



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.

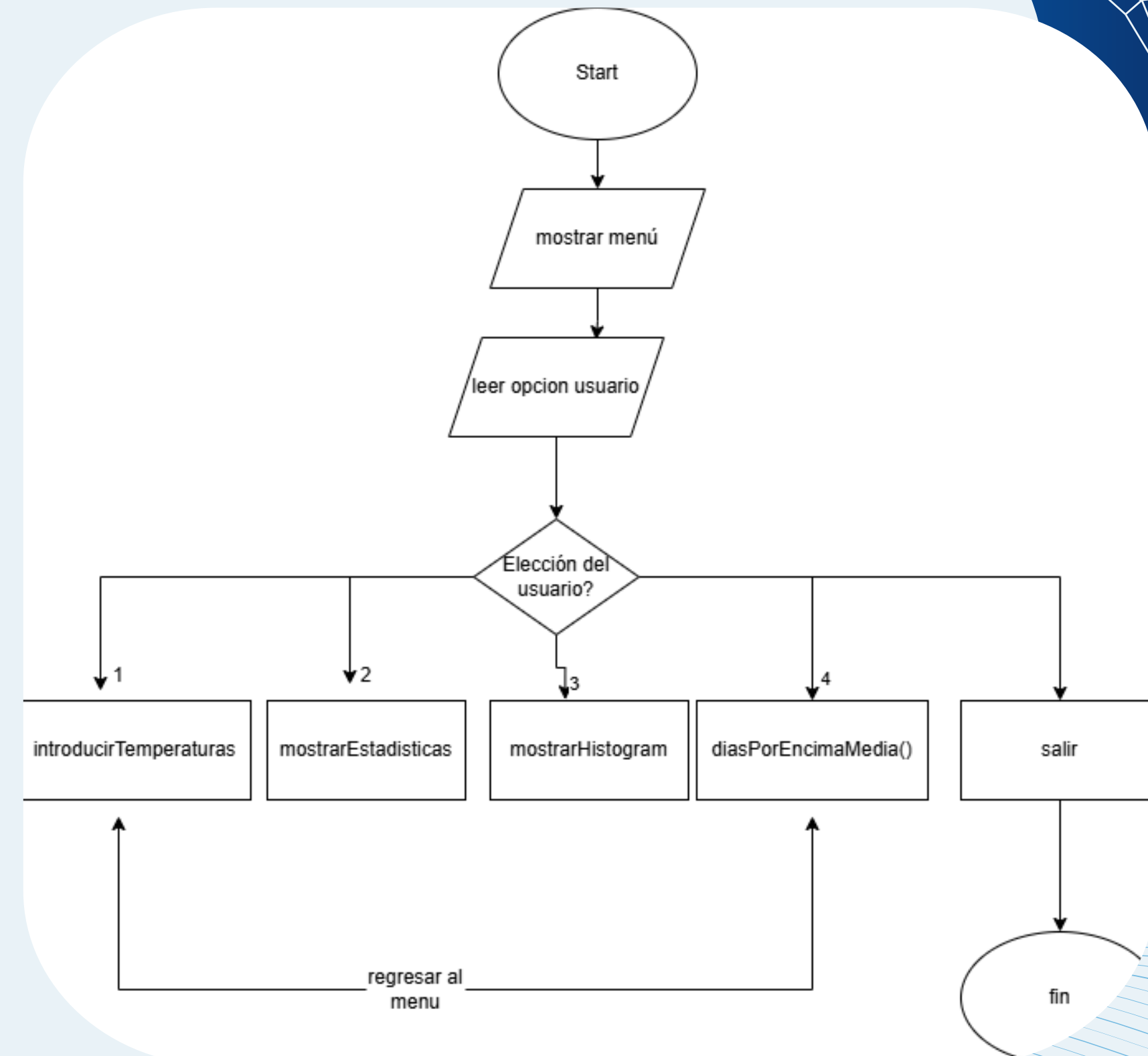
El proyecto combina la lógica de programación con el uso de herramientas de desarrollo.

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

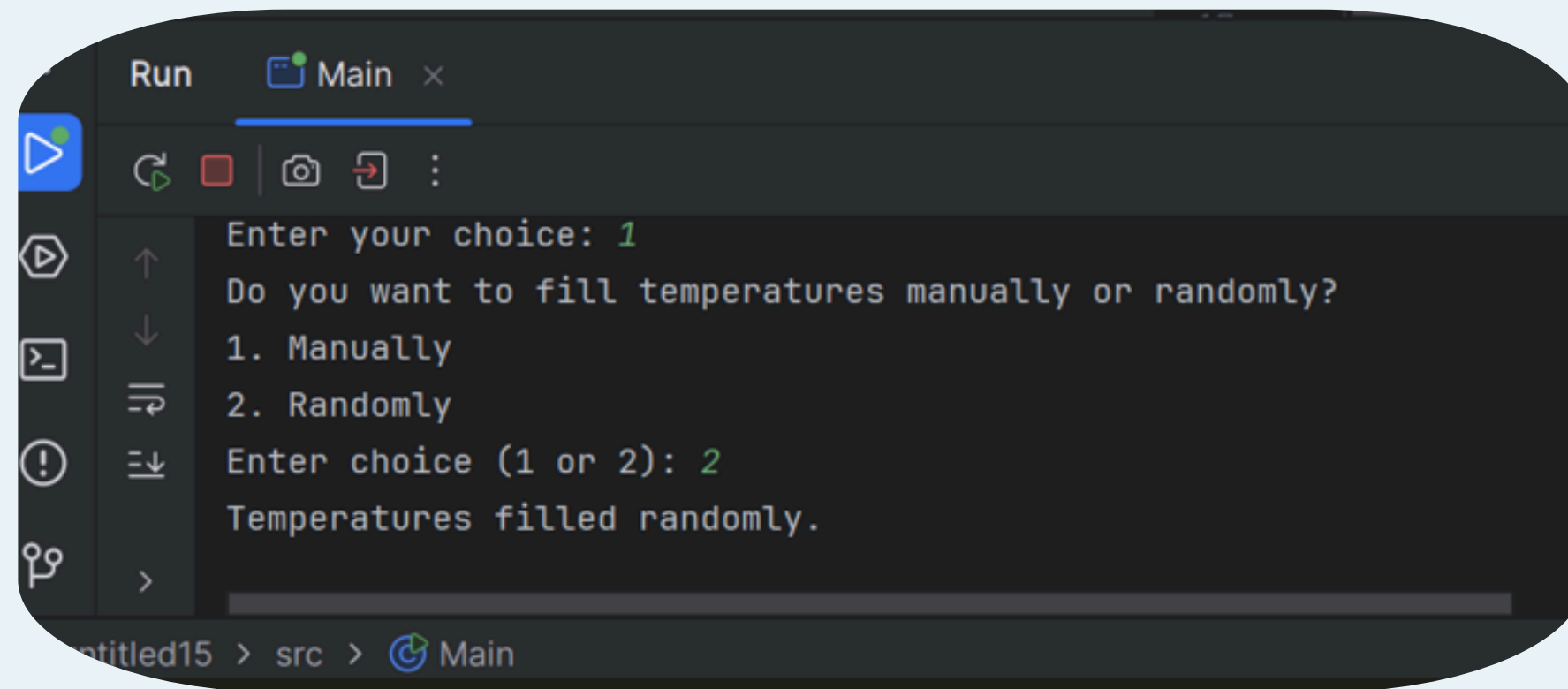
ESTRUCTURAS DE CONTROL

- **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



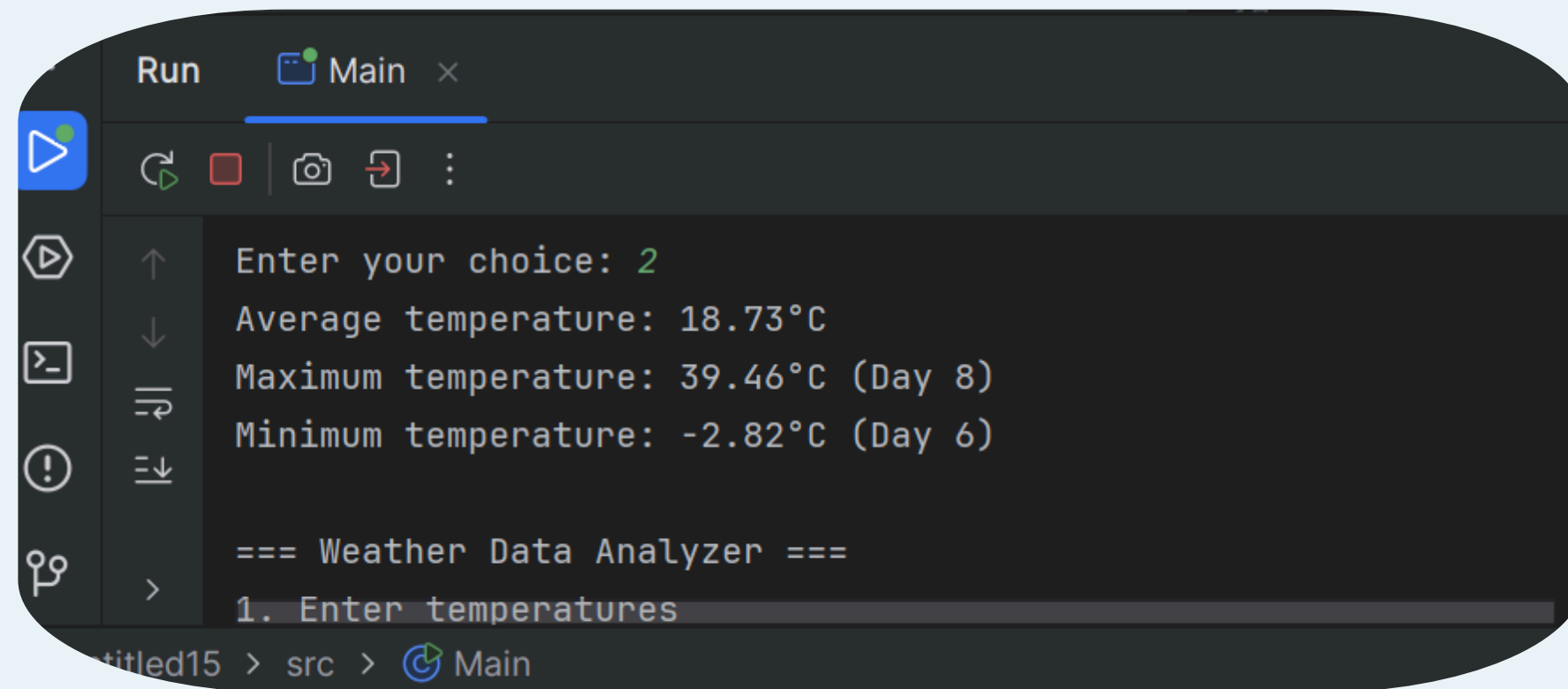
The screenshot shows a code editor window titled "Main" with a dark theme. The code is as follows:

```
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 output shows the user has entered '1' for the first prompt and '2' for the second prompt, resulting in the message "Temperatures filled randomly." The editor's status bar at the bottom indicates the file path is "untitled15 > src > Main".

DEMOSTRACIÓN

- Estadísticas mostradas



The screenshot shows a code editor window with a dark theme. The title bar at the top says "Run" and "Main". Below the title bar is a toolbar with icons for running, stopping, and other actions. The main area of the editor displays the following text:

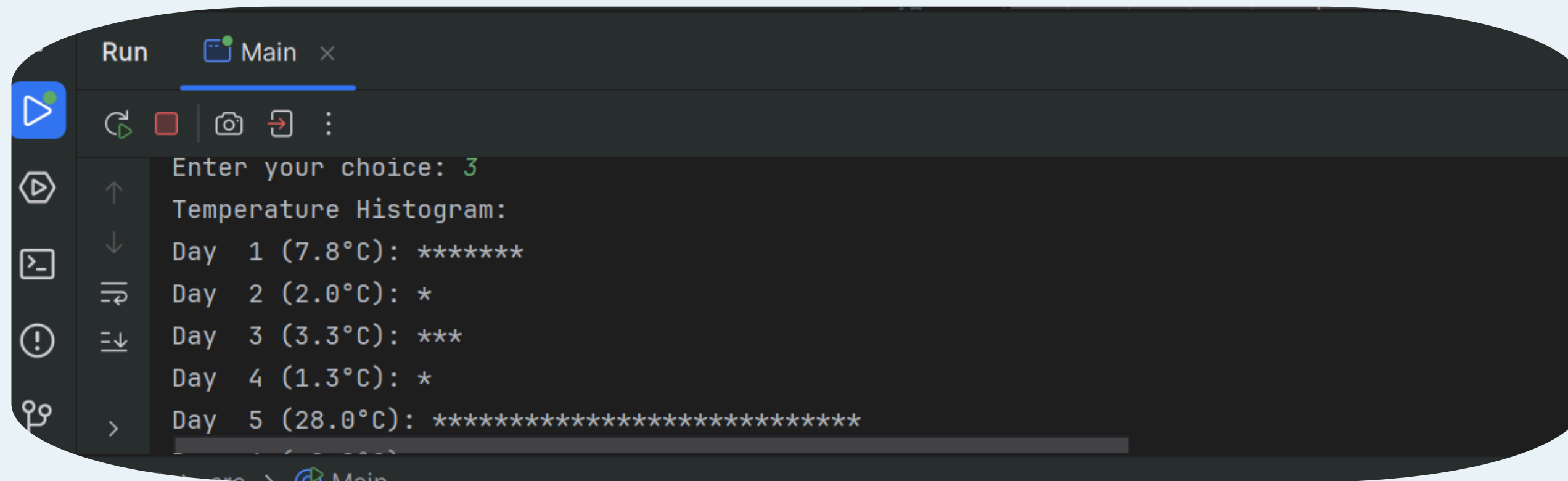
```
Enter your choice: 2
Average temperature: 18.73°C
Maximum temperature: 39.46°C (Day 8)
Minimum temperature: -2.82°C (Day 6)

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

At the bottom of the editor, the file path is shown as "titled15 > src > Main".

DEMOSTRACIÓN

- Histograma generado



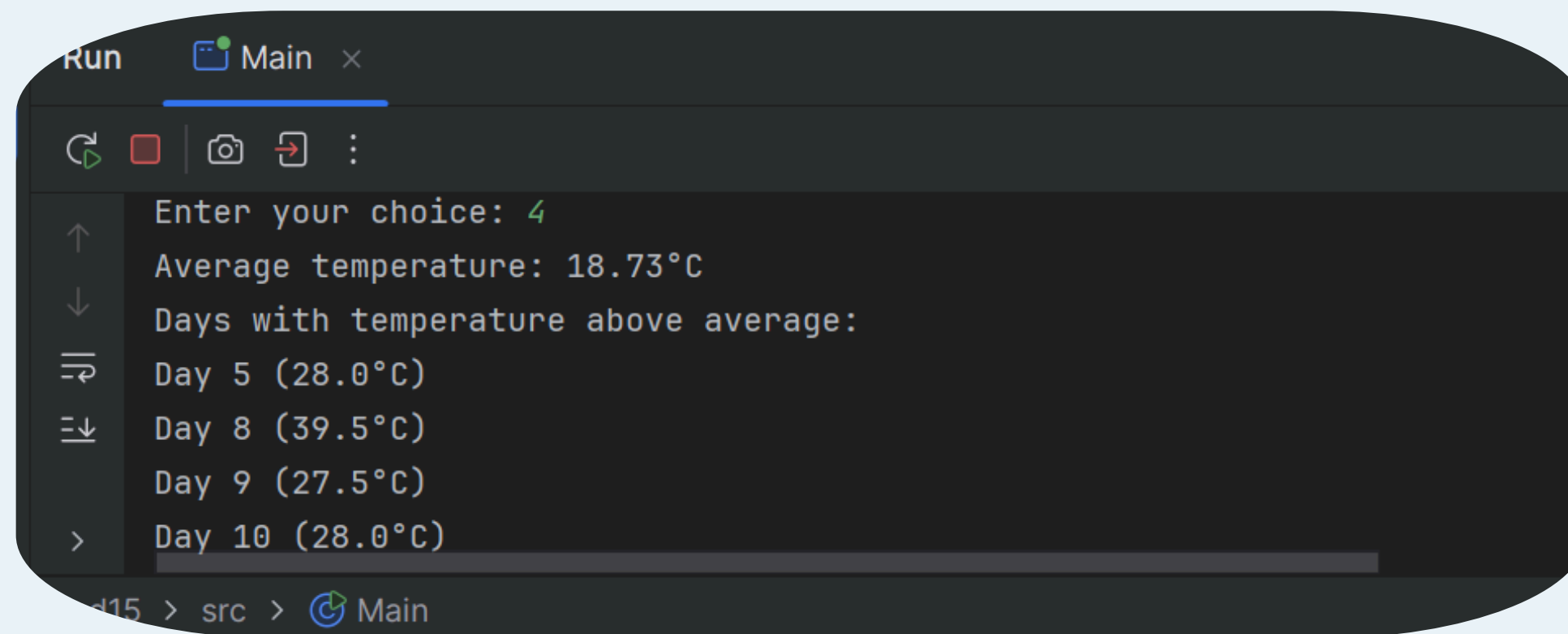
The screenshot shows a code editor window with a dark theme. The title bar at the top says "Run" and "Main x". Below the title bar is a toolbar with icons for running, stopping, and other functions. The main area of the editor displays the following text:

```
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): *****
```

The text is displayed in a monospaced font. The "Enter your choice: 3" line is in green, indicating it was the last line of input. The "Temperature Histogram:" line is in white. The subsequent lines show the temperature for each day and the corresponding number of asterisks representing the histogram bars.

DEMOSTRACIÓN

- Dias por encima de la media



The screenshot shows a terminal window with a dark background. The title bar at the top reads 'Run' and 'Main'. Below the title bar is a toolbar with icons for running, stopping, and other actions. The main area of the terminal displays the following text:

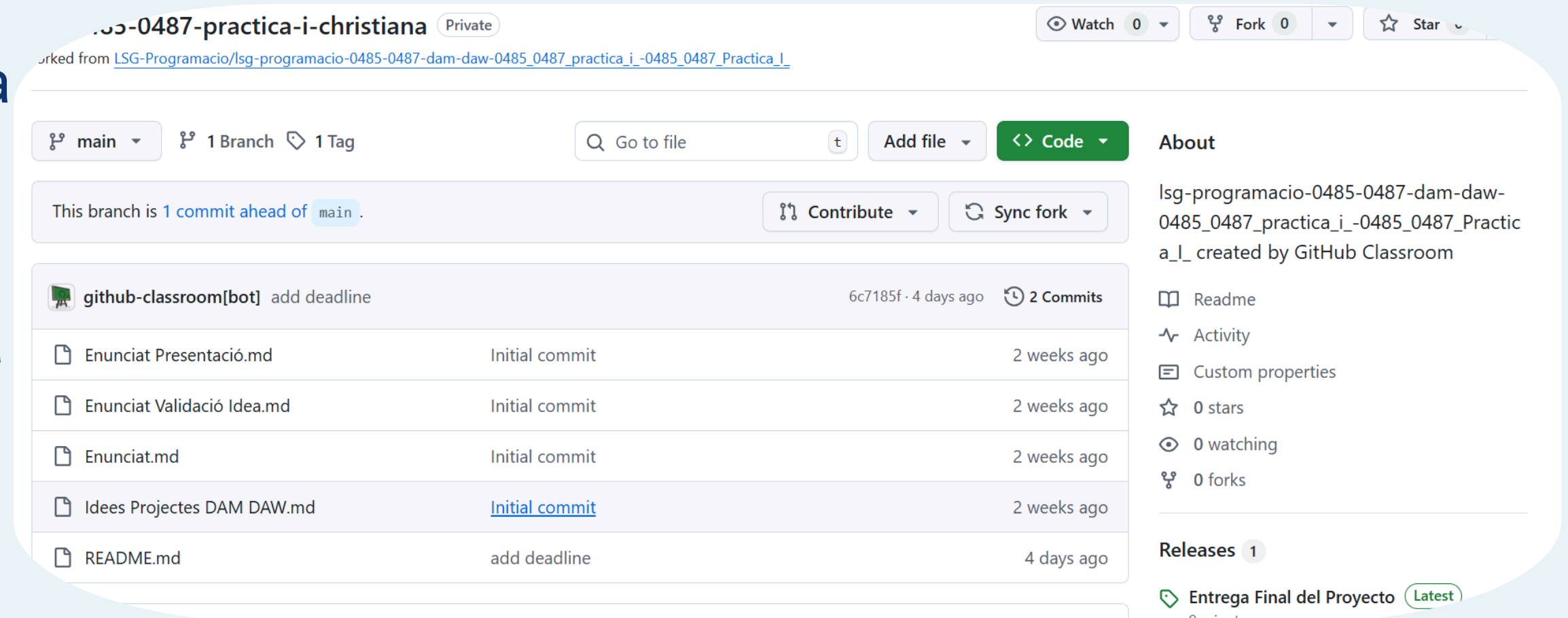
```
Enter your choice: 4
Average temperature: 18.73°C
Days with temperature above average:
Day 5 (28.0°C)
Day 8 (39.5°C)
Day 9 (27.5°C)
Day 10 (28.0°C)
```

At the bottom of the terminal, the path '15 > src > Main' is visible.

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



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**