



CS-3.1	Enunciado de Prueba	Año:	2023
Especialidad:	590-107 Informática		

Prueba	Prueba 1A / Primera parte del Ejercicio en caso de Acc 3-4 y 5	Acceso:	Todos
---------------	--	----------------	-------

OPCIÓN A

Ejercicio 1: Dada la clase Nodo, implementar en lenguaje Java o C++, y haciendo uso de programación orientada a objetos, una pila dinámica de Nodos, sin utilizar clases predefinidas (como por ejemplo la clase STACK).

Se debe mantener la filosofía de esta estructura de datos, optimizando el uso de la memoria.

Debe comentarse el código de manera clara.

Para la implementación en Java:

```
public class Nodo {  
  
    private int valor;  
  
    private Nodo siguiente;  
  
    public void Nodo(){  
        this.valor = 0;  
        this.siguiente = null;  
    }  
  
    public int getValor() {  
        return valor;  
    }  
  
    public void setValor(int valor) {  
        this.valor = valor;  
    }  
  
    public Nodo getSiguiente() {  
        return siguiente;  
    }  
  
    public void setSiguiente(Nodo siguiente) {  
        this.siguiente = siguiente;  
    }  
}
```



Para su implementación en C++:

Archivo Nodo.h

```
#include <iostream>
using namespace std;

class Nodo
{
private:
    int valor;
    Nodo* siguiente;
public:
    void Nodo();
    int getValor();
    void setValor (int valor);
    Nodo* getSiguiente();
    void setSiguiente(Nodo* siguiente);
};
```

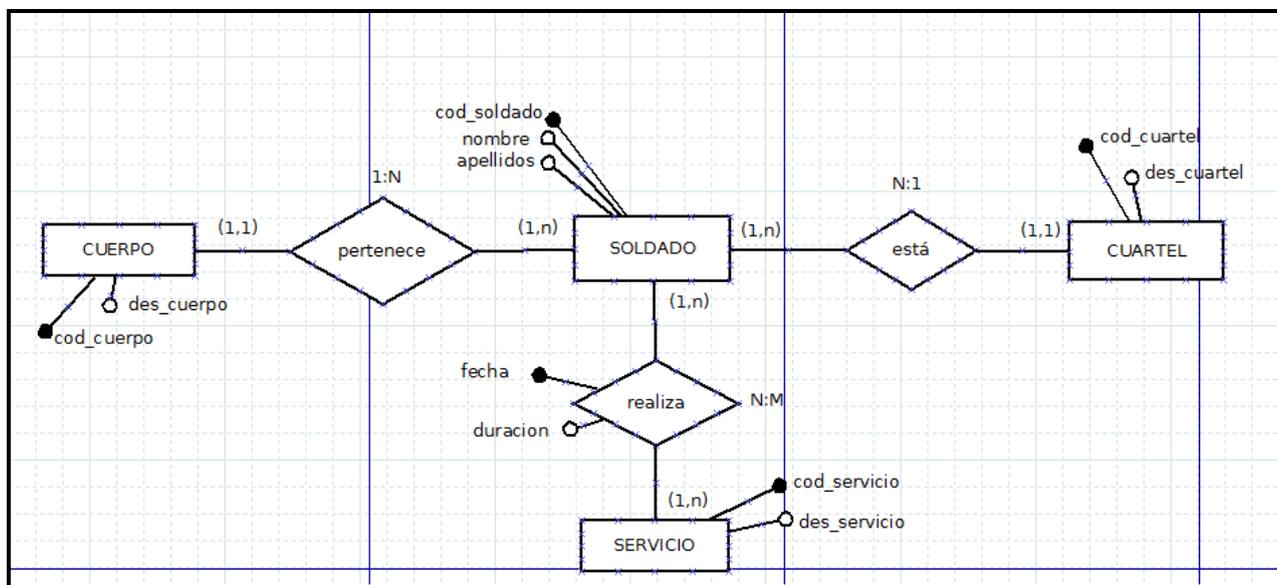
Archivo Nodo.cpp

```
#include "Nodo.h";
void Nodo::Nodo()
{
    this->valor = 0;
    this->siguiente = null;
}
int Nodo::getValor()
{
    return valor;
}
void Nodo::setValor(int valor)
{
    this->valor = valor;
}
Nodo *Nodo::getSiguiente()
{
    return siguiente;
}
void Nodo::setSiguiente(Nodo* siguiente)
{
    this->siguiente = siguiente;
}
```

La pila de Nodos debe contener los siguientes métodos:

- **Constructor de la clase.**
- **void apilar(int valor);** // Introduce un nuevo elemento en la pila
- **void desapilar();** // Elimina el elemento que se encuentra en la cima
- **int cima();** // Devuelve el valor del primer elemento
- **boolean buscar(int referencia);** // Devuelve true si la referencia se encuentra en la pila y false en caso contrario
- **void eliminar(int referencia);** // Borra de la pila el nodo cuyo valor coincida con la referencia que nos pasan, la pila debe quedar en el orden original sin el valor borrado, no se puede alterar el orden de los nodos almacenados.
- **void listar();** // Muestra el valor de los elementos apilados
- **void vaciar();** // Destructor de la clase.

Ejercicio 2: Partiendo del siguiente modelo entidad/relación, correspondiente a una parte muy sencilla de los servicios realizados por los soldados del ejército de un país.



Se pide:

1. Diseñar el modelo lógico relacional derivado del mismo, subrayando las claves primarias de cada tabla e indicando de manera clara a qué clave primaria apunta cada clave ajena.
2. Partiendo del modelo relacional anteriormente resuelto, escribir las sentencias SQL necesarias para dar solución a cada uno de los siguientes apartados, justificando la respuesta dada:
 - a. Suponemos que estamos trabajando con una base de datos MySQL. Los campos *cod_soldado*, *cod_cuarterl* y *cod_cuerpo* son autoincrementales enteros de longitud 11. Realizar las siguientes acciones:
 - i. Crear la tabla correspondiente al soldado, teniendo en cuenta que *nombre* y *apellidos* son cadenas de longitud variable 100 no nulos. Se restringirán actualizaciones y borrados en las claves ajenas.
 - ii. Insertar los siguientes cuarteles.
 - Descripción del cuarterl: Valdemoro.
 - Descripción del cuarterl: Cuatro Caminos.
 - b. Mostrar el número de soldados que hay en cada cuarterl. Los campos a mostrar serán *des_cuarterl* y el número de soldados que hay en él. Ordena ascendente por *des_cuarterl*.



- c. Mostrar los cuerpos en los que no hay ningún soldado. El campo a mostrar será *des_cuerpo*. Ordena descendente por *des_cuerpo*. Se debe usar un subselect para resolver esta consulta.
- d. Mostrar el servicio o servicios que más horas han durado. El campo a mostrar será *des_servicio*. Ordena ascendente por *des_servicio*.
- e. Mostrar las horas totales de servicios que ha realizado cada soldado. Los campos a mostrar serán *apellidos* del soldado, *nombre* del soldado y la suma de las horas totales que ha realizado en todos sus servicios. Ordena ascendente por *apellidos* y *nombre* del soldado.
- f. Mostrar los soldados cuyas horas totales de servicios realizados sean superiores a la media del total de horas realizadas por los soldados que han realizado servicios. Los campos a mostrar serán *apellidos* del soldado, *nombre* del soldado y el total de horas de servicios que ha realizado. Ordena ascendente por *apellidos* y *nombre* del soldado.

Valoración:

- **Ejercicio 1:** Máximo 4 puntos
- **Ejercicio 2:** Máximo 4 puntos.
- Plantea un procedimiento y una progresión correcta en el desarrollo de los ejercicios. Es riguroso y claro en las técnicas utilizadas: Máximo 1 punto
- Limpieza y orden en la presentación de los ejercicios: Máximo 1 punto.



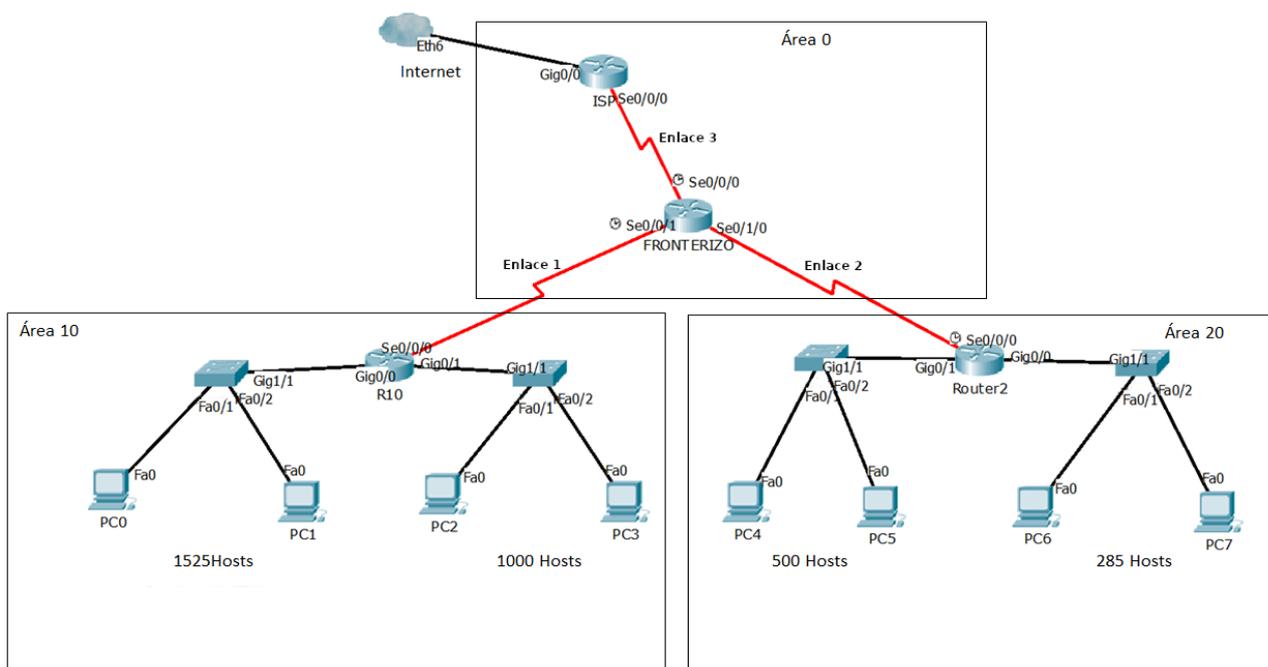
CS-3.1	Enunciado de Prueba	Año:	2023
Especialidad:	590-107 Informática		

Prueba	Prueba 1A / Primera parte del Ejercicio en caso de Acc 3-4 y 5	Acceso:	Todos
---------------	--	----------------	-------

OPCIÓN B

Ejercicio 1:

La empresa OPOSINF S.A dispone de la dirección de red 172.33.0.0/16 para repartir entre sus diferentes sedes y su ISP le proporciona la dirección 201.1.1.0/28. Su esquema de red responde a la siguiente topología (con más detalle en página 3):



Se pide:

1. Calcular el rango de direcciones IP que se debe asignar a cada una de las subredes y los enlaces entre routers, partiendo de la dirección asignada a la empresa y los hosts necesarios en cada subred. Justifica la técnica utilizada.

Para el cálculo se debe hacer uso de técnicas que permitan el menor desperdicio posible de direcciones IPs.

- Subred 1: 1525 hosts
- Subred 2: 1000 hosts
- Subred 3: 500 hosts
- Subred 4: 285 hosts



Para la distribución de IPs se rellenará una tabla con la siguiente información:

Nombre Subred	Dirección Subred	Máscara subred	1ª dirección útil	Última dirección útil	Dirección de broadcast de la subred
....

2. **Escenario 1:** El router FRONTERIZO utiliza **enrutamiento estático**, de tal manera que hay comunicación entre todos los hosts de la empresa y todos ellos tengan salida a Internet a través del ISP. Realiza mediante comandos la configuración de dicho router partiendo del terminal que se muestra en la imagen inferior y justificando la solución.

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)#
Router (config)#
Router (config)#
Router (config)#
```

Copy

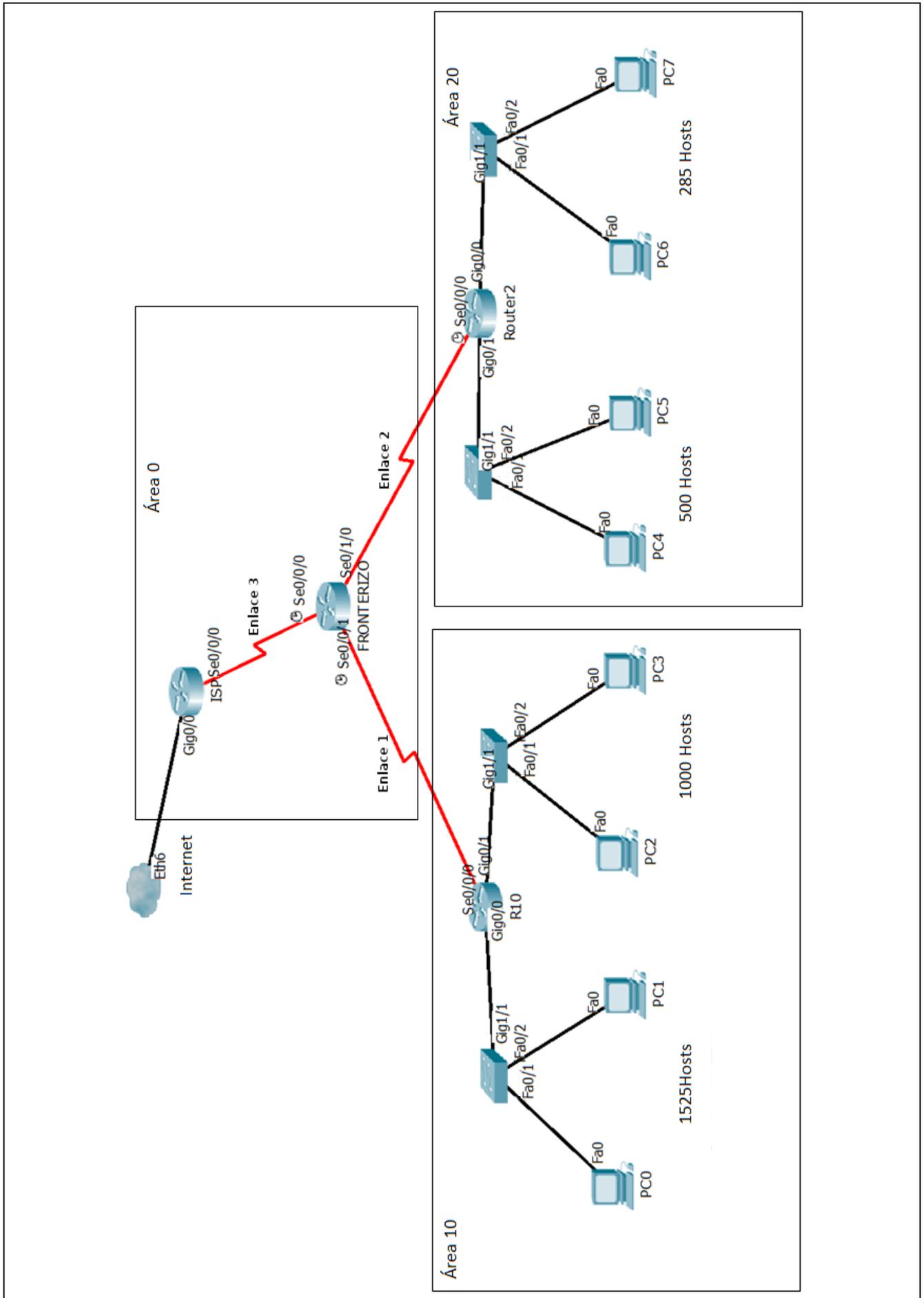
Paste

3. **Escenario 2:** El router R10 y el router FRONTERIZO utilizan el protocolo de **enrutamiento dinámico OSPF**. Realiza mediante comandos la configuración de ambos routers partiendo del terminal que se muestra en la imagen inferior, de tal manera que no exista comunicación entre los hosts que se encuentran en distintas áreas y que todos los equipos salgan a Internet a través el ISP. Justifica la solución.

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)#
Router (config)#
Router (config)#
Router (config)#
```

Copy

Paste



**Ejercicio 2:**

Realizar un programa en JAVA o C que almacene en una estructura de datos estática (matriz bidimensional) los litros de lluvia diarios caídos en la ciudad de Plasencia durante el mes de febrero. No se podrán utilizar las clases predefinidas para el manejo de matrices. Supondremos que el mes tiene 28 días y que el día 1 del mes es un lunes, por lo que el mes estará formado por 4 semanas exactas de 7 días.

Se debe comentar todo el código que se implemente.

1. Implementa una función que muestre un menú de opciones. Mostrará por pantalla las opciones, leerá la entrada de teclado y, en función de ésta, llamará a la función que realice la operación correspondiente. Después de que el usuario seleccione una opción y se realice la operación elegida, volverá a mostrarse el menú para que el usuario pueda volver a seleccionar otra opción, y así sucesivamente hasta que el usuario pulse 5 y se finalice la ejecución. Ejemplo de ejecución:

```
**** ESTADÍSTICA DE LLUVIAS EN PLASENCIA - FEBRERO ****
Elija una opción:
1 - Rellenar los litros de agua por metro cuadrado diarios.
2 - Mostrar los litros diarios de agua del mes.
3 - Mostrar la media de litros diarios del mes.
4 - Mostrar el día o días más lluviosos.
5 - Salir del programa.
```

2. Implementa una función que permita al usuario rellenar la matriz con los litros de lluvia diarios, pidiéndolos uno a uno. Ejemplo de ejecución:

```
**** ESTADÍSTICA DE LLUVIAS EN PLASENCIA - FEBRERO ****
Elija una opción:
1 - Rellenar los litros de agua por metro cuadrado diarios.
2 - Mostrar los litros diarios de agua del mes.
3 - Mostrar la media de litros diarios del mes.
4 - Mostrar el día o días más lluviosos.
5 - Salir del programa.
1
Semana 1
  Lunes: 3
  Martes: 0
  Miércoles: 0
  Jueves: 1
  Viernes: 2
  Sábado: 0
  Domingo: 9
Semana 2
  Lunes: 0
  Martes: 1
  Miércoles: 1
  Jueves: 2
  Viernes: 2
  Sábado: 9
  Domingo: 0
Semana 3
  Lunes: 0
  Martes: 0
  Miércoles: 0
  Jueves: 1
  Viernes: 4
  Sábado: 4
  Domingo: 2
```

```
Semana 4
Lunes: 1
Martes: 0
Miércoles: 0
Jueves: 2
Viernes: 1
Sábado: 1
Domingo: 0
- Tabla de lluvias rellena -
```

3. Implementa una función que muestre los litros registrados para cada día de cada semana del mes. Ejemplo de ejecución:

```
**** ESTADÍSTICA DE LLUVIAS EN PLASENCIA - FEBRERO ****
Elija una opción:
1 - Rellenar los litros de agua por metro cuadrado diarios.
2 - Mostrar los litros diarios de agua del mes.
3 - Mostrar la media de litros diarios del mes.
4 - Mostrar el día o días más lluviosos.
5 - Salir del programa.
2
Semana 1
Lunes:3  Martes:0  Miércoles:0  Jueves:1  Viernes:2  Sábado:0  Domingo:9
Semana 2
Lunes:0  Martes:1  Miércoles:1  Jueves:2  Viernes:2  Sábado:9  Domingo:0
Semana 3
Lunes:0  Martes:0  Miércoles:0  Jueves:1  Viernes:4  Sábado:4  Domingo:2
Semana 4
Lunes:1  Martes:0  Miércoles:0  Jueves:2  Viernes:1  Sábado:1  Domingo:0
```

4. Implementa una función que muestre la cantidad media de litros de lluvia del mes. Ejemplo de ejecución:

```
**** ESTADÍSTICA DE LLUVIAS EN PLASENCIA - FEBRERO ****
Elija una opción:
1 - Rellenar los litros de agua por metro cuadrado diarios.
2 - Mostrar los litros diarios de agua del mes.
3 - Mostrar la media de litros diarios del mes.
4 - Mostrar el día o días más lluviosos.
5 - Salir del programa.
3
La media de litros de lluvia diarios del mes ha sido de 1,64 litros.
```

5. Implementa una función que muestre el día o los días más lluviosos del mes. Ejemplo de ejecución:

```
**** ESTADÍSTICA DE LLUVIAS EN PLASENCIA - FEBRERO ****
Elija una opción:
1 - Rellenar los litros de agua por metro cuadrado diarios.
2 - Mostrar los litros diarios de agua del mes.
3 - Mostrar la media de litros diarios del mes.
4 - Mostrar el día o días más lluviosos.
5 - Salir del programa.
4
El día o días más lluviosos del mes fueron:
El Domingo de la Semana 1 con 9 litros.
El Sábado de la Semana 2 con 9 litros.
```

Valoración:

- **Ejercicio 1:** Máximo 4 puntos
- **Ejercicio 2:** Máximo 4 puntos.
- Plantea un procedimiento y una progresión correcta en el desarrollo de los ejercicios. Es riguroso y claro en las técnicas utilizadas: Máximo 1 punto
- Limpieza y orden en la presentación de los ejercicios: Máximo 1 punto.