

# **ANALIZADOR DE TEMPERATURAS MENSUALES**

Christiana Adeola Ogunjobi

# ÍNDICE

- 1. Problema a resolver
- 2. Objetivos del proyecto
- 3. Menú principal
- 4. introducirTemperaturas
- 5. mostrarEstadísticas
- 6. Estructuras de datos
- 7. Estructuras de control
- 8. Demostración
- 9. Control de versiones (GITHUB)
- 10. Dificultades y aprendizajes
- 11. Mejoras futuras
- 12. Conclusión

# PROBLEMA A RESOLVER

En muchos casos necesitamos analizar temperaturas registradas durante un mes completo. Se requiere una aplicación sencilla que permita:

- Introducir temperaturas manualmente o de forma aleatoria.
- Calcular estadísticas básicas.
- Mostrar un histograma.
- Indicar qué días superan la media.

El objetivo es crear una herramienta simple pero útil para analizar datos de temperatura.

# OBJETIVOS DEL PROYECTO

- Practicar programación estructurada (sin POO).
- Utilizar arrays, bucles y condicionales.
- Validar la entrada del usuario.
- Implementar un menú con **switch**.
- Aplicar control de versiones con GitHub.

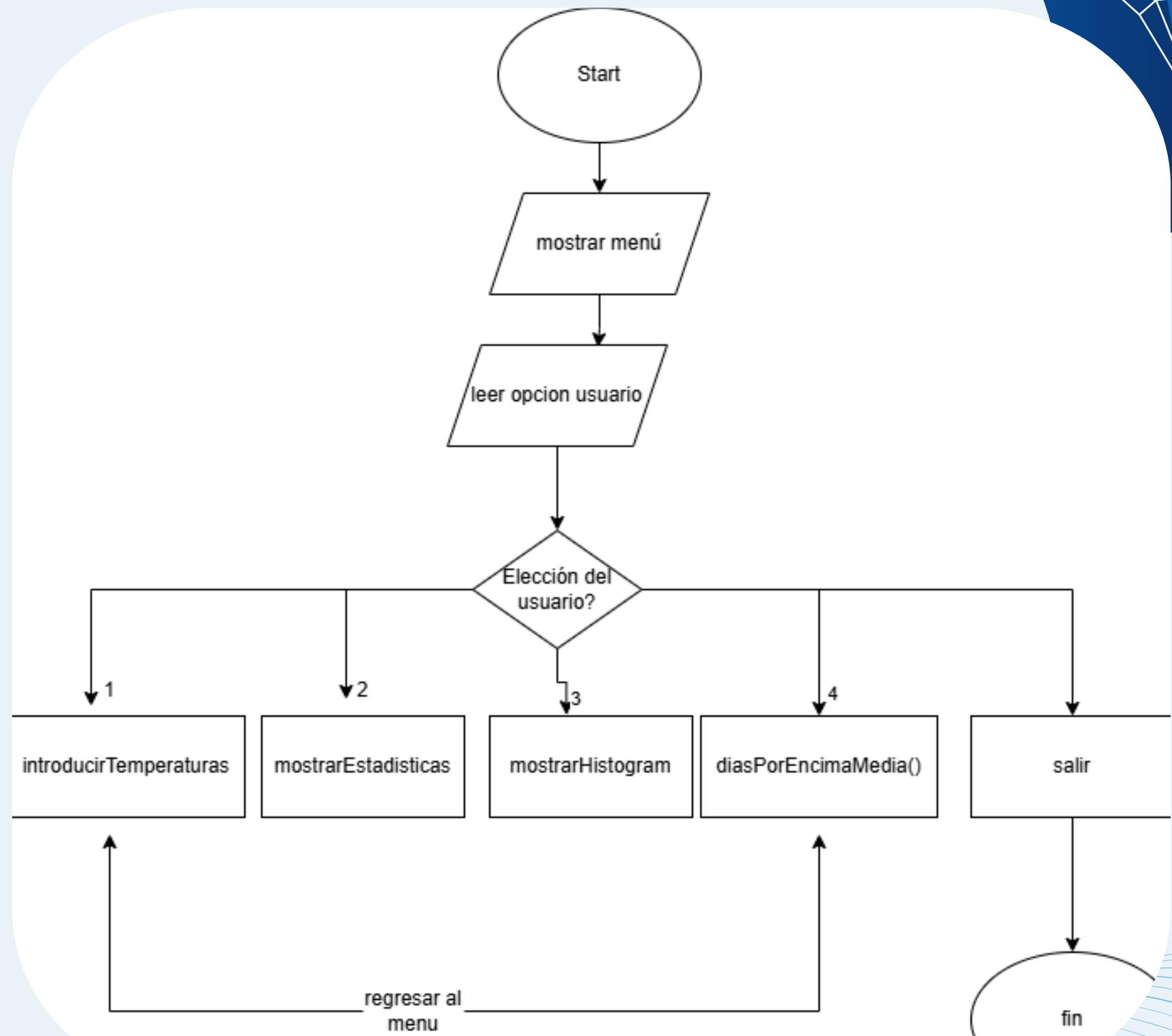
El proyecto combina la lógica de programación con el uso de herramientas de desarrollo.

# MENÚ PRINCIPAL

Opciones del menú:

- Introducir temperaturas
- Mostrar estadísticas
- Mostrar histograma
- Días por encima de la media
- Salir

El menú se implementa con un bucle  
do-while y un switch dentro del método main



# INTRODUCIR TEMPERATURAS

- El usuario elige entrada manual o aleatoria.
- Un bucle rellena el array.
- Se valida la entrada para evitar errores.

Este método inicializa el array de temperaturas. Todas las demás funciones dependen de él.

# MOSTRAR ESTADÍSTICAS

- Recorre las 30 temperaturas.
- Calcula:
  - Media
  - Máxima y día correspondiente
  - Mínima y día correspondiente

Este método es el núcleo analítico del programa.

# ESTRUCTURAS DE DATOS

- Array estático: **double[] temperaturas = new double[30]**
- Variables auxiliares:
  - **sum, max, min**
  - **diaMax, diaMin**
  - **media**

Se utilizan solo arrays y variables simples, tal como exige la práctica

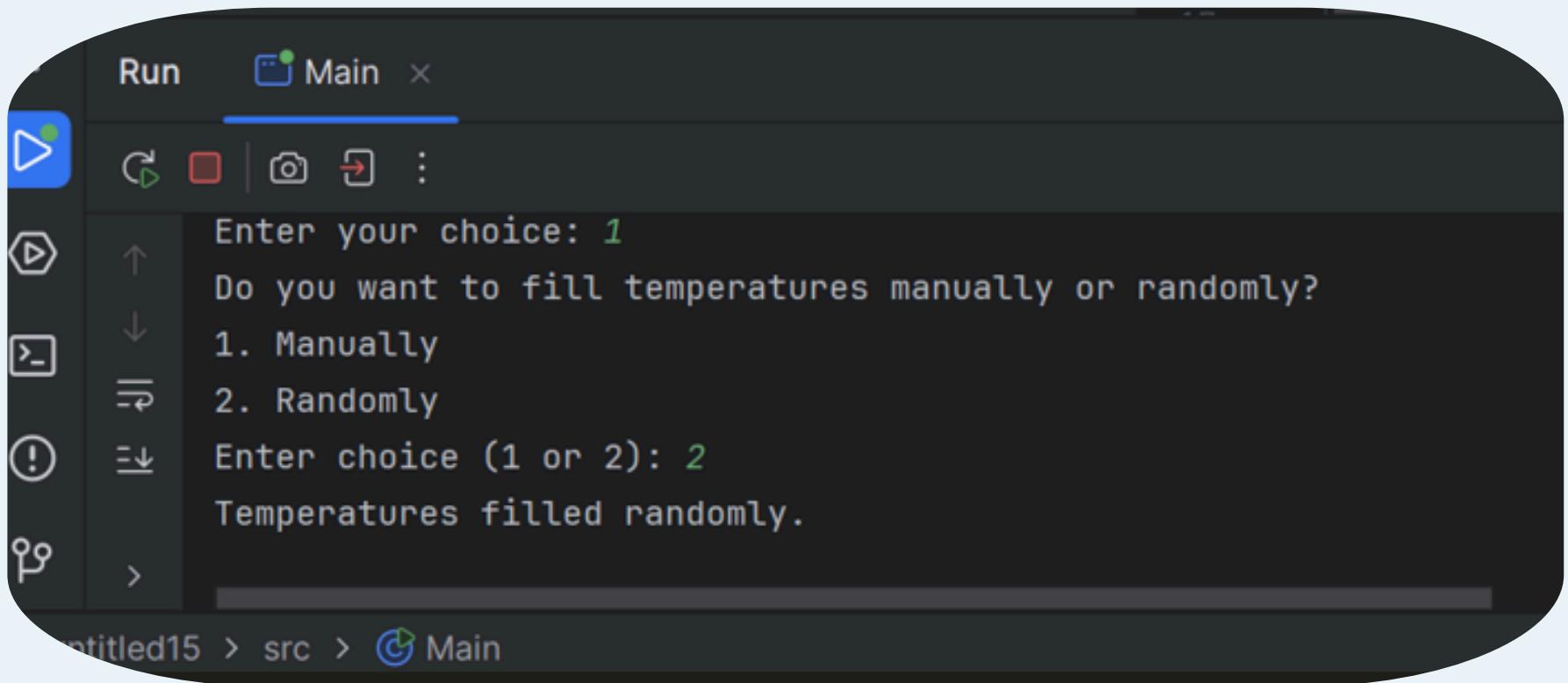
# ESTRUCTURAS DE CONTROL

- **if / else** → comparaciones de máximos y mínimos
- **switch** → menú principal
- **while / for** → recorrido de las temperaturas
- Validación de entrada con **hasNextDouble()**

El proyecto demuestra dominio de las estructuras básicas de control

# DEMOSTRACIÓN

- Introducción de temperaturas



The screenshot shows a terminal window titled "Main". The window has a dark theme with light-colored text. It displays the following interaction:

```
Enter your choice: 1
Do you want to fill temperatures manually or randomly?
1. Manually
2. Randomly
Enter choice (1 or 2): 2
Temperatures filled randomly.
```

The terminal window is part of a larger interface with a navigation bar at the top and a file tree at the bottom.

# DEMOSTRACIÓN

- Estadísticas mostradas

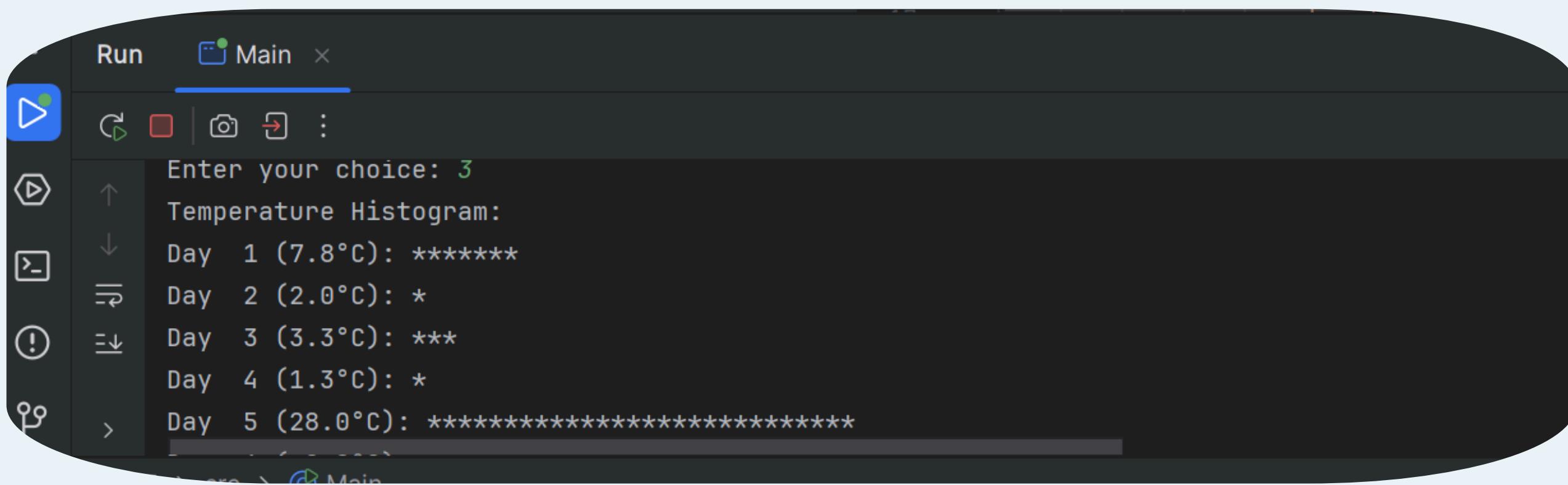
```
Run Main x
Run Main
Main

Enter your choice: 2
Average temperature: 18.73°C
Maximum temperature: 39.46°C (Day 8)
Minimum temperature: -2.82°C (Day 6)

== Weather Data Analyzer ==
1. Enter temperatures
```

# DEMOSTRACIÓN

- Histograma generado



The screenshot shows a terminal window titled "Main". The user has entered the command "3" and the program has generated a temperature histogram for five days. The output is as follows:

```
Enter your choice: 3
Temperature Histogram:
Day 1 (7.8°C): *****
Day 2 (2.0°C): *
Day 3 (3.3°C): ***
Day 4 (1.3°C): *
Day 5 (28.0°C): *****
```

# DEMOSTRACIÓN

- Días por encima de la media



```
Run Main ×
Run | Main | : 
Enter your choice: 4
Average temperature: 18.73°C
Days with temperature above average:
Day 5 (28.0°C)
Day 8 (39.5°C)
Day 9 (27.5°C)
Day 10 (28.0°C)
```

d15 > src > Main

Estas capturas muestran el funcionamiento real de la aplicación

# CONTROL DE VERSIONES

- Uso del repositorio de la clase.
- Commit frecuentes
- Ramas por funcionalidad
- Release final creada

The screenshot shows a GitHub repository page for a private repository named 'lsg-0485-practica-i-christiana'. The repository was forked from 'LSG-Programacio/lsg-programacio-0485-0487-dam-daw-0485\_0487\_practica\_i\_-0485\_0487\_Practica\_I'. The main branch has 1 commit ahead of it. The repository contains several files: 'Enunciat Presentació.md', 'Enunciat Validació Idea.md', 'Enunciat.md', 'Idees Projectes DAM DAW.md', and 'README.md'. The commits are all from the 'github-classroom[bot]' user, with the first three being 'Initial commit' and the last two being 'add deadline'. The repository has 0 stars, 0 forks, and 0 watching. A release titled 'Entrega Final del Proyecto' was made 9 minutes ago.

File	Commit Type	Time Ago
Enunciat Presentació.md	Initial commit	2 weeks ago
Enunciat Validació Idea.md	Initial commit	2 weeks ago
Enunciat.md	Initial commit	2 weeks ago
Idees Projectes DAM DAW.md	Initial commit	2 weeks ago
README.md	add deadline	4 days ago

Se siguieron buenas practicas de control de versiones durante todo el desarrollo.

# DIFICULTADES Y APRENDIZAJES

- Validación de datos del usuario
- Corrección de bucles
- Diseño de diagramas de flujo
- Organización modular del código

Este proyecto me ayudó a mejorar mi pensamiento lógico y mis habilidades de depuración.

# MEJORAS FUTURAS

- Exportar datos a un archivo
- Implementar interfaz gráfica
- Comparar varios meses
- Añadir más estadísticas avanzadas

# CONCLUSIÓN

Este proyecto me permitió aplicar los principales conceptos de programación vistos en clase y desarrollar un programa completo y funcional. Practiqué el uso de menús, métodos, bucles, condicionales y arrays.



**GRACIAS POR LA  
ATENCIÓN**