

# C2\_TP1

## 1) Lectura de señales biomédicas

Se desea analizar una señal de pulsioxímetro que devuelve un `uint16_t` codificado de la siguiente forma:

- **8 bits más significativos** → frecuencia cardíaca (en bpm)
- **8 bits menos significativos** → porcentaje de saturación de oxígeno (SpO<sub>2</sub>)

Escriba una función `void mostrar_lectura(uint16_t dato)` que imprima:

```
Frecuencia: XX bpm | SpO2: YY%
```

## 2) Datos de sensores integrados

En un sistema biomédico se usan estructuras para almacenar datos provenientes de múltiples sensores. Se desean guardar, en una misma estructura:

- ID (nro entero)
- Temperatura
- Presión
- Humedad

1. Declare un array de `sensorData` para 5 sensores.
2. Escriba una función `void cargar_sensor(sensorData arr[], int n)` que solicite por teclado los valores del sensor.
3. Escriba una función `void mostrar_promedios(sensorData arr[], int n)` que calcule y muestre los promedios de **temperatura, presión y humedad** de todos los sensores cargados.

### 3) Composición de vectores en señales electromagnéticas

En telecomunicaciones, las componentes ortogonales de un vector pueden combinarse para obtener una señal resultante.

1. Escriba una función que reciba dos vectores (cada uno con componentes `x` e `y`, tipo `float`) y devuelva un tercer vector que sea la **suma vectorial**.
  2. Escriba otra función que calcule el **módulo** del vector resultante.
  3. Imprima el vector final y su módulo.
- 

### 4) Timestamp de eventos biomédicos

Se almacenan eventos con timestamps de 3 campos: hora, minutos y segundos.

1. Escriba una función que reciba dos timestamps (struct con `hh`, `mm`, `ss`) y imprima la **diferencia entre ambos** en formato `"hh:mm:ss"`.
  2. Si segundo timestamp no es posterior al primero, imprima "Formato Invalido".
- 

### 5) Registro de paciente monitorizado

Un sistema de monitoreo biomédico almacena, por cada paciente, lo siguiente:

- **ID del paciente**
- **Datos personales:** nombre, edad y sexo
- **Últimas lecturas** de sensores: frecuencia cardíaca, temperatura corporal y presión arterial

```
struct DatosPersonales {  
    char nombre[30];  
    int edad;  
    char sexo;  
};  
  
struct Lectura {  
    float frecuenciaCardiaca;
```

```
float temperatura;  
float presion;  
};  
  
struct Paciente {  
    int id;  
    struct DatosPersonales datos;  
    struct Lectura ultimaLectura;  
};
```

1. Escriba una función que reciba un `Paciente` y lo guarde en un archivo `pacientes.dat`
  2. Escriba una función que lea todos los Pacientes de un archivo e imprima los promedios de la edad, frecuencia cardiaca, temperatura y presión
- 

## 6) Uso eficiente de memoria

Una central de comunicación puede recibir **dos tipos de mensajes**:

- **Texto** (máx. 50 caracteres)
- **Lecturas de sensores**, que incluyen:
  - ID del sensor (int)
  - Valor leído (float)

Se necesita almacenar ambos tipos de datos en la misma estructura, pero **optimizando el uso de memoria**.

a) Defina la estructura adecuada

b) Escriba una función que reciba un `Mensaje` y muestre el contenido según el tipo.