítulo, explicaremos el marco legal que define la anonimización de los datos, así como loas diferentes técnicas que se usan para conseguir de forma efectiva esa anonimización.

El grupo de trabajo sobre protección de datos fue creado en el artículo 29 de la Directiva 95/46/CE. Se trata de un órgano consultivo europeo independiente que aborda cuestiones relativas a la protección de datos y la intimidad. Sus funciones se describen en el artículo 30 de la Directiva 95/46/CE y el artículo 15 de la Directiva 2002/58/CE. Las correspondientes funciones de secretaría son ejercidas por la Dirección C (Derechos Fundamentales y Ciudadanía de la Unión) de la Comisión Europea, Dirección General de Justicia, B-1049 Bruselas, Bélgica, Oficina No. MO-59 02/013.

A la luz de la Directiva 95/46/CE y de otros instrumentos jurídicos pertinentes de la UE, la anonimización es el resultado de un tratamiento de los datos personales realizado para evitar de forma irreversible su identificación. En este proceso, los responsables del tratamiento deben considerar distintos aspectos y valorar todos los medios que puedan utilizarse razonablemente para la identificación de los datos (ya sea por el responsable del tratamiento o por terceros).

La anonimización implica un tratamiento posterior de los datos personales. Por tanto, debe satisfacer el requisito de compatibilidad teniendo en cuenta las circunstancias y los fundamentos jurídicos de dicho tratamiento. Por otra parte, aunque los datos anonimizados se encuentren fuera del alcance de la legislación sobre protección de datos, es posible que los interesados tengan derecho a protección en virtud de otras disposiciones legales (como las que protegen la confidencialidad de las comunicaciones). El dictamen 05/2014 expone la solidez de cada técnica aplicando tres criterios:

- ¿Se puede singularizar a una persona?
- ¿Se pueden vincular registros relativos a una persona?
- ¿Se puede inferir información relativa a una persona?

La conclusión del dictamen es que las técnicas de anonimización pueden aportar garantías de privacidad y usarse para generar procesos de anonimización eficientes, pero solo si su aplicación se diseña adecuadamente, lo que significa que han de definirse con claridad los requisitos previos (el contexto) y los objetivos del proceso para obtener la anonimización deseada al mismo tiempo que se generan datos útiles.

La solución óptima debe decidirse caso por caso y puede conllevar la combinación de diversas técnicas, aunque siempre respetando las recomendaciones prácticas que se formulan en ese documento.

Por último, los responsables del tratamiento deben ser conscientes de que un conjunto de datos anonimizado puede entrañar todavía riesgos residuales para los interesados. Efectivamente, por una parte, la anonimización y la reidentificación son campos de investigación activos en los que se publican con regularidad nuevos descubrimientos y, por otra, incluso los datos anonimizados, como las estadísticas, pueden usarse para enriquecer los perfiles existentes de personas, con la consiguiente creación de nuevos problemas de protección de datos. En suma, la anonimización no debe contemplarse como un procedimiento esporádico, y los responsables del tratamiento de datos deben evaluar regularmente los riesgos existentes.

### 5.1.1 Perspectiva legal

En la Directiva 95/46/CE, el considerando 26 hace mención a la anonimización y excluye los datos anonimizados del alcance de la legislación sobre protección de datos: *Considerando que los principios de la protección deberán aplicarse a cualquier información relativa a una persona identificada o identificable; que, para determinar si una persona es identificable, hay que considerar el conjunto de los medios que puedan ser razonablemente utilizados por el responsable del tratamiento o por cualquier otra persona, para identificar a dicha persona; que los principios de la protección no se aplicarán a aquellos datos hechos anónimos de manera tal que ya no sea posible identificar al interesado; que los códigos de conducta con arreglo al artículo 27 pueden constituir un elemento útil para proporcionar indicaciones sobre los medios gracias a los cuales los datos pueden hacerse anónimos y conservarse de forma tal que impida identificar al interesado;*

Esto implica que para anonimizar cualesquiera datos es necesario eliminar de ellos los elementos suficientes para que no pueda identificarse al interesado. El análisis de las referencias a la anonimización en los principales instrumentos jurídicos de la UE sobre protección de datos permite poner de manifiesto cuatro características fundamentales:

- La anonimización puede ser el resultado de un tratamiento de datos personales realizado para impedir de forma irreversible la identificación del interesado.

- Pueden considerarse varias técnicas de anonimización, sin que la legislación europea contenga ninguna norma prescriptiva.

- Hay que dar importancia a los elementos contextuales: debe considerarse *el conjunto de los medios que puedan ser razonablemente utilizados* para la identificación por parte del responsable del tratamiento o de un tercero, prestando especial atención a lo que se entiende, en el estado actual de la técnica, como *medios que puedan ser razonablemente utilizados* (dado el incremento de la potencia de los ordenadores y de las herramientas disponibles).

- La anonimización lleva implícito un factor de riesgo que ha de tenerse en cuenta al evaluar la validez de las técnicas de anonimización, incluidos los posibles usos de los datos *anonimizados* mediante estas, además de considerarse asimismo la gravedad y probabilidad del riesgo.

Cabe añadir que la anonimización ha de ajustarse a las restricciones legales que recuerda el Tribunal de Justicia de la Unión Europea en su sentencia sobre el asunto C-553/07 (College van burgemeester en wethouders van Rotterdam contra M.E.E. Rijkeboer) y que se refieren a la necesidad de conservar los datos en un formato identificable a fin de que puedan ejercerse, por ejemplo, los derechos de acceso por parte de los interesados. En concreto, el Tribunal señala que *el artículo 12, letra a), de la Directiva [95/46/CE] obliga a los Estados miembros a garantizar un derecho de acceso a la información sobre los destinatarios o categorías de destinatarios a quienes se comunican los datos y al contenido de la información comunicada, no sólo para el presente, sino también para el pasado. Corresponde a los Estados miembros fijar un plazo de conservación de dicha información, así como el acceso correlativo a ésta, guardando un justo equilibrio entre, por un lado, el interés del afectado en proteger su intimidad, concretamente a través de las distintas vías de intervención y de recurso previstas por la Directiva y, por otro, la carga que la obligación de dicha información puede representar para el responsable del tratamiento.*

Existe cierta confusión en torno al concepto que aparece en la legislación que menciona *el conjunto de los medios razonablemente utilizados*. En este sentido, el grupo de trabajo ya ha aclarado que la Directiva propone la razonabilidad de los medios usados como criterio para evaluar si el tratamiento de anonimización es suficientemente sólido, es decir, si la identificación es

*razonablemente* imposible [6]. Afectan directamente a la identificabilidad el contexto y las circunstancias particulares de cada caso.

### 5.1.2 Perspectiva técnica

Desde una prespectiva técnica hay que tener en cuenta dos cosas. Por una parte el análisis de los atributos. Así, clasificamos los atributos en tres tipos:

- Identificadores: Atributos que representan datos que identifican de forma única o casi única al individuo. Son identificadores los nombres y apellidos, DNI, dirección postal, correo electrónico, etc.
- Cuasi-Identificador: Atributos que representan datos que, sin que identifiquen a la persona por si mismos, si son críticos y, en conjunto, pueden identificar finalmente al individuo. Son ejemplos de cuasi-identificadores la fecha de nacimiento, la fecha y hora de la visita, el código postal, etc.
- Atributos sensibles: Atributos que no contienen información que pueda ser crítica a nivel de identificación, pero que si contienen información sensible a nivel de privacidad y son datos que no pueden ser vinculados al individuo. p.e. enfermedades, tratamientos médicos etc.

Por otra parte existen tres riesgos clave para analizar si la identificación es razonablemente imposible:

- Singularización: la posibilidad de extraer de un conjunto de datos algunos registros (o todos los registros) que identifican a una persona.
- Vinculabilidad: la capacidad de vincular como mínimo dos registros de un único interesado o de un grupo de interesados, ya sea en la misma base de datos o en dos bases de datos distintas. Si el atacante puede determinar (p. ej., mediante un análisis de correlación) que dos registros están asignados al mismo grupo de personas, pero no puede singularizar a las personas en este grupo, entonces la técnica es resistente a la singularización, pero no a la vinculabilidad.
- Inferencia: la posibilidad de deducir con una probabilidad significativa el valor de un atributo a partir de los valores de un conjunto de otros atributos.

A partir de este análisis y teniendo en cuneta el uso que se quiere dar a los datos, existen un conjunto de técnicas para proceder a la anonimización

de los datos y que se presentan en las siguientes secciones. Cabe destacar el concepto de **privacidad diferencial**. La privacidad diferencial es la automatización, en un sistema de datos como por ejemplo una base de datos, del conjunto de técnicas usadas para la anonimización. De esta manera, el sistema crea "vistas" que se pueden adaptar al nivel de accesso del usuario o al objetivo de la investigación, aplicando un conjunto u otro de técnicas para anonimizar los datos.

## 5.2 Pseudo-anonimización

La pseudo-anonimización se define en el GDPR como "el procesamiento de datos personales de tal manera que los datos ya no puedan atribuirse a un sujeto de datos específico sin el uso de información adicional, siempre que dicha información adicional se mantenga por separado y sujeta a y medidas organizativas para garantizar la no atribución a una persona identificada o identificable (artículo 4 (3b)) [7].

A diferencia de la anonimización, las técnicas de pseudo-anonimización no eximirán a los responsables del tratamiento del ámbito del GDPR por completo. Sin embargo, ayuda a las organizaciones a cumplir con sus obligaciones en materia de protección de datos, en particular los principios de 'minimización de datos' y 'limitación de almacenamiento' (artículos 5 (1c) y 5 (1e) [8]), y en el caso de procesamiento con fines de investigación contribuye a la creación de un conjunto de datos anonimizado.

Existen 5 técnicas en la pseudoanonimización:

- Borrar un atributo: Se trata de eliminar del conjunto de datos un atributo. Por ejemplo el nombre del paciente. En este caso se pierde la información contenida en este campo de forma definitiva y no se puede recuperar.

- Cifrado con clave: Se trata de cifrar un atributo del conjunto de datos. Por ejemplo el nombre del centro médico o Hospital. En este caso, la información no está disponible para el usuario de los datos pero si que se puede recuperar mediante el uso de la clave de cifrado

- Función Hash: Se trata de codificar la información de un atributo mediante una función Hash. Las funciones Hash son funciones de una sola dirección. Eso quiere decir que dados unos datos, el resultado de aplicar la función a esos datos siempre dará el mismo valor, sin embargo, dado el valor resultante no se puede recuperar el valor original. Se podría aplicar por ejemplo al código de médico. De esta manera dado el código

de médico podríamos saber que datos se le corresponden, pero sin él, no es posible determinar esa información.

Garantías ofrecidas por la pseudo-anonimización:

- Singularización: todavía es posible distinguir los registros de las personas, ya que para una entrada (persona), los valores del resto de atributos se corresponden exactamente a los originales.
- Vinculabilidad: la vinculabilidad seguirá siendo trivial entre registros que utilicen el mismo atributo pseudonimizado para referirse a la misma persona. Incluso si se utilizan diferentes atributos pseudonimizados para el mismo interesado, la vinculabilidad puede ser posible mediante otros atributos. Solo si no se puede utilizar ningún otro atributo en el conjunto de datos para identificar al interesado y si se han eliminado todos los vínculos entre el atributo original y el atributo pseudonimizado (incluso mediante la eliminación de los datos originales), no habrá referencias cruzadas obvias entre dos conjuntos de datos que utilizan diferentes atributos pseudonimizados.
- Inferencia: Los ataques de inferencia a la identidad real de un sujeto de datos son posibles dentro del conjunto de datos o en diferentes bases de datos que usan el mismo atributo pseudonimizado para un individuo, o si los seudónimos se explican por sí mismos y no enmascaran la identidad original del sujeto de datos adecuadamente.

En conclusión, la pseudo-anonimización és una gran ayuda y una de las técnicas más usadas, especialmente para los campos identificadores. Sin embargo, no garantiza por si sola ninguno de los requisitos y debe ser complementada con otras técnicas. En la Figura 5.1, se puede ver un ejemplo de uso de técnicas de pseudo-anonimización en un conjunto de datos hospitalarios.

## 5.3 Randomización

La randomización trata de introducir un componente aleatorio a los datos de manera que no se pueda revertir el proceso y conseguir los datos originales. Evidentemente, los atributos deben ser elegidos de forma cuidadosa para mantener el valor de los datos.

### 5.3.1 Añadido de Ruido

Se trata de modificar los atributos de un conjunto de datos de manera que sean menos precisos sin afectar la distribución estadística de los mismos y

| | | | |
|---|---|---|---|
| COD_EPISODIO | INTEGER | Código único de episodio | cifrado |
| COD_PACIENTE | INTEGER | Código de paciente (Número de Historia Clínica NHC) | Eliminado |
| FECHA_ATENCION | DATE | Fecha de atención DD/MM/YYYY HH24:MI:SS. Fecha en la que el paciente es atendido por el personal asistencial | |
| SERVICIO_INGRESO | VARCHAR2(8 BYTE) | Servicio de ingreso | Eliminado |
| COD_MOTIVO_ATENCION | VARCHAR2(8 BYTE) | Código motivo de atención | Cifrado |
| DES_MOTIVO_ATENCION | VARCHAR2(60 BYTE) | Descripción motivo de atención | Eliminado |
| SERVICIO_ALTA | VARCHAR2(8 BYTE) | Servicio de alta | Eliminado |
| MEDICO_ALTA | INTEGER | Médico de alta | Cifrado |
| COD_MOTIVO_ALTA | VARCHAR2(8 BYTE) | Código motivo de alta | Cifrado |
| DES_MOTIVO_ALTA | VARCHAR2(60 BYTE) | Descripción motivo de alta | Eliminado |
| OBSERVACIONES | VARCHAR2(255 BYTE) | Observaciones | Eliminado |
| COD_HOSPITAL | VARCHAR2(8 BYTE) | Código del hospital al que se ingresa o se traslada el paciente | Cifrado |
| DES_HOSPITAL | VARCHAR2(200 BYTE) | Descripción del hospital al que se ingresa o se traslada el paciente | Eliminado |

Figure 5.1: Ejemplo de un informe de anonimización sobre datos hospitalarios

mantener el valor para el posterior análisis. Por ejemplo se podría introducir un ruido de $+/-5cm$ en la altura de los pacientes. Este valor por si solo probablemente no cambia sustancialmente el valor de los datos (siempre y cuando no se esté haciendo un estudio precisamente de la altura) a la vez que introduce un nivel alto de anonimización.

Garantías ofrecidas por el añadido de ruido:

- Singularización: aún es posible distinguir los registros de un individuo (quizás de una manera no identificable) aunque los registros sean menos confiables.
- Vinculabilidad: aún es posible vincular los registros de la misma persona, pero los registros son menos confiables y, por lo tanto, un registro real se puede vincular a uno agregado artificialmente (es decir, el resultado de añadir el ruido). En algunos casos, una atribución incorrecta puede exponer a un sujeto de datos a un nivel de riesgo significativo e incluso mayor que uno correcto.
- Inferencia: Los ataques de inferencia pueden ser posibles, pero la tasa de éxito será mucho menor y algunos falsos positivos (y falsos negativos) son plausibles.

En conclusión, se trata de una muy buena técnica para bajar mucho las probabilidades de cualquiera de los 3 tipos de riesgo.

### 5.3.2 Permutación

Consiste en intercambiar algunos atributos entre diferentes personas. Por ejemplo, se podría intercambiar los códigos postales que corresponden a una misma región.

Garantías ofrecidas por la permutación:

- Singularización: al igual que con el añadido de ruido, aún es posible distinguir los registros de una persona, pero los registros son menos fiables.
- Vinculabilidad: si la permutación afecta atributos y cuasi-identificadores, puede evitar la vinculación correcta de atributos tanto interna como externamente a un conjunto de datos, pero aún así permitir la vinculación incorrecta, ya que una entrada real puede asociarse a un sujeto de datos diferente.
- Inferencia: las inferencias aún se pueden extraer del conjunto de datos, especialmente si los atributos están correlacionados o tienen fuertes relaciones lógicas; sin embargo, sin saber qué atributos se han permutado, el atacante debe considerar que su inferencia se basa en una hipótesis errónea y, por lo tanto, solo es posible la inferencia probabilística.

En conclusión, la permutación es una gran defensa contra la inferencia.

## 5.4 Generalización

La generalización trata de eliminar o modificar algunas entradas para evitar que atributos con valores poco comunes sirvan para identificar a un individuo. Por ejemplo, imaginemos un conjunto de datos de pacientes que contienen el nombre de su enfermedad a la vez que contienen otros datos que podrían ser críticos (p.e. domicilio o la fecha y hora de la visita, etc.). En el caso de enfermedades comunes, la cantidad de registros associados será grande, y por lo tanto no identificable, pero qué pasa con las enfermedades raras? Si ese registro corresponde a pocas personas es fácil identificar a los individuos.

### 5.4.1 Agregación y $k$-anonimidad

La agregación consiste en modificar algunos atributos de manera que la granularidad de la información sea más general y se conformen grupos de entradas,

para un mismo valor del atributo, más pobladas. Por ejemplo, se puede agregar la información de la ciudad para convertirla en una información de región. (p.e. comarca o provincia).

La $k$-anonimidad, a diferencia de la agregación, no es estrictamente una técnica de anonimización, sino una prueba (test) del nivel de anonimización de un conjunto de datos. Sin embargo, si decimos que un conjunto de datos son $k$-anonimos, podemos aplicar los mismos análisis sobre la Singularización, Vinculabilidad e Inferencia. La unidad de evaluación y estudios tecnológicos de la agencia española de protección de datos [9] lo describe de la siguiente manera: *"Se dice que un individuo es k-anónimo dentro del conjunto de datos en el que se encuentra incluido si, y sólo si, para cualquier combinación de los atributos cuasi-identificadores asociados, existen al menos otros $k-1$ individuos que comparten con él los mismos valores para esos mismos atributos [10].Hay que tener en cuenta que la k-anonimidad no se centra en los atributos sensibles de los registros, sino en los atributos cuasi-identificadores que pueden permitir la vinculación."*

Además nos añade que: *"De este modo, la probabilidad de identificar a un individuo concreto en base a ese conjunto de cuasi-identificadores es como máximo 1/k, por lo que para garantizar un bajo riesgo de reidentificación debe garantizarse un valor mínimo de k cuando se pretende llevar a cabo el diseño de un proceso de anonimización o disociación de datos."*

La manera de mejorar la $k$-anonimidad es mediante la agregación de los atributos cuasi-identificadores hasta conseguir que como mínimo $k$ entradas compartan los mismos valores de los cuasi-identificadores.

Garantías ofrecidas por la agregación:

- Singularización: debido a que ahora $k$ usuarios comparten los mismos atributos, ya no debería ser posible señalar a un individuo dentro de un grupo de $k$ usuarios.

- Vinculabilidad: si bien la vinculabilidad es limitada, sigue siendo posible vincular registros por grupos de $k$ usuarios. Entonces, dentro de este grupo, la probabilidad de que dos registros correspondan a los mismos cuasi-identificadores es $1/k$ (que podría ser significativamente más alta que la probabilidad de que tales entradas sean desvinculables).

- Inferencia: El principal defecto del modelo $k$-anonimato es que no evita ningún tipo de ataque de inferencia. De hecho, si todos los $k$ individuos están dentro de un mismo grupo, entonces si se sabe a qué grupo pertenece un individuo, es trivial recuperar el valor de esta propiedad.

En conclusión, la agregación es una herramienta muy pontente para proteger de la singularización y la vinculabilidad, pero no de la inferencia.

### 5.4.2 Diversidad-$l$ y cercanía-$t$

Debido a que la $k$-anonimidad no protege de los ataques de inferencia, hace falta definir mejores métricas para evaluar el nivel de anonimización. En este sentido aparece la diversidad-$l$ [11]. De hecho, hay varios ataques que pueden romper fácilmente la $k$-anonimidad:

- Ataque de homogeneidad: este ataque aprovecha el caso en el que todos los valores de un atributo sensible dentro de un conjunto de $k$ registros son idénticos. En tales casos, aunque los datos se hayan $k$-anonimizado, el valor sensible para el conjunto de $k$ registros puede predecirse exactamente.

- Ataque de conocimiento de fondo: este ataque aprovecha una asociación entre uno o más atributos de cuasi-identificador con el atributo sensible para reducir el conjunto de valores posibles para el atributo sensible. Por ejemplo, precisamente en [11] demostraron que saber que los ataques cardíacos ocurren a una frecuencia reducida en pacientes japoneses podría usarse para reducir el rango de valores para un atributo sensible de la enfermedad de un paciente.

En [12] se expone el principio de la diversidad-$l$: *"Se dice que una clase de equivalencia tiene diversidad l si hay al menos l valores "bien representados" para el atributo sensible. Se dice que una tabla tiene diversidad l si cada clase de equivalencia de la tabla tiene diversidad l".* Es decir, para cualquier grupo de quasi-identificadores iguales, los atributos sensibles de ese grupo deben tener como mínimo $l$ valores diferentes.

Sin embargo, la diversidad-$l$ no tiene en cuenta la distribución probabilística de los atributos sensibles. Para mejorar este modelo aparece la cercanía-$t$. En [12] se define la cercanía-$t$ como: *"Se dice que una clase de equivalencia tiene cercanía-t si la distancia entre la distribución de un atributo sensible en esta clase y la distribución del atributo en toda la tabla no es más que un umbral t. Se dice que una tabla tiene cercanía-t si todas las clases de equivalencia tienen cercanía-t".*

Garantías ofrecidas por la diversidad-$l$ y la cercanía-$t$:

- Singularización: al igual que la $k$-anonimidad, la diversidad-$l$ y la cercanía-$t$ pueden garantizar que los registros relacionados con un individuo no se puedan destacar en la base de datos.

- Vinculabilidad: la diversidad-$l$ y la cercanía-$t$ no son una mejora sobre el $k$-anonimato con respecto a la desvinculación. El problema es el mismo que con cualquier grupo: la probabilidad de que las mismas

| | | | |
|---|---|---|---|
| COD_PEDIDO | INTEGER | Código del pedido | Cifrado |
| PER_PETICIONARIA | INTEGER | Persona peticionaria | Cifrado |
| UNI_PETICIONARIA | VARCHAR2(16 BYTE) | Unidad peticionaria | Diversidad 10 |
| PER_RECEPTORA | INTEGER | Persona receptora | Cifrado |
| UNI_RECEPTORA | VARCHAR2(16 BYTE) | Unidad receptora | Diversidad 10 |
| FECHA_PEDIDO | DATE | Fecha petición | Agregación simple |
| HORA_PEDIDO | VARCHAR2(8 BYTE) | Hora petición | |
| FECHA_RESOLUCION | DATE | Fecha resolución | |
| HORA_ RESOLUCION | VARCHAR2(8 BYTE) | Hora resolución | |
| COD_PACIENTE | INTEGER | Código de paciente | Cifrado |
| VISITA_PETICION | INTEGER | Fecha límite en la que resolver la petición | |
| VISITA_RESULTADO | INTEGER | Hora límite en la que resolver la petición | |
| ESTADO_PEDIDO | VARCHAR2(8 BYTE) | Estado del pedido | |

Figure 5.2: Ejemplo de un informe de anonimización sobre datos hospitalarios

entradas pertenezcan a un mismo sujeto de datos es mayor que $1/N$ (donde $N$ es el número de sujetos de datos en la base de datos).

- Inferencia: La principal mejora de la diversidad-$l$ y la cercanía-$t$ sobre la $k$-anonimidad es que ya no es posible configurar ataques de inferencia contra una base de datos $l$-diversa o $t$-cerrada con un 100% de confianza.

En la Figura 5.2 se muestra un ejemplo de informe de anonimización de datos hospitalarios donde hay atributos sensibles con diversidad-10.

# Bibliografía


[1] Agencia española de protección de datos, unidad de evaluación y estudios tecnológicos. la k-anonimidad como medida de la previacidad. https://www.aepd.es/sites/default/files/2019-09/nota-tecnica-kanonimidad.pdf. Accessed: 2021-10-18.

[2] Directiva 95/46/ce del parlamento europeo y del consejo, de 24 de octubre de 1995, relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos. https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:es:HTML. Accessed: 2021-10-18.

[3] General data protection law (gdpr) of the european union. article 4. https://www.privacy-regulation.eu/en/article-4-definitions-GDPR.htm. Accessed: 2021-10-18.

[4] General data protection law (gdpr) of the european union. article 5. https://www.privacy-regulation.eu/en/article-5-principles-relating-to-processing-of-personal-data-GDPR.htm. Accessed: 2021-10-18.

[5] General data protection law (gdpr) of the european union. recital 26. https://www.privacy-regulation.eu/en/recital-26-GDPR.htm. Accessed: 2021-10-18.

[6] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and -diversity.

[7] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24, 2006.

[8] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

# Chapter 6

# Compresión de Datos I

*Joan Bartrina*

A lo largo de esta unidad vamos a ver de forma superficial bastantes conceptos relacionados con la compresión de datos. Motivaremos su aplicación, introduciremos distintas unidades de almacenamiento, veremos algunas definiciones sobre imágenes y vídeos. Además, se presentaran distintos términos relacionados con la eficiencia de compresión, como comparar dos imágenes (en términos de calidad) y se introducirán distintas técnicas de compresión con y sin pérdida.

## 6.1 Por qué la compresión de datos?

Actualmente los sistemas de adquisición de datos son más usados en el día a día de un centro médico. Des de la adquisición de datos en los procesos de generación de informes médicos, facturación y sistemas de gestión de citas, etc, hasta la recolección de imágenes médicas como radiografías, mamografías, tomografías computacionales, angiografias, etc. Uno de los principales problemas de los centros médicos es el de cómo almacenar toda esta información de forma adecuada.

De un modo muy general en flujo de los datos en un sistema de compresión es el descrito por la figura 6.1. Los informes, archivos xml, facturas, albaranes, imágenes, y vídeos son comprimidos mediante un "compresor". Este compresor debe de ser específico para cada uno del tipo de datos a comprimir, ya que no existe un sistema de compresión genérico para cualquier tipo de datos (para más información se sugiere una lectura al Pigeon Theorem [13]). Un vez comprimidos los datos, estos se encontraran en un formato específico como: zip, rar, jpeg, jpg, jp2, avi, mpeg, mp4, mp5, hevc, etc. Para poder visualizar los datos correctamente deberemos descomprimirlos

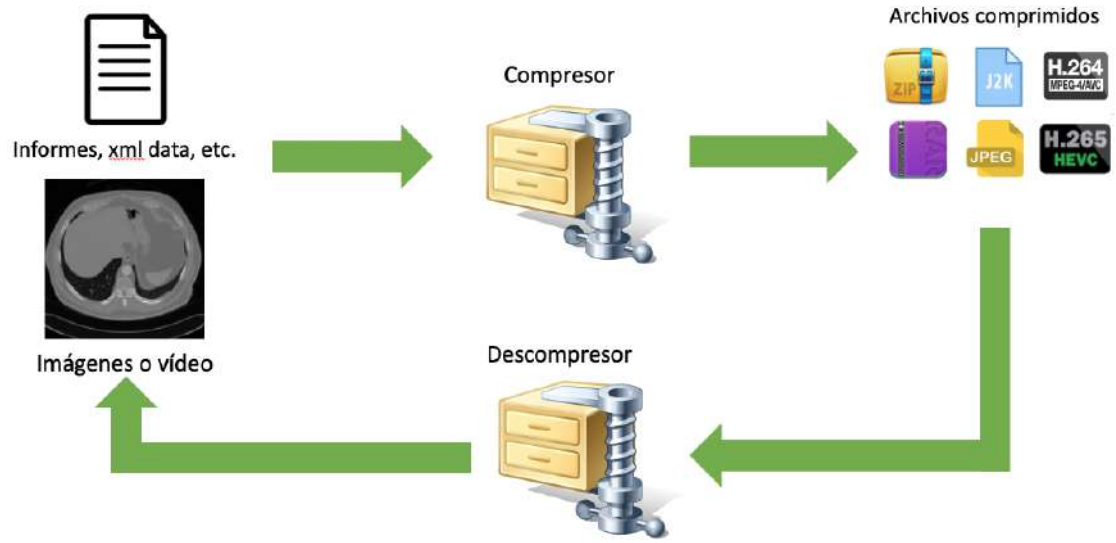



Figure 6.1: Flujo de datos.

previamente, motivo por el cual es esencial conocer el sistema de compresión utilizado, para poder descomprimir el archivo y poder visualizar los datos de forma correcta.

## 6.2 Unidades de almacenamiento en sistemas informáticos

En los sistemas informáticos los datos se almacenan utilizando bits, un bit es un 0 o 1. Una secuencia de 8 bits es 1 byte, 1024 bytes es un kilobyte (Kb), 1024 kb son un megabyte (Mb). En [14] se describen las distintas unidades de medida para sistemas informáticos.

Un secuencia de 0s y 1s nos permite almacenar y/o representar los datos deseados. Por ejemplo, en el caso de un archivo de texto debemos almacenar distintos caracteres: a, A, b, B, c, C, 1, 2, 3, 4, @, #, $, etc.

Para almacenar estos datos debemos utilizar una codificación específica, por ejemplo y para facilitar la comprensión del resto del documento utilizaremos la codificación ASCII [15]. Esta codificación almacena los datos que se van a imprimir por la pantalla, ya sean números o caracteres, en secuencias de 8 bits. La tabla 6.1 muestra para algunos caracteres, el valor decimal, y la secuencia de 0s y 1s, conocida como codificación binaria, que se almacena. Para los caracteres utilizaremos la codificación ASCII (Glyph 1967).

| Carácter | Codificación decimal | Codificación binaria |
|---|---|---|
| @ | 64 | 0100 0001 |
| A | 65 | 0100 0001 |
| C | 67 | 0100 0011 |
| P | 80 | 0101 0000 |
| S | 83 | 0101 0011 |
| 0 | 48 | 0011 0000 |
| 1 | 49 | 0011 0001 |
| . | . | . |
| . | . | . |
| . | . | . |

Tabla 6.1: Algunos ejemplos de la relación entre caracteres, valor decimal y codificación binaria para ASCII (Glyph 1967).

Las ventajas de la utilización de la compresión de datos son bastante directas, ya que esta nos permite almacenar en el mismo espacio (p.e Megabytes) más información, hecho que nos permite transmitir la misma información en un menor tiempo o utilizando menos ancho de banda (redes 3G, 4G o 5G). De aquí es importante destacar que lo importante es como se almacena una misma información (p.e una secuencia de bits) de modo que esta se pueda representar con una secuencia binaria de menor longitud. Para ello, utilizamos una codificación. Es importante diferenciar entre datos e información. Por ejemplo, supongamos que alguien nos envía dos veces una mismo correo electrónico con una imagen médica que ocupa 500MB. Tenemos 2 veces la misma imagen, ocupando 1000MB de datos pero sólo 500MB nos aportan información.

Calcular lo que no puede ocupar un archivo de texto o una imagen dependerá de dos factores: 1) cantidad de caracteres para un texto o cantidad de píxeles o samples para una imagen. Y, 2) tipo de datos almacenados en un carácter o píxel. Vamos a ver unos ejemplos.

1. **Archivo de texto:** supongamos que tenemos un archivo de 100 páginas, cada página contiene 40 lineas y cada linea 50 caracteres. El número de caracteres del libro será de $800 \times 50 \times 60 = 2400000$. Si cada carácter se almacena utilizando la codificación ASCII necesitaremos 8 bits para cada carácter, por lo que le número de bits necesarios será de $2400000 \times 8 = 19200000$ bits. Podemos cambiar la medida de capacidad a megabyte (Mb) para hacerlo más inteligible a

$$MB = \frac{bits}{8 * 1024 * 1024} \tag{6.1}$$

De modo que, el libro ocuparía un espacio en disco **sin comprimir** de 2.28 Mb.

2. **Imagen:** en este caso, en lugar de un archivo de texto disponemos de una tomografía computacional (CT). Las CTs se caracterizan por ser una secuencia de imágenes 2D obtenidas de una sección del cuerpo humano. Si cada una de las secuencias 2D ocupa $1024 \times 1024$ y esta formada por 220 imágenes 2D, el total de muestras (samples) de la CT son $1024 \times 1024 \times 220 = 230686720$.

   Ahora nos falta por definir la cantidad de información almacenada en una muestra. En este caso, supongamos que el dispositivo adquiere datos que van de un rango comprendido entre el 0 y 1024. Para conocer cuantos bits necesitamos para almacenar cada muestra utilizaremos el concepto de entropía [16], concretamente la siguiente función la función:

   $$\log_2(\text{rango}) = \text{bits}. \tag{6.2}$$

   En donde el rango es la diferencia entre el valor máximo y mínimo adquirido por el dispositivo, en nuestro ejemplo máximo es 1024 y mínimo 0, de modo que necesitaremos $log_2(1024) = 10$ bits para almacenar cada muestra[1]. Como no podemos utilizar $8 + 2$ bits para almacenar una muestra, tendremos que guardar cada una de ellas en 2 bytes. En consecuencia, al final, la CT que hemos definido ocupa un total de

   $$\frac{230686720 * 2 * 8}{8 * 1024 * 1024} = 440 \text{ Mb} \tag{6.3}$$

3. **Vídeo:** en los vídeos es importante tener en cuenta el concepto de "frame rate". El frame rate nos indica el nombre de imágenes que vemos por segundo, en inglés frames per second (fps). Por ejemplo, si disponemos de una angiografia de 1024 filas y 1024 columnas con una profundidad de 12 bits per sample y un fps de 24 y de una duración de 5 minutos, esta angiografia sin comprimir ocupará

   $$\frac{1024 * 1024 * 2 * 24 * 5 * 60}{8 * 1024 * 1024 * 1024} = 1,75 \text{ Gb} \tag{6.4}$$

Ahora bien, es importante utilizar las técnicas de compresión adecuadas en función de los datos a comprimir. Es muy diferente comprimir datos de texto, imágenes o vídeos.

[1]Este mismo concepto se aplica al almacenar texto, donde si tenemos que el valor máximo es 127 y el mínimo 0, necesitaremos $log_2(127) = 8$ bits para cada letra a almacenar.

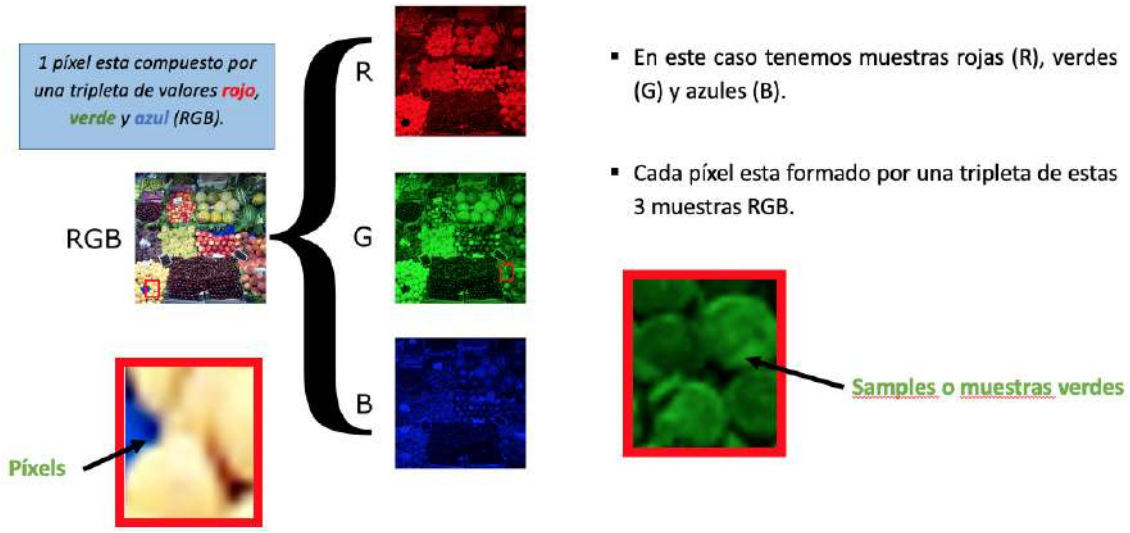



Figure 6.2: Diferencia entre píxel y sample.

### 6.2.1 Sample vs Píxel

A lo largo de esta sección hemos hablado de sample o muestra, pero habitualmente en imágenes digitales hablamos de píxeles. Es importante distinguir la diferencia entre estos dos términos. Mientras que un sample es una muestra única, un píxel es un conjunto de samples de una misma posición espacial. Supongamos una imagen como un vector bidemensional $X_{ij}$, donde $i$ y $j$ identifican las filas y columnas, respectivamente. Cada uno de los $X_{ij}$ se conoce como sample, mientras que el conjunto de valores del vector $X_i$ se conoce como píxel. La figura 6.2 muestra un ejemplo visual de la diferencia entre píxel y sample.

## 6.3 Eficiencia de la compresión

Para medir la eficiencia de las distintas técnicas de compresión se utilizan diferentes métricas, las dos más comunes son el "Compression Ratio (CR)" y los "Bits per sample (bps)". El CR que se define como el ratio entre los datos sin comprimir y comprimidos, y es expresa mediante la siguiente fórmula:

$$CR = \frac{\text{Uncompressed Size}}{\text{Compressed size}}, \tag{6.5}$$

por otro lado los bps nos indican los bits necesarios para representar cada

muestra, y se define como:

$$bps = \frac{\text{Compressed Size (in bits)}}{\text{Number of samples to compress}}, \tag{6.6}$$

Habitualmente los sistemas de compresión **sin pérdida** o inglés **lossless**, obtienen unas tasas de compresión de 3 a 1. Es decir necesitan 1 bit para cada 3 de la muestra original, de modo que su CR es aproximadamente 3. En algunos casos esta tasa de compresión no es suficiente para ello se recorre a utilizar técnicas de **compresión con pérdida** o en inglés **lossy**. Estas técnicas obtienen CR de 4, 5, 10, 100 o más, pero en contrapartida introducen cierta distorsión por lo que no se pueden recuperar los datos originales. Evidentemente a mayor CR, más distorsión y menos fiabilidad sobre los datos originales tendremos. Veamos dos ejemplos a partir de los datos definidos anteriormente.

1. **Compresión texto:** Tenemos que cada uno de los caracteres se almacena utilizando 8 bits. Si necesitamos almacenar la palabra "CASA", nuestra computadora almacenara "67,65,83,65" en codificación decimal y en codificación binaria tendremos las siguientes secuencias "01000011, 01000001, 01010011, 01000001", siguiendo la codificación de la tabla 6.1.

   Supongamos: 1) que hemos definido un sistema de compresión lossy, obtenemos CR $>$ 3, pero no somos capaces de recuperar los datos originales. Y, 2) que la pérdida introducida no nos permite recuperar los 2 últimos bits de cada carácter. De modo que, al descomprimir recuperaremos los siguientes datos:

   "010000XX,
   010000XX,
   010100XX,
   010000XX",

   donde la X son bits que no conocemos, de modo que los debemos "suponer". El hecho de desconocer estos bits NO nos permite recuperar los datos originales, por lo que introduciremos algún tipo de pérdida de información. En el caso de la compresión de texto, vamos a ver de forma muy simple que perder información no es asumible. Por ejemplo, si los datos desconocidos los recuperamos todos como 0 recuperaremos los siguientes datos:

   "01000000,
   01000000,

01010**00**,
01000**00**",

En decimal esto seria equivalente a "64, 64, 80, 64". Utilizando la codificación ASCII [15] (ver tabla 6.1) obtenemos los siguientes caracteres "@, @, P, @". Como se puede observar introducir pérdida en un archivo de texto recupera unos datos que son totalmente distintos a los originales, haciendo que las técnicas lossy no sean factibles para la compresión de texto.

2. **Compresión imágenes:** en este caso disponemos de la imagen CT descrita anteriormente que ocupa 440 Mb. Cada uno de las muestras o píxeles se encuentra almacenado en 16 bits, de modo que para representar un valor de muestro de 1323 almacenaremos la secuencia binaria "**0000** 0101 0010 1011", donde los 4 primeros bits son conocidos como bits de relleno.

   La imagen original ocupa 16 bps, ya que por cada sample necesitamos 16 bits. La representación binaria de los valores decimales nos permite ir refinando el valor decimal a medida que conocemos más bits. Esto traducido en compresión es utilizado por las técnicas lossy para introducir pérdida degradando progresivamente la imagen.

   Supongamos: cómo en el ejemplo anterior que 1) hemos definido un sistema de compresión lossy, obtenemos CR $> 3$, pero no somos capaces de recuperar los datos originales. Y, 2) que la pérdida introducida no nos permite recuperar los N últimos bits de cada carácter. La tabla 6.2 muestra la relación entre los N bits que NO se conocen –debido a la pérdida introducida por el sistema de compresión–, la secuencia binaria obtenida, los bps necesarios para los datos, el valor decimal recuperado y el error que estamos cometiendo respecto al valor original de 1323. Es evidente que nivel analítico el hecho de introducir pérdida hace que los valores no se recuperen de forma correcta, pero a nivel visual que impacto tiene? hasta que punto podemos ir introduciendo pérdida?

   La Figura 6.3 (a) muestra una componente (slice) de una CT. Mientras que las figuras (b), (c) y (d) muestran la misma slice habiendo introducido una pérdida de 2, 6 y 8 bits. Se puede apreciar, que a nivel visual una pérdida de 2 e incluso 6 bits no produce ningún deterioro significativo a nivel visual. Ahora bien, con una pérdida de 8 bits si podemos apreciar pérdida de calidad visual –sobretodo si hacemos zoom–.

| N | bps | Secuencia binaria | Valor decimal recuperado | Error |
|---|---|---|---|---|
| 0 | 16 | 0000 0101 0010 1011 | 1323 | 1323 - 1323 = 0 |
| 1 | 15 | 0000 0101 0010 101**0** | 1322 | 1323 - 1322 = 1 |
| 2 | 14 | 0000 0101 0010 10**00** | 1320 | 1323 - 1320 = 3 |
| 3 | 13 | 0000 0101 0010 1**000** | 1320 | 1323 - 1320 = 3 |
| 4 | 12 | 0000 0101 0010 **0000** | 1312 | 1323 - 1312 = 11 |
| 5 | 11 | 0000 0101 001**0 0000** | 1312 | 1323 - 1312 = 11 |
| 6 | 10 | 0000 0101 00**00 0000** | 1280 | 1323 - 1280 = 43 |
| 7 | 9 | 0000 0101 0**000 0000** | 1280 | 1323 - 1280 = 43 |
| 8 | 8 | 0000 0101 **0000 0000** | 1280 | 1323 - 1280 = 43 |
| 9 | 7 | 0000 010**0 0000 0000** | 1024 | 1323 - 1024 = 299 |
| 10 | 6 | 0000 01**00 0000 0000** | 1024 | 1323 - 1024 = 299 |
| 11 | 5 | 0000 0**000 0000 0000** | 0 | 1323 - 0 = 11 |

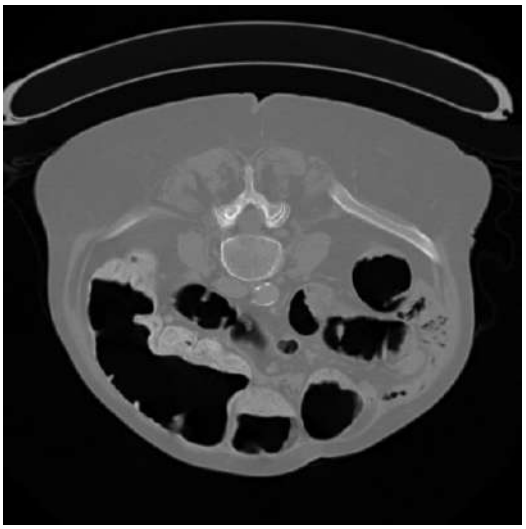
Tabla 6.2: Ejemplo de pérdida de información según la cantidad de bits desconocida debido a la compresión lossy.

Para medir el error introducido durante el proceso de compresión sobre los datos originales utilizaremos métricas de distorsión, concepto que abordaremos en la siguiente sección.

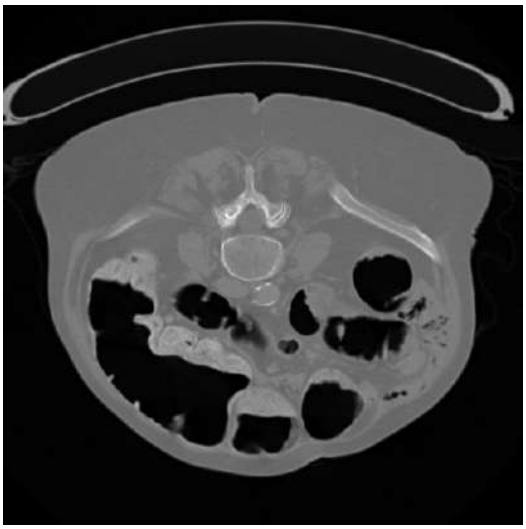
*El resto de la materia de las unidades de compresión se centraran, principalmente, en la compresión de imágenes y/o vídeos. Esto viene motivado ya que los datos generados por los sistemas sanitarios son mayoritariamente imágenes o vídeos, siendo estos el gran problema a solucionar en cuanto a un almacenamiento y transmisión por la red lo más eficiente posible.*

## 6.4 Principios básicos para la compresión

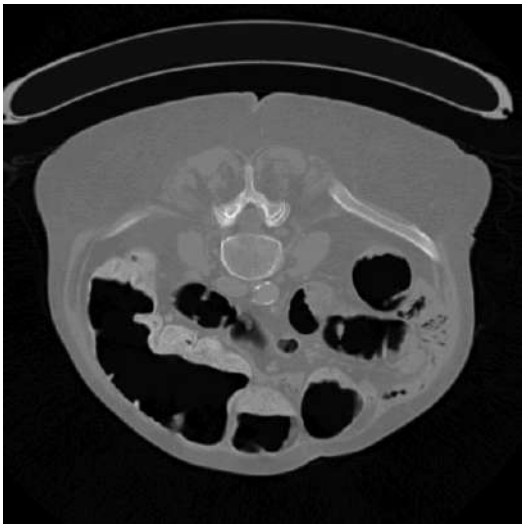
El objetivo de la compresión es la de mostrar la misma información utilizando una codificación que a su vez emplee menos símbolos y en consecuencia menos



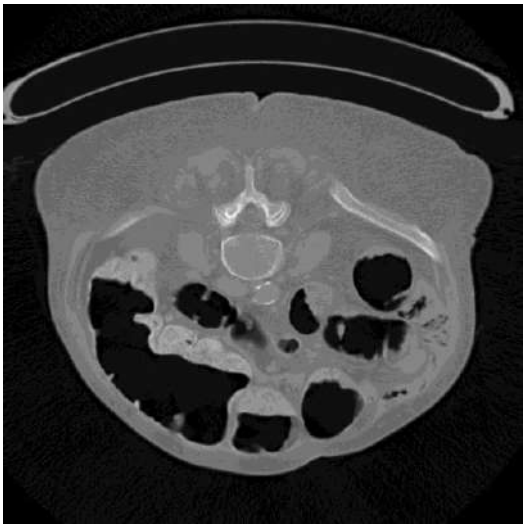
(a) imagen original



(b) imagen sin conocer los 2 últimos bits



(c) imagen sin conocer los 6 últimos bits



(d) imagen sin conocer los 8 últimos bits

Figure 6.3: Ejemplo visual recuperación imágenes.

espacio que la original. De modo que la codificación convertirá un carácter de un lenguaje natural en un símbolo de otro sistema de representación.

La información contenida en un mensaje es proporcional a la cantidad de bits que se requieren como mínimo para representar al mensaje. Para medir la información que nos aporta un mensaje utilizamos el concepto de entropía. La entropía indica la cantidad de bits por símbolo necesarios para representar esa información. Este concepto ya lo hemos introducido anteriormente pero

ahora vamos a ver como se calcula.

La entropía de una señal $X$ se define como:

$$\mathrm{H}(X) = -\sum_i p(X_i)\ log_2\ p(X_i), \tag{6.7}$$

donde $p(X_i)$ es la probabilidad del símbolo $i$.

Por ejemplo, supongamos que queremos codificar el mensaje "ABAB", en este caso tenemos un mensaje de 4 caracteres de 2 símbolos distintos la "A" y la "B". De modo que la probabilidad de B en el mensaje es $P(\mathrm{A}) = \frac{2}{4} = \frac{1}{2}$ y para la "B" los mismo $P(\mathrm{B}) = \frac{2}{4} = \frac{1}{2}$. De modo que la entropía para este mensaje seria de

$$\mathrm{H}(X) = -(\frac{1}{2}log_2\frac{1}{2} + \frac{1}{2}log_2\frac{1}{2}) = 1 \text{ bits por sample (bps)}, \tag{6.8}$$

de modo que podemos codificar cada símbolo con 1 bit, (1 bps). De modo que la entropia nos indica los bits necesarios para codificar cada símbolo. En este caso, si asignamos un 0 a la "A" y un 1 a la "B" el mensaje codificado seria 0101. De modo que hemos codificado el mensaje ABAB en 0101, pasando de necesitar 8 * 4 bits = 4 bytes a sólo 4 bits.

## 6.5 Redundancia y compresión

La **redundancia** es el principio básico de la compresión. Para ser capaces de poder codificar de forma que se pueda representar la misma información con menos datos es necesario encontrar la redundancia y explotarla de alguna forma. De forma muy simple podemos entender la redundancia como **porciones del mensaje predictibles a partir de porciones anteriores**.

Supongamos que tenemos la siguiente secuencia de caracteres a codificar:

AAAAAAAAAA BBBBBBBBBB XXXXXXXXXX TTTTTTTTTT

en este caso tenemos un total de 43 caracteres 10 As, 10 Bs, 10 Xs, 10 Ts y 3 espacios en blanco. Cada carácter se almacena en 1 byte de modo que necesitamos 43 bytes, y estamos utilizando un total de 8 bps (bits per sample).

Uno de las primeras técnicas de compresión fué el **Run Length Encoding** (RLE) [17]. El RLE tanto se utiliza para texto como imágenes o vídeos. El RLE consiste, básicamente en substituir secuencias de caracteres por un único carácter seguido del número de repeticiones. Para la secuencia anterior tendríamos:

*10A* *10B* *10X* *10T*

donde los * se utilizan como elementos separadores de las palabras código. A modo de ejemplo *10A* nos indica que tenemos 10 As consecutivas, seguidas de un espacio en blanco, a continuación 10 Bs consecutivas, seguidas de otro espacio en blanco, y así sucesivamente. En este caso, observamos que con sólo 20 bytes podemos codificar la secuencia original. 20 bytes / 43 samples = 3,72 bps, mucho menor que 8 bps.

Ahora bien que pasaría si tuviésemos la siguiente secuencia a codificar:

AA AA AA AA AA BB BB BB BB BB XX XX XX XX XX TT TT TT TT TT

esta secuencia esta formada por 69 caracteres y su codificación con RLE no daría la siguiente salida:

*2A* *2A* *2A* *2A* *2A* *2B* *2B* *2B* *2B* *2B* *2X* *2X* *2X* *2X* *2X* *2T* *2T* *2T* *2T* *2T*

En este caso se puede observar fácilmente que el RLE en lugar de comprimir lo que hace es expandir, es decir necesitamos más bytes para representar la información con la codificación de el RLE que si no la hubiésemos codificado. Después del RLE obtenemos una secuencia que requiere de 99 bytes. 99 bytes / 69 samples = 11,47 bps que es mayor a 8 bps, de modo que estamos expandiendo!!.

Fácilmente se puede observar que el problema del RLE es cuando encuentra muchas palabras y no ha una secuencia seguida de caracteres. Para evitar estos problemas en 1984 Terry Welch desarrolla el LZW (Lempel-Ziv-Welch) [17] en donde se define un diccionario que busca de secuencias de símbolos (busca patrones), sustituyendo las palabras aprendidas por referencias al diccionario. Consideremos nuevamente la secuencia a codificar:

AA AA AA AA AA BB BB BB BB BB XX XX XX XX XX TT TT TT TT TT

la ejecución del algoritmo de codificación LZW da como salida:

AA *[12][3]*BB *[12][3]*XX *[12][3]*TT *[12][3]*

otra vez, los * se utilizan como elementos separadores de las palabras código. A modo de ejemplo “AA *[12][3]*” nos indica que a partir del espacio en blanco retrocedemos 3 posiciones y ponemos 12 copias de los 3 siguientes caracteres, obteniendo como resultado:

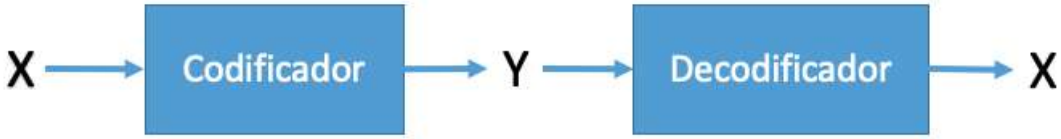



Figure 6.4: Esquema simple codificación y decodificación.

AA AA AA AA AA BB *[12][3]*XX *[12][3]*TT *[12][3]*

trivialmente podemos repetir la operación para BB, XX y TT, obteniendo la secuencia original. en este caso hemos codificado la secuencia original utilizando 44 bytes, por lo que los bps = 44 bytes * 8 bits / 69 samples = 5,10 bps!

Un aspecto que nos debe quedar muy claro es que estos codificadores **RLE y LZW no introducen ningún tipo de pérdida de información, recuperando los datos originales que entran en el codificador**. De aquí en adelante cuando hablemos de codificadores, haremos referencia a técnicas que permiten recuperar los datos originales. La figura 6.4 nos muestra un esquema simple del proceso de codificar una señal $X$, obtener unos datos codificados $Y$ y a continuación decodificar los datos $Y$ obteniendo los originales $X$.

### 6.5.1 Redundancia en imágenes digitales

Hasta este momento hemos visto como podemos explotar la redundancia aplicando una técnica de codificación aplicado a secuencia de caracteres. Pero como bien sabemos, las imágenes y vídeos digitales se estructuran en samples/píxeles (ver 6.2 y cada uno de estos samples son valores numéricos. Ahora, nos debemos preguntar si podemos aplicar antes de la codificación alguna técnica que nos permita reducir la información a codificar? Al tratarse de imágenes, la respuesta es Sí, y vamos a verlo mediante otro sencillo ejemplo.

La idea principal de la compresión de imágenes se basa en que los píxeles vecinos están altamente correlacionados, es decir, que son muy parecidos. Esta correlación se la conoce como correlación espacial. Supongamos que tenemos los siguientes samples de una imagen digital $X$:

12 17 14 19 21 26 23 29 41 38 31 44 46 57 53 50 60 58 55 54 52 51 56 60

en este caso las técnicas de RLW y LZW no serian efectivas ya que no existen secuencias repetidas, notase que:

- sólo 2 samples tienen el mismo valor,
- y la entropía de $H(X) = -(\frac{2}{24}log_2\frac{2}{24} + 22(\frac{1}{24}log_2\frac{1}{24})) = 4,51$bps

### 6.5.2 Predicción

La primera técnica para explotar la correlación espacial que vamos a ver va a ser la predicción. Veamos esta técnica mediante un sencillo, pero ilustrativo, ejemplo. Si calculamos la diferencia de los samples adyacentes, 17 - 12, 14 - 17, 19 - 14, 21 - 19, etc. Obtendremos la unos nuevos datos $X'$

12 5 -3 5 2 5 -3 6 12 -3 -7 13 2 11 -4 -3 10 -2 -3 1 -2 -1 5 4

Es importante destacar que en este caso lo que se hace es realizar una predicción $P$ y calcular un residuo $X'$. Siendo $X$ el vector de los datos originales e $X_i$ el valor de $X$ en la posición $i$ del vector, entonces la predicción $P_i = X_{i-1}$. En este caso, estamos "suponiendo" que el siguiente sample será muy parecido al actual de modo que el valor resultante sera próximo al cero y constante, obteniendo una entropía $H(X') < H(X)$. Después de aplicar la predicción podemos observar que:

- hay más de 4 samples con el mismo valor,
- y la entropía de $H(X') = -((\frac{5}{24}log_2\frac{5}{24}) + (\frac{4}{24}log_2\frac{4}{24}) +$

  $+ 3(\frac{2}{24}log_2\frac{2}{24}) + 9(\frac{1}{24}log_2\frac{1}{24})) = 3,51$bps

A este proceso se le conoce como decorrelación espacial. En la literatura existen distintas técnicas de decorrelación espacial las cuáles se pueden clasificar como técnicas **predictivas** o basadas en **transformada**. La que acabos de ver es una técnica basada en predicción, ya que lo que hacemos es predecir el sample siguiente mediante el anterior mediante:

$$X'_i = \begin{cases} X'_0 = X_0 & \text{si } i = 0, \\ X_{i-1} - X_i \text{ si } i > 0, \end{cases} \tag{6.9}$$

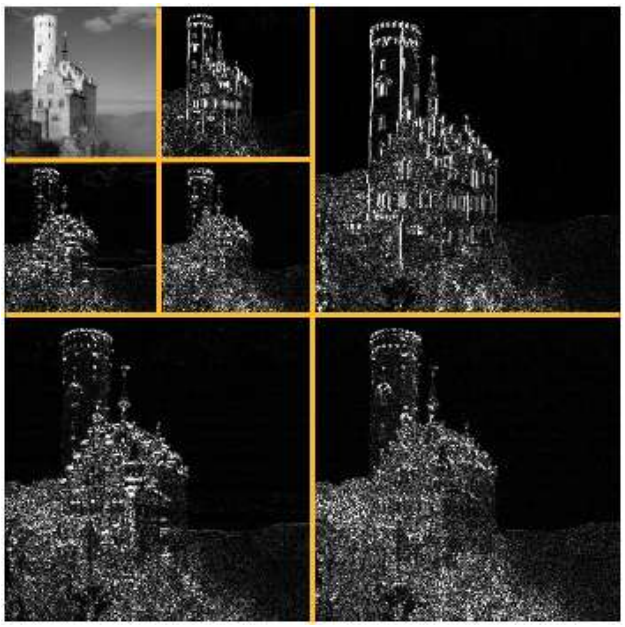


Figure 6.5: Ejemplo de transformada wavelet de 3 niveles de resolución

De modo que, este es un proceso fácilmente reversible utilizando la siguiente función:

$$X_i = \begin{cases} X_0 = X'_0 & \text{si } i = 0, \\ X'_i + X_{i-1} & \text{si } i > 0, \end{cases} \tag{6.10}$$

Para determinar si un predictor funciona de forma eficiente se puede utilizar la entropía de la señal de resultante de la predicción $X'$ o también se puede utilizar la Sum of Absolute Differences (SAD)

$$\text{SAD} = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} |X'_{i,j}|, \tag{6.11}$$

como menor sea el valor de SAD nos indicara que el predictor realiza una mejor predicción de la señal original $X$.

### 6.5.3 Transformada Wavelet

Por otro lado, y como hemos comentado tenemos los métodos basados en transformada, los cuales utilizan formulación matemática para transformar

los datos originales $X$ en $X'$ y a posteriori codificar $X'$. Las transformadas wavelet se han hecho un hueco muy importante en los sistemas de compresión, debido a su alta capacidad de decorrelación y la descomposición por niveles de resolución. El hecho de obtener distintos niveles de resolución permiten descomprimir estos niveles de forma independiente, aportando un gran versatilidad a la hora de transmitir y/o descomprimir de forma parcial una imagen de grandes dimensiones. La figura 6.5 muestra la aplicación de una descomposición de 3 niveles de una transformada wavelet. La transformada wavelet transforman la señal de manera que la divide en dos partes, una parte contiene los datos a una menor resolución, denotada como $L$. Mientras que la otra parte contiene los detalles, denotados como $H$, y son necesarios para recuperar los datos originales.

Una de las transformadas wavelet más simples es la transformada de Haar [18]. Los datos transformados a baja resolución $L$ se obtienen mediante la ecuación 6.12, y los detalles $H$ se obtienen a través de la ecuación 6.13.

$$L_{2n}^{j+1} = \frac{L_{2n}^{j} + L_{2n+1}^{j}}{2} \tag{6.12}$$

$$H_{2n}^{j+1} = \frac{L_{2n}^{j} - L_{2n+1}^{j}}{2} \tag{6.13}$$

La Figura 6.6 muestra un ejemplo de aplicación de la transformada Haar en una descomposición de 3 niveles para un vector de datos

$$X = [\,1, 2, 3, 4, 5, 6, 7, 8\,],$$

obtenemos la siguiente señal transformada

$$X' = [\,4.5, -2, -1, -1, -0.5, -0.5, -0.5, -0.5]$$

Vamos a ver un ejemplo pasos que se aplican en cada uno de los niveles:

- Inicialmente tenemos la señal original $X$, después de aplicar un primer nivel del filtro obtendremos $L1$ y $H1$. Nos encontramos en $j = 0$, para calcular los valores de $j = 1$ y $n = 0$, siendo $n$ la posición del vector $X$ que se va actualizando. Las operaciones para calcular $L_0^1$, $L_1^1$, $H_0^1$, $H_1^1$ son:

$$L_0^1 = \frac{1+2}{2} = 1.5 \;, L_1^1 = \frac{3+4}{2} = 3.5 \tag{6.14}$$

$$H_0^1 = \frac{1-2}{2} = -0.5 \;, H_1^1 = \frac{3-4}{2} = -0.5 \tag{6.15}$$

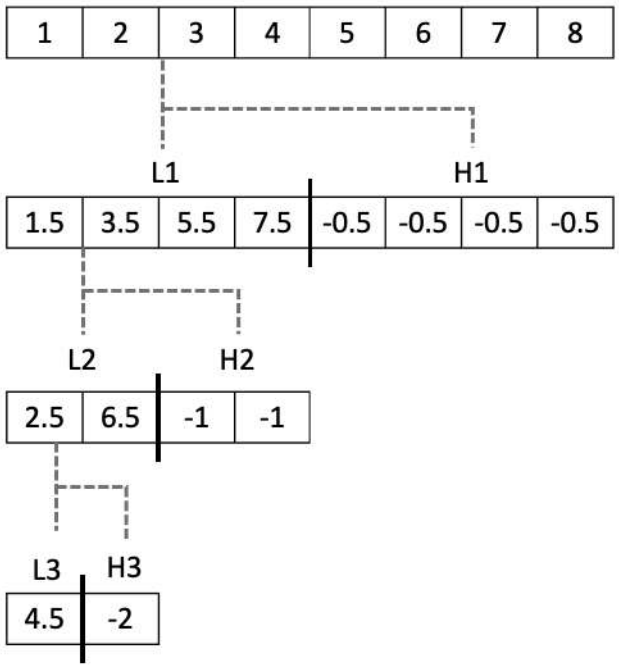



Figure 6.6: Ejemplo de descomposición de transformada Haar en 3 niveles de resolución

- Sobre la señal $L1$ podemos volver a aplicar un nivel de transformada, obteniendo el nivel de resolución 2 y sus respectivos detalles, denotados como $L2$ y $H2$.

Es importante mencionar que la señal que se debe almacenar al final es $L3, H3, H3, H1$ ya que a partir de $L3$ y $H3$ podremos recuperar $L2$, a partir de $L2$ y $H2$ podremos recuperar $L1$ y junto con $H1$ tendremos $X$. si deseamos recuperar $L2$ tendremos:

$$L_0^2 = \frac{4.5 + -2}{=} 2.5 \ , L_1^2 = \frac{4.5 - -2}{=} 6.5 \tag{6.16}$$

De modo que este es un proceso reversible, el cual nos permite ir recuperando los datos a distintos niveles de resolución. Ahora bien, en este ejemplo hemos aplicado la transformada Haar sobre un vector, pero las imágenes digitales son matrices (vectores de 2 dimensiones). En el caso de imágenes digitales, se aplica primero un nivel de transformada wavelet sobre el eje horizontal y a continuación sobre el eje vertical. Si se desean más niveles se vuelve a realizar la operación sobre la señal $L$. La figura 6.6 muestra un ejemplo de aplicación de la transformada Haar en una imagen, en la cual se

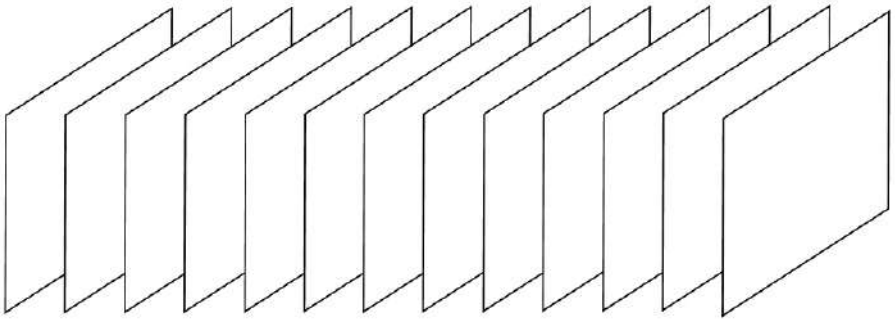


Figure 6.7: Ejemplo de frame rate con fps = 13.

ha realizado una descomposición de 3 niveles. En el mundo de la compresión se han introducido distintas transformadas wavelets, han que tener presente que algunas introducen pérdida y otras NO, permitiendo recuperar los datos originales.

Las dos técnicas vistas para explotar la redundancia espacial se aplican para la codificación de imágenes digitales. Cabe destacar que si deseamos comprimir un vídeo existe también mucha redundancia entre los frames consecutivos. Un frame, es una imagen digital en un vídeo, el número de imágenes que visualizamos por segundo viene determinado por el "frame rate". Lo más habitual es tener un frame rate por segundo fps = 24, de modo que cada segundo tenemos 24 imágenes digitales, esta densidad de frames por segundo hace que frames contiguos sean muy parecidos, a menos que haya movimientos bruscos (como en los deportes). La Figura 6.7 muestra un ejemplo esquemático de 13 frames capturados en un segundo, produciendo una secuencia de vídeo con un fps = 13, donde cada rectángulo representa un frame.

La reducción de la correlación entre los distintos frames las podemos hacer mediante 4 técnicas distintas:

### 6.5.4 Subsampling

Durante la adquisición de imágenes de un vídeo, dependiendo del movimiento efectuado por los objetos que aparecen en el vídeo, se puede dar el caso que la diferencia en frames consecutivos sea muy reducida. Subsampling se basa en considerar que si hay muy poco movimiento los frames consecutivos serán altamente parecidos, de modo que se pueden descartar algunos frames. En la Figura 6.8 se muestra un ejemplo de un vídeo con un subsampling de 2, es decir cada 2 frames uno es descartado. Esto tiene lugar durante

el proceso de compresión, evidentemente el descodificador deberá volver a generar los frames descartados –los azules– utilizando los frames adyacentes –los blancos–.

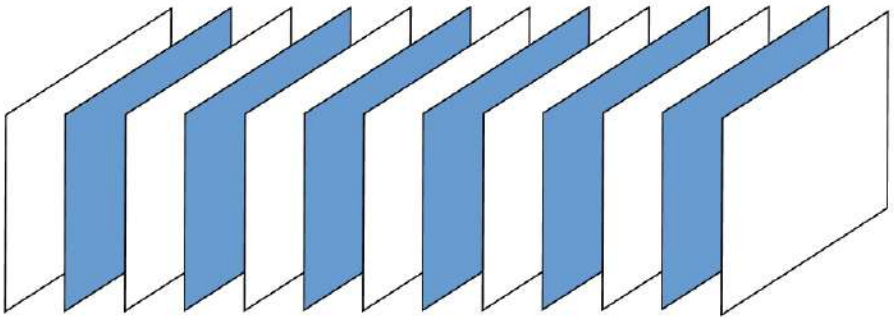


Figure 6.8: Ejemplo de subsampling

### 6.5.5 Diferencia de bloque

Mientras que la técnica del subsampling se basa en considerar que frames consecutivos son prácticamente iguales, la diferencia de bloque considera que frames consecutivos son mayoritariamente muy parecidos. Por eso motivo la diferencia de bloque codifica (opcionalmente) diferencias de una misma zona espacial. Primeramente se dividen los frames en bloques (B) –esta estructura puede ser rectangular–. Cada bloque B del frame $f_i$ se compara con el $f_{i+1}$. Si la diferencia, utilizando la métrica de SAD es pequeña, el bloque se codifica indicando sus coordenadas y la sus diferencias sample a sample con su bloque correspondiente, obteniendo $R_B$, caso contrario se codifica $f_{i+1}$. Matemáticamente se describe de la siguiente forma:

$$R_B = \begin{cases} f_{i+1} - f_i & \text{si } SAD < T = 0, \\ f_{i+1} & \text{si } SAD \geq T, \end{cases} \tag{6.17}$$

donde $T$ es un threshold determinado por el usuario especialista en compresión de datos. La figura 6.9 muestra dos frames con una subdivisión por bloques lógica, donde se resalta (en rojo) los bloques de una misma región espacial que deben analizarse para decidir si se codifican las diferencias o los samples de $f_{i+1}$

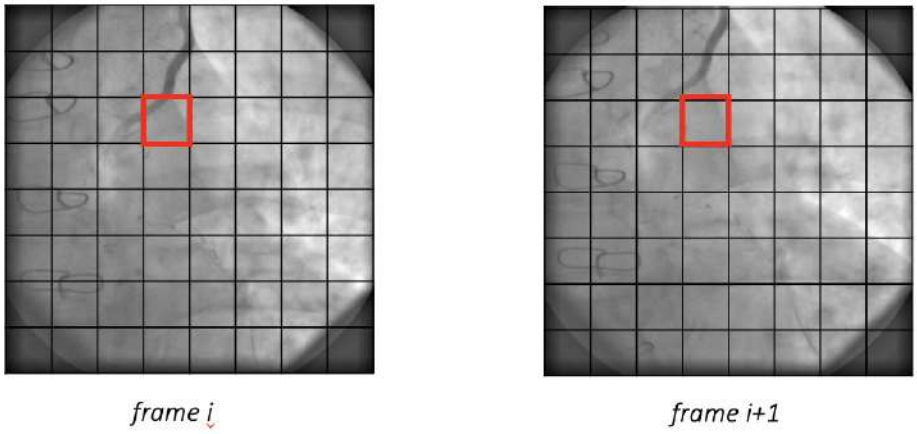



Figure 6.9: Ejemplo de diferencia de bloque

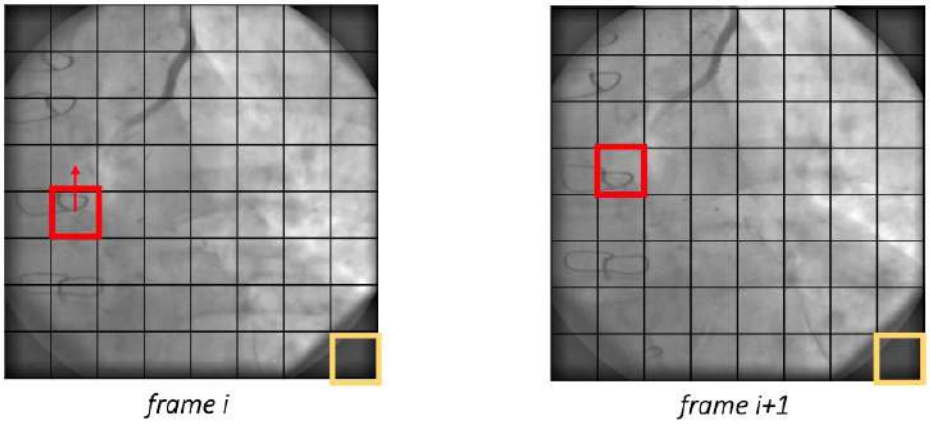



Figure 6.10: Ejemplo de compensación de movimiento

### 6.5.6 Compensación de movimiento

En una película, puede haber una translación de todo el frame en horizontal, vertical o diagonal. La compensación de movimiento pretende utilizar esta translación en todo el frame codificando las diferencias entre los bloques, siempre que estos sean parecidos. En el momento de codificar es necesario almacenar también el vector que nos indica el desplazamiento que ha hecho el bloque, de este modo podremos recuperar la información correctamente. En la compensación de movimiento, sólo almacenamos un vector por frame, considerando que todos los bloques se les aplica el mismo desplazamiento.

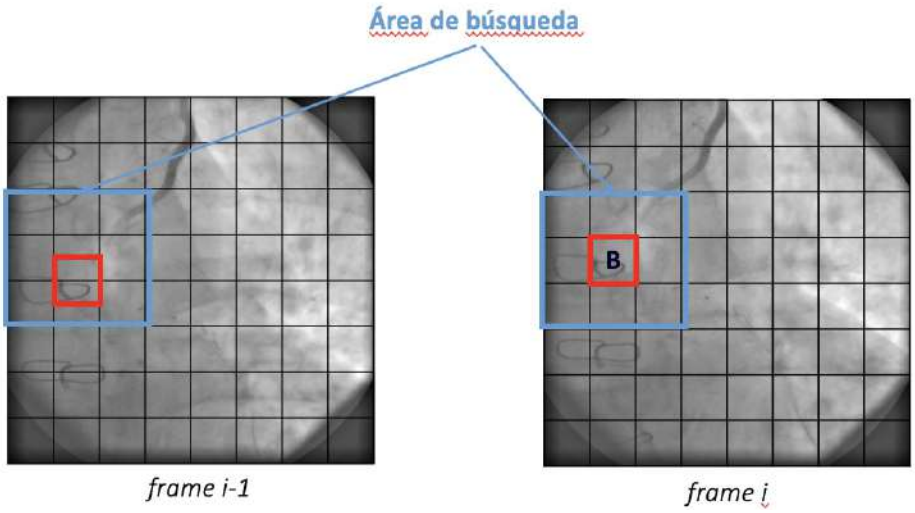



Figure 6.11: Ejemplo de búsqueda de bloque

### 6.5.7 Búsqueda de bloque

Esta técnica sume un mayor movimiento que el considerado por al diferencia de bloque. Supongamos una película, en esta se puede observar que la diferencia entre fotogramas consecutivos la diferencia es mínima, debido a que sólo se mueven algunos objetos de la escena, la cámara o ambos a la vez. El objetivo de esta técnica es buscar donde se han movido los objetos entre fotogramas consecutivos. Para ello, primeramente necesitamos realizar una subdivisión lógica de los frames en bloques $B_j$. Para cada bloque $B_j$ del frame $f_{i+1}$, debemos buscar el bloque más parecido a el minimizando la función

$$MIN(SAD(B_j^{f_{i+1}}, B_n^{f_i})\forall j, \ , \tag{6.18}$$

donde $B_n^{f_i}$ es el bloque $n$ del frame $f_i$ a codificar y $B_j^{f_{i-1}}$ son todos los posibles bloques a comparar del frame anterior. La figura 6.10 muestra un ejemplo gráfico de la técnica de compensación de movimiento, donde el bloque B resaltado en rojo del frame $f_i$ es el bloque a codificar. Es fácilmente observable que realizar dicha búsqueda puede llevar mucho tiempo, en función del tamaño del frame y los bloques. Considerando que entre frames consecutivos los objetos no se habrán movido grandes distancias, esto permite reducir la era de búsqueda y así reducir el tiempo de compresión. Para ello se define el "Área de busqueda" como una región adyacente al bloque a codificar donde realizaremos la búsqueda. En la figura 6.11 se muestran los dos frames el

bloque a codificar y la área de búsqueda.

# Bibliography


[1] Matthias Jarke, Maurizio Lenzerini, Yannis Vassiliou, and Panos Vassiliadis. *Fundamentals of Data Warehouses*. Springer-Verlag, Berlin, Heidelberg, 2nd edition, 2001.

[2] Jonathan I Maletic and Andrian Marcus. Data cleansing: Beyond integrity analysis. In *Iq*, pages 200–209. Citeseer, 2000.

[3] Aaron L Statham, Dario Strbenac, Marcel W Coolen, Clare Stirzaker, Susan J Clark, and Mark D Robinson. Repitools: an r package for the analysis of enrichment-based epigenomic data. *Bioinformatics*, 26(13):1662–1663, 2010.

[4] AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of data integration*. Elsevier, 2012.

[5] Merinda McLure, Allison V Level, Catherine L Cranston, Beth Oehlerts, and Mike Culbertson. Data curation: a study of researcher practices and needs. *portal: Libraries and the Academy*, 14(2):139–164, 2014.

[6] `https://www.privacy-regulation.eu/en/recital-26-GDPR.htm`. Accessed: 2021-10-18.

[7] `https://www.privacy-regulation.eu/en/article-4-definitions-GDPR.htm`. Accessed: 2021-10-18.

[8] `https://www.privacy-regulation.eu/en/article-5-principles-relating-to-processing-of-personal-data-GDPR.htm`. Accessed: 2021-10-18.

[9] `https://www.aepd.es/sites/default/files/2019-09/nota-tecnica-kanonimidad.pdf`. Accessed: 2021-10-18.

[10] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[11] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24, 2006.

[12] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and -diversity.

[13] `https://en.wikipedia.org/wiki/Pigeonhole_principle`. Accessed: 2021-09-01.

[14] `https://es.wikipedia.org/wiki/Byte`. Accessed: 2021-09-01.

[15] `https://en.wikipedia.org/wiki/ASCII`. Accessed: 2021-09-01.

[16] `https://en.wikipedia.org/wiki/Entropy_(information_theory)`. Accessed: 2021-09-01.

[17] `https://es.wikipedia.org/wiki/LZW`. Accessed: 2021-09-01.

[18] `https://en.wikipedia.org/wiki/Haar_wavelet`. Accessed: 2021-09-01.

# Chapter 7

# Compresión de Datos II

*Joan Bartrina*

## 7.1 Métricas de distorsión

La calidad de imagen puede degradarse debido a distorsiones introducidas durante el proceso de compresión. La comunidad científica ha definido distintas medidas analíticas con el fin de medir la calidad las imágenes.

En nuestro escenario disponemos de los datos originales, por ejemplo la imagen sin distorsión, hecho útil ya que permite utilizarla como referencia para medir la calidad de la imagen comprimida y posteriormente descomprimida. De modo que podremos evaluar la calidad de las imágenes comprimidas, una versión sin comprimir de la imagen proporciona una referencia útil. En estos casos, puede utilizar métricas de calidad de referencia completa para comparar directamente la imagen de destino y la imagen de referencia.

Los algoritmos de referencia completa comparan la imagen de entrada con una imagen de referencia sin distorsión. Las medidas de distorsión más comunes son Mean Square Error (MSE), el Peak Signal Noise Ratio (PSNR) y el Peak Absolute Error (PAE). A continuación vamos a ver como se calculan cada una de estas dos medidas de distorsión y calcularemos un ejemplo.

### 7.1.1 Mean Squared Error

En procesamiento de imágenes, el Mean Square Error (MSE) es una métrica que mide la fidelidad entre los datos originales $X$ y los comprimidos y posteriormente descomprimidos, es decir, los datos recuperados $X'$. El MSE mide el promedio de los errores al cuadrado, es decir, calcula la diferencia entre el valor original $X$ y el recuperado $X'$, y esa diferencia es elevada al cuadrado.

Supongamos que $X = \{X_{i,j} | i = 1, 2, ..., M \wedge j = 1, 2, ..., N\}$ y $X' = \{X'_{i,j} | i = 1, 2, ..., M \wedge j = 1, 2, ..., N\}$ son dos señales discretas finitas, en nuestros casos imágenes visuales, donde M es el número de muestras de la señal (píxeles, cuando las señales son imágenes), mientras que $X_{i,j}$ y $X'_{i,j}$ son los valores de $X$ y $X'$, respectivamente. El MSE entre dos señales se calcula mediante la siguiente expresión

$$\text{MSE}(X, X') = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (X_{i,j} - X'_{i,j})^2. \tag{7.1}$$

De esta expresión se puede observar fácilmente que si las dos señales $X$ y $X'$ (imágenes) son iguales el valor de MSE es 0. Por el otro lado, como mayor sea al valor del MSE más distorsión entre $X$ y $X'$ habrá.

### 7.1.2 Peak Signal Noise Ratio

El Peak Signal Noise Ratio (PSNR) es una métrica derivada del MSE. En este caso relaciona la máxima energía posible de la señal y el ruido –el MSE– que afecta a su representación. Debido a que muchas señales tienen un gran rango dinámico, el PSNR se expresa generalmente en escala logarítmica. Como unidades se utiliza el decibelio (dB). El uso más habitual del PSNR es como medida cuantitativa de la calidad de la reconstrucción de imágenes comprimidas. Para su calculo, es necesario calcular previamente el MSE entre dos señales $X$ y $X'$. El PSNR se define como:

$$\text{PSNR}(X, X') = 10 Log_{10} \frac{L^2}{\text{MSE}}, \tag{7.2}$$

donde $L$ es el rango dinámico. Por ejemplo, para imágenes donde cada muestra se presenta con 8 bits por sample, $L = 2^8 - 1 = 255$. Para imágenes de 12 bits por sample, aunque se almacenen en 2 bytes (16 bits), su rango dinámico es $L = 2^{12} - 1 = 4095$.

En este caso, y a diferencia del MSE, si el PSNR es $\infty$ indica que las señales $X$ y $X'$ son iguales, mientras que un valor más próximo al 0 indica que las imágenes son más distintas.

### 7.1.3 Peak Absolute Error

La última métrica interesante en los procesos de compresión es el Peak Absolute Error (PAE) que no indica cual es el error máximo en valor absoluto que

| | | | | |
|---|---|---|---|---|
| 23 | 25 | 1790 | 123 | 125 |
| 230 | 125 | 789 | 124 | 156 |
| 231 | 235 | 1432 | 173 | 185 |
| 233 | 225 | 1290 | 134 | 124 |
| 233 | 215 | 1490 | 236 | 122 |

Tabla 7.1: Señal $X$

| | | | | |
|---|---|---|---|---|
| 22 | 24 | 1789 | 122 | 124 |
| 228 | 124 | 788 | 122 | 154 |
| 230 | 234 | 1430 | 172 | 184 |
| 232 | 224 | 1288 | 132 | 122 |
| 232 | 214 | 1488 | 234 | 120 |

Tabla 7.2: Señal $X'$

se esta cometiendo entre la señal original $X$ y la recuperada $X'$ y se expresa de la siguiente forma

$$\text{PAE}(X, X') = \text{MAX}(|X - X'|) \tag{7.3}$$

### 7.1.4 Ejemplos

En esta subsección vamos a calcular las métricas descritas anteriormente, el MSE, PSNR y PAE entre dos señales $X$ y $X'$. La tablas 7.1 y 7.2 contienen los datos utilizados como señales $X$ y $X'$, respectivamente, y que serán utilizadas para el calculo de las métricas descritas anteriormente en modo de ejemplo.

1. **Calculo MSE**:

   $\text{MSE}(X, X') = \frac{1}{5*5}((23-22)^2 + (25-24)^2 + (1790-1789)^2 + (123-122)^2 + (125-124)^2 ... (236-234)^2 + (122-120)^2) = \frac{55}{25} = 2,2.$

2. **Calculo PSNR**:

   Para el calculo del PSNR debemos primero averiguar el rango dinámico

de los datos. Para ello buscamos el valor máximo en $X$, MAXIMO($X$) = 1790. Calculamos los bits necesarios para representar el 1790 en binario mediante $\lceil Log_2(1790) \rceil = 11$ bits.

$\text{PSNR}(X, X') = 10 Log_{10} \frac{2^{11}-1}{55} = 10 Log_{10} \frac{2047}{55} = 48,81\text{dB}$.

3. **Calculo PAE**:

   $\text{PAE}(X, X') = \text{MAX}(|23-22|, |25-24|, |1790-1789|, |123-122|, |125-124|, ..., |236-234|, |122-120|) = 2$, indicando que el error máximo que se esta cometiendo entre los datos originales y los recuperados es de 2 unidades.

## 7.2 Compresión lossless y lossy: el pipeline

Como hemos visto anteriormente los métodos de compresión se pueden clasificar en dos categorías principales, métodos de compresión sin pérdida (**lossless**) y con pérdida (**lossy**). Además cada uno de estos compresores pueden utilizar técnicas de transformada o de predicción para explotar la redundancia de la imagen y así reducir su entropía.

Las **técnicas lossy** son aquellas que no recuperan los datos originales, es decir, introducen algún tipo de pérdida durante el proceso de compresión que no se puede recuperar. El hecho de introducir esta pérdida nos permite obtener unos factores de compresión mucho más elevados, eso sí, penalizando en como se recuperan los datos, tal y como hemos visto en la figura 6.3. La pérdida durante el proceso de compresión se introduce o bien utilizando utilizando técnicas de **cuantización** o bien utilizando técnicas de **rate control**.

### 7.2.1 Cuantización

Al introducir pérdida las técnicas de compresión se aprovechan de que el ojo humano es bastante bueno percibiendo las pequeñas diferencias en el brillo sobre un área relativamente extensa, pero no es tan bueno distinguiendo la misma variación de intensidad entre samples cercanos. Este echo nos permite poder eliminar estas pequeñas diferencias entre valores. Para ello utilizaremos un cuantizador.

$$\hat{X} = sign(X) \left\lfloor \frac{|X|}{\Delta} \right\rfloor, \tag{7.4}$$

donde $X$ son los valores originales, $\lfloor\ \rfloor$ permite obtener sólo la parte entera (descartando los decimales), $|X|$ devuelve el valor absoluto de $X$, $sign(X)$

nos indica el signo del valor $X$, y $\Delta$ nos indica el paso de cuantización. Si $\Delta = 1$, entonces $X = \hat{X}$, de modo que no se está introduciendo pérdida durante el proceso. Mientras que, lado como mayor sea el valor de $\Delta$ los valores de $\hat{X}$ serán más cercanos a cero y más homogéneos. Al ser más homogéneos la entropía es menor y en consecuencia los bits necesarios son menores.
Vamos a ver un breve ejemplo utilizando datos previamente utilizados. Consideramos de nuevo los datos originales $X$

12 17 14 19 21 26 23 29 41 38 31 44 46 57 53 50 60 58 55 54 52 51 56 60

con una entropía de $H(X) = 4,51$. Si cuantizamos $X'$ utilizando 7.5 con $\Delta = 5$ obtendremos $\hat{X}'$

2 3 2 3 4 5 4 5 8 7 6 8 9 11 10 10 12 11 11 10 10 10 11 12

donde la de entropía $H(\hat{X}') = -(4 \cdot \frac{1}{24} log_2 \frac{1}{24} + 4 \cdot \frac{2}{24} log_2 \frac{2}{24} + \frac{3}{24} log_2 \frac{3}{24} +$

$+ \frac{4}{24} log_2 \frac{4}{24} + \frac{5}{24} log_2 \frac{5}{24}) = 3,22$bps.
Para recuperar los datos debemos aplicar la función inversa del proceso de cuantización, que en este caso se define como:

$$X'' = \Delta \cdot \hat{X}', \tag{7.5}$$

donde obtendremos $X''$ como

10 15 10 15 20 25 20 25 80 35 30 40 45 55 50 50 60 55 55 50 50 50 55 60

si calculamos donde el PAE$(X, X') = 4$. En este punto es importante comentar que siempre se cumpliará que

$$PAE(X, X') \leq \Delta - 1, \tag{7.6}$$

por lo que si $\Delta = 1$ el PAE$(X, X') = 0$ indicando que los datos originales y recuperados, $X$ y $X'$ son iguales. Si $\Delta > 1$ los datos originales y recuperados serán distintos, obteniendo siempre un PAE$(X, X') = \Delta - 1$.

El uso del cuantizador, seguido de un codificador nos permite controlar el error máximo cometido en $X$ y $X'$, pero no podemos controlar el tamaño del archivo final.

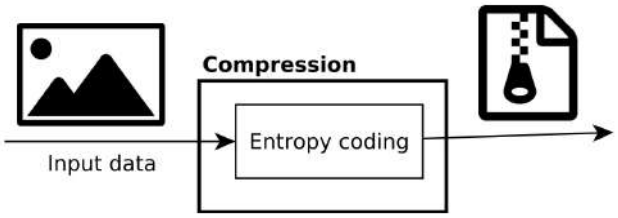



Figure 7.1: Pipeline simple con Entropy encoder

### 7.2.2 Rate control

Las técnicas de rate control, nos permiten tener un control sobre el tamaño del archivo final, de modo que podemos ajustar el tamaño del archivo al ancho de banda del canal por donde se va a transmitir el archivo. Ahora bien, estas técnicas no permiten controlar el error. De forma general las técnicas de rate control, seleccionan cuales son los conjuntos de datos codificados que permiten recuperar las imágenes a una mejor calidad con un determinado rate. El rate, se le conoce como el valor de bps que permite definir el tamaño del archivo comprimido.

El pipeline de un sistema de compresión se le conoce como el conjunto de técnicas que se aplican de forma secuencial que permite comprimir las imágenes. La combinación de estas técnicas permite definir distintos pipelines de compresión con determinadas características. A continuación vamos distintos pipelines en función de las técnicas incluidas en ellos.

#### Pipeline basado en entropy encoder

Permite reducir los datos representando exactamente la misma información. En la literatura podemos encontrar distintos codificadores por entropía como LZ77, LZ78, MQ, entre otros. La figura 7.1 muestra un un método de pipeline simple formado únicamente por un Entropy encoder, por lo que si se aplica de forma inversa permite recuperar los datos originales sin pérdida de información.

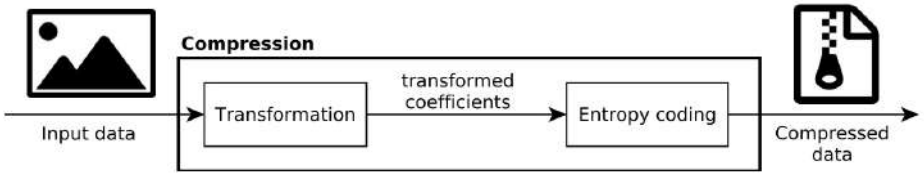



Figure 7.2: Pipeline formado por una transformada y un Entropy encoder

### Pipeline basado en transformada

Como vimos en la unidad anterior las transformadas permiten reducir la entropía de la imagen a codificar. En nuestro caso vimos la transformada wavelet HAAR. Las transformadas wavelet se aplican sobre la imagen, y a continuación los datos transformados suelen ser codificados con mediante un entropy encoder. La figura 7.2 muestra un esquema con estas características. Para recuperar la imagen original debemos deshacer el proceso de forma inversa, de modo que primero utilizaremos un entropy decoder y a continuación aplicaremos la transformación inversa para se recuperar los datos originales.

### Pipeline: Predicción

En un pipeline basado en la predicción obtenemos unos valores residuales $X'$, como vimos en la unidad 6, estos valores a continuación se deben codificar con un codificador por entropía. La figura 7.3 muestra un pipeline basado en la predicción, es importante destactar que en el proceso de compresión el predictor obtiene los residuales $X'$ utilizando el operador resta, mientras que en el descodificador para recuperar la señal original $X$ se utiliza el operador suma.

### Pipeline basado en cuantización y entropy encoder

El pipeline de la figura 7.4 esta formado por un cuantizador y un codificador por entropía. En este caso los datos originales $X$ son cuantizados obteniendo $X'$, posteriormente los datos $X'$ son enviados al codificador por entropía. Esto implica que los datos recuperados al aplicar el descodificador por entropía y el descuantizador no los vamos a recuperar los datos originales, a menos que el paso de cuantización sea de $\Delta = 1$ (fijaos en la fórmula 7.5). De modo que el simple hecho de introducir un cuantizador no implica que no se pueda recuperar la original.

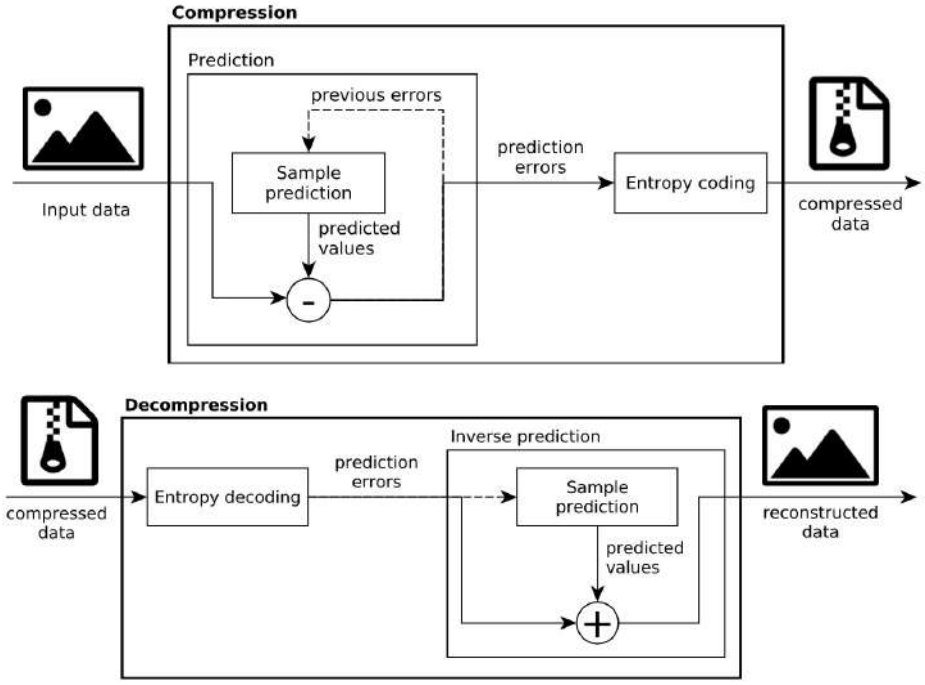



Figure 7.3: Pipeline basado en predicción y entropy encoder

### Pipeline basado en transformación, cuantización y entropy encoder

La figura 7.5 muestra un esquema de compresión basado en transformada wavelet seguido de un cuantizador y un entropy encoder define un esquema de sistema de compresión muy utilizado en la compresión de datos. Este pipeline es utilizado como base en distintos métodos de compresión como JPEG y JPEG2000.

### Pipeline basado predicción, cuantización y entropy encoder

La figura 7.6 muestra el esquema del compresor de un pipeline compuesto por un predictor, un cuantizador y un codificador por entropía. De esta figura, cabe destacar la flecha que va de la caja del cuantizador a la predicción, la cual es fundamental para recuperar la señal de forma adecuada. Notase, que el predictor utiliza unos datos de entrada para estimar una predicción, esta función de predicción debe ser igual en el compresor que en el descompresor. Ahora bien, si se aplica una cuantización con $\Delta > 1$, entonces los datos recuperados $X''$ serán distintos a los originales $X$, de modo que el predictor del compresor utilizaría $X$, mientras que el descompresor utilizaría $X''$. Debido

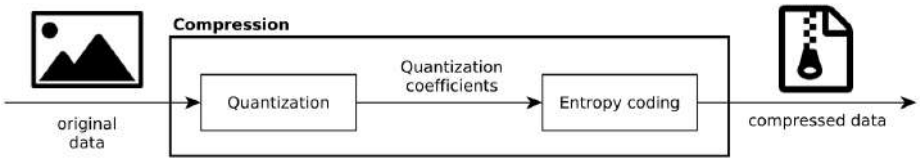



Figure 7.4: Pipeline basado en cuantizador y entropy encoder

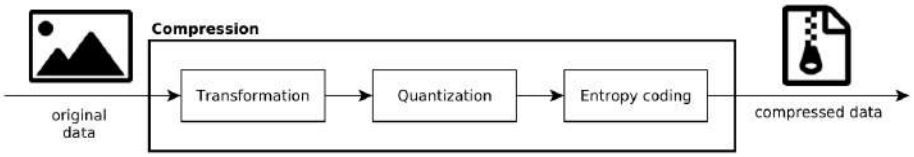



Figure 7.5: Pipeline formado por transformada, cuantizador y entropy encoder

a que la función de predicción es la misma, si los datos son distintos $X$ y $X''$ el resultado de la predicción también lo será, resultando en un reconstrucción totalmente errónea. Para solucionar este problema, en el proceso de **compresión** se realiza la predicción utilizando los datos cuantizados y descuantizados (simulando los datos que dispondrá el predictor del compresor), de este modo ambos utilizaran $X''$ para la predicción, lo que permitirá recuperar los datos correctamente.

Los pipelines basado en predicción son muy frecuentes en sistemas de compresión como JPEG-LS, MPEG4/H.264 y HEVC/H.265. Para MPEG4/H.264 y HEVC/H.265 además se incluyen técnicas de subsampling, diferencia de bloque, compensación de movimiento y búsqueda de bloque (vistas en la unidad anterior). Por este motivo JPEG-LS es mucho más rápida que MPEG4/H.264 o HEVC/H.265 durante el proceso de compresión, para la descompresión los tres métodos son muy rápidos, caso contrario no podríamos reproducir el vídeo en tiempo real para MPEG4/H.264 y HEVC/H.265.

## 7.3 Sistemas de compresión DICOM

En esta sección vamos a ver los sistemas de compresión incluidos en DICOM. No los vamos a ver des del punto de vista técnico, es decir, que técnicas se utilizan para comprir los datos, sino cuales son sus características para determinar cuales debemos utilizar en función de los datos a comprimir y su

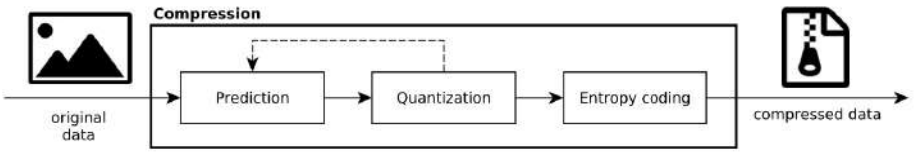
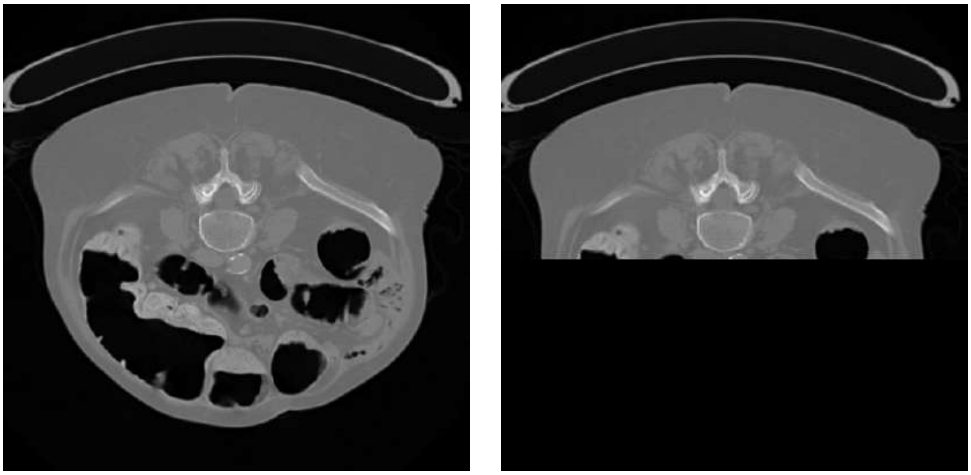



Figure 7.6: Pipeline formado por predicitor, cuantizador y entropy encoder



Imágen original Imágen parcialmente recuperada

Figure 7.7: Ejemplo de recuperación parcial de con un sistema de progresión baseline

aplicación.

### 7.3.1 Run Length Encoding

El método de Run Length Encoding (RLE) es un método de codificación sin pérdida descrito en la sección 6.5. Su aplicación es la de comprimir datos sin pérdida de información. En el caso de recuperación parcial del archivo comprimido los datos que se puedan recuperar van a ser sin pérdida de información. Los datos que no se puedan recuperar los vamos a descodificar a un valor fijo, esto dará como resultado un imágen rdonde se visualizan correctamente las $N$ primeras filas y el resto de filas no se van a recuperar. A este efecto se le denomina recuperación *Baseline*. La Figura 7.7 muestra un ejemplo de progresión baseline.

### 7.3.2 JPEG

Definido por el Joint Photografic Experts Group, el cual le dio nombre al sistema de compresión JPEG. JPEG es sistema de compresión con pérdida. Su uso se encuentra muy extendido en los sistemas fotográficos de uso personal. Para uso médico se tiene que tener en consideración la tasa de compresión. Recordad que altas tasas de compresión no aportaran un impacto sobre la cualidad de la imagen recuperada, de modo que si se realiza un diagnóstico médico este puede verse alterado. Nos permite obtener tasas de compresión de 50 a 1 (la imágen comprimida ocupa 50 veces menos que la original), sin tener mucha pérdida de cualidad visual.

### 7.3.3 JPEG-LS

También definido por el Joint Photografic Experts Group, este nos permite recuperar los datos sin pérdida de información. Su uso ésde gran interés si debemos utilizar las imágenes para un diagnostico médico preciso o utilizando algoritmos de computed aided diagnosis.

### 7.3.4 JPEG2000

Un de los recientes estándares presentado por Joint Photografic Experts Group. Permite la compresión con y sin pérdida. Su uso se encuentra extendido a nivel profesional, ya que tiene ciertas características que lo hacen muy interesante para realizar compresiones interactivas. Esto quiere decir que podemos decidir que cantidad de información queremos transmitir/descodificar, a mayor información transmitida/descodificada mayor sera la calidad de la imagen recuperada.

### 7.3.5 MPEG2, MPEG4/H.264 y HEVC/H.265

Estos tres sistemas de codificación se usa para vídeo. El más común actualmente es el H.264 y H.265. Ambos permiten la codificación sin pérdida de vídeos que capturados con una profundidad de hasta 12 bits por sample. El problema de estos sistemas es que el tiempo de compresión es muy elevado, pero el de descodificación es rápido.

# Bibliography


[1] Matthias Jarke, Maurizio Lenzerini, Yannis Vassiliou, and Panos Vassiliadis. *Fundamentals of Data Warehouses*. Springer-Verlag, Berlin, Heidelberg, 2nd edition, 2001.

[2] Jonathan I Maletic and Andrian Marcus. Data cleansing: Beyond integrity analysis. In *Iq*, pages 200–209. Citeseer, 2000.

[3] Aaron L Statham, Dario Strbenac, Marcel W Coolen, Clare Stirzaker, Susan J Clark, and Mark D Robinson. Repitools: an r package for the analysis of enrichment-based epigenomic data. *Bioinformatics*, 26(13):1662–1663, 2010.

[4] AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of data integration*. Elsevier, 2012.

[5] Merinda McLure, Allison V Level, Catherine L Cranston, Beth Oehlerts, and Mike Culbertson. Data curation: a study of researcher practices and needs. *portal: Libraries and the Academy*, 14(2):139–164, 2014.

[6] `https://www.privacy-regulation.eu/en/recital-26-GDPR.htm`. Accessed: 2021-10-18.

[7] `https://www.privacy-regulation.eu/en/article-4-definitions-GDPR.htm`. Accessed: 2021-10-18.

[8] `https://www.privacy-regulation.eu/en/article-5-principles-relating-to-processing-of-personal-data-GDPR.htm`. Accessed: 2021-10-18.

[9] `https://www.aepd.es/sites/default/files/2019-09/nota-tecnica-kanonimidad.pdf`. Accessed: 2021-10-18.

[10] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[11] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24, 2006.

[12] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and -diversity.

[13] `https://en.wikipedia.org/wiki/Pigeonhole_principle`. Accessed: 2021-09-01.

[14] `https://es.wikipedia.org/wiki/Byte`. Accessed: 2021-09-01.

[15] `https://en.wikipedia.org/wiki/ASCII`. Accessed: 2021-09-01.

[16] `https://en.wikipedia.org/wiki/Entropy_(information_theory)`. Accessed: 2021-09-01.

[17] `https://es.wikipedia.org/wiki/LZW`. Accessed: 2021-09-01.

[18] `https://en.wikipedia.org/wiki/Haar_wavelet`. Accessed: 2021-09-01.

# Chapter 8

# Seguridad de los datos. Conceptos básicos de seguridad y cifrado

*Ramon Martí*

## 8.1 Introducción

En el entorno de los Datos de Salud, y en Internet en general, la seguridad es una característica necesaria, importante pero no fácil de conseguir. Es necesaria porque se manejan datos personales; es importante porque la información tiene mucho valor; pero no es fácil de conseguir porque hay que entender cuándo y cómo los usuarios, los ordenadores, los servicios y las redes deben confiar los unos en los otros. Todo ello, en un entorno muy heterogéneo.

En este escenario, habrá distintos elementos a proteger: hardware, software o datos. Por lo que hace referencia a esta unidad y a la siguiente nos centraremos principalmente en la seguridad de los datos, que se puede tratar en un sentido muy amplio y que puede incluir (se verá con más detalle más adelante), entre otros:

- La protección contra la lectura del contenido de los datos,
- la integridad de los datos frente a las modificaciones,
- la protección contra la denegación de servicios,
- o el acceso a la información y los servicios solo para usuarios autorizados.

Los tres primeros puntos se verán en esta unidad, mientras que este último, junto con la seguridad en Big Data, se verá en la siguiente.

Es necesario destacar que la seguridad debe estar en todos los niveles. Un único punto débil puede comprometer la seguridad de todo el sistema. En parte por este motivo, hay que tener claro que, a pesar de todos los mecanismos que se verán, la seguridad absoluta no es posible, y que generalmente solo se conseguirá que los ataques sean más difíciles de realizar y que tengan una probabilidad de éxito más baja.

En la actualidad, la información se almacena de forma digital en ficheros (o dentro de bases de datos, que también son ficheros), los cuales se deben almacenar de forma segura y restringir el acceso solo al personal autorizado, pero también muy a menudo se deben intercambiar, y por tanto transmitir, entre dispositivos (dispositivos captadores, ordenadores, dispositivos móviles, etc.). Es por eso que en esta unidad y la siguiente nos centraremos en la seguridad en el almacenamiento, acceso y comunicación de los datos, que ya se verá comparten características comunes.

De todas formas, antes de continuar, veremos qué se entiende por fichero.

**¿Qué es un fichero?**

Se entiende por "fichero" o "archivo", todo conjunto de datos, cualquiera que fuese su forma o modalidad de su creación, almacenamiento, organización y acceso. Los tipos de fichero dependerán del tipo de usuario o empresa (clientes, personal, candidatos, proveedores, etc.). En el caso del entorno sanitario, principalmente nos centraremos en la información de los pacientes (que será de carácter personal), y en este módulo nos ocuparemos de los ficheros informáticos, donde la información está almacenada digitalmente.

En este capítulo nos centraremos a describir los aspectos generales sobre la Seguridad de los datos, que también aplican a Big Data. Aunque se incluyen algunos apartados con cierto contenido "técnico", el objetivo es solo dar una idea general del tema, sin entrar en mucho detalle. Con tal fin, empezaremos hablando brevemente del Reglamento General de Protección de Datos (*General Data Protection Regulation*, GDPR), continuaremos con la definición de las principales amenazas de seguridad sobre los datos, la base de los principales mecanismos de seguridad para su protección, finalizando con los mecanismos de gestión de claves.

## 8.2 Reglamento General de Protección de Datos (*General Data Protection Regulation*, GDPR)

Tal y como ya se ha comentado en la Unidad 4, en el entorno de la Salud, es habitual tratar con información personal, con la cual, al ser confidencial, es imprescindible seguir las normativas relativas a la protección de datos.

El Reglamento General de Protección de Datos, RGPD (*General Data Protection Regulation*, GDPR) [5] es el reglamento relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos. Es una normativa a nivel europeo, por lo que cualquier empresa europea o aquellas empresas que tengan negocios en la Unión Europea que manejen información personal de cualquier tipo deberán acogerse a la misma.

### 8.2.1 Niveles de seguridad

El RGPD define tres niveles de seguridad (básico, medio y alto), y especifica qué tipo de tratamiento se tiene que realizar en cada caso. Para los documentos con información personal, como es el caso de los datos de salud en el entorno sanitario, se debe aplicar el nivel de seguridad alto. En concreto, para este nivel alto de seguridad se debe, entre otros:

- A nivel de empresa, poseer un Documento de Seguridad donde se detalle todo el procedimiento realizado para proteger los ficheros.
- A nivel de todo el personal, conocer sus funciones y obligaciones sobre la gestión de incidencias y protección de datos.
- Llevar un registro minucioso sobre las incidencias.
- Guardar en un formato ininteligible cualquier contraseña.
- Realizar copias de seguridad al menos una vez a la semana de toda la información para poder restaurarla en caso de pérdida o destrucción.
- Realizar una auditoría, ya sea interna o externa, por lo menos cada dos años.
- Realizar el cifrado de datos.
- Conservar varias copias de seguridad en diferentes equipos informáticos.
- Controlar y almacenar todos los accesos de los usuarios y conservarlos durante al menos dos años.

- Almacenar toda la información no automatizada bajo clave, pudiendo realizar solamente copias de esa información el personal autorizado.

### 8.2.2 Vulneraciones más destacadas

Centrándonos en el campo de la salud, y para no cometer los mismos errores, es interesante mencionar los casos más destacados de vulneración de los deberes de seguridad y secreto por parte de centros sanitarios a partir del Informe de cumplimiento de la LOPD en Hospitales [6]:

- Difusión de datos de pacientes a través de redes de intercambio de archivos P2P
- Datos de salud abandonados en contenedores de la vía pública
- Almacenamiento de documentación clínica en áreas no restringidas al público y en dependencias al alcance de cualquiera
- Pérdida de historiales clínicos al proceder a la automatización de las historias y no adoptar medidas de seguridad
- Utilización de los datos sanitarios para finalidades no autorizadas y comunicación indebida a terceros.

### 8.2.3 Recomendaciones

En el mismo documento, se numeran las principales recomendaciones de seguridad a tener en cuenta, entre las que destacan las siguientes:

- Custodiar la documentación clínica de pacientes cuando esta no se encuentre archivada impidiendo que pueda ser accedida por terceros.
- Adoptar medidas para evitar la pérdida o sustracción de la documentación durante su transporte.
- Almacenar los archivos físicos de historias clínicas en áreas con acceso protegido mediante clave o dispositivo equivalente y en archivadores que dispongan de mecanismos que obstaculicen su apertura.
- Registrar todos los accesos realizados a los historiales clínicos.
- Aplicar procedimientos de disociación de los datos de carácter personal en los tratamientos de datos que hayan sido externalizados.

- Etc.

Tal como ya hemos dicho en la introducción, por lo que hace referencia a la aplicación de seguridad a nivel general, esta unidad se centrará en el Cifrado de datos, mientras que en la siguiente unidad se verán los mecanismos de control de accesos.

## 8.3 Escenario y Personajes: Alice, Bob y Trudy

Una vez dada una pequeña pincelada sobre el RGPD, empezamos ya con los apartados de seguridad informática. En este entorno, en general el escenario en que nos encontramos es que existe un usuario A que desea almacenar o intercambiar información para que acceda a ella de forma segura un usuario B, ante el peligro que un intruso T (o usuario malicioso M) pueda interceptarla. Para facilitar la lectura y comprensión de la documentación, normalmente se utilizan, y utilizaremos, los nombres Alice, Bob y Trudy (o Mallory) (Figura 8.1) para referirnos estos tres personajes, teniendo en cuenta que en la literatura se pueden encontrar además otros personajes con otros nombres y roles complementarios o más específicos [1].

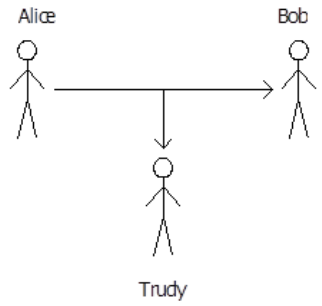



Figure 8.1: Personajes en el escenario de la seguridad

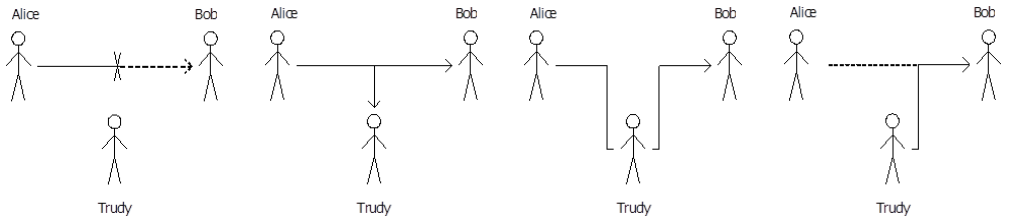
Cabe tener en cuenta que tanto Alice, Bob como Trudy pueden ser personas reales o simplemente dispositivos (ordenadores, *routers*, etc.) que se quieren intercambiar información (o actuar de forma maliciosa como en el caso de Trudy).

## 8.4 Amenazas (*Threats*)

Empezaremos el apartado más técnico de esta unidad listando y describiendo brevemente las distintas amenazas de seguridad que pueden presentarse en

los sistemas de computadores y en las comunicaciones (Figura 8.2), y las preguntas que nos darán lugar a la detección del problema de seguridad:

- **Interrupción**: Trudy interrumpe el intercambio de mensajes (o datos) de Alice a Bob y el mensaje no llega a Bob. ¿Puede asegurarse Alice que el mensaje llegará a Bob? (Figura 8.2a)
- **Interceptación (*eavesdropping*)**: Trudy intercepta el contenido e intenta acceder a él. ¿Puede asegurarse Alice que los datos solo los leerá Bob? (Figura 8.2b)
- **Modificación**: Trudy intercepta el contenido y lo modifica antes que Bob lo reciba. ¿Puede asegurarse Bob que el contenido de los datos recibidos es el original? (Figura 8.2c)
- **Fabricación**: Trudy genera un mensaje (o datos) en nombre de Alice y lo envía a Bob. ¿Puede asegurarse Bob que los datos los ha enviado realmente Alice? (Figura 8.2d)
- **Rechazo**: Alice y Bob se intercambian mensajes y llegan a un acuerdo, pero posteriormente uno de los dos, o ambos, rechaza haber hecho el intercambio. ¿Pueden Alice y Bob protegerse contra el rechazo unilateral o mutuo?
  - ¿Puede Bob asegurar que Alice es el único que le ha podido enviar los datos?
  - ¿Puede asegurarse uno de los dos que el otro ha leído los datos, y por tanto ha aceptado su contenido?
- **Denegación de servicio (*Denial of Service*, DoS)**: Trudy inutiliza un servicio. ¿Puede Bob, que está ofreciendo un servicio, estar seguro de que Alice podrá acceder a él?
- **Reutilización (*Replay*)**: Trudy intercepta un mensaje válido (p. ej. una orden de pago), y lo reutiliza posteriormente. ¿Pueden Alice o Bob estar seguros que un mensaje suyo no se reutilizará?
- **Secuestro de conexión (*Hickjacking*)**: Trudy se apodera de una conexión abierta entre Alice y Bob. Podría incluir falsificación de dirección IP. ¿Pueden estar seguros Alice y Bob que nadie les secuestrará la conexión?
- **Privacidad**: Alice dispone de datos personales almacenados y Trudy accede a ellos. ¿Puede Alice asegurar que datos con información suya no le identificarán?




(a) Interrupción (b) Interceptación (c) Modificación (d) Fabricacion

Figure 8.2: Principales amenazas de seguridad

## 8.5 Servicios de seguridad

Para hacer frente a las amenazas anteriores, es necesario aplicar distintos servicios de seguridad, dependiendo de la amenaza que se desee evitar. A continuación se listan y describen brevemente los principales servicios de seguridad:

- **Confidencialidad**: Proporciona secrecidad de los datos intercambiados entre dos partes frente a terceras partes.
- **Integridad**: Protege la información de alteraciones no autorizadas.
- **Autenticación**: Garantiza que el origen de los datos es lo que dice ser.
- **No repudio**: Evita que alguna parte implicada en un intercambio de información niegue haber participado en dicho intercambio.
- **Disponibilidad**: Garantiza que un servicio no es bloqueado para los usuarios legítimos.
- **No reutilización**: Previene que terceras partes capturen y reutilicen más tarde información intercambiada entre dos partes.
- **Autorización / Control de acceso**: Controla que el acceso a los recursos solo sea posible para las partes que tengan derecho de acceso.
- **Anonimización**: Elimina la información de identificación personal de los conjuntos de datos.

Tal como se verá, en general, para obtener muchos de dichos servicios de seguridad se usarán diferentes Métodos de Cifrado de la información.

En esta unidad nos centraremos en los mecanismos que proporcionan Confidencialidad, Integridad, Autenticación, y No repudio, teniendo en cuenta

que la Anonimización ya ha sido tratada en la unidad 6, y la Autorización se verá en la unidad 9.

También cabe destacar que una organización, antes de aplicar seguridad en la red, debe calcular los riesgos y desarrollar una política de seguridad clara sobre el acceso y protección de la información.

## 8.6 Amenazas y Servicios de seguridad

Si analizamos las amenazas y servicios descritos anteriormente, se ve que para cada amenaza existe uno o más servicios de seguridad adecuados, que se listan a continuación:

- Interrupción: Control de los recursos, (Autorización), (Disponibilidad)
- Interceptación: Confidencialidad
- Modificación: Integridad
- Fabricación: Autenticación
- Denegación de Servicio: Disponibilidad
- Reutilización: No reutilización
- Rechazo: No repudio
- Privacidad: Anonimización

## 8.7 Ataques pasivos y activos

Respecto a los ataques, se pueden separar en pasivos y en activos.

Los ataques pasivos intenta aprender o hacer uso de la información del sistema, pero no afectan a los recursos del mismo. El objetivo del atacante es espiar las transmisiones y obtener la información que se transmite, y por este motivo son difíciles de detectar. En este grupo estaría la divulgación del contenido de los mensajes, pero también el análisis del tráfico en la red para poder así extraer información o patrones.

Por otro lado existen los ataques activos, que conllevan alguna modificación del flujo de datos o la creación de un flujo falso. En este caso sí que es posible detectarlos y se deben intentar evitar, o al menos recuperarse de ellos. Dentro de este grupo estarían los ataques de Repetición, Suplantación, Modificación y Denegación de servicio.

## 8.8 Mecanismos de seguridad

La mayor parte de Mecanismos de seguridad de los datos que veremos en este capítulo del módulo se basarán en Métodos (fórmulas) de cifrado y descifrado de dichos datos o mensajes utilizando claves criptográficas.

Para su descripción usaremos la siguiente nomenclatura:

- $K$, $Ke$, $Kd$: Clave, Clave de cifrado, Clave de descifrado
- $M$: Mensaje o datos a cifrar
- $X$: Mensaje o datos cifrados
- $E()$: Mecanismo o Fórmula de encriptación o cifrado
- $D()$: Mecanismo o Fórmula de desencriptación o descifrado
- $H()$: Función de *Hash* o resumen

Es muy importante destacar que cuando en este módulo se habla de mensajes, también incluiría cualquier tipo de datos o fichero sobre los cuales deseáramos aplicar seguridad. Asimismo, cuando se habla del intercambio de dichos mensajes (o datos o ficheros), normalmente nos centramos en la transmisión de los mismos a través de la red, pero la seguridad que se describe en la transmisión también aplicaría a su almacenamiento (discos duros, SSD, lápiz USB, etc.).

Referente a la encriptación, podremos obtener un mensaje (o datos) cifrado, $X$, utilizando una función de encriptación, $E()$, sobre un mensaje (o datos), $M$, con una clave de encriptación, $Ke$.

$$X = E(Ke, M)$$

Asimismo, se podrá obtener de nuevo el mensaje (o datos) original, $M$, utilizando una función de desencriptación, $D()$, sobre ese mensaje (o datos) encriptado, $X$, con una clave de desencriptación, $Kd$.

$$M = D(Kd, X) = D(Kd, E(Ke, M))$$

Aparte, también se utilizarán las funciones de *hash*, que a partir de un mensaje permiten obtener una cadena de longitud fija, y relativamente corta, que identifica de forma única un mensaje.

$$H(M)$$

Tal como se verá más adelante, existen diferentes Métodos de Cifrado que se utilizarán dependiendo de los servicios de seguridad que se deseen obtener.

Pero antes de continuar presentaremos qué son y cómo se representan los Mensajes, qué se entiende por Clave criptográfica, y qué son los mecanismos de Cifrado por bloques y por flujo.

### Cifrar/Descifrar vs Encriptar/Desencriptar

Aunque inicialmente para "Transcribir en guarismos, letras o símbolos, de acuerdo con una clave, un mensaje o texto cuyo contenido se quiere proteger." y su inversa se debían utilizar los términos cifrar y descifrar, en la actualidad ya se aceptan también los anglicismos encriptar y desencriptar, por lo que en esta unidad se utilizarán indistintamente.

### Mensaje

Desde el punto de vista de la informática, un mensaje está formado por una secuencia de caracteres, que a su vez cada uno de los cuales se representa como una secuencia de 8 o más bits (0 o 1) y que se pueden tratar como números. Por lo tanto, un mensaje será una secuencia de bits, y también se corresponderá con un número.

Pongamos por ejemplo la palabra "Hola". Los 4 caracteres de esta palabra se pueden representar, cada uno de ellos con 8 bits:

H: 01001000; o: 01101111; l: 01101100 ; a: 01100001

La palabra "Hola", será la secuencia de los 4 caracteres:

Hola: 01001000011011110110110001100001

Uniendo todos los bits de los 4 caracteres obtenemos un número binario, que también se puede representar en decimal.

Hola = 01001000011011110110110001100001 = 1215261793

Una vez se tiene la representación binaria numérica de un mensaje, o unos datos, ya se podrá aplicar sobre este todo tipo de funciones matemáticas como las asociadas a la encriptación y a la desencriptación.

Cabe destacar, que una vez realizado todo el tratamiento utilizando esta representación numérica binaria, se realizará el paso inverso para obtener caracteres (y por tanto palabras y frases).

### Clave criptográfica

Una clave criptográfica es "similar a una contraseña", pero normalmente es de mayor longitud y se crea mediante algoritmos diseñados para que sea difícil de adivinar. Suele basarse en datos aleatorios o pseudoaleatorios [4].

Cada fórmula de encriptación requerirá unas características para las claves a usar y por tanto para la generación de dichas claves. Además, con igual algoritmo, cuantos más bits (más larga) tenga una clave (normalmente son de 128 bits o más), más segura será, ya que será más difícil de descubrir por parte de un atacante.

### Cifrado por bloques y por flujo

Las funciones de cifrado se dividen en función de cómo se procesa el mensaje (o datos) a tratar: Cifrado por bloque y Cifrado por flujo.

En el Cifrado por bloques el mensaje a cifrar se procesa en bloques de k bits. Por ejemplo, si k = 128, el mensaje se divide en bloques de 128 bits y cada bloque se cifra de forma independiente.

Por otra parte, el Cifrado de flujo convierte el texto plano en texto cifrado tomando 1 bit del texto plano cada vez.

En esta unidad, la mayoría de las funciones de cifrado actuales que se mencionarán serán de cifrado de bloque.

## 8.9 Ataques básicos

En lo referente a los ataques, los más básicos son los que se centran en intentar descubrir la clave o contraseña.

El primer tipo de ataque es el *Ataque de fuerza bruta*, en el que el atacante prueba tantas claves o contraseñas que le es posible con la esperanza de acabar adivinándolas correctamente.

El segundo tipo de ataque es el *Ataque de diccionario*, que estaría más centrado en descubrir contraseñas que en claves, que se basa en probar cadenas de un listado preestablecido, normalmente derivado de una lista de palabras como la de un diccionario (de ahí el nombre de Ataque de diccionario). En este caso se prueban menos contraseñas, ya que solo se prueba las que se consideran más probables. A menudo tiene éxito porque muchas personas tienen la tendencia a elegir contraseñas cortas que son palabras ordinarias o contraseñas comunes, o variantes simples obtenidas, por ejemplo, añadiendo un dígito o un carácter de puntuación. Este ataque es relativamente fácil de derrotar, con contraseñas que no sea una simple variante de una palabra que se encuentre en cualquier diccionario o lista de contraseñas de uso común.

## 8.10 Técnicas de cifrado

En lo referente a las técnicas de cifrado, el objetivo es utilizar fórmulas de encriptación, desencriptación y *hash*, así como mecanismo de generación de claves, con las características adecuadas, pero a la vez públicos y conocidos por todo el mundo.

Siguiendo el segundo principio de Kerckhoffs, "La efectividad del sistema no debe depender de que su diseño permanezca en secreto." Es decir, la seguridad no se basa en esconder los algoritmos, sino en utilizar algoritmos públicos pero demostradamente seguros, y sí en esconder la clave, que en caso de sufrir un ataque siempre se puede cambiar.

A continuación se listan algunas de las características principales de los buenos mecanismos de cifrado:

- La cantidad de secrecidad debe decidir la cantidad de trabajo para la encriptación y desencriptación
- El conjunto de claves del algoritmo de encriptación ha de ser libre de complejidad
- La implementación del proceso ha de ser lo más simple posible
- Los errores en la encriptación no deben propagarse y causar corrupción de la información posterior del mensaje
- La longitud del texto (o datos) encriptado no debe ser (mucho) más largo que el texto del mensaje original

Dentro de las técnicas de cifrado tendremos el Cifrado simétrico (o privado) y el Cifrado asimétrico (o público), además de las funciones de *Hash* y de la Firma digital.

### 8.10.1 Cifrado Simétrico o Privado

En el Cifrado simétrico o privado, las entidades a comunicar, Alice y Bob, comparten la misma clave $K$.

Alice cifra el mensaje (o datos) $M$, con la función de encriptación, $E()$ y utilizando la clave $K$, dando lugar al mensaje encriptado $X$ (Figura 8.3).

$$X = E(K, M)$$

Bob recibe el mensaje encriptado $X$, y lo descifra con una función de desencriptación, $D()$, utilizando la misma clave que ha usado Alex y que él también conoce, obteniendo así el mensaje (o datos) original $M$:

$$M = D(K, X)$$

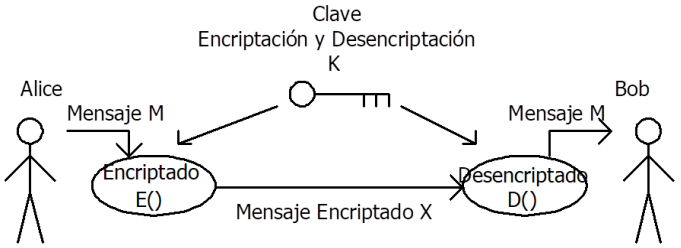



Figure 8.3: Cifrado simétrico

Con este cifrado simétrico se consigue (si nadie no deseado no llega a conocer la clave):

- Confidencialidad: Si se intercepta el mensaje, no podrá desencriptarlo y por tanto acceder a su contenido
- Integridad: Si se intercepta, tampoco no se podrá modificar, ya que antes se debería desencriptar

### Desplazamiento: Cifrado César

Un ejemplo de cifrado simétrico es el que se conoce como Cifrado César, o como cifrado por desplazamiento, código de César o desplazamiento de César [2]. El método debe su nombre a Julio César, quien se dice lo usaba para comunicarse con sus generales.

Es una de las técnicas de cifrado más sencillas y más ampliamente conocidas. Es un cifrado por desplazamiento donde cada letra del texto claro se sustituye por otra letra que esté un determinado número fijo de posiciones desplazadas en el alfabeto. Por ejemplo, con decalaje (clave) 3, la A se sustituiría por la D, la B por E, y así.

Se puede representar alineando 2 alfabetos; donde el alfabeto cifrado es el alfabeto en claro aplicando una rotación a izquierda o derecha n posiciones. El número n de posiciones que se desplaza un alfabeto respecto al otro es el valor que se utilizaría como clave. La misma clave se utilizaría para encriptar como para desencriptar, pero las funciones de encriptar y desencriptar serían distintas, ya que una sería rotación a la izquierda, y la otra a la derecha (daría igual cual es cual, mientras emisor y receptor se pongan de acuerdo).

Ejemplo: Rotación hacia izquierda 4 posiciones (parámetro de decalaje, 4, que se utiliza como clave):

```
Texto en claro: abcdefghijklmnopqrstuvwxyz
```

```
Texto cifrado:  defghijklmnopqrstuvwxyzabc
```

Este mecanismo es fácil de atacar porque solo hay que probar las 25 posibles claves (desplazamientos).

Para cifrar un mensaje, hay que buscar cada letra en la línea "texto en claro" y escribir la letra de la línea "texto cifrado". Para descifrarlo, al revés.

```
Texto en claro: bob.  i love you.  alice
Texto cifrado:  ere.  l oryh arx.  dolfh
```

### Sustitución: El escarabajo de oro (Edgar Allan Poe)

Otro mecanismo simple de cifrado simétrico es la sustitución, en la que se sustituye un carácter por uno (o más) caracteres (o símbolos). En este caso, podemos utilizar el alfabeto con las sustituciones ya hechas como clave.

Por ejemplo podemos tener la clave `mnbvcxzasdfghjklpoiuytrewq`, que significa que la a se escribe como m, la b como n, etc.

```
Texto en claro: abcdefghijklmnopqrstuvwxyz
Texto cifrado:  mnbvcxzasdfghjklpoiuytrewq
```

Y con esta clave, podemos encriptar (y posteriormente también desencriptar) el mismo texto en claro que en el apartado anterior:

```
Texto en claro:  bob.  i love you.  alice
Texto cifrado:  nkn.  s gktc wky.  mgsbc
```

Este método simple de sustitución es de fácil ataque solo con observar la frecuencia de aparición de los caracteres (en cada idioma hay caracteres que aparecen mucho más que otros), o los patrones de letras que se repiten a menudo (p. ej. "the" en inglés).

Un ejemplo de este tipo de encriptación se puede ver en el libro *El escarabajo de oro* (*The Gold-Bug*) de Edgar Allan Poe, donde se encuentra el mensaje de la Figura 8.4.

En el mismo libro se describe brevemente el análisis de la frecuencia de repetición de los caracteres o símbolos, que da lugar a la obtención la correspondencia entre estos y el texto en claro:

```
Texto en claro: abcdefghijklmnopqrstuvwxyz
Texto cifrado:  52-†81346,709*‡.$();?¶]¢:[
```

Se puede considerar "`52-†81346,709*‡.$();?¶]¢:[`" como la clave, y con su obtención permite el descifrado del texto, que es el siguiente (primero el original en ingles, seguido de una de sus diversas traducciones al castellano):

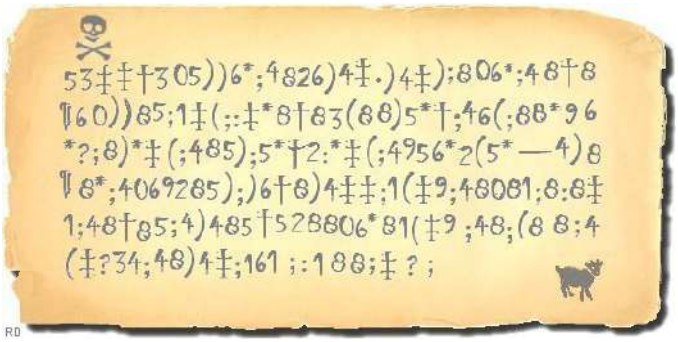



Figure 8.4: Texto (en inglés) cifrado por substitución en *The Gold-Bug*, Edgar Allan Poe

> *A good glass in the bishop's hostel in the devil's seat forty-one degrees and thirteen minutes northeast and by north main branch seventh limb east side shoot from the left eye of the death's head a bee-line from the tree through the shot fifty feet out*
>
> Un buen vidrio desde el hotel del obispo en el asiento del diablo cuarenta y un grados trece minutos norte nordeste tronco principal séptima rama este tiro por el ojo izquierdo de la calavera línea recta desde el árbol siguiendo el tiro cincuenta pies

### Sustitución y permutación: *Data Encryption Standard* (DES)

Otro ejemplo de mecanismo simétrico (ya casi no utilizado por inseguro) es el *Data Encryption Standard*, DES, y a veces también llamado *Data Encryption Algorithm*, DEA).

Este algoritmo es una combinación, fácil de implementar y de relativa rapidez en el cálculo, de sustituciones (que acabamos de ver) y permutaciones (Figura 8.5):

- Sustitución: Sustitución de unos bits por otros, proporcionando confusión
- Permutación: Reordenación de los bits, proporcionando difusión

Cabe resaltar de nuevo que tanto las sustituciones como las permutaciones en si, no son complejas de implementar, y lo que aportará la seguridad al método es su combinación de forma "inteligente" (que tampoco aportará mucha complejidad) junto con el uso de claves secretas (y que no hayan sido comprometidas).

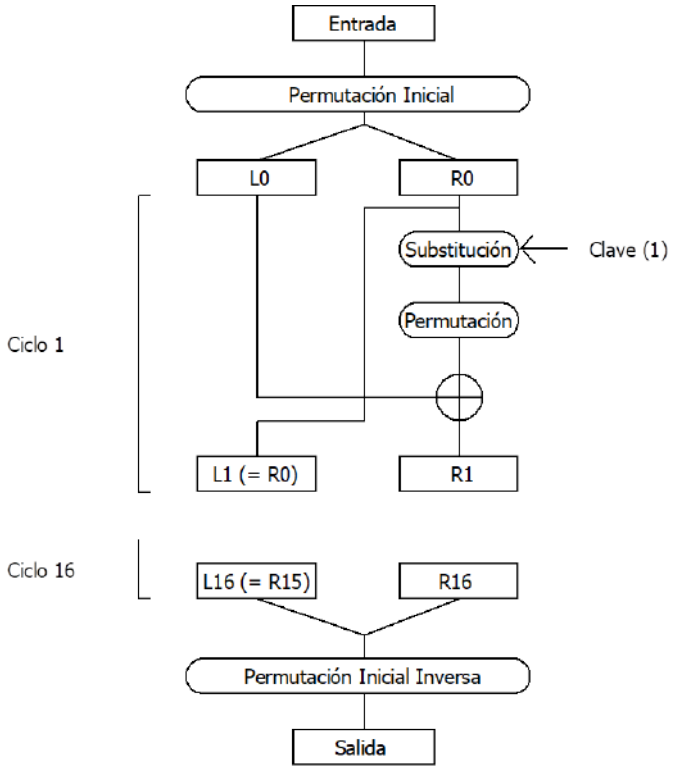



Figure 8.5: Data Encryption Standard (DES)

En el DES, los algoritmos de Sustitución y Permutación son conocidos y ya fijados. Esas 2 técnicas se repiten, 1 detrás de otra, durante 16 ciclos.

El mensaje original se divide y encripta en bloques de 64 bits, y aunque la clave es de 64 bits de longitud (actualmente ya proporciona muy poca seguridad), su longitud efectiva a efectos de seguridad son 56 bits (el resto, 8 bits, son de paridad para control de errores), y a partir de esa clave se generan 16 subclaves de 48 bits cada una de las cuales se utiliza en 1 de los ciclos.

Los algoritmo de encriptación y de desencriptación son exactamente el mismo, pero tomando las 16 subclaves en orden inverso.

De todas formas DES ya no se considera seguro y por este motivo se creó AES, *Advanced Encryption Standard* para en cierto modo sustituirlo. AES procesa los datos en bloques de 128 bits, y soporta claves de 128, 192 o 256 bits.

### 8.10.2 Cifrado Asimétrico o Público

Después de ver el cifrado simétrico, veremos el cifrado asimétrico o público. Su característica es que las claves van en pares complementarios (Figura 8.6), y que aún conociendo una de ellas es imposible deducir la otra:

- 1 clave para encriptar ($Ke$)
- 1 clave para desencriptar ($Kd$)


Figure 8.6: Cifrado asimétrico

En este mecanismo, a cada usuario (p. ej. Alice o Bob), se le asigna un par de dichas claves complementarias:

- $Ks$: Clave Privada o Secreta. Solo la conoce el usuario correspondiente.
- $Kp$: Clave Pública. Compartida por todos los usuarios.

En nuestro escenario, donde tenemos Alice y Bob, habrá 4 claves:

- $KAs$ Clave Secreta de Alice. Solo la conoce Alice.
- $KAp$ Clave Pública de Alice. La conoce todo el mundo.
- $KBs$ Clave Secreta de Bob. Solo la conoce Bob.
- $KBp$ Clave Pública de Bob. La conoce todo el mundo.

Las propiedades más importantes del Cifrado Asimétrico son:

- Si se encripta con una clave pública, $X = E(Kp, M)$, entonces solo se puede desencriptar con la clave secreta complementaria, $M = D(Ks, X)$
- Si se desencripta con la misma clave que se ha encriptado, no se obtiene el mensaje original: $M <> D(Kp, X)$

- También se puede utilizar una clave secreta para encriptar, $Y = E(Ks, M)$, y en este caso solo se puede desencriptar con la clave pública complementaria, $M = D(Kp, Y)$

Aunque se puede utilizar tanto la clave pública como la secreta para encriptar, dependiendo del modo en que se use, el cifrado asimétrico proporcionará 2 tipos distintos de seguridad, que se describen a continuación:

- modo Encriptación
- modo Autenticación

### 8.10.3 Cifrado Asimétrico o Público - Modo Encriptación

La primera opción es el Cifrado asimétrico modo encriptación, en el cual Alice, la emisora, encripta el mensaje (o datos), $M$, con una función de encriptación, $E()$, utilizando la clave pública de Bob, $KBp$, que es el receptor, dando lugar al mensaje encriptado $X$ (Figura 8.7).

$$X = E(KBp, M)$$

Bob desencripta el mensaje (o datos) encriptado, $X$, con una función de desencriptación, $D()$, utilizando su clave secreta $KBs$, y obtiene el mensaje original, $M$.

$$M = D(KBs, X)$$

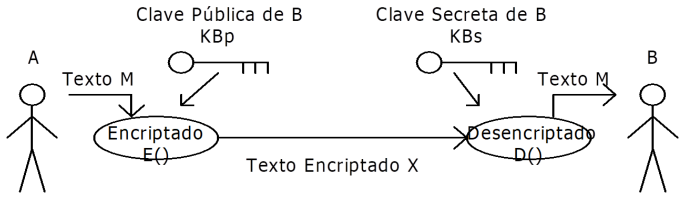



Figure 8.7: Cifrado asimétrico modo encriptación

Este Mensaje solo lo podrá desencriptar, y por tanto leer, el único usuario que posee $KBs$ (la clave complementaria a la usada en la encriptación), que es Bob, y tampoco nadie lo podrá haber modificado (ya que también necesitaría $KBs$ para desencriptarlo antes de modificarlo). Por lo tanto, este mecanismo proporciona:

- Confidencialidad
- Integridad

### 8.10.4 Cifrado Asimétrico o Público - Modo Autenticación

La segunda opción es el Cifrado asimétrico modo autenticación . En este caso, Alice utiliza su propia clave secreta, $KAs$, para encriptar el mensaje (o datos) $M$, con una función de encriptación, $E()$ (Figura 8.8).

$$X = E(KAs, M)$$

Para desencriptarlo, Bob utilizará una función de desencriptación, $D()$, con el mensaje encriptado, $X$, y la clave pública de Alice, $KAp$. Así obtendrá el mensaje original, $M$:

$$M = D(KAp, X)$$

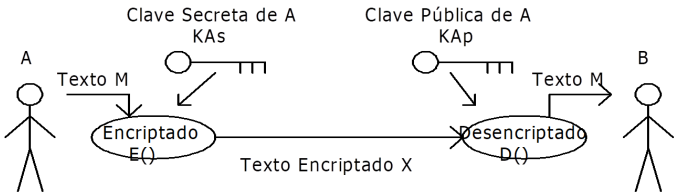



Figure 8.8: Cifrado asimétrico modo autenticación

El mensaje $M$ lo puede obtener cualquiera descifrando $X$, ya que la clave necesaria, $KAp$, es pública. Por tanto, el mensaje no es confidencial.

Pero si se es capaz de descifrar el mensaje $X$ con la clave pública de Alice, $KAp$, se puede demostrar que ese mensaje solo lo puede haber generado el usuario que posee $KAs$, que es Alice. Asimismo, si se ha podido descifrar el mensaje recibido, este será exactamente igual al enviado, ya que tampoco nadie lo podrá haber modificado y vuelto a cifrar ya que no posee $KAs$.

Por lo tanto, este mecanismo proporciona:

- Integridad
- Autenticación de origen

De todos modos, el Cifrado Asimétrico o Público en Modo Autenticación tiene la desventaja que es necesario cifrar todo el mensaje para que al final cualquier usuario pueda acceder a su contenido, ya que la clave de desencriptación es pública.

**Encriptación Rivest-Shamir-Adelman (RSA)**

Uno de los mecanismos de encriptación asimétrica utilizados es el Rivest-Shamir-Adelman (RSA).

Solo como información, este mecanismo opera con aritmética módulo n, y el mensaje es tratado como un número entero sin signo y partido en trozos de como máximo la misma longitud que la longitud de un valor n (valor usado en una fórmula que se mencionará a continuación). Las claves son típicamente de 1024 a 4096 bits. Es decir, los datos se parten en bloques de igual longitud (excepto el último trozo de los datos), y se tratan como números enteros.

En RSA la fórmula de encriptación y desencriptación es la misma y lo que varía, recordar que es un mecanismo asimétrico, son las claves de encriptación, $Ke$, y desencriptación, $Kd$. Por lo que hace referencia a la fórmula, simplemente es coger un bloque del mensaje original (longitud menor que la longitud de n, como se ha dicho), elevarlo a la clave ($Ke$ para el cifrado o $Kd$ para el descifrado), hacer la división por el número n, y obtener el resto de dicha división (la operación para obtener el resto de una división se conoce como operación *módulo*, mod ). Con esta operación se obtiene un bloque cifrado o descifrado según sea el caso.

- $X = M^{Ke} \mod n$
- $M = X^{Kd} \mod n$

En este método también se ve que desde el punto de vista matemático, tanto las funciones de división (para obtener su resto) como la potencia no son operaciones complejas (aunque computacionalmente esta segunda necesita más recursos).

Para que el mecanismo funcione, es importante la generación de claves de manera que se cumpla que el resto de la división del mensaje elevado al producto de las 2 claves de nuevo el mismo mensaje. El mecanismo para la generación de claves que cumpla esta condición es conocido, y basado en la utilización de números primos grandes.

- $M^{Ke*Kd} \mod n = M$.

Si se cumple esta condición, y teniendo en cuenta alguna de las propiedades del módulo (resto) de la división, se puede ver que con RSA el descifrado de un mensaje cifrado, tal como se espera, vuelve a dar el mensaje inicial:

- $M = X^{Kd} \mod n = (M^{Ke})^{Kd} \mod n = M^{Ke*Kd} \mod n = M$

También se puede ver que la encriptación y desencriptación son mutuamente inversos. Es decir, que primero se puede usar una de las claves para encriptar y luego la otra para desencriptar, o se pueden usar en orden inverso.

- $M = X^{Kd} \mod n = (M^{Ke})^{Kd} \mod n = (M^{Kd})^{Ke} \mod n = M^{Kd*Ke} \mod n$

### 8.10.5 Técnica Asimétrica vs Técnica Simétrica

Antes de continuar, es interesante comparar los mecanismos de cifrado simétrico con el asimétrico. En la técnica asimétrica:

+ Hay menor tarea de gestión de las claves.

+ El período de validez de las claves es más largo.

– Las Claves suelen ser más largas.

– Hay menor seguridad.

– Los algoritmos son más complejos y lentos.

Para solucionar algunas de estas desventajas veremos la Firma digital y las Claves de sesión.

### 8.10.6 Firma digital

El primer mecanismo que permite solucionar alguna de las desventajas del uso de mecanismos asimétricos es la Firma digital. El mecanismo de Cifrado asimétrico en modo autenticación proporciona autenticación e integridad, pero no confidencialidad, a costa de tener que cifrar de forma asimétrica (más costosa computacionalmente que la simétrica) todo el documento, al que todo el documento puede acceder .

El mecanismo de firma digial nos proporcionará la misma seguridad que nos proporcionaría una firma física sobre un documento: Autenticación del origen del documento, así como garantía que el documento no ha sido modificado (integridad), pero con menos costo computacional y más rapidez que el mecanismo asimétrico modo autenticación, que de hecho proporciona la misma seguridad.

Para explicar la Firma digital será necesario primero saber qué es una función de resumen (o *hash*).

### Función de resumen (*hash*)

Las funciones de resumen (o *hash*) son unas funciones que transforman una cadena de caracteres en un valor o una clave de longitud fija, generalmente más corta, que representa (y "resuma" de algún modo) el documento (o datos) original, de modo que si se modifica el documento o su *hash*, se puede detectar. El carácter de control del DNI/NIF podría considerarse una versión muy simple del *hash* del número de DNI.

A continuación se listan las características principales del *hashing*:

- El algoritmo de *hashing* se llama función de *hash*.
- El *hashing* es siempre una operación unidireccional, es decir, a partir del *hash* no se debe poder obtener la cadena original.
- Una buena función de *hash* no debería producir el mismo valor de *hash* a partir de 2 cadenas diferentes, ni se debe poder manipular una cadena original para que su modificación tenga el mismo *hash* o un *hash* deseado. De todas formas, es verdad que si se utiliza un *hash* de n bits, tendremos $2^n$ hashes distintos, y si tenemos un número mayor que este de mensajes (o datos), seguro que habrá mensajes con el mismo *hash*. Con el ejemplo del NIF/DNI, se utilizan solo 23 caracteres distintos para la letra de control, con lo cual, con 24 o más DNI seguro que al menos hay 2 con la misma letra.
- Las funciones de *hash* no han de ser desconocidas (y no lo son) ni de mucha complejidad matemática.
- Algunas funciones de *hash* pueden complementarse con el uso de una clave simétrica.

Desde el punto de vista de seguridad, las funciones de *hash*, como permiten detectar modificaciones, proporcionan Integridad, pero con poca complejidad matemática. No proporcionan confidencialidad porque solo son un resumen que acompaña al documento (o datos) original (que sí puede ir encriptado), ni autenticación de origen ya que cualquiera puede calcular un *hash* porque las funciones son públicas.

En el caso de las funciones de *hash* que se complementan con una clave que se haya intercambiado al inicio de forma autenticada, se puede considerar que en cierta manera el *hash* también proporciona autenticación (al menos autenticación que Alice o Bob, que disponen de la clave intercambiada, lo ha generado, y no ha sido Trudy).

### Firma digital

Una vez explicadas las funciones de *hash*, ya se puede continuar con la Firma digital. Al igual que en los documentos físicos, Alice envía a Bob un mensaje (o datos) $M$ junto con su firma $S$, donde:

- $M$ es el mensaje (o datos) que se intercambian, que puede ser enviado sin encriptar, $M$, o encriptado, $X$, dependiendo si se desea confidencialidad o no
- $H()$ es una función de *hash* conocida y pública
- $S$ es la Firma digital

Por lo que hace referencia a la Firma digital, S, simplemente es la encriptación con un mecanismo asimétrico del *hash* del mensaje a enviar, $H(M)$, utilizando la clave secreta de Alice, el emisor, $KAs$. Como el cálculo del *hash* es poco costoso, ya se puede apreciar que el hecho de encriptar solo dicho *hash*, y no el documento entero, es mucho menos costoso y más rápido que el Cifrado asimétrico en modo autenticación.

Si Alice quiere proporcionar confidencialidad (y de hecho también proporcionaría integridad) al mensaje, lo puede hacer cifrando con una función simétrica o con una función asimétrica con la clave pública de Bob, $KBp$.

Alice envía el mensaje (sin encriptar o encriptado) más la firma: M o X + S (Figura 8.9):

> $M$ o $X = E(K, M)$ o $X = E(KBp, M)$: Texto o Texto Encriptado (Confidencialidad + Integridad)
>
> $S = E(KAs, H(M))$: Firma digital (Autenticación + Integridad)

Bob recibe un mensaje $M'$ (sin encriptar) o $X'$ (encriptado) y una firma $S'$, que no sabe si son realmente el mensaje y la firma originales:

> $M'$ o $X'$ ($M'$ encriptado)
>
> $S'$ ($H(M)'$ encriptado)

Entonces, Bob podrá desencriptar la firma recibida, $S'$, con una función de desencriptado, $D()$, y la clave pública de Alice, $KAp$, y si la función no da ningún error, se obtiene así el *hash* del mensaje original, $H(M)$.

En caso de que el mensaje esté encriptado, Bob lo puede desencriptar con una clave simétrica, $K$, o su clave secreta, $KBs$, según sea el caso.

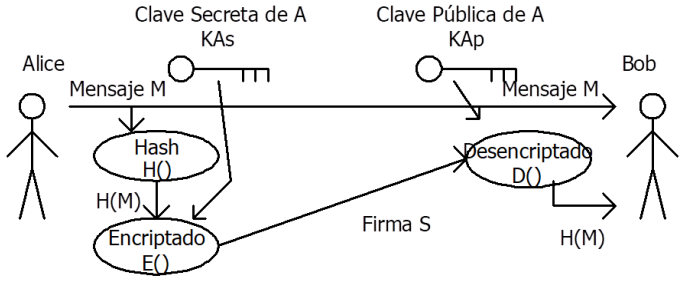



Figure 8.9: Firma digital

Y una vez obtenido el mensaje, o datos $M'$ (ya sea porque se ha recibido sin encriptar como si se ha recibido encriptado y se ha desencriptado), Bob ya puede calcular su *hash*, $H(M')$. Si el *hash* del mensaje recibido y el *hash* extraído de la firma son idénticos, es que el mensaje recibido es exactamente el mismo que se ha enviado, por lo que se tiene integridad. Asimismo, como se ha usado la clave pública de Alice, $KAp$, para obtener el *hash* a partir de la firma, se puede asegurar que la firma (y por tanto el documento (o datos) cuyo *hash* está en su interior) solo lo puede haber generado Alice, y por tanto también se obtiene autenticación de origen.

$$M' \text{ o } M' = D(K, X') \text{ o } M' = D(KBs, X')$$

$$H(M) = D(KAp, S')$$

$$H(M') =? H(M)$$

Por lo tanto, con la Firma digital se consigue:

- Integridad
- Autenticación de origen

Además, si se ha encriptando el mensaje, también se consigue Confidencialidad.

Se ve que con la Firma digital se obtienen la misma seguridad que en el cifrado asimétrico modo autenticación, pero con menos cómputo, ya que el cálculo del *hash* tiene muy poca complejidad, y posteriormente solo se encripta dicho *hash*, que es de longitud muy inferior al mensaje completo.

### No Rechazo

El servicio de seguridad para evitar el rechazo de un mensaje (o datos), ya sea por parte del emisor (rechazar haber hecho el envío o generado unos datos)

como por parte del receptor (rechazar haber recibido o leído un mensaje) tiene mucha relación con la autenticación que se ha visto en el mecanismo de firmas digitales.

Con este fin, habría 2 posibles opciones:

La primera es el uso de un Notario (o Tercera Parte de Confianza, *Trusted Third Party*, TTP) (Figura 8.10), que actúe como intermediario en el intercambio de información, y que de fe de dicho intercambio.

La comunicación con notario requiere los servicios de autenticación, integridad y "Reconocimiento" mutuo.

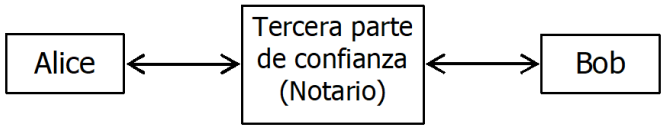



Figure 8.10: Notario / *Trusted Thir Party* (TTP)

La segunda opción es no utilizar ningún notario en el intercambio, y sustituirlo por una prueba de origen y una prueba de entrega (Figura 8.11).

La prueba de origen la puede generar Alice firmando, $S_A$, el mensaje (o datos) que envía:

$$M \text{ o } X \ (= E(KBs, M) \text{ o } E(K, M))$$

$$S_A = E(KAs, H(M))$$

Asimismo, la prueba de entrega (o reconocimiento de entrega) constaría de la firma por parte de Bob, $S_B$, del mensaje recibido, y opcionalmente adjuntando también el mismo mensaje:

$$\text{opcionalmente, } M \text{ o } X \ (= E(KAs, M) \text{ o } E(K, M))$$

$$S_B = E(KBs, H(M))$$

En caso de desearse confidencialidad, el mensaje se intercambiaría encriptado con una técnica simétrica o asimétrica cifrada con la clave pública del receptor.

### 8.10.7 Clave de sesión

El segundo método para solucionar la desventaja de los requerimientos de cómputo de los mecanismos asimétricos son las claves de sesión

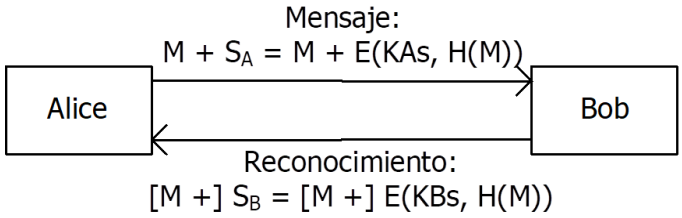



Figure 8.11: No rechazo, sin notario

Una Clave de sesión es una clave simétrica de un solo uso que se utiliza para cifrar todos los mensajes en una sesión (o conexión o intercambio) de comunicación. Para su funcionamiento se requiere una fase inicial de intercambio seguro de la Clave (simétrica) de sesión, que si se realiza utilizando un mecanismo asimétrico nos proporcionará autenticación de los usuarios involucrados. Esa Clave de sesión simétrica servirá para el cifrado del resto de mensajes, proporcionando confidencialidad e integridad a los mismos.

Las claves de sesión, tal como se ha visto, proporcionarán autenticación e integridad, pero lo importante es que la principal carga computacional será la del mecanismo simétrico utilizado en la fase de intercambio de mensajes.

Como curiosidad, este es el mecanismo que se utiliza en los navegadores web cuando se utiliza el protocolo HTTPS, y en la barra superior, a la izquierda de la URL, aparece un candado cerrado en verde. Primero hay una autenticación del servidor (con mecanismo asimétrico) y un intercambio seguro de 4 claves de sesión (una para encriptación del texto y otra para autenticación en *hash*, para los intercambios de Alice a Bob, pero también de Bob a Alice). Una vez intercambiadas las claves, el intercambio de datos se realiza con esas claves.

Finalmente, cabe resaltar que existen mecanismos para el intercambio seguro de claves a través de un medio potencialmente seguro, como puede ser el mecanismo de intercambio de claves Diffie-Hellman. En este mecanismo, Alice y Bob generan un número secreto cada uno, y se intercambian "información parcial" de su número. Con el número secreto propio más la "información parcial" recibida, Alice y Bob pueden generar el mismo número compartido, que puede ser utilizado como clave simétrica. En este mecanismo, aunque el canal no sea seguro y Trudy vea las 2 "informaciones parciales" intercambiadas, nunca podrá generar el número compartido final (la clave simétrica).

## 8.11 Resumen de técnicas

Una vez descritas las principales técnicas de seguridad a continuación se resume la seguridad que proporcionan.

- Técnica simétrica: Confidencialidad, Integridad
- Técnica asimétrica modo encriptación: Confidencialidad, Integridad ($KBp$)
- Técnica asimétrica modo autententicación: Autenticación de Origen, Integridad ($KAs$) (No confidencialidad.) y No Rechazo
- Firma digital: Autenticación de Origen, Integridad ($KAs$) (No confidencialidad.) y No Rechazo

### 8.11.1 Tendencias

Como se ve, hay distintas técnicas que proporcionan la misma seguridad, pero tal como se ha visto algunas son mejores o más eficientes que otras. A continuación se listan las técnicas más habituales, junto con algunos de los algoritmos que las implementan

- Confidencialidad e integridad: Técnica simétrica (AES, [DES,] 3DES, ...)
- Autenticación y no rechazo: Técnica asimétrica y Firma digital (RSA, Elgamal, DSA [firmas digitales], Criptografía de curvas elípticas, RSA-DSA, ...)
- Integridad: *Hash* (MD5, SHA-1, ...)

## 8.12 Distribución y gestión de claves

Uno de los problemas más importantes en las comunicaciones seguras es la gestión y distribución de las claves, ya que hay que conseguir que las entidades o usuarios implicados en la comunicación obtengan de forma segura (confidencial y íntegra), las claves necesarias para poder encriptar y desencriptar los datos

Dependiendo de la técnica (simétrica o asimétrica), se tendrá una serie de ventajas y desventajas en cuanto a la distribución y gestión de claves.

Supongamos N usuarios, y se quiere añadir 1 nuevo usuario (N+1), es necesario intercambiar las claves necesarias con el resto de usuarios.

Referente a la distribución de claves en la técnica simétrica, se requiere la distribución de forma privada a cada uno de los N usuarios, de 1 clave compartida con N+1. Para tal fin, se requiere establecer canales seguros para distribuir las claves que deben garantizar la seguridad posterior.

En la técnica asimétrica, las claves distribuidas son las públicas (K1p, ..., KNp), que es tan sencillo como publicar dichas claves. De esta manera, la distribución siempre se realiza de forma segura. Esta técnica es la más conveniente para la distribución.

En cuanta a la gestión de claves, es necesaria para establecer algún mecanismo que asegure (certifique) la pertenencia de las claves distribuidas al usuario indicado en exclusiva. Por este motivo es necesario definir una política de seguridad para la comunidad de entidades y usuarios perteneciente a un dominio gestionado por la misma autoridad

En la técnica simétrica la gestión la realizarán los Centros de distribución de claves (*Key Distribution Center*, KDC), mientras que en la técnica Asimétrica serán las Autoridades de certificación (*Certification Authority*, CA) que utilizarán el mecanismo de Infraestructura de clave pública (*Public Key Infrastructure*, PKI).

#### Key Distribution Center

Tal como se ha comentado, los Centros de distribución de claves (*Key Distribution Center*, KDC) son los encargados de la gestión de las claves en la técnica simétrica. Son centros que almacenan de forma segura todas las claves correspondientes a la comunidad, y deben disponer de seguridad, ya que ha de proteger todas las claves de la comunidad.

Los KDE comparte una clave simétrica secreta diferente con cada usuario registrado (clave que se puede instalar manualmente en el servidor cuando un usuario se registra). El KDC conoce la clave secreta de cada usuario y cada usuario puede comunicarse de forma segura con él utilizando esta clave. El conocimiento de esta clave permite a un usuario obtener una clave segura para comunicarse con cualquier otro usuario registrado.

## 8.13 Infraestructura de clave pública (*Public Key Infrastructure*, PKI)

La Infraestructura de Clave Pública (*Public Key Infrastructure*, PKI) es el conjunto de mecanismos utilizado para la gestión de las claves asimétricas. Este incluye la combinación de procesos, algoritmos y estructuras de datos

que permiten asegurar la Identidad de los participantes en un intercambio de datos mediante la utilización de técnicas criptográficas

La seguridad en PKI se basa en el concepto de Certificado digital, que se verá a continuación, con lo que una PKI es un conjunto de hardware, software, personas, políticas y Procedimientos necesarios para crear, gestionar, distribuir, utilizar, almacenar y revocar dichos certificados digitales.

### 8.13.1 Certificado digital

Un Certificado digital es un mecanismo para establecer una relación fiable entre una Clave pública y la Identidad de su propietario. Es una estructura de datos ("fichero") que incluye información (Figura 8.12), entre otras, de la Identidad del titular del certificado, la Clave pública de dicho titular, el periodo de validez de la clave, el algoritmo de firma de la clave, etc., todo ello firmado digitalmente por una Autoridad de Certificación (que es un Tercero de confianza) con su clave secreta, KCAs.

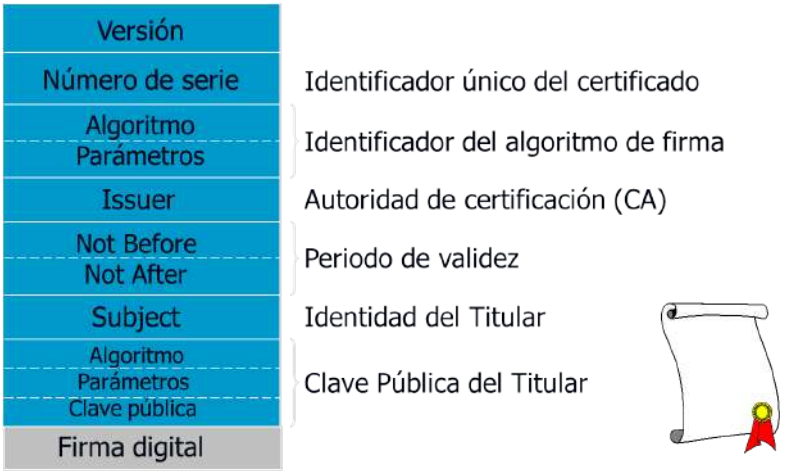



Figure 8.12: Campos principales de un Certificado digital

Un Certificado digital, a diferencia de lo que su nombre podría indicar, no es un mecanismo que permita autenticar un documento, sino que es el mecanismo (un "fichero") para guardar y distribuir de forma segura y autenticada las claves públicas de los usuarios.

Cabe resaltar que para comprobar la autenticidad de un Certificado digital, lo único necesario, tal como se ha visto anteriormente en el apartado de Firmas digitales, es comprobar la firma utilizando la clave pública del firmante, en este caso, de la Autoridad de Certificación, KCAp.

De hecho, cuando hasta ahora hemos hablado de "se intercambia una clave pública", lo que en realidad se intercambia es un certificado en cuyo interior hay, entre otros campos, la clave pública junto el nombre del usuario asociado a dicha clave, y todo ello firmado digitalmente por una Autoridad de Certificación que le proporciona autenticación e integridad.

### 8.13.2 Autoridad de Certificación (CA)

A partir de la estructura de los certificados, se puede ver que el sistema de PKI se basa en la firma que incluyen los certificados, que les proporciona autenticación e integridad. Las Autoridades de Certificación (Certification Authority, CA), son entidades que se pueden considerar como Terceras Partes de confianza (*Trusted Third Party*, TTP) que se encargan de la emisión y revocación de los Certificados digitales. Estas son las encargadas de firmar los certificados, validando así su contenido (incluida la identidad del titular y su clave pública).

Para que todo el sistema de PKI funcione y sea confiable, la clave secreta de una CA se ha de mantener, y se mantiene, bajo estrictas medidas de seguridad.

#### Jerarquía de Certificación

Uno de los "problemas" de la infraestructura de PKI es en qué Autoridades de Certificación confiamos? ¿Como estamos seguros que la clave pública de una Autoridad de certificación, KCAp, que firma un certificado pertenece realmente a ella? La clave pública de toda autoridad de certificación, como toda clave pública, se distribuirá también a través de un certificado firmado, por lo general por otra Autoridad de certificación, dando lugar a una estructura jerárquica entre Autoridades de certificación.

El modelo se basa en una "pocas" Autoridades de Certificación muy conocidas y muy confiables, llamadas CA raíz, que publican su Certificado digital con su clave pública firmada por ellas mismas (con clave secreta de la misma CA). Es lo que se conoce como un Certificado autofirmado. La seguridad se basa en que son tan conocidas que su Certificado también es muy conocido y público (p. ej. los navegadores web ya incluyen los Certificados digitales de dichas Autoridades de Certificación).

Estas Autoridades de Certificación raíz, proveen de certificados a otras CA subordinadas para que generen y firmen certificados ya sea para otras CA para que también generen y firmen certificados, o directamente generen certificados para usuarios finales. Esta jerarquía de Autoridades de certificación, cada una con su certificado firmado por la CA superior (excepto la

CA raíz que es autofirmado, es decir, firmado por ella misma), y también con usuarios finales con sus certificados, se puede ver en la Figura 8.13. En dicha figura, los jueces representan Autoridades de Certificación, los pergaminos representan certificados digitales, y el sello que incluyen, su firma. Excepto la Autoridad de Certificación superior (CA raíz), que ha firmado ella misma su certificado, el resto de certificados, tanto CA como usuarios, están firmados por la CA de la capa inmediatamente superior.

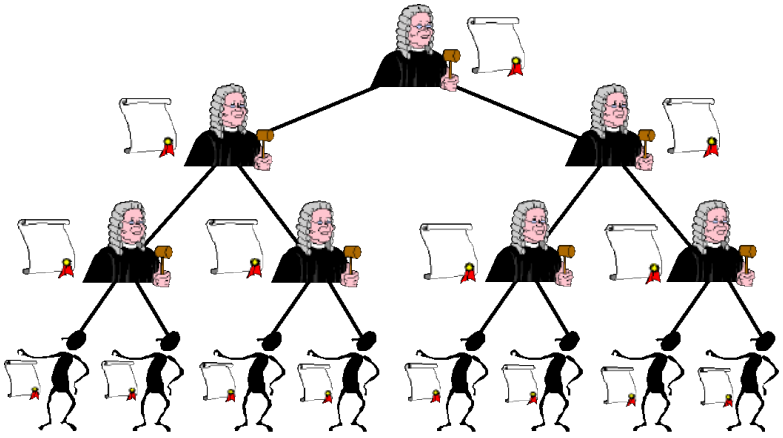


Figure 8.13: Jerarquía de Autoridades de Certificación (CA)

### Lista de Revocación de Certificados (*Certificate revocación List*, CRL)

Tal como se ha visto, todo Certificado incluye su periodo de validez, pero puede ser necesario revocarlo antes del fin de su validez (igual que sucede con las tarjetas de crédito) por motivos diversos:

- El titular del certificado ha cambiado su afiliación
- La clave secreta del titular se ha visto comprometida
- La clave secreta de la CA se ha comprometido

Es aquí donde aparece el concepto de Lista de Revocación de Certificados (*Certificate Revocación List*, CRL). Tal como su nombre indica, es una lista pública de certificados revocados por la CA emisora de los certificados, firmada por esa misma CA.

Por este motivo, cualquier firma o autenticación realizada con una clave privada asociada a dichos certificados una vez ya revocados, no tiene validez ni se puede confiar en la autenticación (ni integridad o confidencialidad si es el caso).

**Publicación de Material Criptográfico**

Finalmente, cabe destacar que aunque los Certificados son públicos y han de ser fáciles de acceder, con lo que la publicación de certificados, tanto los de los usuarios como también los de las CA, es necesaria para cifrado. Para dicho fin, las CA pueden utilizar el web, o un mecanismo (protocolo) conocido como Directorio (protocolos LDAP o X.500), que no entraremos a detallar en este módulo.

### 8.13.3 Autoridad de Registro (*Registration Authority*, RA)

Para que el mecanismo de PKI funcione, es necesaria, además de la Autoridad de Certificación, una segunda entidad: una Autoridad de Registro (*Registration Authority*, RA). Una RA es la entidad responsable de la verificación de los datos que se han de añadir al certificado, incluyendo la identidad del titular del certificado, y los permisos para los usos autorizados del certificado (extensiones), así como de la gestión del soporte (formato) en el que el titular guardará sus claves privadas.

La Política de Certificación (*Certificate Policy*) es un documento público que establece de forma no ambigua los mecanismos que la CA establece con las RAs en el proceso de certificación

## 8.14 *Pretty Good Privacy*, PGP

A parte del mecanismo basado en la Infraestructura de clave pública (*Public Key Infrastructure*, PKI), es interesante comentar *Pretty Good Privacy*, PGPl como una alternativa. PGP es un programa informático de cifrado y descifrado de datos que proporciona privacidad y autenticación criptográfica para la comunicación de datos. PGP y otros productos similares siguen el estándar OpenPGP (RFC 4880) [3].

Al igual que en PKI, cada clave pública está vinculada a un nombre de usuario y/o a una dirección de correo electrónico.

La primera versión se basaba en una red de confianza (*web of trust*), frente al sistema X.509 que se ha visto, que utiliza un enfoque jerárquico basado en la autoridad del certificado. De todas formas, las versiones actuales del cifrado PGP incluyen ambas opciones.

### 8.14.1 Red de confianza)

El modelo de Red de confianza de PGP (Figura 8.14) es un modelo de confianza descentralizado donde unos usuarios confían de otros usuarios, a diferencia del de PKI que es un modelo jerárquico basado en la confianza en Autoridades de Certificación.

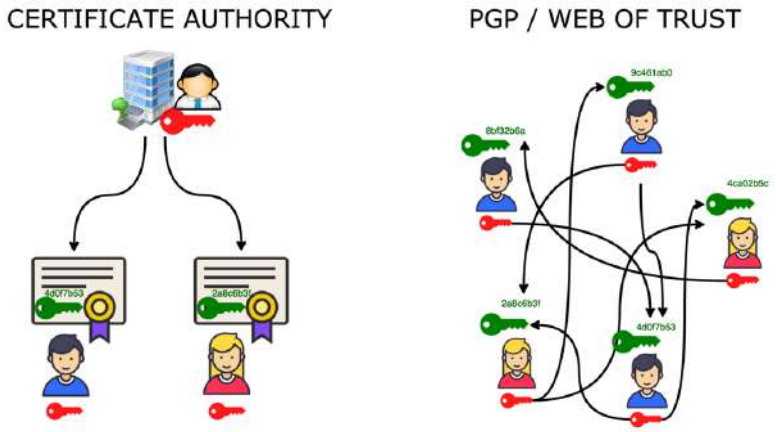



Figure 8.14: Red de confianza PGP

En PGP son mismos los usuarios los que firman las claves de otros usuarios en los cuales confían. De esta manera, los usuarios confían en las claves firmadas por los usuarios en los que ellos confían, y una clave tendrá tantas firmas como usuarios que confían en ella (Figura 8.15).


Figure 8.15: Una clave PGP incluye las firmas de los usuarios que confían en ella

Cuando un usuario confía en alguien, añade su Firma digital a la clave de este, diciéndole al resto de usuarios que "confío en esta persona", por lo que si la gente confía en ti (o en el Dr. Müller) es probable que también

confíe en la persona cuya clave tú has firmado. Cuantas más firmas, más probabilidades hay de hablar con la persona real.

## 8.15 Resumen y Conclusiones

Para finalizar este capítulo, solo hace falta recordar que se han presentado los conceptos básicos de seguridad y cifrado, teniendo en cuenta que en el entorno de los Datos de Salud la seguridad es primordial para el almacenaje y acceso seguro a datos de salud privados.

Primero se ha visto brevemente el Reglamento General de Protección de Datos (GDPR), centrándonos en el entorno de la Salud.

A continuación se han descrito las principales amenazas de seguridad, sus servicios, y los mecanismos de encriptación que proporcionan dichos servicios, finalizando con los mecanismos de gestión de claves.

# Bibliografía


[1] Alice y bob. https://es.wikipedia.org/wiki/Alice_y_Bob. Último acceso: 01/10/2021.

[2] Cifrado césar. https://es.wikipedia.org/wiki/Cifrado_C%C3%A9sar. Último acceso: 01/10/2021.

[3] Openpgp message format (rfc 4880). https://datatracker.ietf.org/doc/html/rfc4880. Último acceso: 01/10/2021.

[4] Password vs key. https://simplicable.com/new/password-vs-key. Último acceso: 01/10/2021.

[5] Reglamento (ue) 2016/679 del parlamento europeo y del consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la directiva 95/46/ce (reglamento general de protección de datos). https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX:32016R0679. Último acceso: 01/10/2021.

[6] Agencia Española de Protección de Datos. Informe de cumplimiento de la lopd en hospitales. https://web.archive.org/web/20170930031728/http://www.agpd.es/portalwebAGPD/revista_prensa/revista_prensa/2010/notas_prensa/common/octubre/INFORME_HOSPITALES.pdf, https://slideplayer.es/slide/3916814/. Último acceso: 01/10/2021.

[7] J. Kurose and K. Ross. *Computer Networking: A Top-Down Approach, eBook, Global Edition*. Pearson Education, 2018.

[8] W. Stallings and T. Case. *Business Data Communications: Infrastructure, Networking and Security*. Allways learning. Pearson, 2013.

# Chapter 9

# Seguridad de los datos. Tecnologías y riesgos para la seguridad en Big Data

*Ramon Martí*

## 9.1 Introducción

Tal como ya se ha visto anteriormente, Big Data (macrodatos) [7, 12] hace referencia a conjuntos de datos tan grandes y complejos que precisan de aplicaciones informáticas no tradicionales de procesamiento de datos para tratarlos adecuadamente.

Una vez vistos los Conceptos básicos de seguridad y cifrado, este capítulo se centrará en, primero ver los mecanismos de control de acceso a los datos, para posteriormente describir las amenazas y riesgos de seguridad y privacidad de Big Data, con sus 10 principales retos y algunas recomendaciones generales.

## 9.2 Control de acceso

Al ser los datos de Salud una información privada, disponer de mecanismos para controlar el acceso a los mismos es un aspecto fundamental. La función primordial del "control de acceso" es controlar los 3 siguientes aspectos (Figura 9.1):

- qué sujeto (activo) tiene acceso
- a qué objeto (pasivo) (archivos, directorios, hardware, etc.)

- y qué operación de acceso (crear, leer, modificar, borrar, ejecutar, etc.) se le permite ejecutar.


Figure 9.1: Aspectos del control de acceso

Dicha funcionalidad se puede proporcionar de distintas maneras, y nos centraremos en el mecanismo que se conoce como Control de acceso discrecional (*Discretionary Access Control*, DAC) y sus características, aunque también se describirán muy brevemente el Control de acceso obligatorio (*Mandatory Access Control*, MAC) y el Control de acceso basado en roles (*Role Based Access Control*, RBAC).

### 9.2.1 Granularidad

Un aspecto a tener en cuenta en el control de acceso a los datos es la granularidad [9], que es el tamaño en que se subdividen los datos.

Para el control de acceso, cada una de estas unidades mínimas corresponderá a un objeto sobre el cual podremos controlar el acceso. En el caso de información médica estos objetos podrían ser, p. ej. Grupos de documentos o informes, documentos o informes, partes de documentos o informes o incluso campos de documentos o informes.

Una granularidad más fina (tamaño más pequeño) ofrece ventajas en cuanto a la flexibilidad del control de acceso o procesamiento de datos al tratar cada campo de datos de forma aislada si es necesario. Sin embargo, conlleva una sobrecarga en la entrada, almacenamiento y control de acceso de dichos datos. Un problema de rendimiento causado por una granularidad excesiva puede no revelarse hasta que la escalabilidad se convierta en un problema.

### 9.2.2 Control de acceso discrecional (*Discretionary Access Control*, DAC)

El Control de acceso discrecional (*Discretionary Access Control*, DAC) se denomina discrecional porque se basa en la discrecionalidad del propietario. Es decir, es el propietario del objeto quien especifica las normas de acceso sobre el mismo.

El acceso a los objetos de datos (archivos, directorios, etc.) se permite en función de la identidad de los usuarios, y se definen reglas de acceso explícitas que establecen quién puede, o no, ejecutar qué acciones sobre qué recursos.

En el Control de acceso discrecional se puede dar a los usuarios la capacidad de transmitir sus privilegios a otros usuarios, donde la concesión y revocación de privilegios está regulada por una política administrativa. Esto hace que sea flexible en cuanto a la especificación de políticas, y que sea ampliamente implementado en las plataformas multiusuario estándar (Unix, NT, Novell, etc.).

#### Control de acceso obligatorio (*Mandatory Access Control*, MAC)

Además del DAC, también tenemos otros mecanismos como el Control de acceso obligatorio (*Mandatory Access Control*, MAC).

En el Control de acceso obligatorio, es el sistema (y no los usuarios) quien especifica qué sujetos pueden acceder a qué objetos de datos. El modelo MAC suele utilizarse en entornos en los que la confidencialidad es de suma importancia, como instituciones militares. Ejemplos de sistemas comerciales basados en MAC son SELinux (*Security-Enhanced Linux*) y Trusted Solaris.

### 9.2.3 Matriz de control de acceso (*Access control Matrix*, ACM)

Para la implementación del Control de acceso discrecional, se utiliza la Matriz de control de acceso. Este modelo abstracto describe con precisión el estado de protección a través de una matriz que incluye los derechos de los sujetos sobre los objetos.

### 9.2.4 Modelo de la matriz de control de acceso (*Access Control Matrix Model*)

La Matriz de control de acceso describe el estado de protección de un sistema, identificando los objetos, sujetos y acciones.

El estado del sistema se define mediante 3 parámetros, una tripleta, (S, O, A) (Figura 9.2). S es el conjunto de sujetos, que son la entidad que solicita un servicio y que tanto puede ser un usuario ("persona") como un proceso ("programa"). O es el conjunto de objetos a los que se puede acceder. Asimismo, R (de `Right`) son los derechos de acceso. Finalmente, A es la matriz de acceso, donde cada uno de sus elementos (celdas, A[si, oj]) indican los derechos de acceso que cada uno de los sujetos tienen sobre cada uno de los objetos.

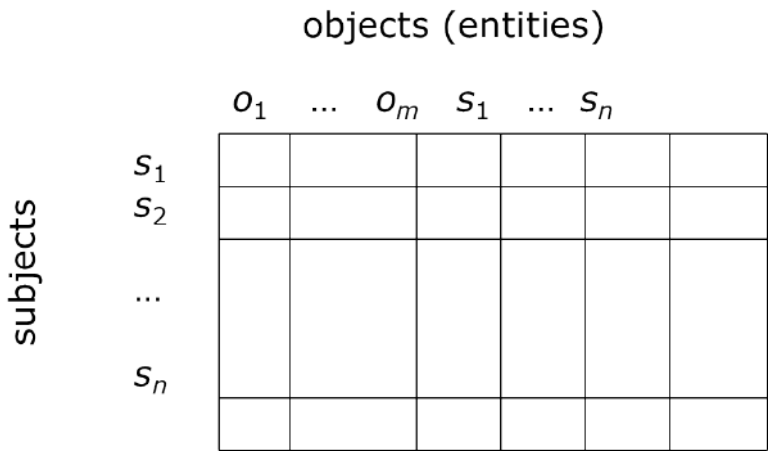



Figure 9.2: Matriz de control de acceso

Por ejemplo, la entrada A[s, o] de la matriz de control de acceso son los privilegios de acceso del sujeto s sobre el objeto o.

- Sujetos S = s1,...,sn
- Objetos O = o1,...,om
- Derechos R = r1,...,rk
- Entradas A[si, oj] $\subseteq$ R
- A[si, oj] = rx, ..., ry significa que el sujeto si tiene los derechos rx, ..., ry sobre el objeto oj

Por ejemplo, en la Figura 9.3, Ann (el sujeto) sobre el fichero File 1 (*subject*), dice (entrada A[Ann, File1]) que es la dueña (*own*) y tiene los permisos de lectura (*read*) y escritura (*write*), sobre el fichero 2 (entrada A[Ann, File2]) tiene los derechos (permisos) de lectura y escritura, y sobre el programa Program 1 (entrada A[Ann, Program1]) tienen el derecho de ejecución(*execute*). Sobre el fichero File 3 (entrada A[Ann, File3]) no tiene ningún derecho de acceso.

- Conceder permisos: Insertar valores en las entradas de la matriz
- Revocar permisos: Eliminar valores de las entradas de la matriz
- Comprobar permisos: Verificar si la entrada relacionada con un sujeto s y un objeto o contiene un modo de acceso determinado

| | File 1 | File 2 | File 3 | Program 1 |
|---|---|---|---|---|
| Ann | own<br>read<br>write | read<br>write | | execute |
| Bob | read | | read<br>write | |
| Carl | | read | | execute<br>read |

Figure 9.3: Ejemplo de Matriz de control de acceso

### 9.2.5 Implementación de la matriz de control de acceso

El Control de acceso discrecional es un modelo abstracto y que sobre papel proporciona una visión clara de los derechos de acceso. De todas formas, con el fin de reducir el tamaño de la información, la Matriz de control de acceso se almacena "eliminando" las celdas de la tabla en las cuales ningún sujeto no tiene ningún derecho sobre un objeto (es decir, eliminando las celdas vacías). Con este fin, la Matriz de control de acceso se puede implementar principalmente de tres maneras: Tabla de autorizaciones, Lista de control de acceso (por columnas) y Lista de capacidades (por filas).

#### Tabla de autorizaciones

En la Tabla de autorizaciones (Figura 9.4), la información se reduce a 3 columnas (sujetos, acciones, objetos), con 1 fila para cada acción permitida a un usuario para un objeto. Los objetos sobre los cuales un usuario no tiene acciones permitidas no aparecen en la tabla. Este mecanismo es utilizado generalmente en los Sistemas de gestión de bases de datos, SGBD (*Database management system*, DBMS).

> **Qué es Sistema de gestión de bases de datos, SGBD**
>
> Un Sistema de gestión de bases de datos, SGBD es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos.
>
> Habitualmente la información se encuentra almacenada en tablas relacionadas entre ellas, y luego este sistema de gestión se conoce como Sistema de gestión de bases de datos relacionales, SGBDR (*Relational database management system*, RDBMS).

| User | Access mode | Object |
|---|---|---|
| Ann | own | File 1 |
| Ann | read | File 1 |
| Ann | write | File 1 |
| Ann | read | File 2 |
| Ann | write | File 2 |
| Ann | execute | Program 1 |
| Bob | read | File 1 |
| Bob | read | File 3 |
| Bob | write | File 3 |
| Carl | read | File 2 |
| Carl | execute | Program 1 |
| Carl | read | Program 1 |

Figure 9.4: Tabla de autorización

### Lista de control de acceso (*Access Control List*, ACL)

En la Lista de control de acceso (*Access Control List*, ACL) (Figura 9.5) la matriz se almacena por columnas, habiendo 1 lista para cada objeto. En cada una de estas listas hay 1 entrada para cada sujeto que tiene algún permiso de acceso al objeto, donde se indica las acciones que este puede ejercer sobre el objeto.

### Lista de capacidades (*Capability List*)

Finalmente, en la Lista de capacidades (*Capability List*)(Figura 9.6) la matriz se almacena por filas (sujetos), y se implementa con una lista para cada sujeto. En cada una de estas listas hay 1 entrada para cada objeto al cual el usuario tiene algún derecho de acceso, donde se indica el acceso que el usuario está autorizado a ejercer sobre dicho objeto.

### ACLs vs Lista de Capacidades

Tanto ACL como las Listas de capacidades proporcionan control de acceso, pero con ACL es más inmediato comprobar la posesión de autorizaciones sobre un objeto, pero ¿qué sucede si se quiere comprobar sobre un sujeto?. Mientras que con las listas de capacidades es más inmediato determinar los privilegios de un sujeto, pero ¿si se quiere comprobar los de un sujeto?.

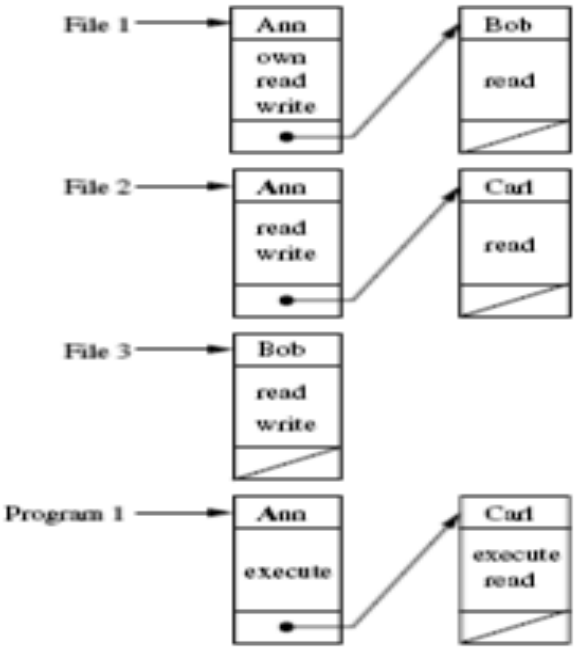



Figure 9.5: Lista de control de acceso

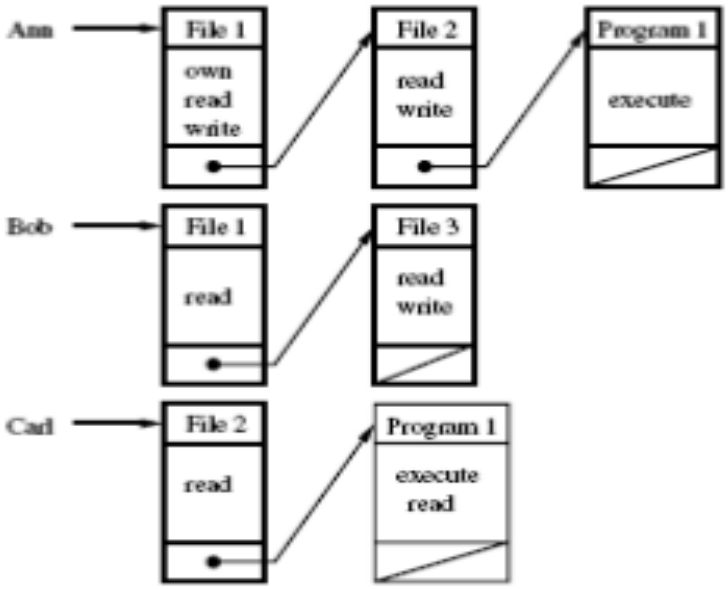



Figure 9.6: Lista de capacidades

### 9.2.6 Vulnerabilidades de las políticas discrecionales

Para finalizar, cabe mencionar que una de las vulnerabilidades de las políticas discrecionales es el hecho que, tal como se ve en la tabla de accesos, no hay distinción entre sujetos (entidad que solicita el acceso a un objeto) y usuarios ("sujeto humano"), cuando en la realidad el sujeto puede ser un usuario, pero también un proceso ("programa").

Otra vulnerabilidad es que todos estos mecanismos solo controlan los permisos para el acceso a la información, pero no hay control sobre el flujo e intercambio de dicha información.

Finalmente, otra vulnerabilidad tiene relación con los programas maliciosos, p. ej. un troyano, ya que estos obtienen los permisos del sujeto que lo ha ejecutado.

### 9.2.7 Características adicionales del DAC

La flexibilidad del DAC se ve reforzada por el hecho que admite la implmentación de distintas visiones de los permisos (complementarias entre ellas):

- Positivo o negativo
- Fuertes o débiles
- Implícitos o explícitos
- Basado en el contenido

#### Permisos positivos y negativos

Los sistemas se pueden basar en permisos positivos y en permisos negativos.

En los permisos positivos, la política predeterminada por defecto es que los sujetos no tienen acceso a los objetos, y el permiso les da acceso los mismos.

Por contra, en los permisos negativos, por defecto los sujetos tienen acceso a los objetos, y el permiso es para denegarles el acceso.

Ambos enfoques son útiles para especificar excepciones a una política determinada y para imponer un control más estricto sobre determinados elementos de datos cruciales

De todas formas, este tipo de permisos puede llevar a conflictos de autorización, donde se deberá tomar una decisión sobre su solución:

- No hay conflictos

- Los permisos negativos tienen prioridad
- Los permisos positivos tienen prioridad
- Ninguno de los dos tiene prioridad
- Los permisos más específicos tienen prioridad

#### Permisos fuertes y débiles

La siguiente clasificación de permisos son los fuertes y los débiles. Los permisos fuertes no se pueden sobrescribir, mientras que los débiles pueden ser sobrescritos por otros permisos fuertes y débiles.

#### Permisos implícitos y explícitos

Algunos modelos admiten permisos implícitos (o heredados). Estos permisos son otorgados a un objeto por ser hijo de un objeto padre. Los permisos implícitos pueden derivarse, ya sea por un conjunto de reglas de propagación que explotan las jerarquías de sujetos, objetos y privilegios, o mediante un conjunto de reglas de derivación definidas por el usuario

Por otra parte, los permisos explícitos son establecidos por defecto cuando se crea el objeto, o por acción del usuario y no son heredados.

#### Permisos basados en el contenido

Finalmente tenemos el control de acceso basado en el contenido, que condicionan el acceso a un objeto determinado en función de su contenido. Este tipo de permisos son principalmente relevantes para los sistemas de bases de datos.

Por ejemplo, en un RDBMS que soporte el control de acceso basado en el contenido es posible autorizar a un sujeto a acceder a la información solo de aquellos empleados cuyo salario no sea superior a 30K €.

### 9.2.8 Control de acceso basado en roles (*Role Based Access Control*, RBAC)

Tal como su nombre indica, el Control de acceso basado en roles (*Role Based Access Control*, RBAC) es un mecanismo de control de acceso de los usuarios a la información (o a equipos) que se basa en los "roles que los usuarios individuales dentro de la organización". Su idea principal es controlar y mantener de forma centralizada los derechos de acceso que reflejen las directrices de protección de la organización.

Este mecanismo es útil porque la combinación de usuarios y permisos tiende a cambiar con el tiempo, pero los permisos asociados a un rol son más estables. Así, el acceso depende del rol/función, no de la identidad. Primero se predefinen las relaciones rol-permiso y luego es sencillo asignar usuarios a los roles predefinidos.

Ejemplo: Alice es contable y tiene acceso a los datos financieros. Si se va y contratan a Betty como nueva contable, ahora es Betty quien tiene ahora acceso a esos datos. La función de "contable" es la que dicta el acceso, no la identidad individual.

RBAC está diseñado para la separación de funciones permitiendo definir los roles que necesitan los usuarios para el acceso a los objetos.

Al igual que en el caso de sujetos individuales, la cuestión clave es si los roles los asigna una autoridad central (en cuyo caso RBAC es una forma de MAC); o si son los propios usuarios los que definen los roles que tienen acceso a sus objetos (igual que en DAC), lo que puede llevar a múltiples roles por objeto y absolutamente ninguna semántica en los roles.

#### Ventajas de RBAC

El Control de acceso basado en roles tiene muchas ventajas a destacar:

- Permite una gestión eficaz de la seguridad, ya que los roles son administrativos, que ya disponen de una jerarquía de roles.
- El principio del mínimo privilegio permite minimizar los daños.
- Dispone de restricciones en la separación de deberes para evitar el fraude.
- Permite la agrupación de objetos.
- Es neutral a las políticas de control de acceso (como tal, no define políticas concretas), lo que proporciona generalidad. Esto permite soportar tanto políticas de Control de acceso obligatorio (*Mandatory Access Control*, MAC) como las de Control de acceso discrecional (*Discretionary Access Control*,DAC).

## 9.3 Amenazas de seguridad en Big Data

Una vez vistos los mecanismos de control de acceso, ya podemos continuar con la seguridad de Big Data. Por su naturaleza, se puede ver que existen muchos aspectos donde puede existir un riesgo de seguridad:

- Gran cantidad de datos: Dificultad de su manejo y gestión
- Muchas fuentes: No todas las fuentes tienen por qué ser fiables
- Información personal: La información a menudo no es anónima (y se ha de proteger)
- Comunicaciones de datos: Los canales no siempre son fiables
- Muchos ordenadores almacenando y accediendo a los datos: Algún ordenador podría ser malicioso
- Software accediendo y analizando datos: Algún software también podría ser malicioso

## 9.4 Protección en Big Data

A partir de la descripción anterior, se ve que la seguridad de Big Data conlleva sus propios retos, además de ser un objetivo de alto valor. No es que la seguridad de Big Data sea fundamentalmente diferente de la seguridad de los datos tradicionales, pero sí tiene sus particularidades, que incluyen:

- Los **datos** recogidos, agregados y analizados para el análisis de Big Data.
- La **infraestructura** utilizada para almacenar y albergar los datos Big Data.
- Las **tecnologías** aplicadas para analizar los datos estructurados y no estructurados de Big Data.

## 9.5 Importancia de la seguridad en Big Data

La inclusión de seguridad en Big Data es importante por diversos motivos.

El primero es evitar el acceso accidental con el fin que los usuarios no se perjudiquen a sí mismos. Esta no es toda la seguridad necesaria, sino solo un primer paso fundamental que no requiere una autenticación fuerte.

El segundo motivo es detener a los usuarios malintencionados. Con este fin, la seguridad sí que tiene que ser real, ya que el código de análisis de los datos puede ser malicioso. Cabe tener en cuenta que la confianza implícita no evita la amenaza interna.

Finalmente, el tercer motivo, es el hecho que Big Data accede y combina datos de distintas bases de datos requiere una confianza en los datos en un entorno multiusuario.

## 9.6 CSA [ampliado]: Los diez principales retos de seguridad y privacidad de Big Data

En el documento [8], la *Cloud Security Alliance*, CSA, define *Los diez principales retos de seguridad y privacidad de Big Data*, numerados del 1 al 10, agrupados en 4 grandes bloques, y que son los que se verán en este apartado (Figura 9.7): Seguridad de la infraestructura, Privacidad de los datos, Gestión de datos e Integridad y seguridad reactiva.

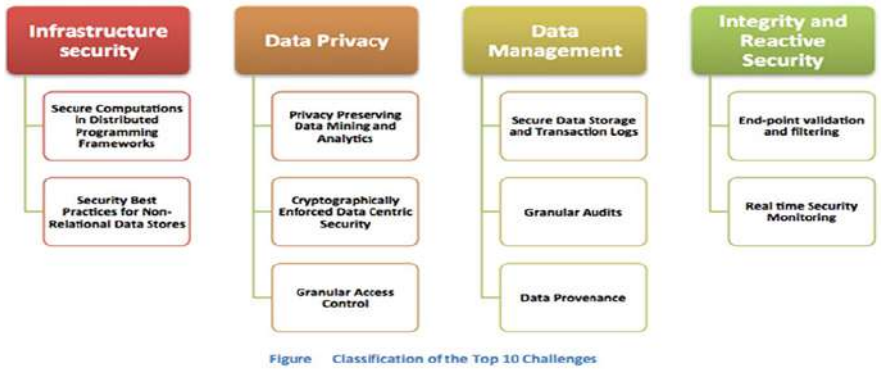



Figure 9.7: Clasificación de la CSA de los diez principales retos de seguridad y privacidad de Big Data

A continuación se describen brevemente los 4 grandes bloques, con los 10 retos.

### 9.6.1 Seguridad de las infraestructuras

El primer bloque es la Seguridad en las infraestructuras. En lo referente a este tema, se definen 2 puntos: 1. Cálculos seguros en marcos de programación distribuidos; y 2. Mejores prácticas de seguridad para almacenes de datos no relacionales.

**1. Cálculos seguros en marcos de programación distribuida**

Para el análisis de Big Data es necesario utilizar la programación y almacenaje distribuidos, es decir con muchos ordenadores conectados entre ellos, y accediendo cada uno a solo a parte de los datos o a todos ellos.

En este caso, y con el fin de evitar ataques, es necesario tanto asegurarse que los programas que analizan los datos no son maliciosos, como añadir seguridad a los mismos datos para que no puedan ser accedidos por programas maliciosos.

**2. Mejores prácticas de seguridad para los almacenes de datos no relacionales**

Big Data utiliza Bases de datos no relacionales, NoSQL, muy distintas de las Bases de datos relacionales que son las que se utilizan habitualmente en otros entornos. El problema de estas Bases de datos no relacionales es que no se diseñaron con la seguridad en mente, con lo que las soluciones de seguridad aún no son muy maduras.

Como las bases de datos NoSQL no proporcionan ningún soporte de seguridad explícito en la base de datos, una opción por la que optan los desarrolladores suele ser integrar la seguridad en el *middleware* (el software entre los Programas y la base de datos en sí). Sin embargo, los aspectos de agrupación (*clustering*, varios equipos con parte de los datos del usuario trabajando al unísono como un solo sistema) de las bases de datos NoSQL plantean retos adicionales para la solidez de dichas prácticas de seguridad.

### 9.6.2 Privacidad de los datos

El segundo bloque es la Privacidad de los datos. En este caso los puntos relacionados son: 6. Minería y análisis de datos escalables y componibles que preservan la privacidad; 7. Seguridad centrada en los datos reforzada criptográficamente; y 8. Control de acceso granular.

**6. Minería y análisis de datos escalables y componibles que preservan la privacidad**

El Big Data puede potencialmente permitir la invasión de la privacidad, el marketing invasivo, la disminución de las libertades civiles y el aumento del control estatal y empresarial.

La anonimización de los datos para el análisis no suele ser suficiente para mantener la privacidad del usuario. Por este motivo es importante establecer

directrices y recomendaciones para evitar la divulgación involuntaria de la privacidad.

**7. Seguridad centrada en los datos reforzada criptográficamente**

Tal como ya se ha comentado, hay 2 enfoques para controlar el acceso, y por tanto la visibilidad, de los datos a los distintos sujetos (usuarios). El primero es limitar el acceso físico de dichos usuarios al sistema, mientras que el segundo es cifrar los datos de manera que solo quien disponga de clave pueda acceder a ellos. Ambos enfoques tienen ventajas y desventajas.

Históricamente, el primero es más sencillo de implementar y, combinado con la comunicación protegida por criptografía, es el estándar para la mayoría de las infraestructuras informáticas y de comunicación. De todas formas, expone una superficie de ataque mucho mayor, ya que existen muchas ataques centrados en eludir el control de acceso y acceder directamente a los datos.

La protección de los datos mediante el cifrado tiene una superficie de ataque más pequeña y mejor definida. Aunque es posible realizar ataques para extraer las claves secretas, estos ataques son mucho más difíciles de ejecutar.

**8. Control de acceso granular**

La propiedad de seguridad que importa desde el punto de vista del control de acceso es el secreto: evitar el acceso a los datos por parte de personas que no deberían tener acceso.

El control de acceso granular regula los permisos de acceso a los usuarios no sobre la totalidad de la información, sino solo sobre partes de la misma. Ya se ve que este control de acceso ofrece a los gestores de los datos más precisión a la hora de compartir datos sin comprometer el secreto, pudiéndose aplicar distintas políticas para cada una de esas partes.

Cabe destacar que con los mecanismos de acceso de grano grueso (partes de tamaño grande), para garantizar una seguridad sólida los datos se deben pasar a una categoría más restrictiva.

### 9.6.3 Gestión de datos

Como tercer bloque hay la Gestión de datos, para la cual se definen 3 puntos: 3. Almacenamiento seguro de datos y registros de transacciones; 9. Auditorías granulares; y 10. Procedencia de los datos.

### 3. Almacenamiento seguro de datos y registros de transacciones

En Big Data, normalmente los datos y los registros de transacciones se almacenan en medios de almacenamiento de varios niveles. En estos sistemas se utiliza distintos tipos de medios de almacenamiento para crear múltiples niveles para almacenar diferentes tipos de datos, para así reducir los costes totales de almacenamiento y mejorar el rendimiento y la disponibilidad. El movimiento manual de los datos entre niveles ofrece al responsable de TI un control directo sobre qué datos se mueven y cuándo. Sin embargo, a medida que el tamaño del conjunto de datos crece de forma exponencial, caso de Big Data, la escalabilidad y la disponibilidad hacen necesaria la nivelación automática (*auto-tiering*) para la gestión del almacenamiento.

Las soluciones de nivelación automática no hacen un seguimiento de dónde están almacenados exactamente los datos, lo que plantea nuevos retos para la seguridad del almacenamiento de dichos datos. Por ese motivo son imprescindibles nuevos mecanismos para frustrar el acceso no autorizado y mantener una disponibilidad constante.

### 9. Auditorías granulares

Con la supervisión de la seguridad en tiempo real (véase posteriormente el punto 5), el objetivo es la detección y notificación en el momento en que se produce un ataque. Pero en realidad, no siempre será así (p. ej., con nuevos ataques, o con ataques reales no detectados).

La auditoría no es algo nuevo, pero el alcance y la granularidad pueden ser diferentes en contextos de seguridad en tiempo real como en el caso de Big Data. Por ejemplo, en este contexto hay más objetos de datos, que probablemente (pero no necesariamente) están distribuidos.

Para descubrir un ataque no detectado, es necesario disponer de información de auditoría, con lo que dicha información es crucial para entender lo que ocurrió y lo que salió mal. También es necesaria para el cumplimiento de requisitos, la regulación y la investigación forense.

### 10. Procedencia de los datos

En Big Data, los metadatos con el origen y procedencia (cronología de la propiedad, custodia o ubicación) de los datos crecerán en complejidad debido a grandes gráficos de procedencia generados por entornos de programación de aplicaciones Big Data que lo soporten.

El análisis de estos grandes gráficos de procedencia para detectar dependencias de metadatos para aplicaciones de seguridad y/o confidencialidad, aunque es computacionalmente intensivo, será necesario.

Para la recopilación segura de la procedencia, debe integrarse una técnica de autenticación rápida y ligera en la infraestructura ya existente.

### 9.6.4 Integridad y seguridad reactiva

Finalmente nos centramos en la Integridad y seguridad reactiva, que es la encargada de responder a las amenazas pasadas y presentes, en lugar de anticiparse a los peligros futuros (que sería tarea de la seguridad proactiva). En este bloque se especifican 2 puntos: 4. Validación/filtrado de entradas en el punto final; y 5. Supervisión de la seguridad en tiempo real.

#### 4. Validación/filtrado de entrada de punto final

Muchos usos de Big Data en entornos empresariales requieren la recopilación de datos de diversas fuentes, incluidos los dispositivos finales. Por ejemplo, los sistemas de gestión de eventos e información de seguridad (*security information and event management system*, SIEM) pueden recoger registros de eventos de millones de dispositivos *hardware* y de aplicaciones *software* en una red empresarial.

Un reto clave en el proceso de recopilación de datos es la validación de las entradas. ¿Cómo podemos confiar en los datos? ¿Cómo podemos validar que una fuente de datos de entrada no es maliciosa? ¿Y cómo podemos filtrar las entradas maliciosas de nuestra colección?

La validación y el filtrado de las entradas es un reto de enormes proporciones que plantean las fuentes de entrada no fiables, especialmente con el modelo de "traiga su propio dispositivo" (*bring-your-own-device*, BYOD), donde los empleados llevan sus propios dispositivos personales (portátiles, tabletas, móviles...) a su lugar de trabajo para tener acceso a recursos de la empresa.

#### 5. Supervisión de la seguridad en tiempo real

Big Data y la seguridad no solo se cruzan en la protección de las infraestructuras de Big Data, sino también en el aprovechamiento del mismo análisis de Big Data para ayudar a mejorar la seguridad de otros sistemas.

Uno de los problemas más complejos de la analítica de Big Data es la supervisión de la seguridad en tiempo real, que consta de dos aspectos principales:

(a) La supervisión de la propia infraestructura de Big Data. Un ejemplo es la supervisión del rendimiento y la salud de todos los nodos que componen la infraestructura de Big Data.

(b) La utilización de la misma infraestructura para el análisis de datos con la finalidad de la supervisión de la seguridad. Un ejemplo sería que un proveedor de servicios sanitarios utilizara herramientas de monitorización para buscar reclamaciones fraudulentas o que un proveedor de servicios en la nube utilizara herramientas de Big Data similares para obtener una mejor monitorización de las alertas y del cumplimiento de la normativa en tiempo real.

Estas mejoras podrían proporcionar una reducción del número de falsos positivos y/o un aumento de la calidad de los verdaderos positivos. La supervisión de la seguridad en tiempo real es un reto debido al número de alertas generadas por los dispositivos de seguridad. Estas alertas (correlacionadas o no) conducen a un número masivo de falsos positivos, que a menudo se ignoran debido a la limitada capacidad humana de análisis. Este problema podría incluso aumentar con el Big Data, dado el volumen y la velocidad de los flujos de datos.

Sin embargo, las tecnologías de Big Data pueden ofrecer la oportunidad de procesar y analizar rápidamente diferentes tipos de datos. Estas tecnologías pueden utilizarse para proporcionar, por ejemplo, una detección de anomalías en tiempo real basada en un análisis de seguridad escalable.

### 9.6.5 Recomendaciones generales de seguridad para el Big Data

Una vez descritos los diez principales retos de seguridad y privacidad, finalmente, destacamos algunas recomendaciones generales a seguir por lo que hace referencia al Big Data [5, 2, 14].

*Examinar/Comprobar los proveedores de la nube*: Si se va a almacenar la información de Big Data en la nube, se debe asegurar que el proveedor cuenta con los mecanismos de protección adecuados. Es necesario asegurarse de que el proveedor realiza auditorías de seguridad periódicas y se deben acordar sanciones en caso de que no se cumplan los estándares de seguridad adecuados.

*Crear una política de control de acceso adecuada*: Hace falta crear políticas que permitan el acceso solo a los usuarios autorizados.

*Proteger los datos*: Tanto los datos brutos como el resultado de la analítica deben estar adecuadamente protegidos. Para tal objetivo se debe utilizar el cifrado correspondiente para garantizar que no se filtren datos sensibles.

*Proteger las comunicaciones*: Los datos en tránsito deben estar adecuadamente protegidos para garantizar su confidencialidad e integridad.

Utilizar la supervisión de la seguridad en tiempo real: El acceso a los datos debe ser supervisado. Con el fin de evitar el acceso no autorizado a los datos, debe utilizarse la inteligencia sobre amenazas.

*Anonimizar los datos*: La anonimización de los datos también es importante para garantizar que se abordan los problemas de privacidad y hay que asegurarse de que toda la información sensible se elimina del conjunto de registros recogidos.

*Registrar todo*: Es la única manera de detectar de forma fiable las actividades no autorizadas.

*Utilizar un sistema de Gestión de Eventos e Información de Seguridad (Security information and event management, SIEM)*: El uso de un sistema SIEM permitirá dar sentido a los datos de registro (log), ya que este tipo de análisis no es factible realizarlos manualmente.

*Evaluar los riesgos sobre los datos que se recopilan*: Las organizaciones deberían realizar una evaluación de riesgos sobre los datos que están recopilando. Deben considerar si están recopilando información de los clientes que debe mantenerse privada y establecer políticas adecuadas que protejan los datos y el derecho a la privacidad de sus clientes.

*Considerar la compartición de los datos*: Si los datos se comparten con otras organizaciones, debe considerarse cómo se hace. La divulgación deliberada de datos que resulte ser una violación de la privacidad puede tener un gran impacto en una organización desde el punto de vista de la reputación y la economía.

*Considerar las leyes regionales*: Finalmente, las organizaciones también deben considerar cuidadosamente las leyes regionales sobre el manejo de los datos de los clientes, como la Directiva de Datos de la UE.

## 9.7 Resumen y conclusiones

En este capítulo, primero nos hemos centrado en el Control de acceso, que nos ha permitido conocer brevemente los mecanismos que permiten controlar los permisos que tienen los usuarios sobre los datos.

Finalmente, hemos continuado con las Tecnologías y riesgos para la seguridad en Big Data y se ha visto que es un aspecto muy importante debido a la gran cantidad de datos, de diversas fuentes, a menudo con información personal, con distintos software y ordenadores almacenando y accediendo, a menudo de forma remota, a todos estos datos. Con esta visión del problema, hemos descrito los diez principales retos de seguridad y privacidad, y finalmente se han dado unas breves recomendaciones generales de seguridad.

# Bibliografía


[1] Access control. http://www.utc.edu/center-information-security-assurance/course-listing/4670-lecture6-dac-rbac.ppt. Último acceso: 01/10/2021.

[2] Big data brings big security problems. https://web.archive.org/web/20161009232519/http://www.informationweek.com/big-data/big-data-analytics/big-data-brings-big-security-problems/d/d-id/1252747. Último acceso: 2016-10-09.

[3] Big data: Issues and challenges. http://www.slideshare.net/HarshMishra3/harsh-big-data-seminar-report. Último acceso: 01/10/2021.

[4] Big data security. http://www.slideshare.net/JoeyEcheverria/big-data-security. Último acceso: 01/10/2021.

[5] Big data security - challenges & solutions. https://www.f-secure.com/en/consulting/our-thinking/big-data-security-challenges-and-solutions/. Último acceso: 01/10/2021.

[6] Big data security (fowler). http://sector.ca/portals/17/Presentations13/BigDataSecurity-FOWLER.pdf. Último acceso: 01/10/2021.

[7] Big data (wikipedia). https://en.wikipedia.org/wiki/Big_data. Último acceso: 01/10/2021.

[8] CSA releases the expanded top ten big data security & privacy challenges. https://cloudsecurityalliance.org/media/news/csa-releases-the-expanded-top-ten-big-data-security-privacy-challenges/. Último acceso: 01/10/2021.

[9] Granularity. https://en.wikipedia.org/wiki/Granularity. Último acceso: 01/10/2021.

[10] Granularity. https://security.stackexchange.com/questions/63518/mac-vs-dac-vs-rbac. Último acceso: 01/10/2021.

[11] How to manage big data's big security challenges. http://data-informed.com/manage-big-datas-big-security-challenges/. Último acceso: 01/10/2021.

[12] Macrodatos (wikipedia). https://es.wikipedia.org/wiki/Macrodatos. Último acceso: 01/10/2021.

[13] Top ten big data security and privacy challenges. http://www.isaca.org/Groups/Professional-English/big-data/GroupDocuments/Big_Data_Top_Ten_v1.pdf. Último acceso: 01/10/2021.

[14] Shaobing Wu and Changmei Wang. Big data security framework based on encryption. In Xingming Sun, Zhaoqing Pan, and Elisa Bertino, editors, *Cloud Computing and Security*, pages 528–540, Cham, 2018. Springer International Publishing.