**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**“AÑO DEL FORTALECIMIENTO DE LA SOBERANÍA NACIONAL”**

Logotipo

Descripción generada automáticamente con confianza media

**Ingeniería de Sistemas - 202202**

**Open Source Software - E52B**

**Profesor:**

Carlos Alberto Flores Orihuela

**TB-2:**

**Webflux y Programación Reactiva**

**Grupo 03:**

Luigi Aguirre Charco

Wilber Alex Choque Itusaca

Jordy Peralta Ontaneda

Diana Arrieta Jeri

Shessira Cinthya Orellana Rojas

Aymin Cynthya Roman Apolinario

**LIMA – PERÚ**

**2022**

Contenido

[WEBFLUX Y PROGRAMACIÓN REACTIVA 2](#_Toc119266544)

[1. Que es programación Reactiva 2](#_Toc119266545)

[2. Un esquema de cómo funciona 3](#_Toc119266546)

[3. Para qué sirve, beneficios y desventajas 4](#_Toc119266547)

[4. Introducción a WebFlux 4](#_Toc119266548)

[5. ¿Como se aplica? Ejemplo 5](#_Toc119266549)

[6. Caso Práctico en que se puede aplicar, incluir programade código frontend y backend. 9](#_Toc119266550)

[7. ¿Qué es lo que más les ha impresionado o gustado? 16](#_Toc119266551)

[8. Conclusiones y Recomendaciones. 16](#_Toc119266552)

# **WEBFLUX Y PROGRAMACIÓN REACTIVA**

Realice una investigación sobre la Programación Reactiva, el trabajo debe tratar mínimo lo siguiente:

1. **Que es programación Reactiva**

La programación reactiva es un paradigma enfocado en el trabajo con flujos de datos de manera asíncrona. Su concepción y evolución ha ido ligada a la publicación del Reactive Manifestó, que establecía las bases de los sistemas reactivos, los cuales deben ser:

* Responsivos: aseguran la calidad del servicio cumpliendo unos tiempos de respuesta establecidos.
* Resilientes: se mantienen responsivos incluso cuando se enfrentan a situaciones de error.
* Elásticos: se mantienen responsivos incluso ante aumentos en la carga de trabajo.
* Orientados a mensajes: minimizan el acoplamiento entre componentes al establecer interacciones basadas en el intercambio de mensajes de manera asíncrona.

La motivación para el surgimiento de este paradigma procede de la necesidad de responder a las limitaciones de escalado presentes en los modelos de desarrollo actuales, que se caracterizan por su desaprovechamiento del uso de la CPU debido al I/O, el sobreuso de memoria (enormes thread pools) y la ineficiencia de las interacciones bloqueantes.

**Fuente:** Medina, Jaime, Webinar Tecnológico. Desarrollo y Gestión de Proyectos de Software: Programación Reactiva en Java con Spring

<http://hdl.handle.net/10757/651984>

<https://profile.es/blog/que-es-la-programacion-reactiva-una-introduccion/>

1. **Un esquema de cómo funciona**

Es un paradigma de programación que observa flujos de datos y reacciona a eventos. Por ejemplo, podemos hablar de programación reactiva cuando un usuario de YouTube esta suscrito a un canal, y se le envía una notificación sobre cualquier cambio, comentario o nuevo video. Es el siguiente ejemplo, se detalla cómo funciona este paradigma:



**Fuente:**<https://www.techtarget.com/searchapparchitecture/definition/reactive-programming>

1. **Para qué sirve, beneficios y desventajas**

Para qué sirve:

Sirve para tener un desarrollo más fácil. Sirve para manejar mecanismos como: Los suscriptores pueden indicar el número de datos que quieren o pueden procesar, de manera que el notificador nunca les enviará más de n cantidad de elementos. Los notificadores pueden aplicar diferentes operaciones como buffers o descartar algunos datos, para evitar saturar a los suscriptores lentos.

Beneficios:

Uno de los beneficios es que es escalable. Se atiende muchas peticiones con un conjunto limitado de hilos, que son aprovechados al máximo. Tienen una mejor y más predecible escalabilidad y, en general, tardan menos tiempo en arrancar y empezar a servir peticiones. En general, también consumen menos memoria y CPU, al hacer un uso más eficiente. Puede utilizar un número fijo de hilos

Desventajas:

Entre las desventajas se incluye que el modelo de programación se complica bastante y el código es menos legible, más difícil de mantener y depurar. Se requiere un tiempo de aprendizaje y es más complejo de probar. Otro inconveniente es la falta de librerías reactivas, especialmente en tecnologías con más tiempo en el ecosistema, siendo JDBC el mayor exponente.

**Fuente:** https://www2.deloitte.com/es/es/pages/technology/articles/la-programacion-reactiva-en-spring.html

<http://ferestrepoca.github.io/paradigmas-de-rogramacion/reactive/reactive_teoria/index.html>

1. **Introducción a WebFlux**

WebFlux es un framework parte de Spring 5 que está basado en Project Reactor, nos ofrece anotaciones para poder implementar aplicaciones reactivas, también es una alternativa al tradicional Spring MVC ampliamente usado hasta hoy.

WebFlux nos ofrece las siguientes características:

* Non\_blocking thread: los hilos pueden trabajar de manera concurrente sin tener que esperar a la tarea previa.
* Reactive Stream API: Nos ofrece la posibilidad de crear streams asíncronos que no son bloqueantes.
* Asyncronus data processing: Permite procesar la data de manera asíncrona sin interrumpir la ejecución de la aplicación principal.

**Fuente**: <https://www.cleveritgroup.com/blog/programacion-reactiva-con-spring-webflux-parte-2>

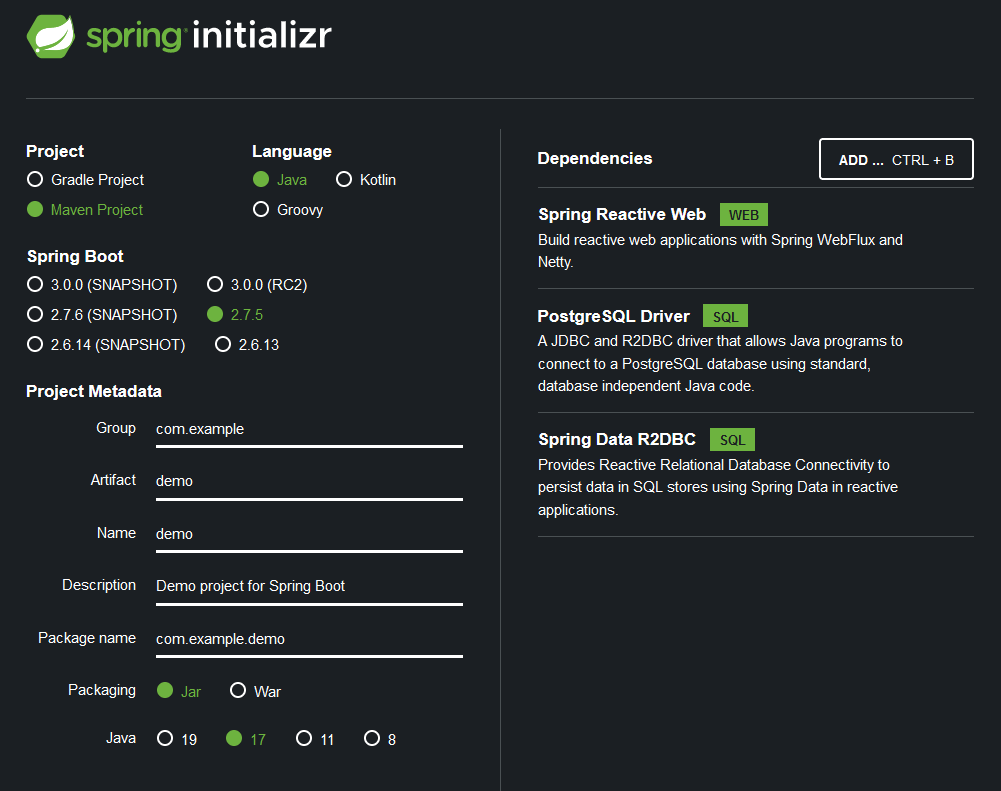
1. **¿Como se aplica? Ejemplo**

Con WebFlux se pueden crear dos formas de implementaciones reactivas:

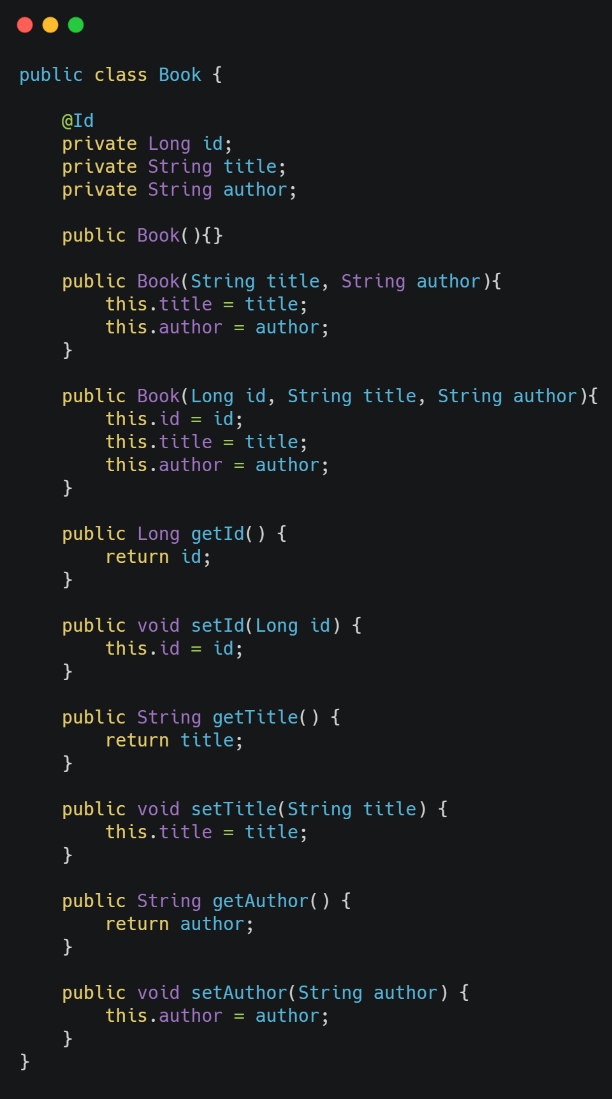
1. Mediante anotaciones de SpringBoot en el controlador muy similar al conocido y clásico Spring MVC, pero ahora haciendo uso de los streams reactivas Flux y Mono.
2. Mediante programación funcional haciendo uso de “functional routing and Handling”.

La gran diferencia está en que la segunda esta activa desde que se inicia la aplicación hasta que esta es finalizada y no requiere decirle a la aplicación mediante anotaciones donde tiene que manejar las solicitudes como se hace usualmente con la anotación @RestController en el clásico Spring MVC.

Para aplicar WebFlux creamos un proyecto y sus dependencias.



1. Mediante anotaciones Sprint boot
   * Entidad



* + Repositorio



* + Service



* + Controller



1. Mediante programación funcional
   * Router



* + Handler



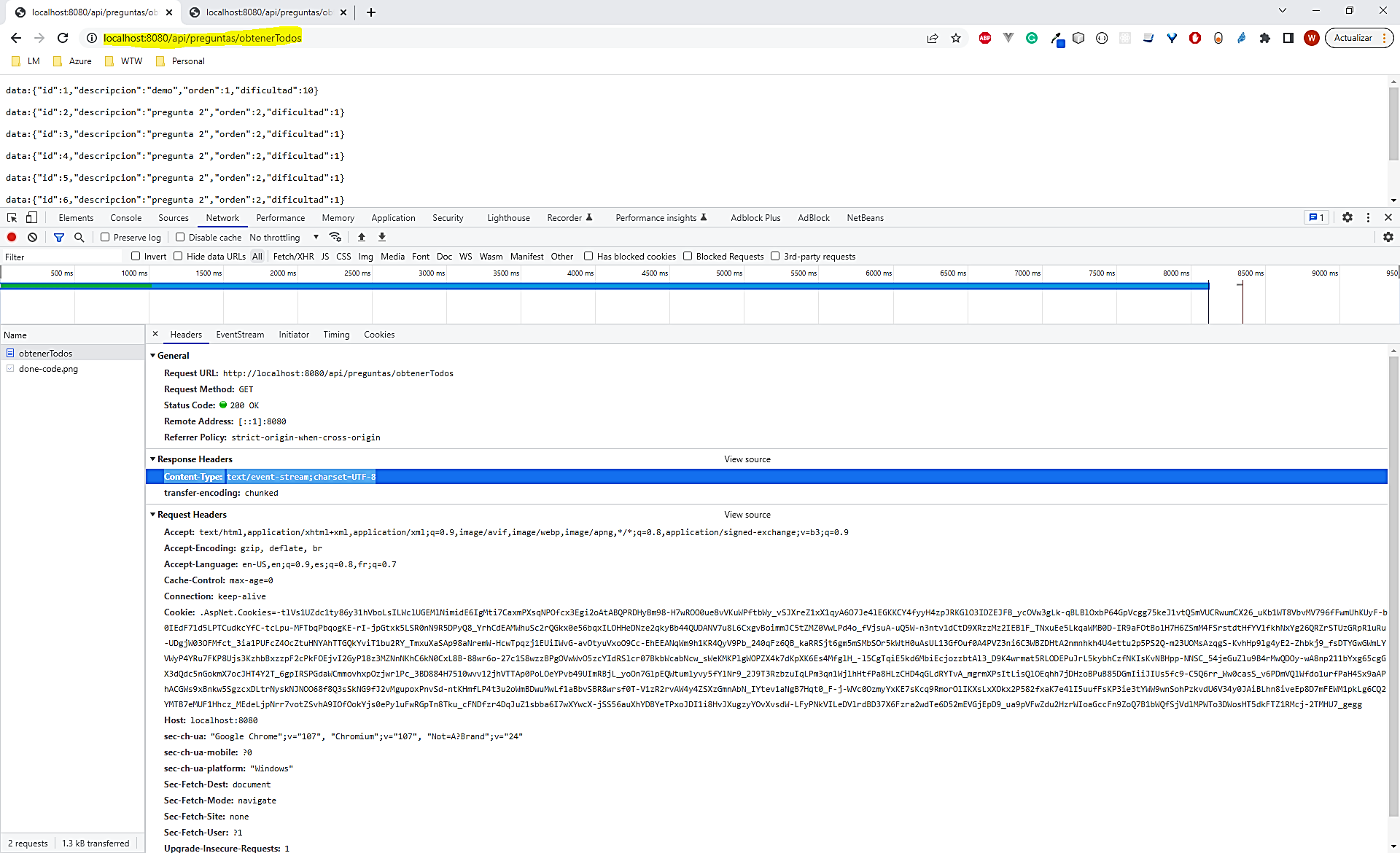
Fuente: <https://www.cleveritgroup.com/blog/programacion-reactiva-con-spring-webflux-parte-2>

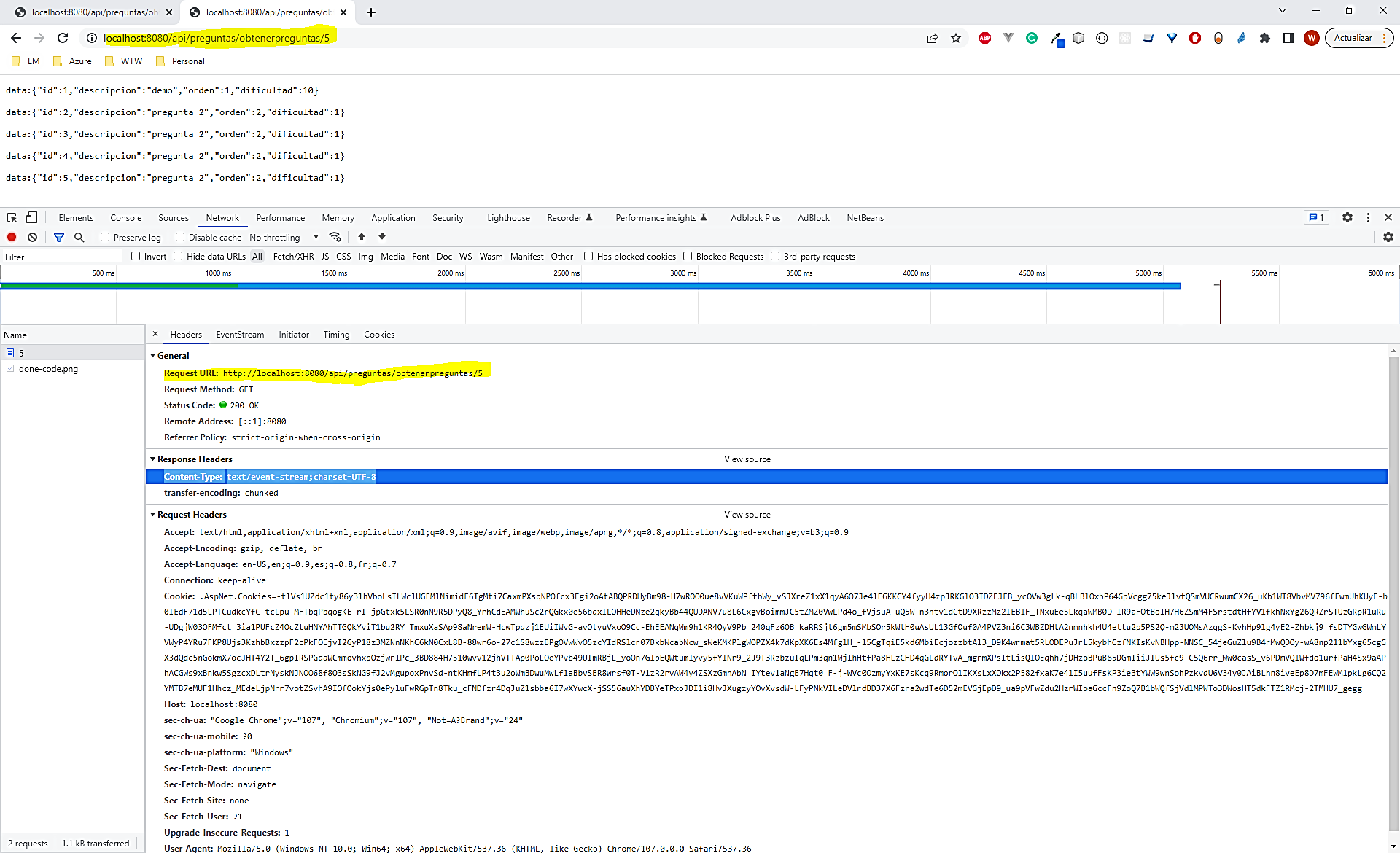
1. **Caso Práctico en que se puede aplicar, incluir programade código frontend y backend.**

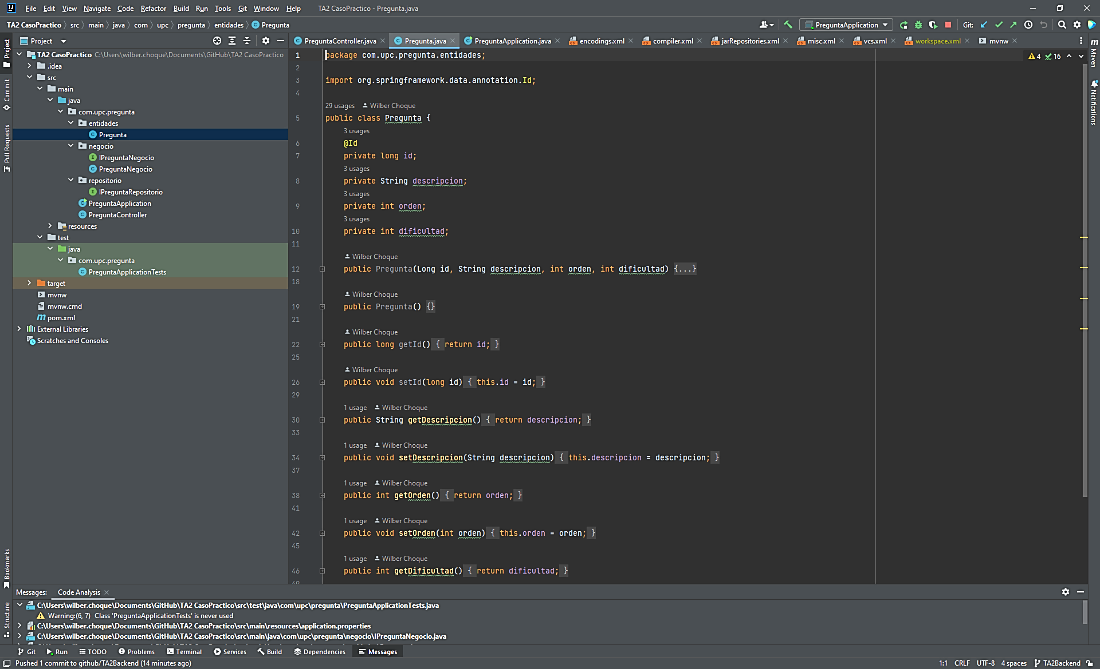
El caso práctico trata de un CRUD de Preguntas, el método principal es el obtener una lista de preguntas con límite para autogenerar un examen con una cantidad de preguntas definida por el usuario.

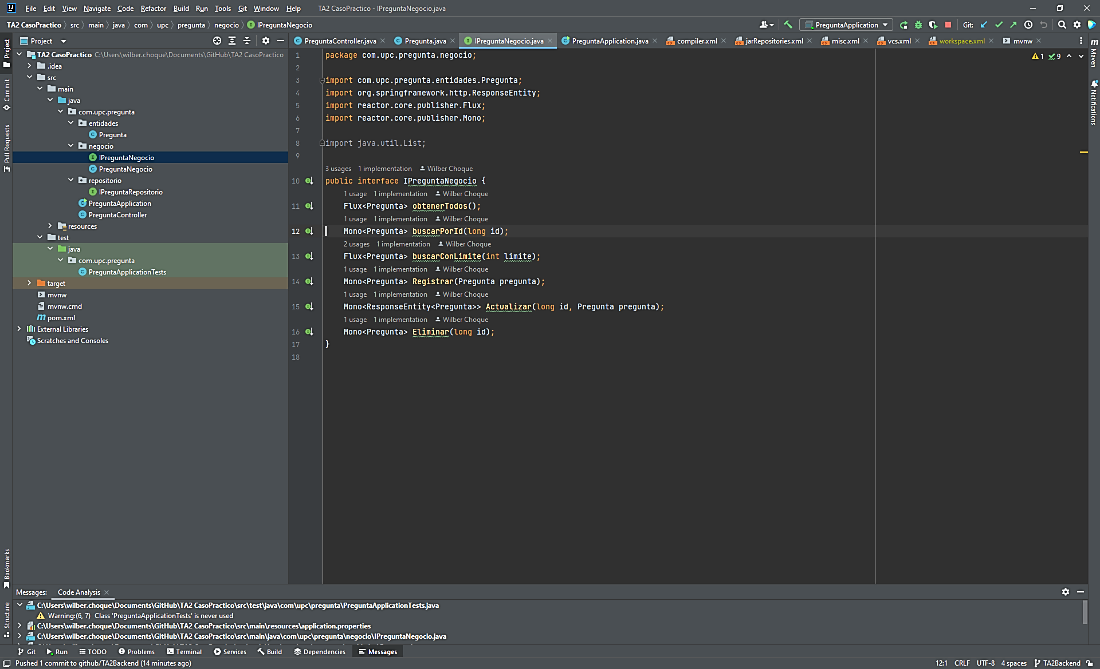
**Fuente:** https://upcedupe.sharepoint.com/:u:/s/OpenSourceSoftware-Grupo3/Edl1N7JNuDROnhzBHkragUQBfIz\_362or0R4QDqcbJuohQ?e=F1HErg

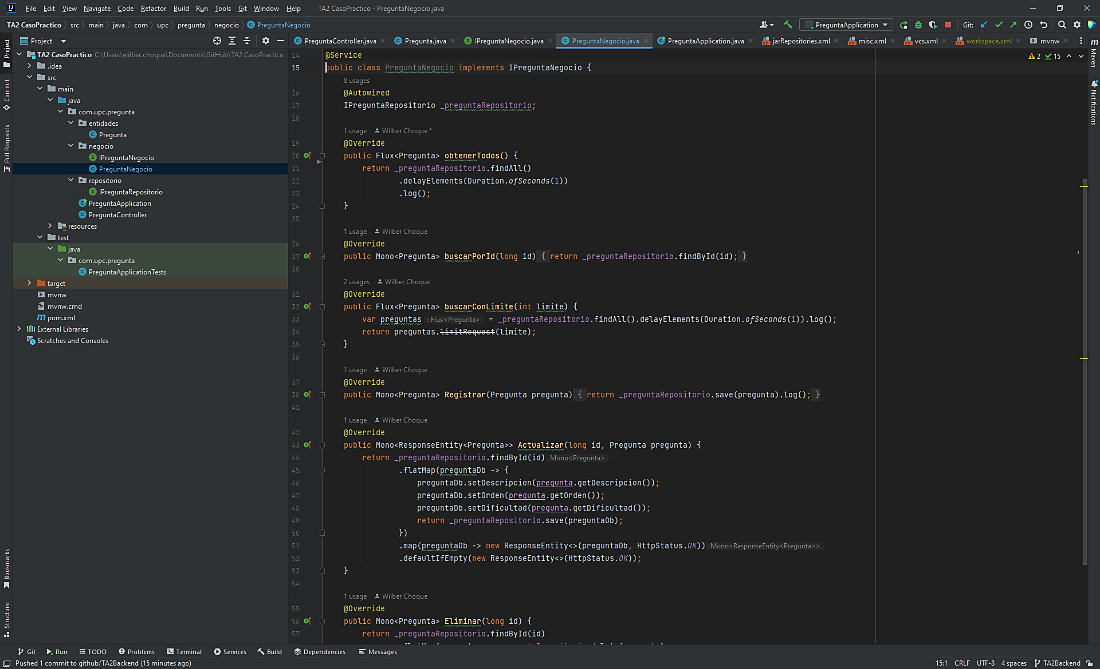
Backend: <https://github.com/wchoque/TA2_CasoPractico>



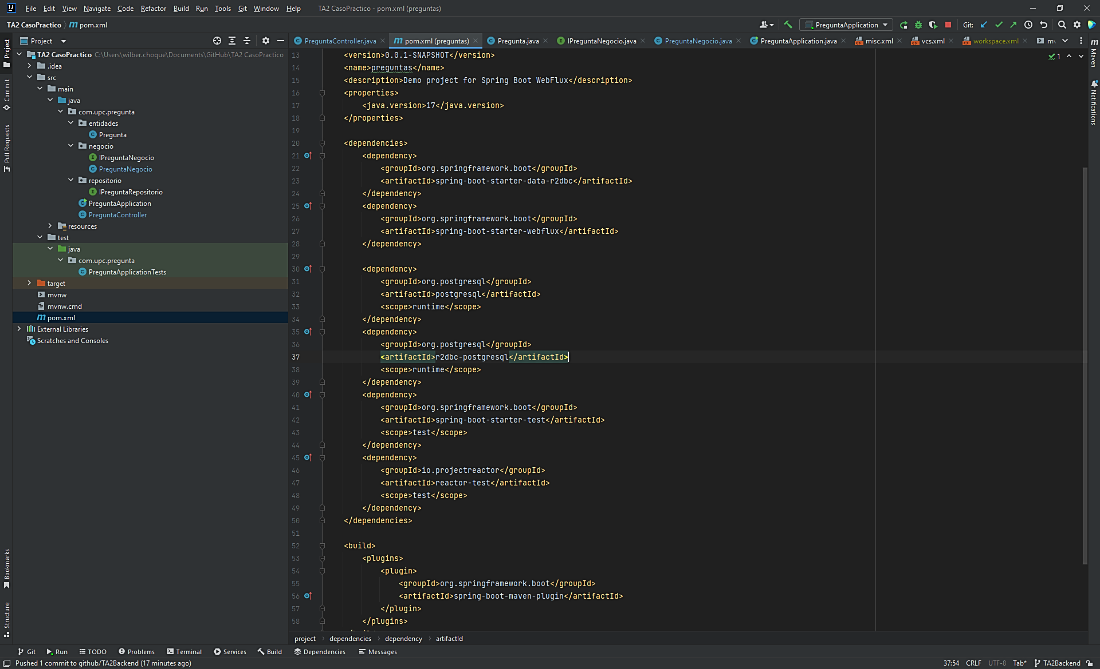




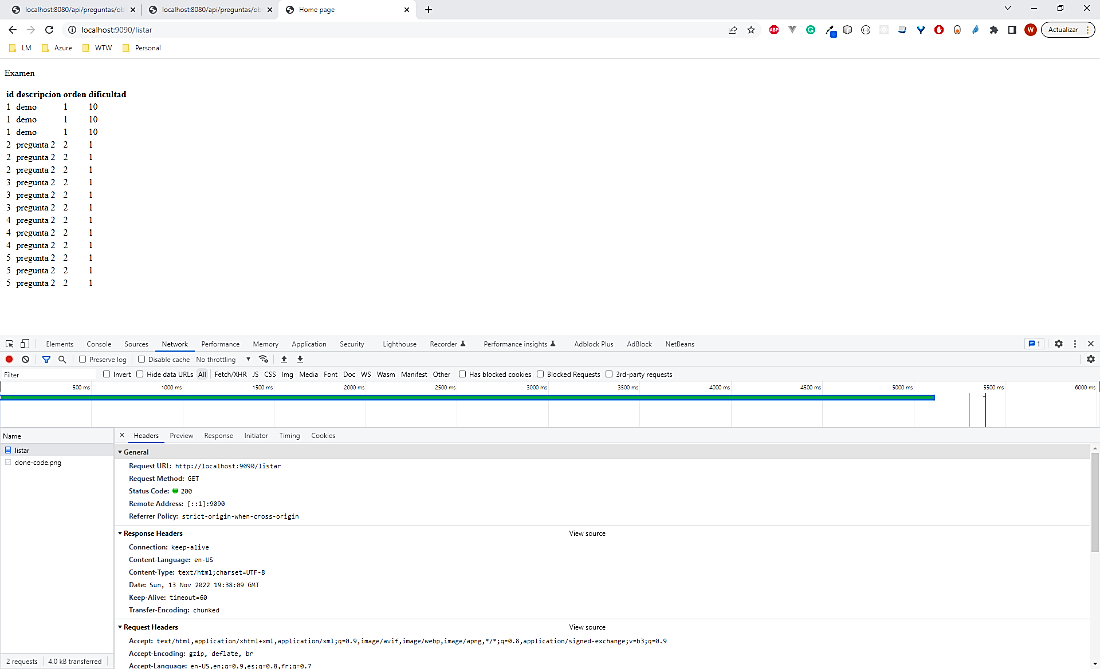


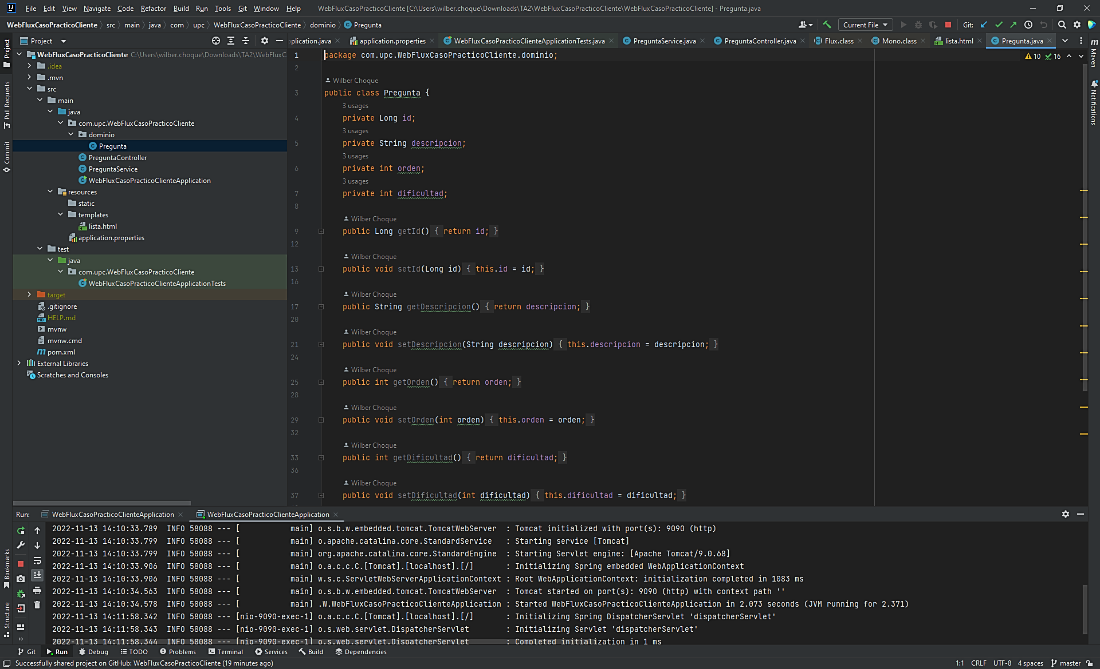


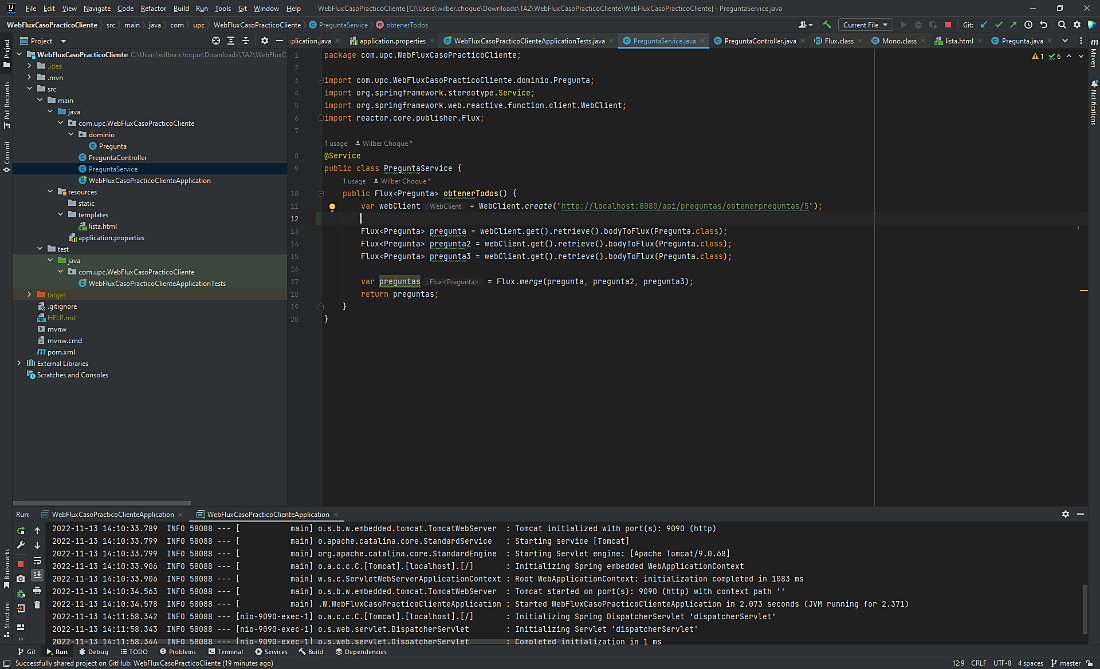


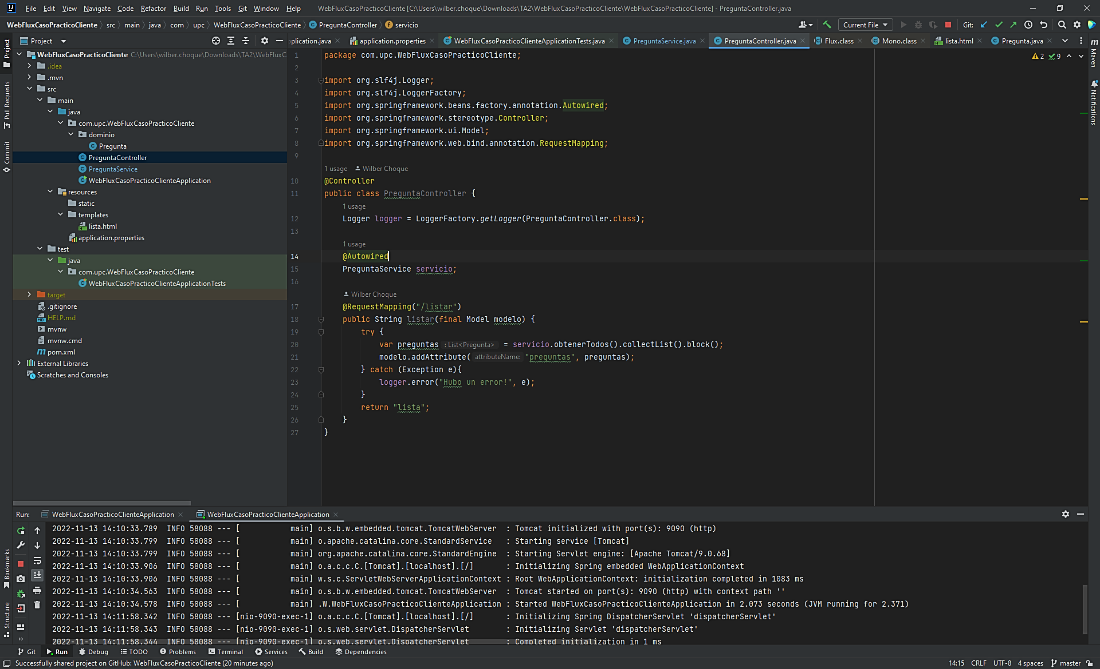


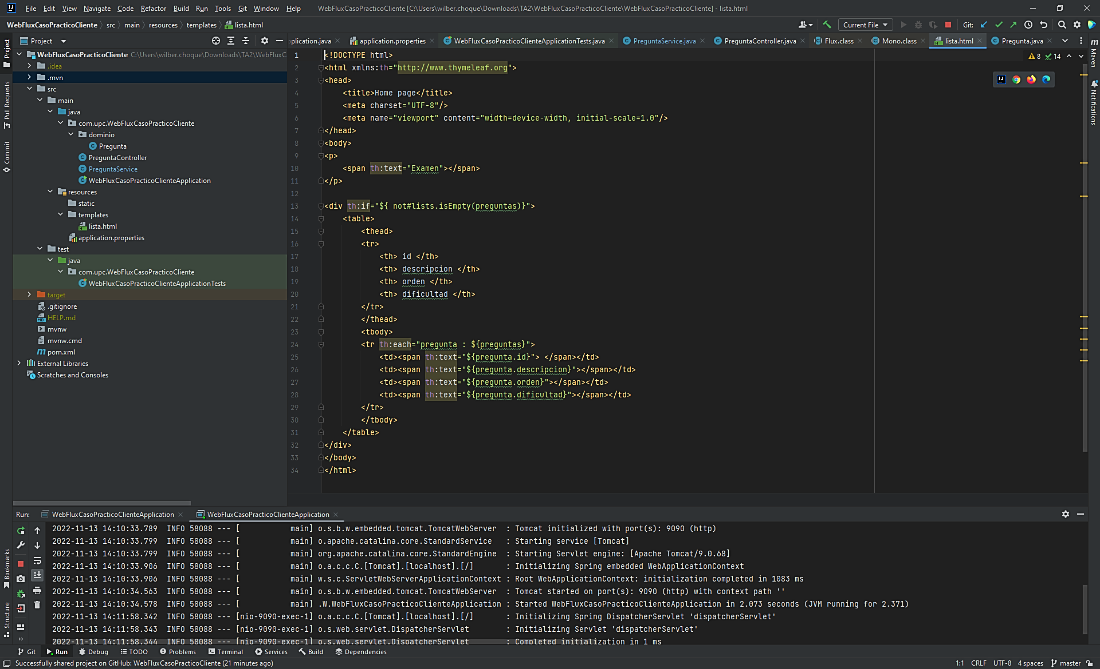
**Frontend:** <https://github.com/wchoque/WebFluxCasoPracticoCliente>

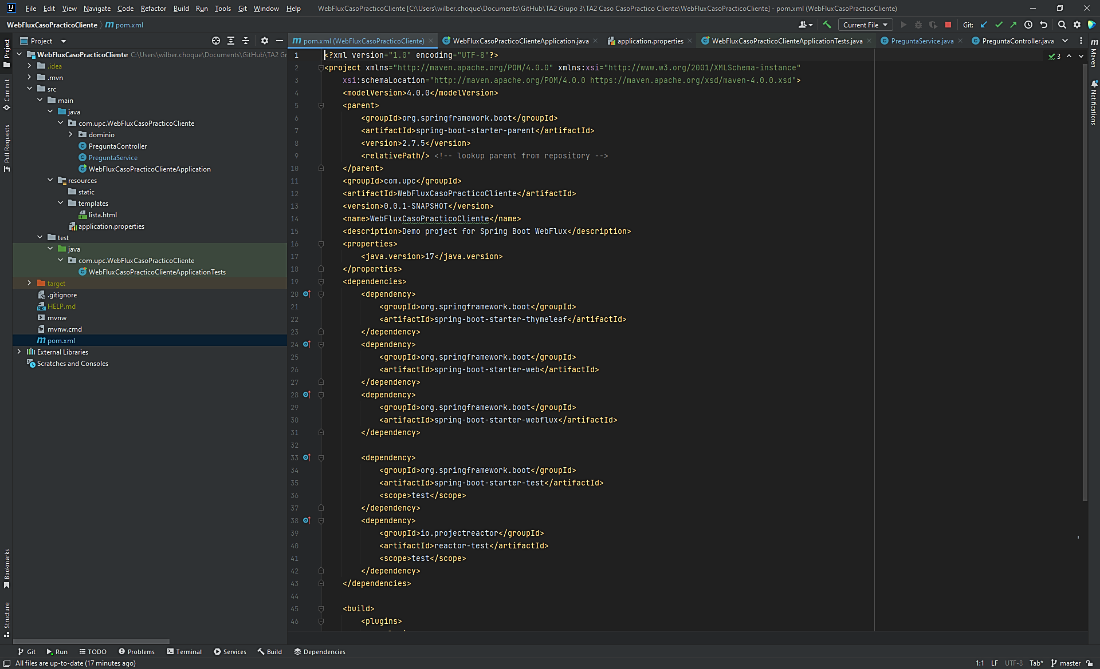












1. **¿Qué es lo que más les ha impresionado o gustado?**

La forma de pensar no es tan imperativa, sino es más funcional y que cada función origina un flujo de datos. (U201924447)

Lo que me ha gustado son las ventajas que se obtiene con la programación reactiva ya que se puede realizar más con menos. (U202112833)

Lo que más me ha impresionado de la programación reactiva es que es posible generarla al crear componentes individuales pero que al integrarse trabajan armoniosamente en conjunto. Además de ellos, programar este tipo de aplicaciones es un reto dado que exige crear módulos flexibles y dinámicos por lo tanto la programación se torna funcional, es decir matemática pura. (U201820727)

1. **Conclusiones y Recomendaciones.**

En Programación Reactiva, en lugar de ir nosotros activamente a buscar los datos, existen unos flujos de datos a los que nosotros reaccionamos cuando recibimos nueva información. (U201924447)

En la programación asíncrona clásica, un hilo que se ejecuta pasa la mayor parte de su tiempo esperando. El hilo que se ejecuta envía una petición, por ejemplo, a la base de datos, luego espera durante un periodo de tiempo la respuesta con los datos, cuando finalmente obtiene una respuesta, continúa trabajando y al finalizar vuelve a estar disponible para otra ejecución. Lo que hace que este recurso se esté usando de forma ineficiente. Aprovechemos esta ventaja de la programación reactiva. (U202112833)

La programación reactiva es capaz de convertir cualquier cosa en un flujo para administrar de forma asíncrona. Además, facilita la tarea de manejo de errores y hace que el código sea más flexible, legible, compatible y fácil de escribir. (U201820727)