***los links oficiales***

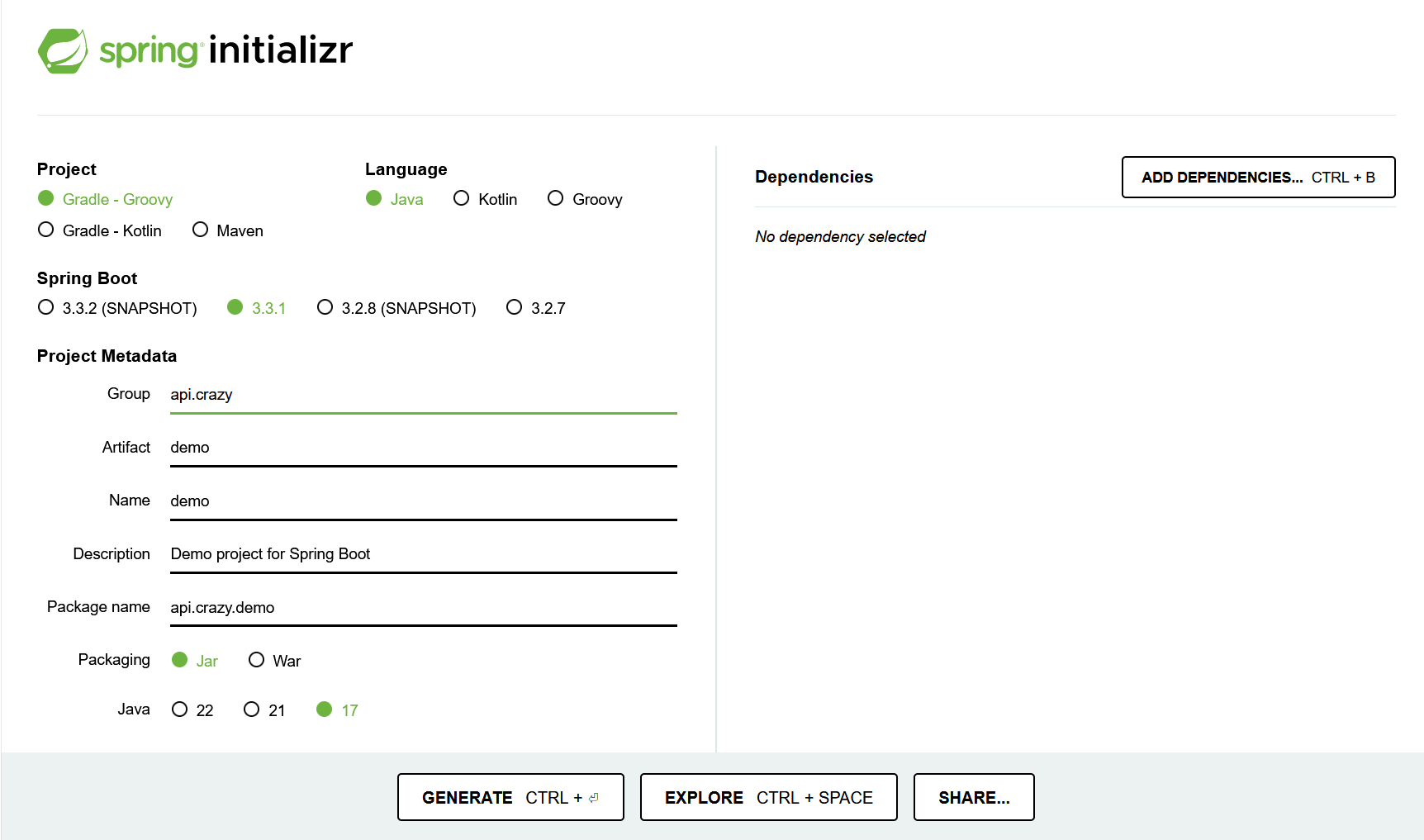
**Descarga Spring Boot:**

<https://start.spring.io/>

**Documentación oficial:**

<https://spring.io/guides/gs/spring-boot>

Configuración del proyecto de java (2024)



Que es gradle

Es un automatizador de código realiza la creación, pruebas, publicación, inclusión de librerías y despliegue del código que nosotros vamos a escribir.

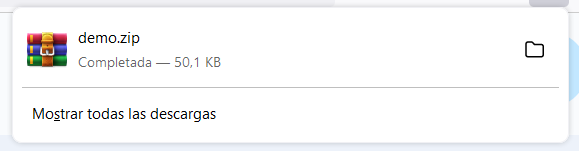
Tiene varias funciones pero lo que nosotros vamos a utilizar es el gestor

…

Vamos a descargar gradle

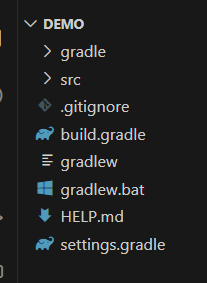
…

Al configurar nuestro proyecto vamos a hacer click en generate y nos descargará un comprimido.



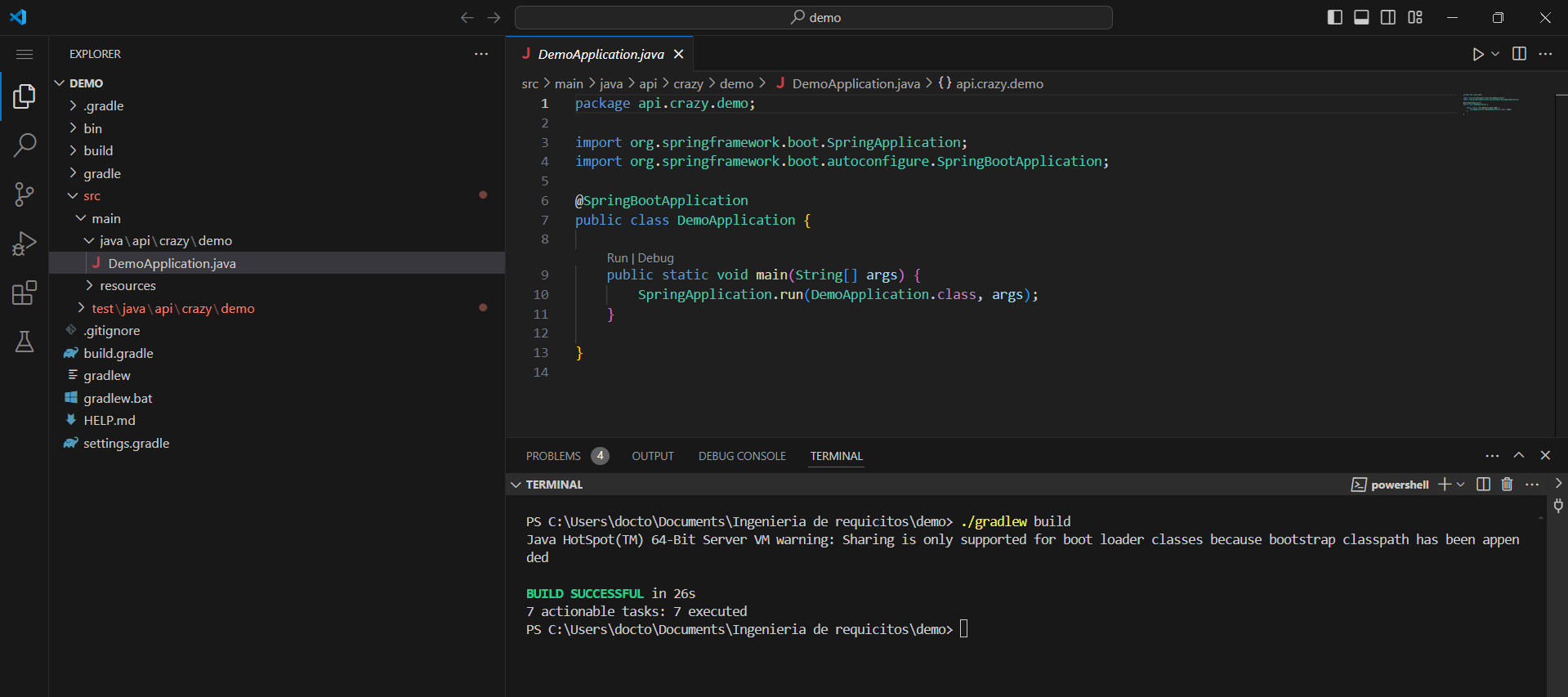
Una vez descomprimido lo abriremos con visual studio code

Y veremos la siguiente estructura de carpetas



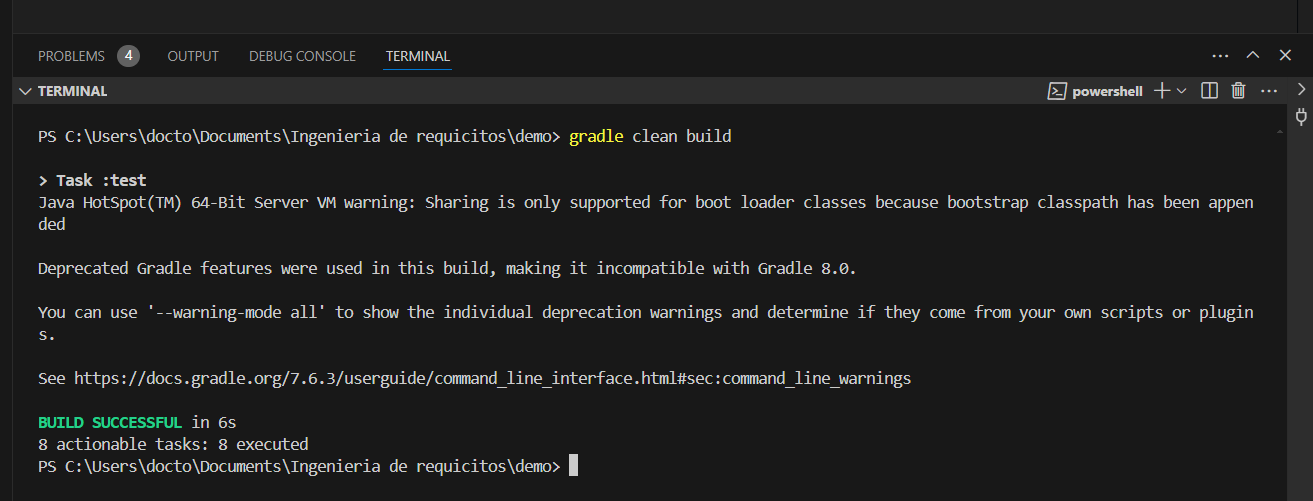
Vamos a compilar el proyecto con el siguiente comando (tardará varios minutos) esté comando va a generar todo el código base.

./gradlew build



Ahora vamos a re - compilar el aplicativo para ver como funciona con el siguiente comando:

gradle clean build



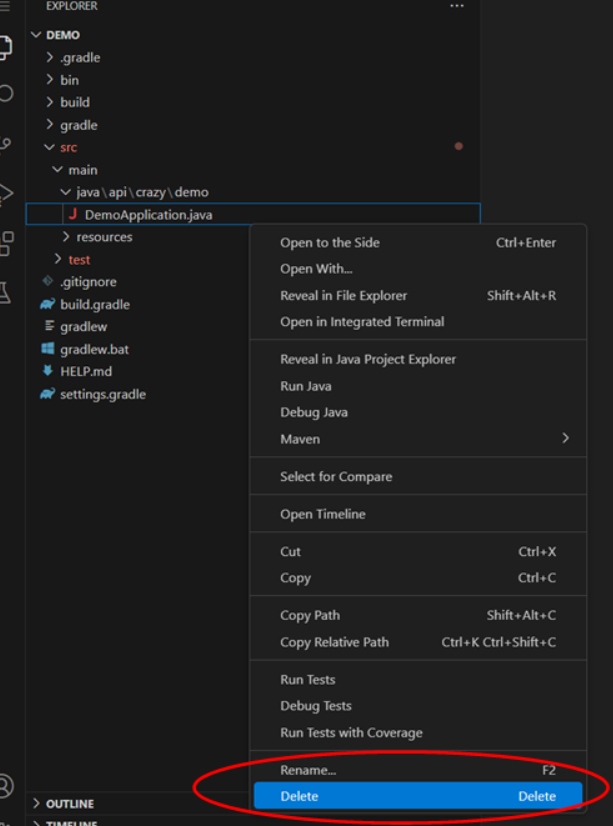
No es imposible correr el proyecto debido a que no tenemos código que ejecutar. Vamos a crear:

* Una aplicación
* Un controlador web

***NOTA: El código lo encontramos en la documentación oficial.***

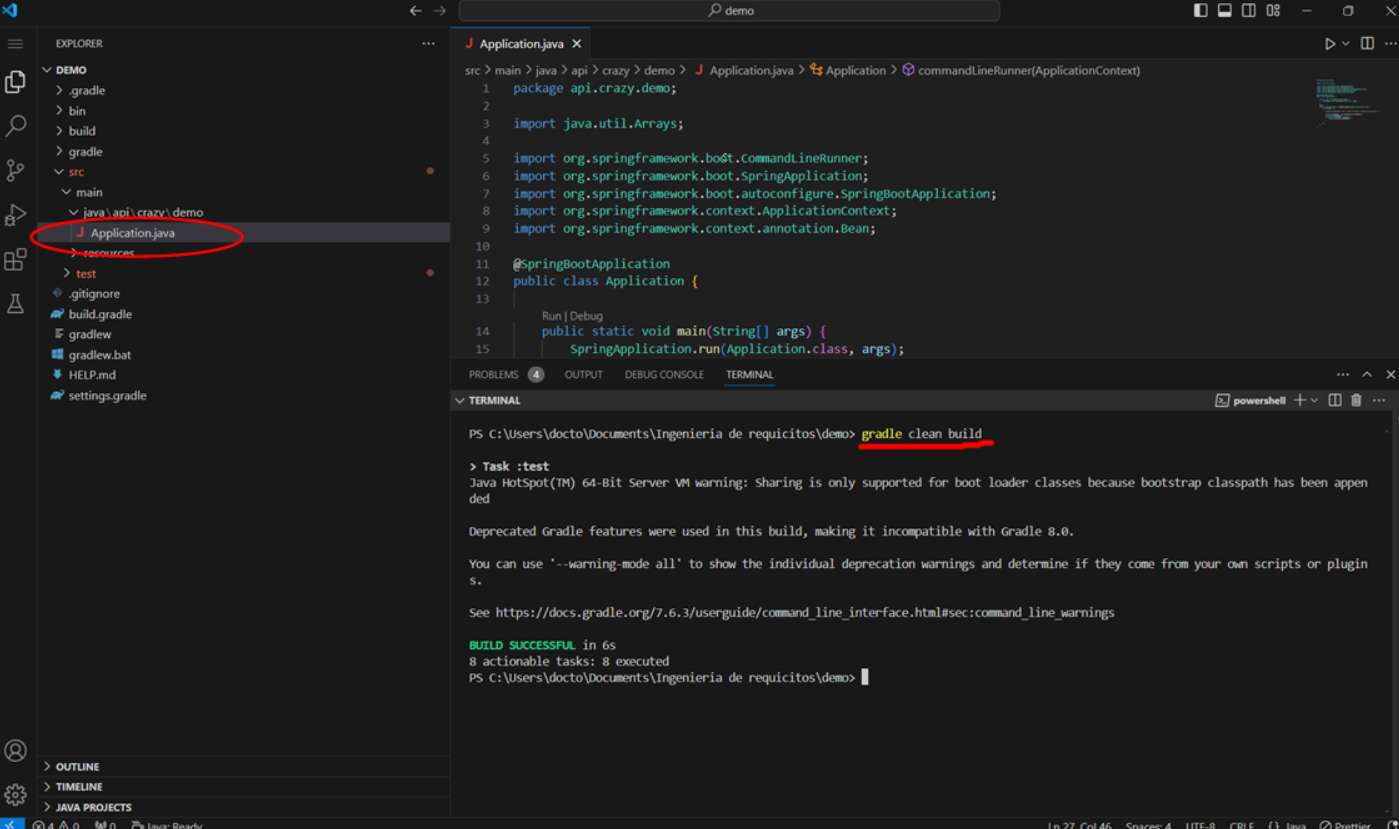
Vamos a crear la aplicación

1 - vamos a borrar el antiguo DemoApplication.java



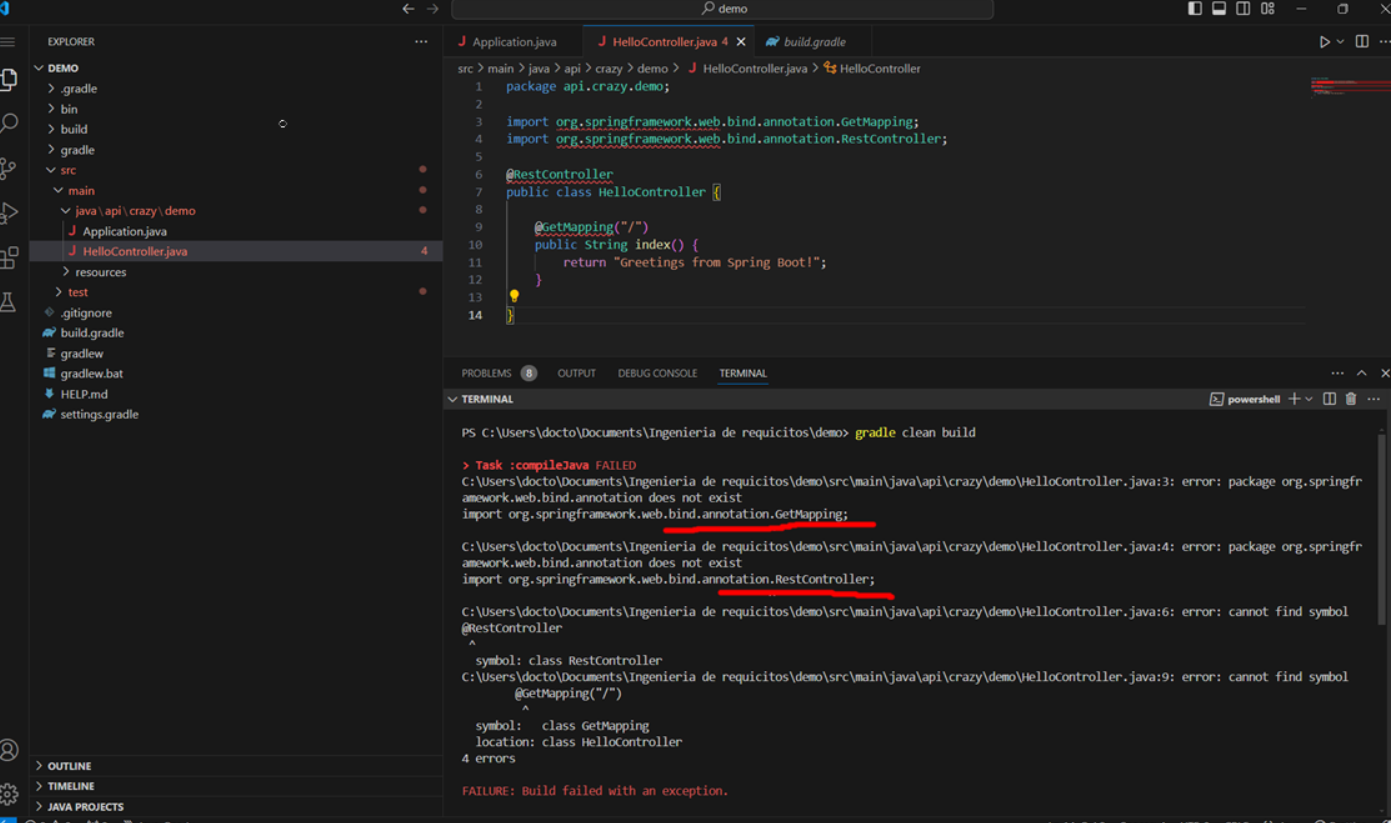
Y vamos a crear nuestra App real y luego a compilar.

demo\src\main\java\api\crazy\demo\Application.java



Un controlador WEB de ejemplo que nos va a exponer un API.

demo\src\main\java\api\crazy\demo\HelloController.java



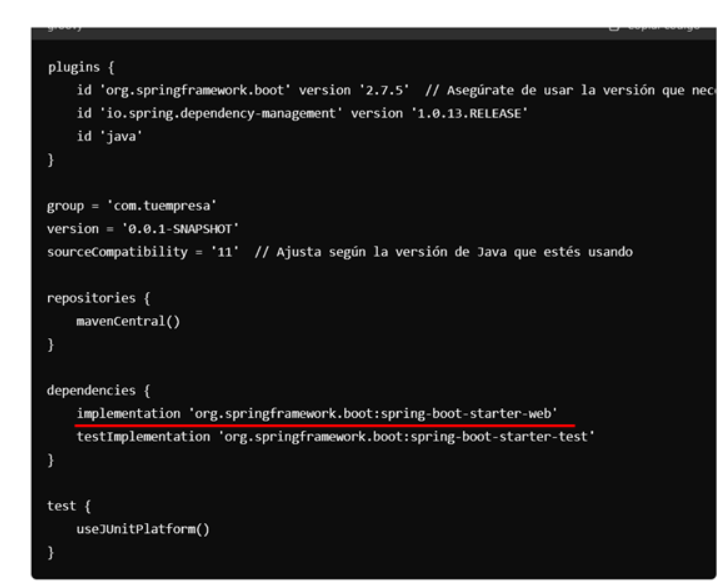
Si nosotros procedemos a compilar vamos a ver 2 errores en las librerías de GetMapping y RestController y eso pasa porque no las hemos instalado.

Para instalar una librería en Gradle solo basta con agregarla al archivo: “build.gradle”

Para saber como se instala se recomienda preguntar a CHAT GPT

PROMT:

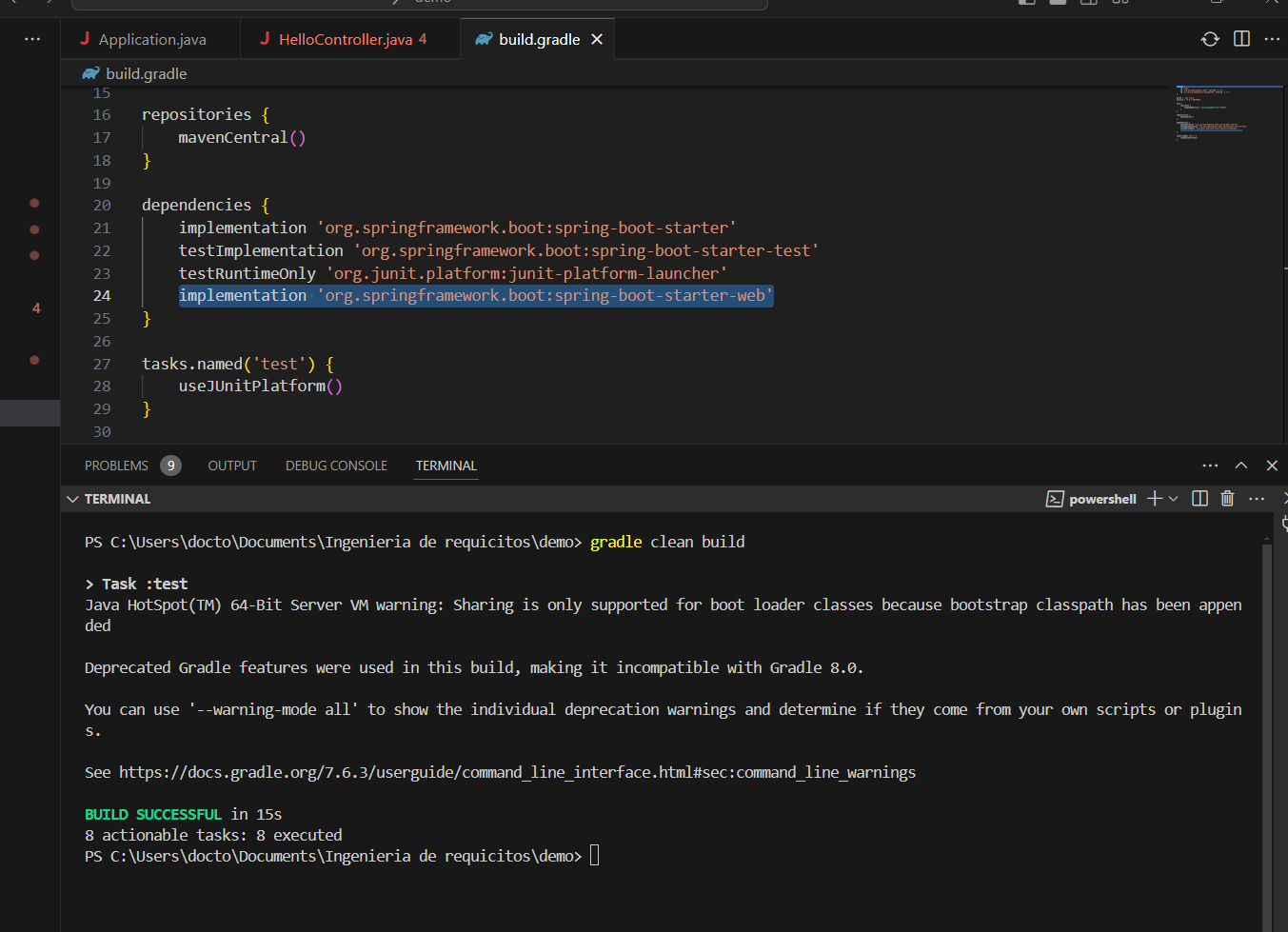
“Ayuda con java en el framework spring boot. como instalar las librerias GetMapping y RestController en gradle.build”



En mi caso solo me falta una dependencia:

implementation 'org.springframework.boot:spring-boot-starter-web'

Y una vez agregamos la dependencia al archivo “gradle.build” vamos a volver a compilar “gradle clean build”



Y para correr el código:

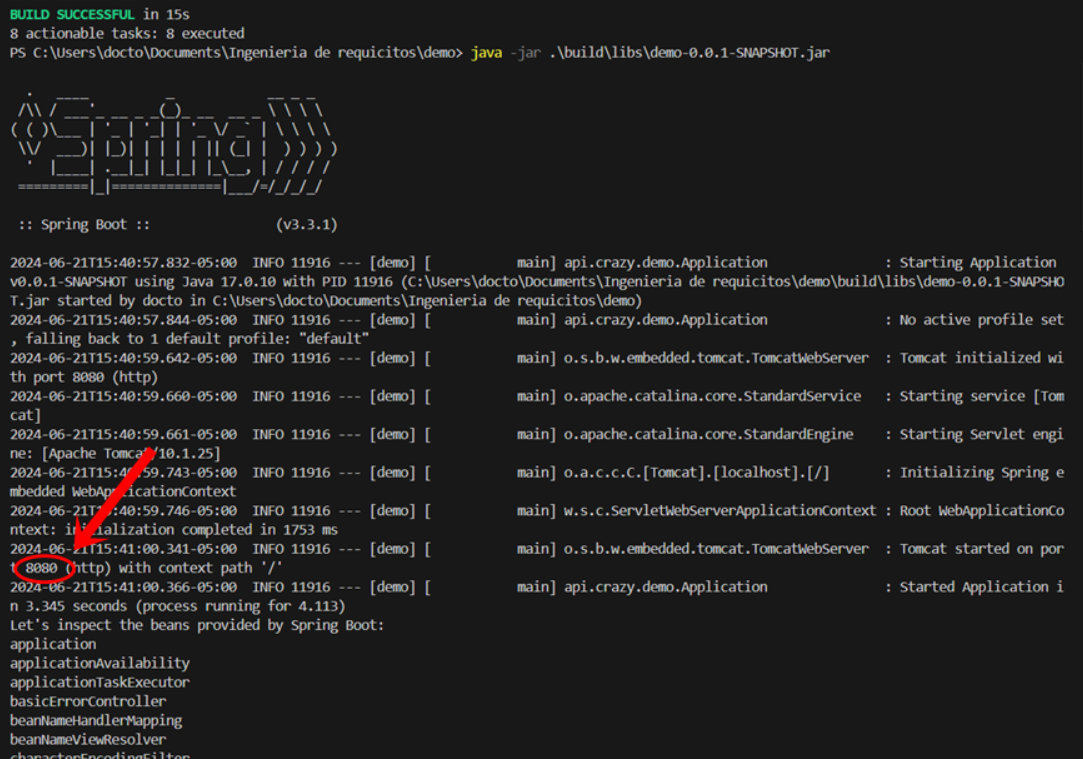
Vamos a ejecutar el archivo que se compiló lo podremos encontrar en:

demo\build\libs\demo-0.0.1-SNAPSHOT.jar

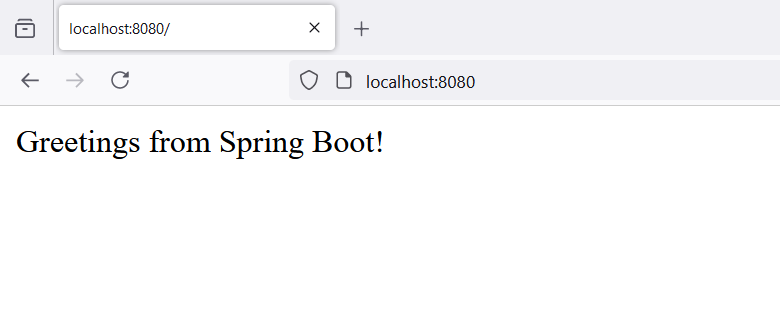
y ese archivo lo vamos a compilar con el siguiente comando:

java -jar .\build\libs\demo-0.0.1-SNAPSHOT.jar

Y vamos a ver un texto muy largo y en dicho texto vamos a encontrar el puerto por el cual se está ejecutando el proyecto



Y ahora vamos al navegador a la dirección: <http://localhost:8080/>

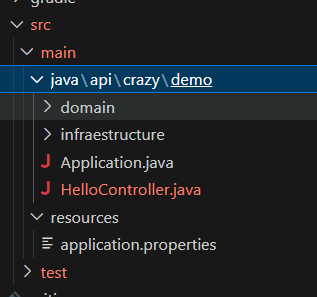


FELICITACIONES … COMPLETAMOS EL HOLA MUNDO EN SPRING BOOT

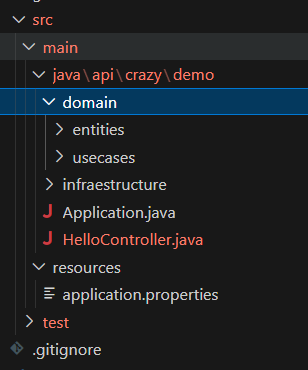
Ahora vamos a ponernos serios

Vamos a crear capas:

* Infraestructura: vamos a colocar los entry points.
* Dominio: vamos a poner los objetos y los casos de uso.

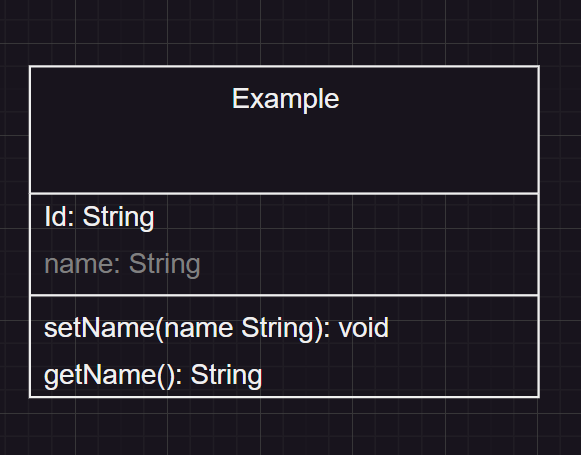
Para crear una capa solo tenemos que crear una carpeta, por ello vamos a crear dos carpetas al nivel de “DEMO”  
  


Dentro de DOMAIN vamos a manejar 2 conceptos:



* entidades: son los objetos de la APP
* casos de uso: es la lógica que hace que la APP funcione.

1 - vamos a crear la siguiente entidad de ejemplo:



2 - Ahora vamos a crear nuestro primer caso de uso:

| Obtener todos los Example | | |
| --- | --- | --- |
| 01-07-2024 + 01 | Felipe Hernandez | |
| Yo como usuario de la plataforma quiero ver en una lista todos los Example disponibles para conocer qué tanto funciona mi aplicación | | |
| 1. Se debe de mostrar ejemplos inventados. | | |

Antes de empezar con los casos de uso tenemos que hablar un poquito sobre la segregación de interfaces y sus beneficios.

Hablar sobre el nombre de la carpeta

Hablar sobre los 3 archivos

…

Lo primero que tenemos que hacer es crear el nombre de la carpeta del caso de uso:

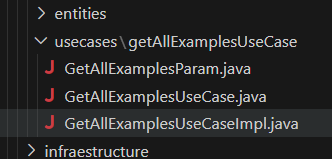
getAllExamplesUseCase

Y luego dentro de la carpeta tenemos que crear 3 archivos:

GetAllExamplesParam.java

GetAllExamplesUseCase.java

GetAllExamplesUseCaseImpl.java



* Explicar los 3 archivos pero con código.

Ahora vamos a tener que forzarlo a inyectar… y eso se hace por medio de el decorador

//Libreria

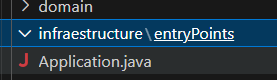
import org.springframework.stereotype.Component;

//Decorador

@Component

Ahora vamos a crear la carpeta de los entrypoints

NOTA: recuerda borrar el archivo “HelloController.java”



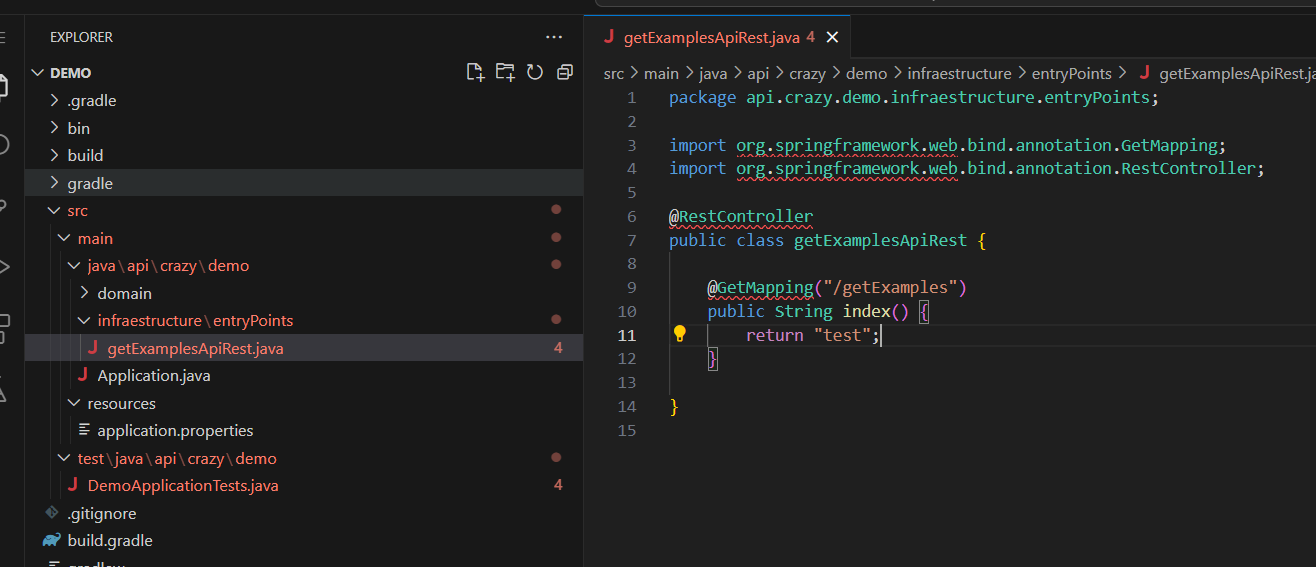
Y adentro vamos a crear el archivo que es capaz de mostrar todos los Example pero antes vamos a probar.

Compilamos:

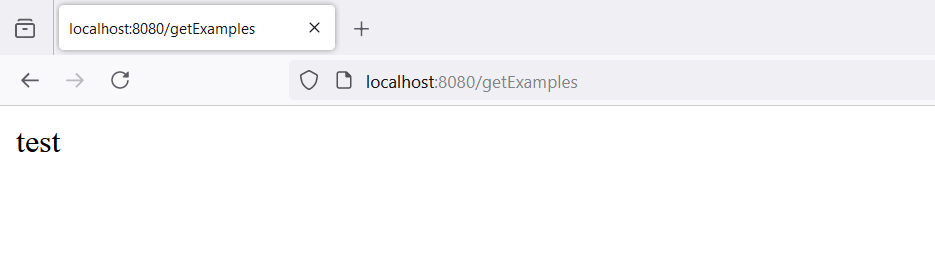
gradle clean build

Ahora vamos a ejecutar

java -jar .\build\libs\demo-0.0.1-SNAPSHOT.jar

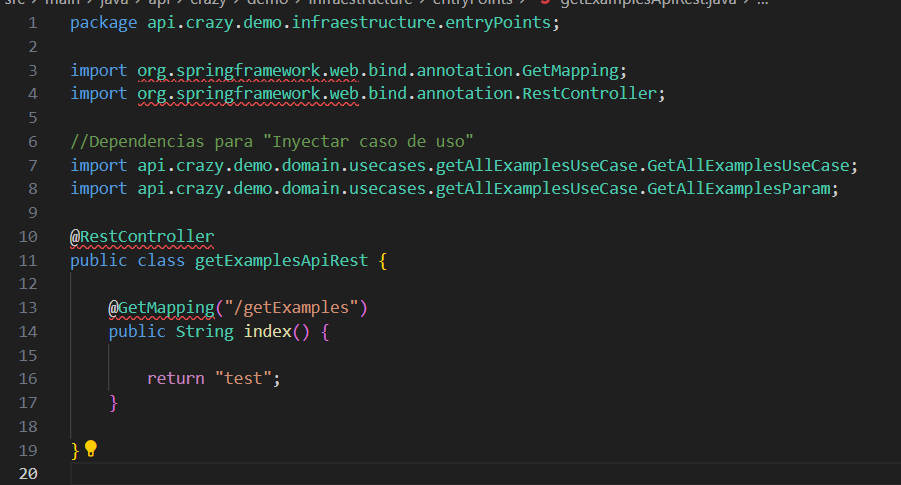


Vamos a probar desde el navegador

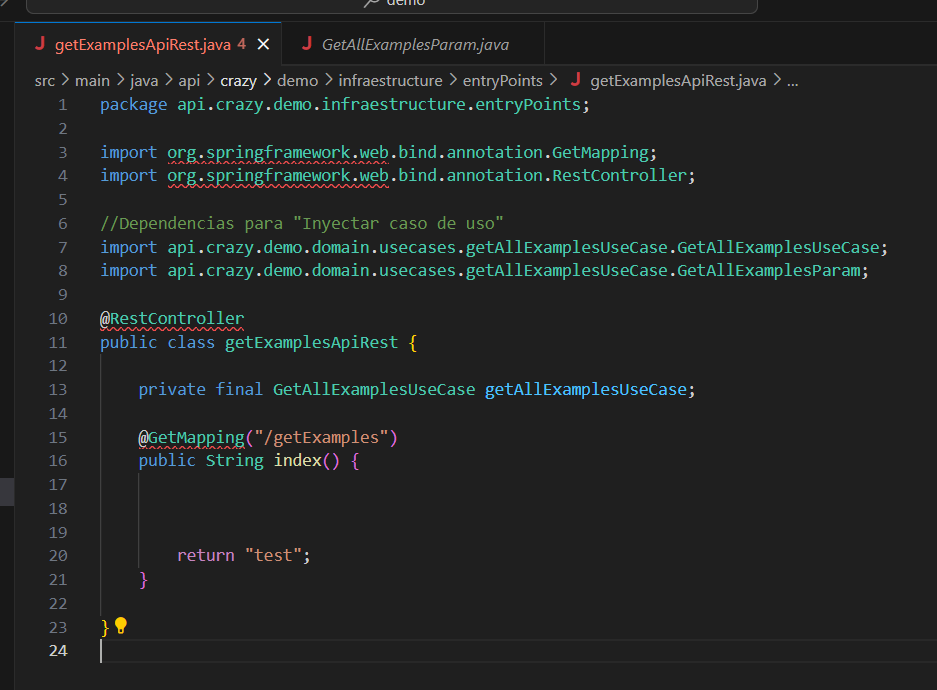


Ahora vamos a inyectar el caso de uso

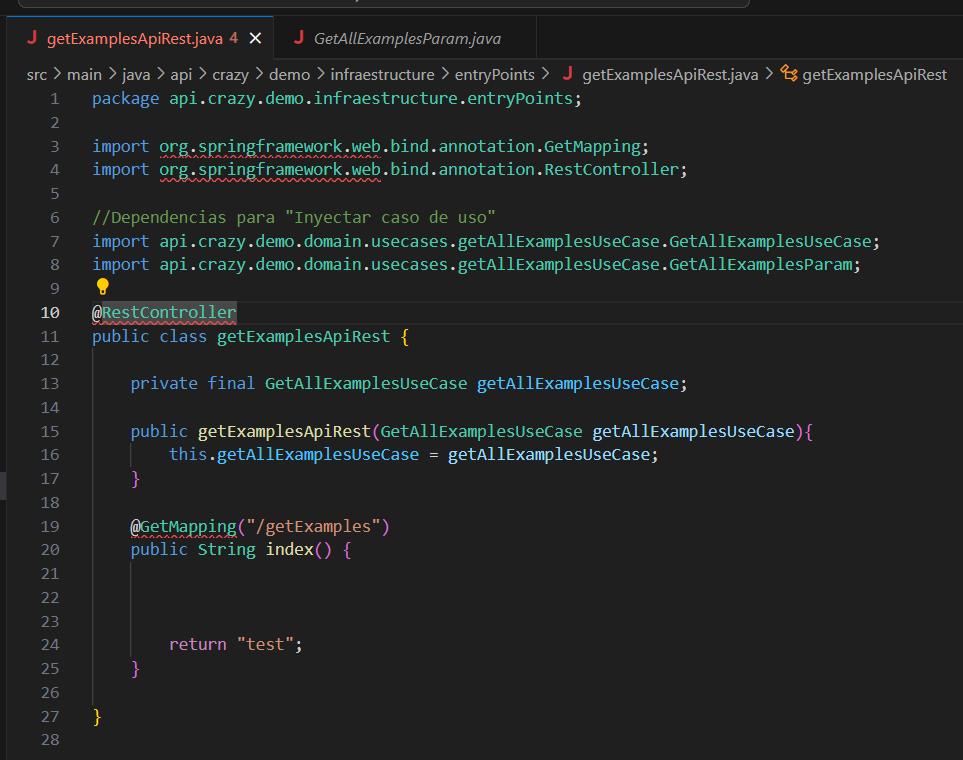
1 - IMPORTAR LAS DEPENDENCIAS



2 - Declarar como static final



3 - crear un constructor e instanciar



4 - llamar al “call” caso de uso que acabamos de inyectar y mandar el param y retornar

Compilamos:

gradle clean build

Ahora vamos a ejecutar

java -jar .\build\libs\demo-0.0.1-SNAPSHOT.jar

Vamos a verlo en el navegador

