

Nombre del centro académico IES Virgen del Carmen. Primer trimestre del 2025/26

**Profesor: Santiago Rodenas Herráiz**

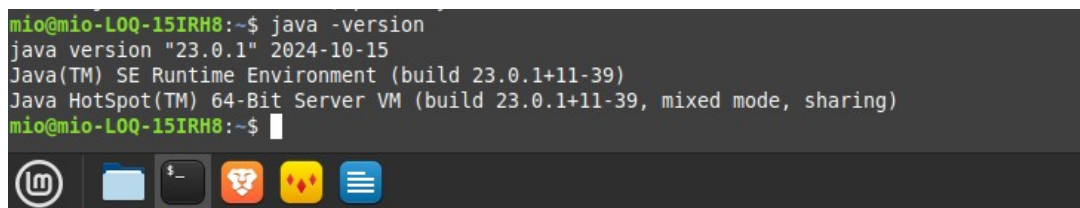
**Correo electrónico: srodher115@g.educaand.es**

# Prueba inicial

## PSP (2º DAM)

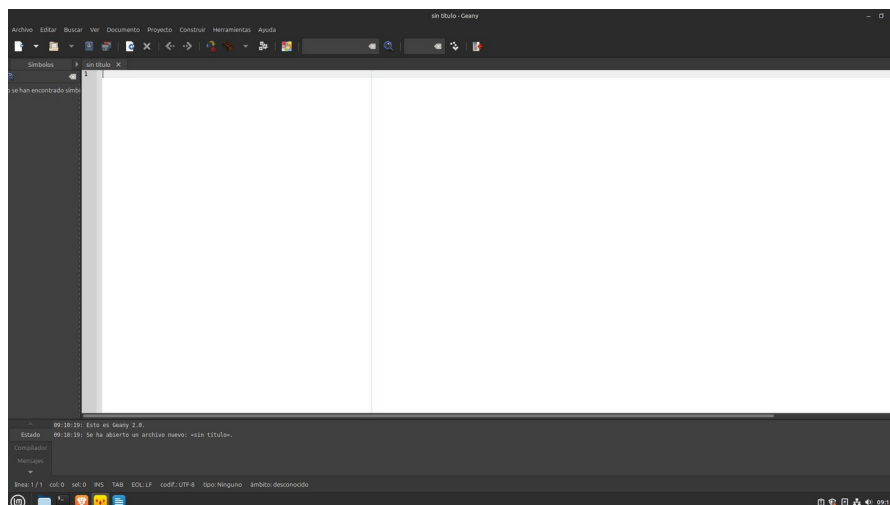
---

1. En vuestro ordenador, verificar la versión de la runtime de java y del compilador. (Realizar captura de pantalla)

A screenshot of a Linux terminal window. The prompt is 'mio@mio-L0Q-15IRH8:~\$'. The user has entered 'java -version'. The output is: 'java version "23.0.1" 2024-10-15', 'Java(TM) SE Runtime Environment (build 23.0.1+11-39)', and 'Java HotSpot(TM) 64-Bit Server VM (build 23.0.1+11-39, mixed mode, sharing)'. The prompt is now 'mio@mio-L0Q-15IRH8:~\$'. The terminal has a dark background and a taskbar at the bottom with icons for a terminal, file manager, and other applications.

```
mio@mio-L0Q-15IRH8:~$ java -version
java version "23.0.1" 2024-10-15
Java(TM) SE Runtime Environment (build 23.0.1+11-39)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.1+11-39, mixed mode, sharing)
mio@mio-L0Q-15IRH8:~$
```

2. Descargar e instalar cualquier editor de JAVA del tipo geany, JEdit, etc. La idea es utilizar un simple editor y no un entorno. (Realizar captura de pantalla)



**3. Explicar en qué consiste la multiprogramación y la monoprogramación.**

**Multiprogramación:** Te permite realizar varios procesos a la vez y por varios hilos

**monoprogramación:** Solo te permite utilizar un proceso a la vez y un solo hilo

**4. Comentar las partes de la arquitectura básica de Voon Neumman.**

**5. Explicar la diferencia entre UC y la ALU.**

**6. Explicar qué es el Kernel de linux y sus funciones.**

Es la parte principal de liux y es donde se encuentra toda la informacion del sistema operativo

**7. Sabes en qué consiste una estructura de PILA en programación.**

**8. Abrir un fichero de texto en gedit y crear una clase que implemente método público con todos los números pares comprendidos entre un min y un max. Deberéis declararos min y max como atributos privados. Compilar el fichero sobre línea de comandos y ejecutar. (Realizar captura de pantalla del código y de su ejecución )**



```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 7.2 /home/mio/Esconditeo/disco duro/DAT2/PSP/TEM1/Pares.java
//
// @author Pedro Gutiérrez Vico
//
import java.util.Scanner; // importación de un paquete externo para poder utilizar la clase Scanner

public class Pares {
    private static int min;
    private static int max;
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Introduce el número mínimo: ");
        min = sc.nextInt();
        System.out.println("Introduce el número máximo: ");
        max = sc.nextInt();
        for (int i = min; i <= max; i++) {
            if (i % 2 == 0) {
                System.out.print(i + " ");
            }
        }
    }
}
```

```
Archivo  Editor  Ver  Buscar  Terminal  Ayuda
mio@mio-LQJ-15IRH8: ~
- javaagent: -jargpath[=<options>]
  Load Java programming language agent, see java.lang.instrument
-splash: <imagepath>
  show splash screen with specified image
  HIGPI scaled images are automatically supported and used
  if available. The unscaled image filename, e.g. image.ext,
  should always be passed as the argument to the --splash option.
  The most appropriate scaled image provided will be picked up
  automatically.
  See the SplashScreen API documentation for more information
@argument files
  one or more argument files containing options
--disable-@files
  prevent further argument file expansion
--enable-preview
  allow classes to depend on preview features of this release
To specify an argument for a long option, you can use --<name>=<value> or
--<name> <value>.

mio@mio-LQJ-15IRH8:~$ java -version
Errors: Could not find or load main class version
Caused by: java.lang.ClassNotFoundException: version
mio@mio-LQJ-15IRH8:~$ java -version
java-version: no se encontró la orden
Orden «java» no encontrada, pero hay 25 similares.
mio@mio-LQJ-15IRH8:~$ java -version
java version "23.0.1" 2024-10-15
Java(TM) SE Runtime Environment (build 23.0.1+11-39)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.1+11-39, mixed mode, sharing)
mio@mio-LQJ-15IRH8:~$ sudo gedit /home/mio/Escritorio/disco duro/DAM 2/PSP/TEMA 1/Pares.java
[sudo] contraseña para mio:
sudo: gedit: orden no encontrada
mio@mio-LQJ-15IRH8:~$ sudo nano /home/mio/Escritorio/disco duro/DAM 2/PSP/TEMA 1/Pares.java
mio@mio-LQJ-15IRH8:~$ sudo nano /home/mio/Escritorio/
base contenedores
base contenedores.zip
brave-browser.desktop
brave-wikipedia:cepnokjaffmgkkgg-Default.desktop
mio@mio-LQJ-15IRH8:~$ sudo nano /home/mio/Escritorio/disco duro/
Calculadora.java
concatenar.java
convertidor.java
DAM1/
DAM2/
OVPedrol.java
mio@mio-LQJ-15IRH8:~$ sudo nano /home/mio/Escritorio/disco duro/DAM2/PSP/TEMA1/
.idea/ PSP/
mio@mio-LQJ-15IRH8:~$ sudo nano /home/mio/Escritorio/disco duro/DAM2/PSP/TEMA1/
.idea/ out/ Pares.java
mio@mio-LQJ-15IRH8:~$ sudo nano /home/mio/Escritorio/disco duro/DAM2/PSP/TEMA1/Pares.java
mio@mio-LQJ-15IRH8:~$ sudo java /home/mio/Escritorio/disco duro/DAM2/PSP/TEMA1/Pares.java
Inserte el numero minimo:
1
Inserte el numero maximo:
97
1 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 mio@mio-LQJ-15IRH8:~$

Archivos recientes:
fichas semanales fct/
juegos/
org.gnome.Dia.desktop
Practica 4.3.docx
med3javaFile1.java
pedrol/
Practicas de wifi y hacking para SWM.docx
Practicas de wifi y hacking para SWM_Pedro y_Javi.pdf
practicas documentos/
practicas paso a paso del tema http.docx
test db.sql
WEB TEST/
WEB TEST.zip
Screenshot 2024-03-07-16-48-38-615_com.jpg
trabajo/
trabajo bootstrap/
trabajo bootstrap.rar
VirtualBox-7.0.14-161095-Win.exe
```

```
import java.util.ArrayList;
import java.util.Scanner; // importación de un paquete externo para poder utilizar la clase Scanner

public class Pares2 {
    private static int min; // 2 usages
    private static int max; // 2 usages

    public static void main(String[] args) {
        Sonpares();
    }

    public static ArrayList<Integer> Sonpares(){ // 1 usage
        Scanner sc = new Scanner(System.in);

        ArrayList resultado = new ArrayList();

        System.out.println("Inserte el numero minimo");
        min = sc.nextInt();
        sc.nextLine();

        System.out.println("Inserte el numero maximo");
        max = sc.nextInt();
        sc.nextLine();

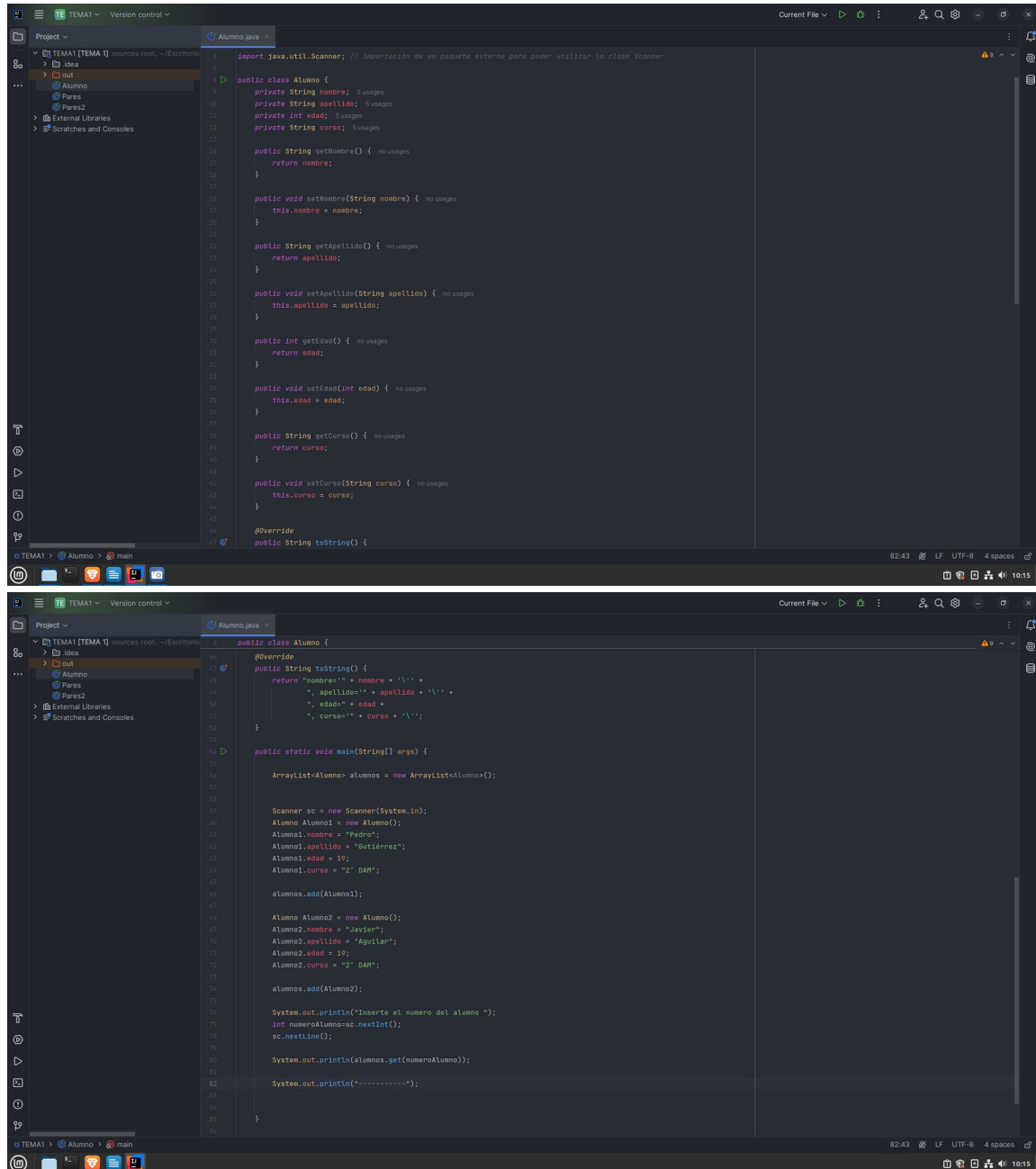
        for(int i = min; i <= max; i++){
            if(i % 2 == 0){
                resultado.add(i);
            }
        }

        System.out.println(resultado);

        return resultado;
    }
}
```

## 9. Implementar una clase Alumno con el nombre, apellidos, edad y curso.

Debéis incorporar el constructor, getters/setters, toString. Implementar una lista dinámica en el main, que permita leer un número de objetos pedidos desde el teclado. Deberéis sacar por pantalla toda la lista de Alumnos.



The image shows two screenshots of an IDE (IntelliJ IDEA) displaying the implementation of the `Alumno` class and its `main` method.

**Top Screenshot: `Alumno.java`**

```
import java.util.Scanner; // importación de un paquete externo para poder utilizar la clase Scanner

public class Alumno {
    private String nombre; // usages
    private String apellido; // usages
    private int edad; // usages
    private String curso; // usages

    public String getNombre() { // no usages
        return nombre;
    }

    public void setNombre(String nombre) { // no usages
        this.nombre = nombre;
    }

    public String getApellido() { // no usages
        return apellido;
    }

    public void setApellido(String apellido) { // no usages
        this.apellido = apellido;
    }

    public int getEdad() { // no usages
        return edad;
    }

    public void setEdad(int edad) { // no usages
        this.edad = edad;
    }

    public String getCurso() { // no usages
        return curso;
    }

    public void setCurso(String curso) { // no usages
        this.curso = curso;
    }

    @Override
    public String toString() {
```

**Bottom Screenshot: `Alumno.java`**

```
    @Override
    public String toString() {
        return "Nombre:" + nombre + '\n' +
            ", apellido:" + apellido + '\n' +
            ", edad:" + edad + '\n' +
            ", curso:" + curso + '\n';
    }

    public static void main(String[] args) {
        ArrayList<Alumno> alumnos = new ArrayList<Alumno>();

        Scanner sc = new Scanner(System.in);
        Alumno Alumno1 = new Alumno();
        Alumno1.nombre = "Pedro";
        Alumno1.apellido = "Gutiérrez";
        Alumno1.edad = 19;
        Alumno1.curso = "2º DAM";

        alumnos.add(Alumno1);

        Alumno Alumno2 = new Alumno();
        Alumno2.nombre = "Javier";
        Alumno2.apellido = "Aguilar";
        Alumno2.edad = 19;
        Alumno2.curso = "2º DAM";

        alumnos.add(Alumno2);

        System.out.println("Inserte el número del alumno ");
        int numeroAlumno = sc.nextInt();
        sc.nextLine();

        System.out.println(alumnos.get(numeroAlumno));

        System.out.println("-----");
    }
}
```

**10. Implementar un método, que lea desde teclado cualquier carácter hasta pulsar intro. Por cada carácter, hay que mandarlo a un fichero llamado texto.txt en el directorio /tmp, donde no aparezca ninguna vocal. Ejemplo: Si leemos el texto “hola que tal”, deberemos ver en el fichero el texto hl q tl.** Sólo se puede consultar la página de **oracle**, para repasar las funciones de lectura y escritura.