

Práctica 1: Repaso programación en Java.

1. Utilizando Java desde una terminal. Realiza los siguiente pasos:

(a) Descargar **JDK** (Java Development Kit) del sitio:

<https://www.oracle.com/java/technologies/downloads/>.

Este paquete contiene todo lo necesario para programar en Java, incluyendo:

- **JRE** (Java Runtime Environment): necesario para ejecutar programas Java.
- **javac**: el compilador de Java (traduce tu código .java en bytecode .class).
- tambien incluye otras herramientas como **java**, **javadoc**, etc.

Elige la versión estable más reciente (por ejemplo, **Java 21 LTS**).

Descarga el instalador adecuado para tu sistema operativo (**Windows, macOS o Linux**).

- (b) **Instalación:** Ejecuta el instalador y sigue las instrucciones para tu sistema operativo.
- (c) **Configuración:** Configura la variable de entorno **PATH** Esto es necesario para que puedas usar **javac** y **java** desde cualquier carpeta en una terminal.

En **Windows**:

- Abre **Panel de Control** → **Sistema** → **Configuración avanzada del sistema**.
- Haz clic en Variables de entorno.
- Busca la variable Path, edítala y agrega:
`C:\ProgramFiles\Java\jdk-<NRO-DE-VERSION>\bin`
- Guarda y cierra.

En **macOS/Linux**:

- Edita tu archivo de configuración (**.bashrc**, **.zshrc** o **.bash_profile**) y agrega:
`export PATH=$PATH:/<RUTA-AL-JDK>/bin`
- Guarda el archivo, cierra todas las terminales que tengas abiertas.

(d) **Verificación:** Abre una consola (**cmd** en Windows, **Terminal** en Mac/Linux)

- Para verificar JRE, escribe en la terminal: `java -version`

debería aparecerte algo como:

```
java version "21.0.2" 2025-01-16 LTS
Java(TM) SE Runtime Environment ...
Java HotSpot(TM) 64-Bit Server VM ...
```

- Para verificar el compilador de java, escribe: `javac -version`

debería mostrarte algo como:

```
javac 21.0.2
```

2. **Compilar y ejecutar con javac:** Abre cualquier editor de texto o si prefieres usar algún editor de código como sublime, atom o notepad++ también puedes hacerlo. Crea un archivo llamado **HelloWorld.java**.

Escribe el siguiente código dentro del mismo:

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello World from Java!");  
    }  
}
```

Abre una terminal, ubícate en la carpeta donde creaste el archivo **HelloWorld.java** y escribe:

- para **compilar**:

```
javac HelloWorld.java
```

Esto genera un archivo HelloWorld.class (bytecode).

- para **ejecutar**:

```
java HelloWorld
```

Atención: Para ejecutar, no escribas .class ni .java al final.

3. Analizando el método **main**

```
public static void main(String args[])
```

En Java, dicho método es el punto de inicio de un programa. Cuando se ejecuta un programa con el intérprete de Java, se comienza llamando al método **main()** de esa clase. A partir de ahí, este método se encarga de invocar a todos los demás métodos necesarios para ejecutar la aplicación.

El perfil del método main contiene los siguientes modificadores:

- **public**
- **static**
- **void**

Investiga y explica que efecto tiene cada uno de estos modificadores sobre el método.

Toma el programa del ejercicio 2,

- (a) Renombra el método **main()** por **principal()**. Guarda los cambios y compila. Dio algún error? Prueba ahora ejecutar. Dio algún error?.
- (b) Prueba ahora lo mismo pero eliminando el **static**. Da algún mensaje de error? Fue en el momento de compilación o ejecución?

- (c) Modifica el tipo de retorno del main de **void** a **int**. Prueba compilar, fíjate el mensaje de error que recibes. Agrega ahora el return para que quede de la siguiente manera:

```
public class HelloWorld {
    public static int main(String[] args){
        System.out.println("Hello World from Java!")
    };
    return 0;
}
```

Compila y ejecuta, da algún mensaje de error? Fue en el momento de compilación o ejecución?

- (d) **Bonus:** Renombra el nombre del archivo de "HelloWorld.java" a "HolaMundo.java", guarda los cambios, compila y ejecuta. Da algún mensaje de error? Fue en el momento de compilación o ejecución?
4. Crea un archivo llamado Greeting.java que contenga el siguiente código java:

```
public class Greeting {
    public static void main(String[] args) {
        if (args.length > 0) {
            String name = args[0];
            System.out.println("Hello, " + name + "!
                               Welcome to Java!");
        } else {
            System.out.println("Please run the program
with your name as an argument.");
        }
    }
}
```

Abre una terminal, ubícate en el directorio donde guardaste el archivo y compila: **javac Greeting.java**. Luego ejecuta lo siguiente y compara los resultados. Qué pasó en cada caso?

- (a) **java Greeting**
 - (b) **java Greeting Pablo**
 - (c) **java Greeting <TU-NOMBRE>**
5. Escribe un programa en Java llamado **Sum** que reciba exactamente dos números como argumentos desde la línea de comandos, los convierta de tipo **String** a **int**, calcule su suma y muestre el resultado en el formato **"La suma es: <resultado>"**. Si no se reciben exactamente dos argumentos, el programa debe mostrar el mensaje **"Por favor ingresa exactamente dos números"**.
6. Vamos a retomar el ejercicio de mediciones de temperaturas trabajado previamente en la materia "Introducción a los algoritmos" durante el primer cuatrimestre.
- (a) Descarga el archivo **MedicionesTemperaturas.java**, provisto en nuestra aula virtual y ubícalo en una nueva carpeta.
 - (b) Crea en la misma carpeta donde copiaste el archivo anterior, un archivo llamado **MedicionesTemperaturasTest.java**.

- (c) Define la clase con el mismo nombre del archivo y agrega un método main.
 - (d) Dentro del main, crea una instancia de **MedicionesTemperaturas** para la fecha 1 de Febrero. Luego agrega las siguientes temperaturas $[-1, 3, 8, 15, 12, 6, 6, 2]$ al objeto.
 - (e) Invoca el método imprimir para dicho objeto. Compila y ejecuta desde una terminal.
7. Extiende la clase `MedicionesTemperaturas.java`, con métodos que calculen:
- (a) el promedio de las temperaturas registradas
 - (b) la máxima temperatura registrada.
 - (c) la mínima temperatura registrada.
 - (d) la amplitud térmica, es decir la diferencia entre la máxima temperatura registrada y la mínima temperatura.

Compila y verifica que las implementaciones realizadas no tienen errores de sintaxis, ni de tipos.

Luego agrega código en el main para probar estos métodos imprimiendo los resultados retornados por los mismos. Compila y ejecuta desde una terminal.

8. Busca la implementación realizada de **ArrayUtils.java**, en la Tarea 6 del Práctico 5, de "Introducción a los algoritmos". En caso que no lo tengas descarga el archivo **ArrayUtils.java**, provisto en nuestra aula virtual y completa las implementaciones de los métodos propuestos.
- Copia el archivo con los métodos implementados en una nueva carpeta.
 - Extiende la clase con un método main para probar las implementaciones realizadas. Compila y ejecuta desde una terminal.