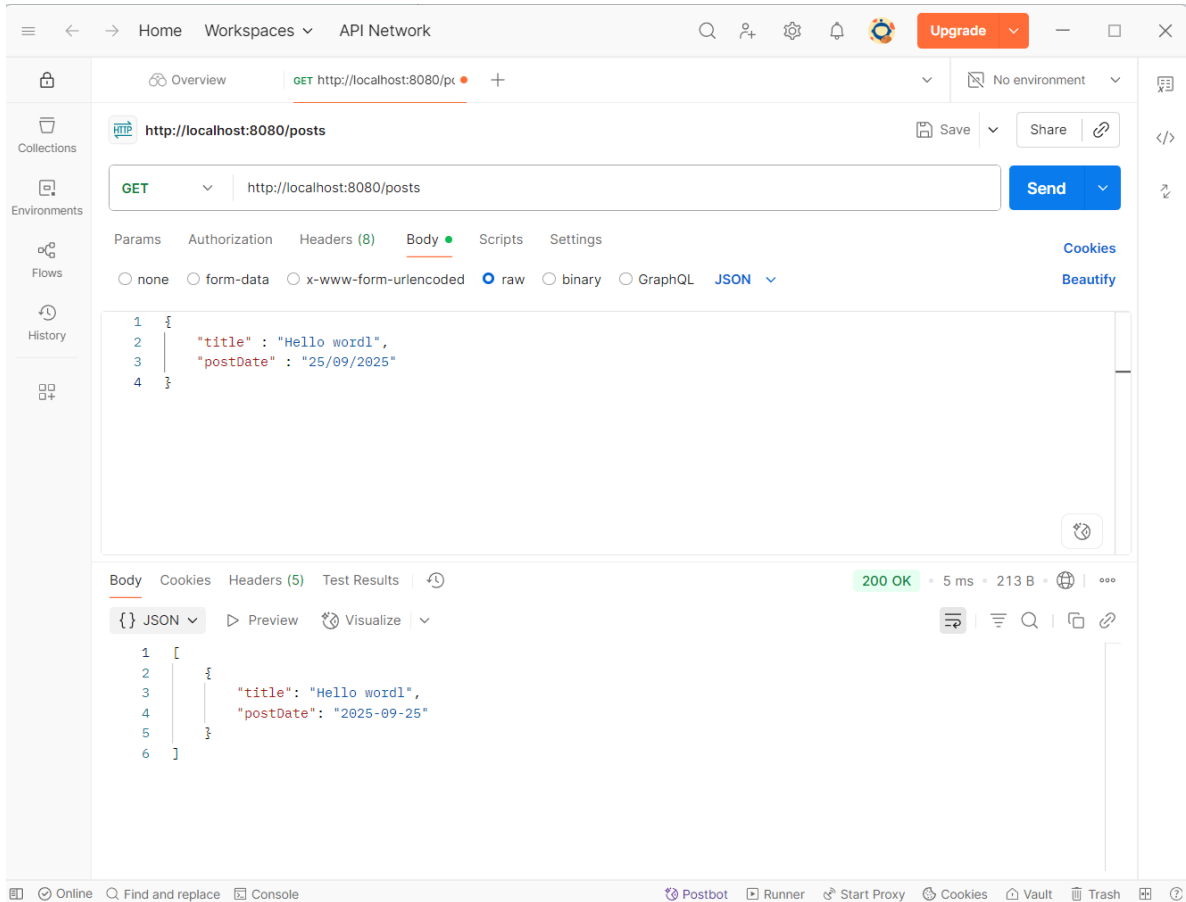


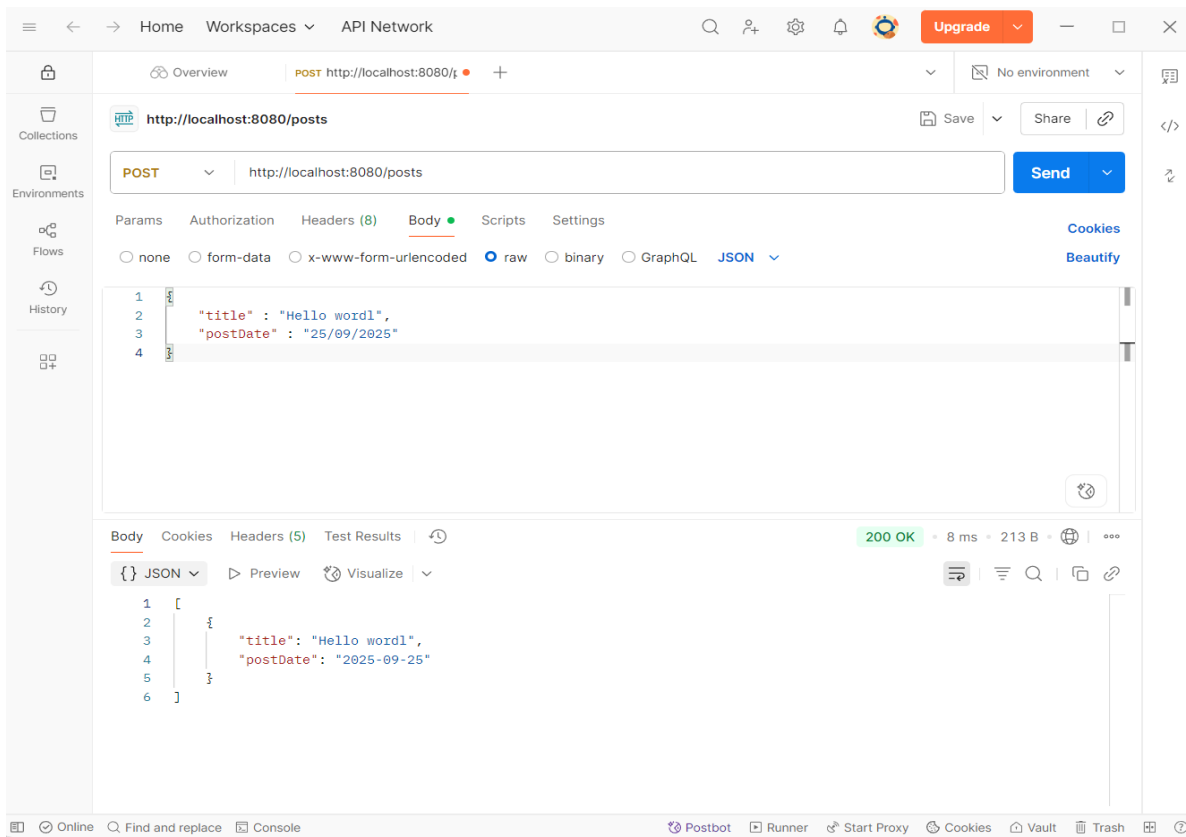
Norman Enmanuel Quijano Amaya QA240640

Guia07

Desarrollo de Aplicaciones con Web Frameworks DWF404 G03L

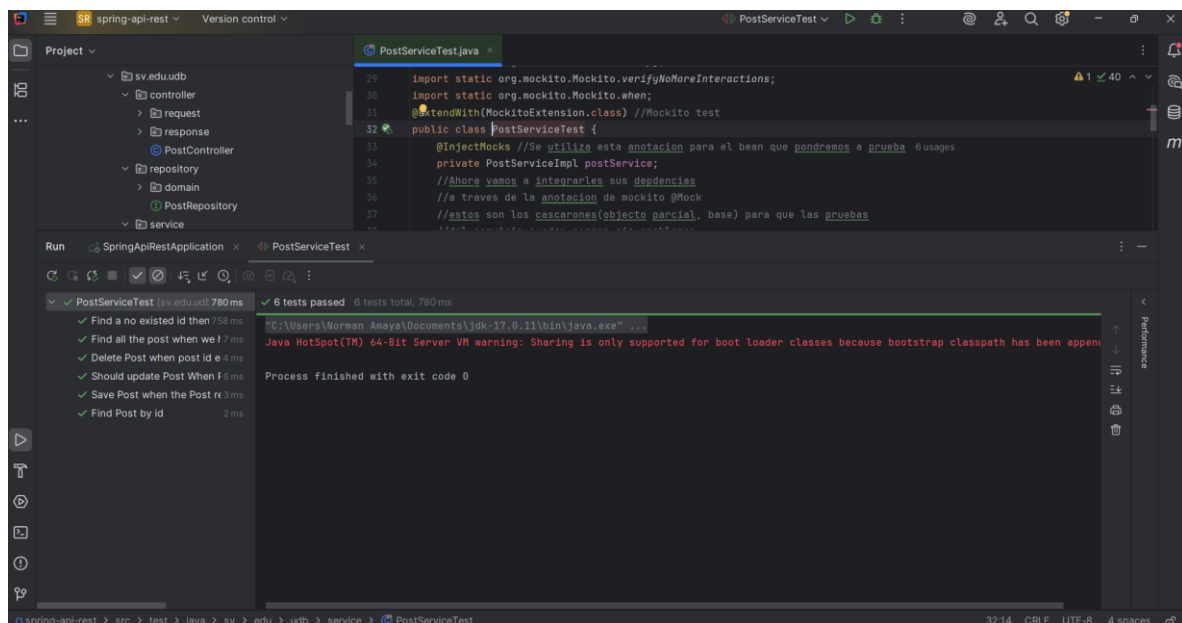


Ya al correr el SpringApplication nos vamos al postman a probar nuestra api rets donde con el método get podemos visualizar lo que registramos con el método post.



En esta captura es el registro de los datos que queremos registrar y de tal manera se va a visualizar cuando pongamos el método get como se mostraba en la captura anterior.

Las pruebas del **repository** se enfocan en validar la capa de persistencia, es decir, la parte que se comunica directamente con la base de datos. Para esto se utiliza la anotación `@DataJpaTest`, que permite probar las operaciones de JPA en un contexto aislado con una base en memoria H2. En este nivel se verifica que los métodos de `PostRepository` (como `findAll`, `findById`, `save` y `deleteById`) funcionen correctamente. Un ejemplo típico sería guardar un objeto `Post` y comprobar que puede recuperarse con `findById`, o bien, eliminar un registro y confirmar que ya no existe en la base.



Las pruebas del **service** validan la lógica de negocio de la aplicación, que normalmente está en la clase `PostServiceImpl`. Aquí no se utiliza una base de datos real, sino que se trabaja con **mocks** y también con **Mockito**. Esto permite simular el comportamiento de dependencias como el `PostRepository` y el `PostMapper`. Con estas pruebas se comprueba que los métodos del servicio ejecuten la lógica esperada: por ejemplo, que al buscar por un ID válido se devuelva un `PostResponse`, que al buscar un ID inexistente se lance una excepción, o que un `PostRequest` se convierta, guarde y retorne correctamente. El enfoque principal está en probar reglas de negocio sin depender de la infraestructura.