

Manual Básico de Ejecución

- **Tecnologías Utilizadas**

- Java 17+
- Spring Boot
- JPA / Hibernate
- Base de datos (H2)
- JUnit / Mockito
- Apache JMeter
- .NET Core 8 (Frontend)

Ejercicios 1, 2 y 3 – Ejecución por Consola

Ejecución

1. Abrir el proyecto en el IDE (IntelliJ / Eclipse)
2. Ejecutar la clase principal:

EvaluacionJohnCalleApplication

La ejecución se realiza desde el método main()

Validación en Consola

Al ejecutar el proyecto, en la consola se visualiza:

✓ **Ejercicio 1 – SOMOS / BA**

- Se imprimen los números del 1 al 100
- Reglas:
 - Múltiplos de 3 → SOMOS
 - Múltiplos de 5 → BA
 - Múltiplos de 3 y 5 → SOMOS BA

✓ **Ejercicio 2 – Anagramas**

- Se valida el ejemplo:
- ACUERDO - ECUADOR

- Resultado esperado en consola:
- ACUERDO Y ECUADOR SI SON ANAGRAMAS

✓ Ejercicio 3 – Ordenamiento

- Arreglo original:
[50, 5, 10, 36, 25, 85, 23, 65]
- Resultado ordenado de mayor a menor:
[85, 65, 50, 36, 25, 23, 10, 5]

Al ejecutar el proyecto, los endpoint ya se encontrarán a la espera de peticiones

Tabla de Endpoints – APIs Spring Boot

#	Implementación	Endpoint	Método	Parámetros	Descripción
1	Posts	/posts/importar	<u>POST</u>	N/A	Consume la API externa de posts y guarda la información en la base de datos usando JPA
2	Posts	/posts/titles/{userId}	GET	userId	Retorna la lista de títulos (title) asociados al userId enviado
3	ExternalUsers	/api/external/users	GET	N/A	Consume la API externa de usuarios y devuelve la lista de usuarios mapeados

APIs Externas Consumidas

#	API Externa	Método	Descripción
1	https://jsonplaceholder.typicode.com/posts	GET	Obtiene la lista de posts para ser almacenados en la base de datos
2	https://jsonplaceholder.typicode.com/users	GET	Obtiene la lista de usuarios externos

APIs Consumidas por el Front (.NET Core 8)

#	Endpoint Backend	Método	Uso en el Front
1	/posts/titles/{userId}	GET	Muestra los títulos de los posts por usuario
2	/api/external/users	GET	Lista los usuarios externos

Dependencia Frontend / Backend

Para que el Frontend en .NET Core 8 funcione correctamente, es obligatorio que el backend desarrollado en Java (Spring Boot) esté en ejecución.

El frontend consume directamente los endpoints expuestos por el backend Java, por lo tanto:

- El servicio Spring Boot debe estar levantado
- El puerto configurado (por defecto 8080) debe estar disponible
- Las URLs del backend deben estar correctamente configuradas en el frontend

Flujo de Ejecución Correcto

1. Iniciar Backend Java (Spring Boot)

- Ejecutar la clase principal del proyecto
- Verificar que el servicio esté activo:
- http://localhost:8080

2. Iniciar Frontend .NET Core 8

- Ejecutar localmente o acceder vía IIS
- El frontend consumirá los endpoints del backend Java