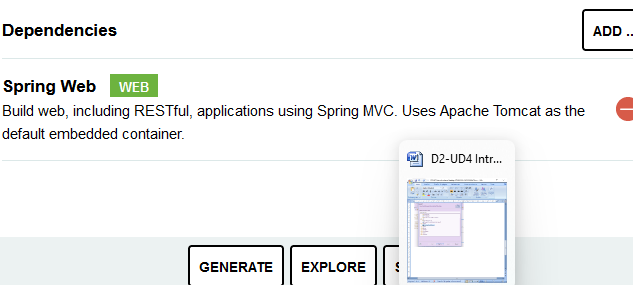
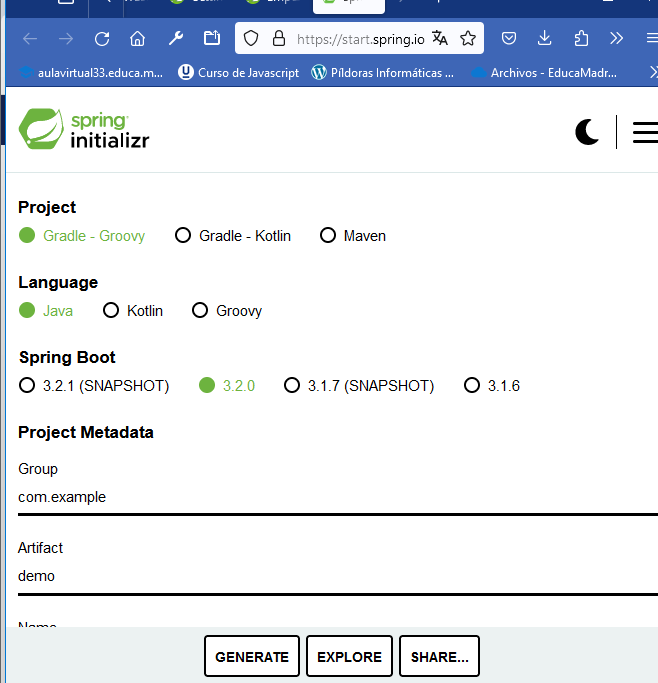
**Tarea 2 – UD4**

**Introducción al testing**

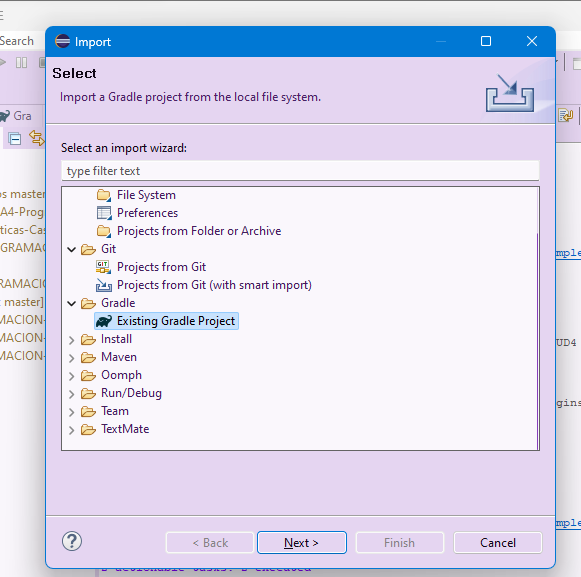
**TUTORIAL:** [**https://spring.io/guides/gs/testing-web/**](https://spring.io/guides/gs/testing-web/)

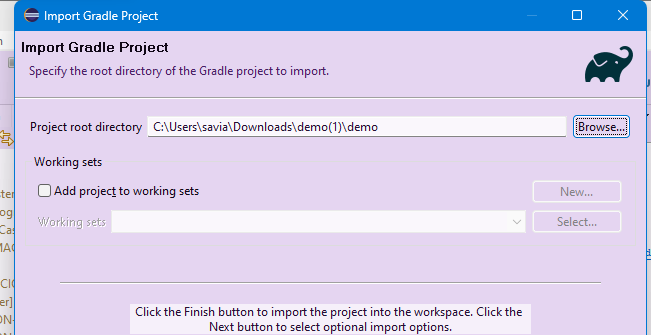
Voy a usar Eclipse.

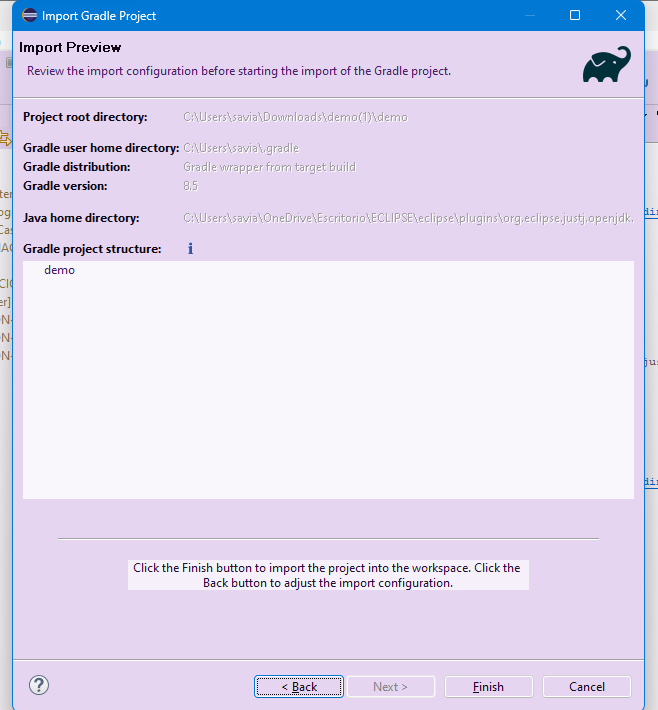
Genero zip de proyecto gradle, seleccionando SpringWeb en dependencias y java como lenguaje:

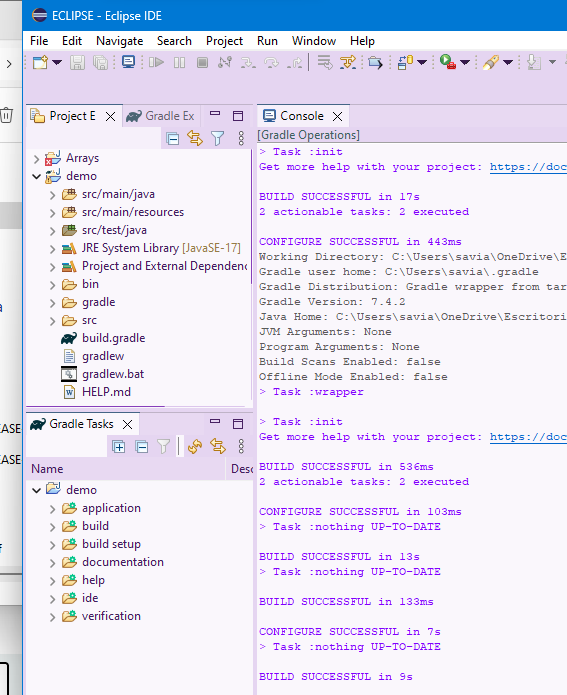


Y lo importo en Eclipse:

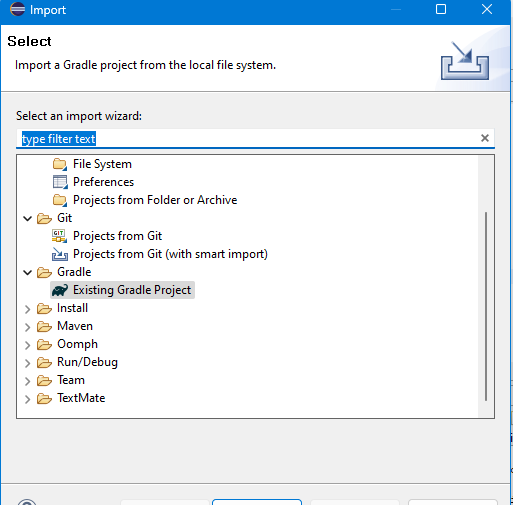


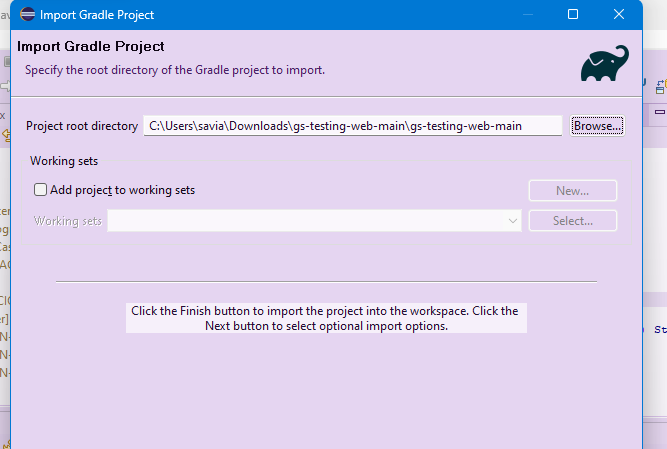




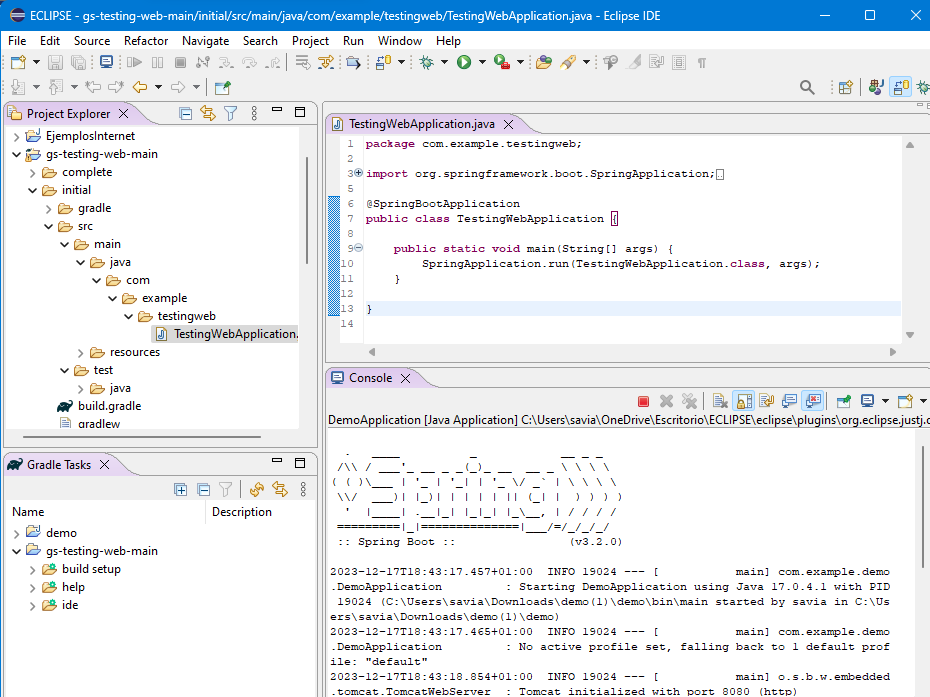


Descargo el repositorio de la guía y lo importo como Gradle Project:



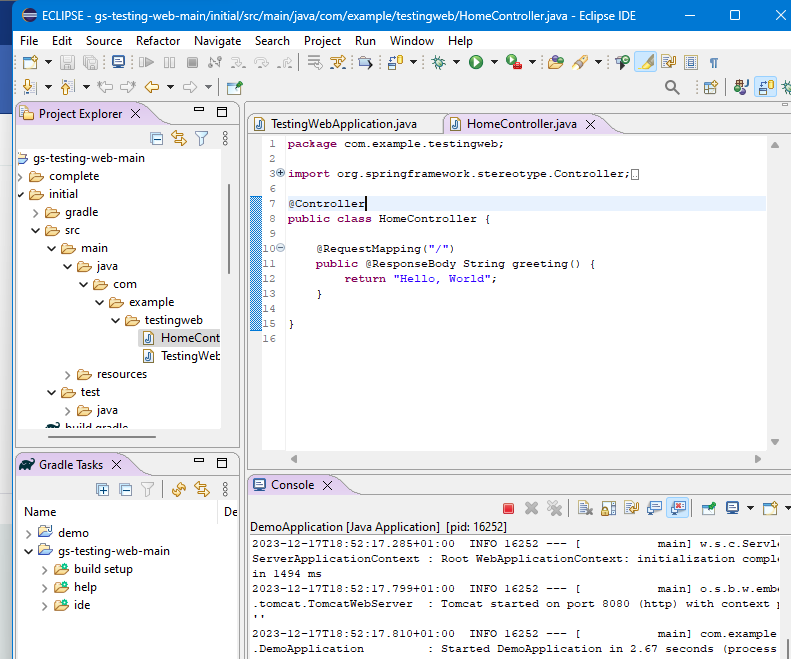


Y comprobamos que se ha importado bien y que se ejecuta:

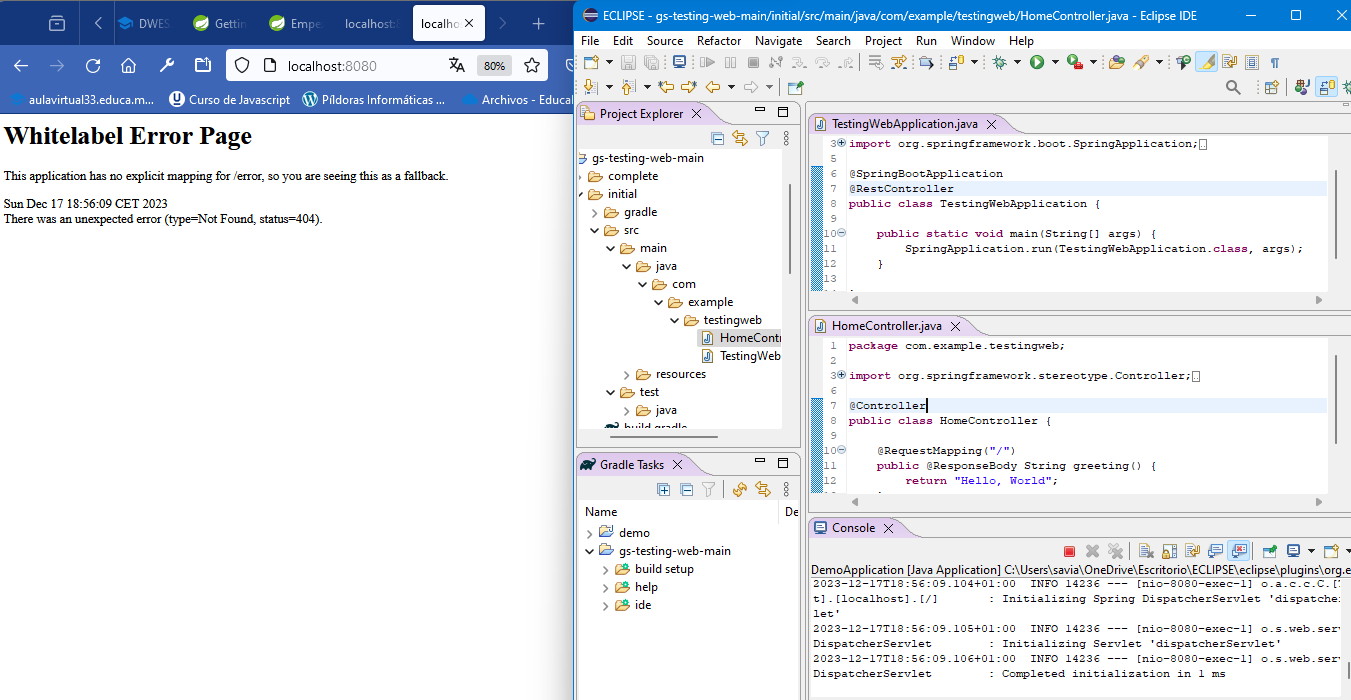


**Crear una aplicación simple:**

Creamos nuevo controlador en la clase HomeController:

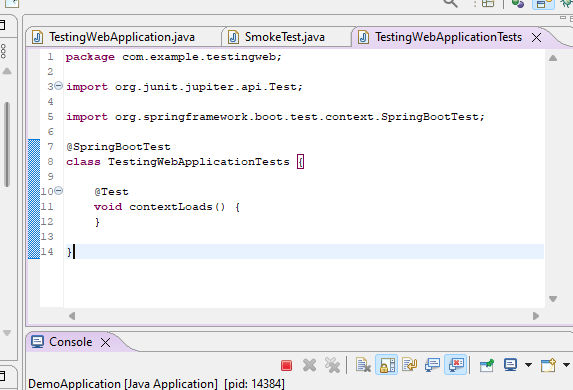


Ahora probamos a carga la página de inicio en el navegador, pero no sale nada, aunque en la consola de eclipse no sale ningún error:



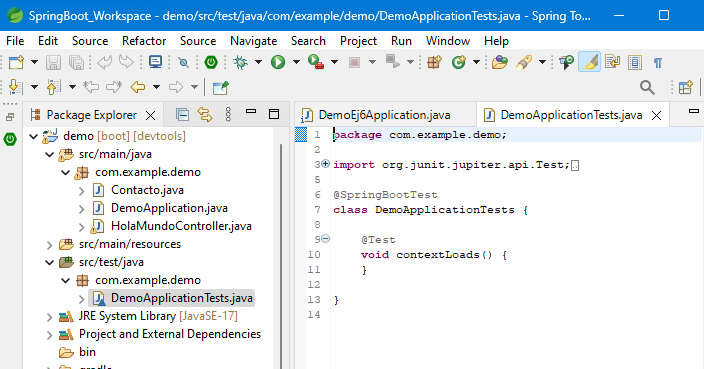
Aún así, sigo con el tutorial:

Voy a automizar la prueba. Escribo una prueba de control en TestingWebApplicationTest.java:



Aquí empecé a tener problemas por lo que voy al paso avanzado usando el código del desafío anterior:

1. **Salta el tutorial hasta el punto “Run the application” pues la parte anterior es lo ya hecho en el desafío anterior.**
2. **Mira si el proyecto ha creado algo en src/test/java. Si estuviera una clase tipo DemoApplicationTests o similar creada automáticamente en tu proyecto. Aprovecha para leer el significado de las anotaciones** **en el tutorial, aunque algunas todavía no las veas.** Así es, hay un archivo, lo abro y veo sus anotaciones:



* @SpringBootTest
  + indica que esta clase debe cargar la configuración de la aplicación Spring Boot.
  + Cuando ejecutas pruebas con @SpringBootTest, Spring Boot configura el contexto de la aplicación y carga todas las configuraciones necesarias para ejecutar las pruebas.
* class DemoApplicationTests:
  + Esta clase representa la suite de pruebas para tu aplicación.
  + Las pruebas específicas se escribirán como métodos dentro de esta clase.
* @Test:
  + indica que el método siguiente es una prueba.
  + El nombre del método sugiere que la prueba verifica si el contexto de la aplicación se carga correctamente.
* void contextLoads() { }:
  + cuerpo de la prueba. El método está vacío, lo que significa que no contiene ninguna lógica de prueba específica.
  + La presencia de este método generalmente se debe a que, en la fase de desarrollo inicial, simplemente se verifica que el contexto de la aplicación se cargue correctamente.

Cuando ejecutas esta prueba (puedes hacerlo desde tu IDE o utilizando herramientas de construcción como Maven o Gradle), Spring Boot cargará el contexto de la aplicación y verificará que no haya errores de configuración. La prueba contextLoads suele ser la primera prueba que se ejecuta para asegurarse de que la aplicación se esté configurando correctamente.

Puedes agregar pruebas adicionales a esta clase a medida que desarrollas tu aplicación para verificar el comportamiento específico de los componentes y servicios

1. **En el apartado “Test the application” compara si el primer código (TestingWebApplicacionTest) a parte del nombre, su contenido es igual o no a la clase del punto anterior o en caso contrario créala. Con un clic derecho encima de la clase dale a Run → Gradle Test y observa el resultado. Simplemente comprueba que la aplicación se levanta. Introduce un error en la aplicación (ej: duplicar un punto de acceso web) y vuelve a correr el test ¿cambia algo?**