**Sprint 3.**

**i. Presentación MVC**

1. Aplicación con persistencia Relacional.

2. Pruebas unitarias de la lógica desarrollada

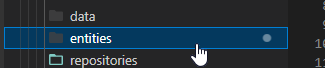
**ii. Informe de retrospectiva**

**iii. Historias de usuario a desarrollar en el sprint 4. (Trello)**

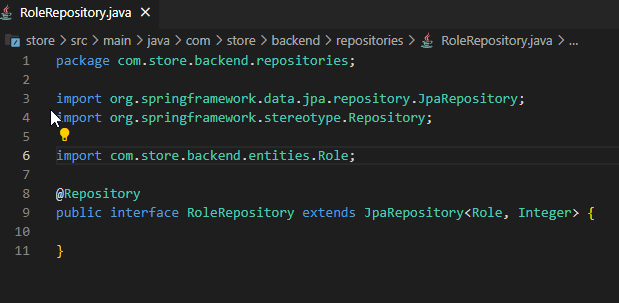
**Desarrollo:**

**Presentación MVC.**

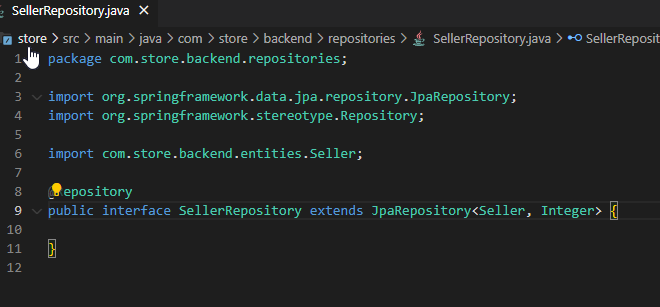
* Creación repositorios, controladores, servicios.

Creamos la carpeta repositories. Para el manejo de persistencia con JPA: 

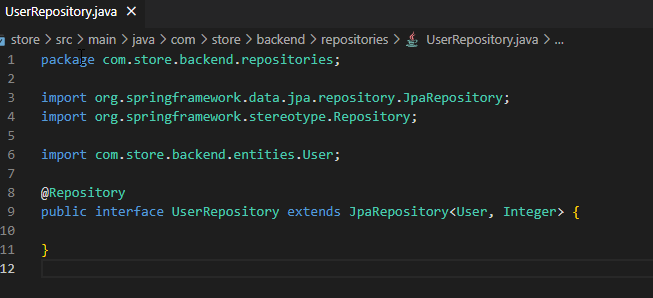
**RoleRepository**

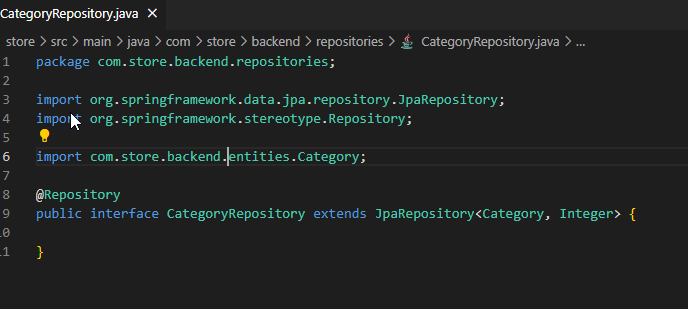


**SellerRepository:**

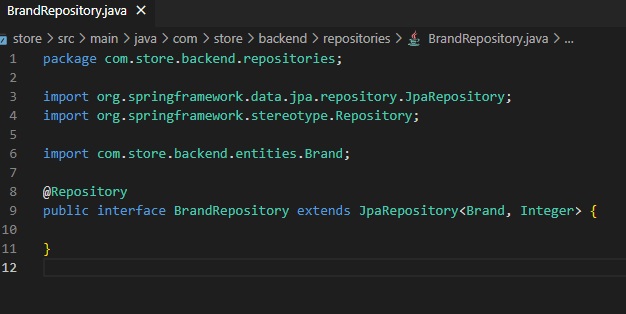


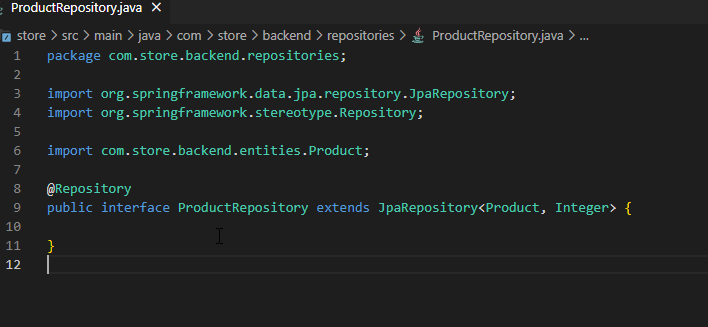
**UserRepository**:

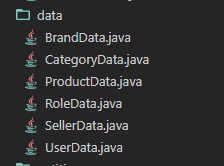
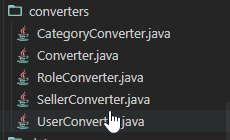
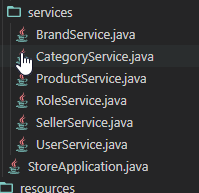


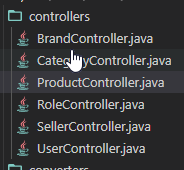
**CategoryRepository** .

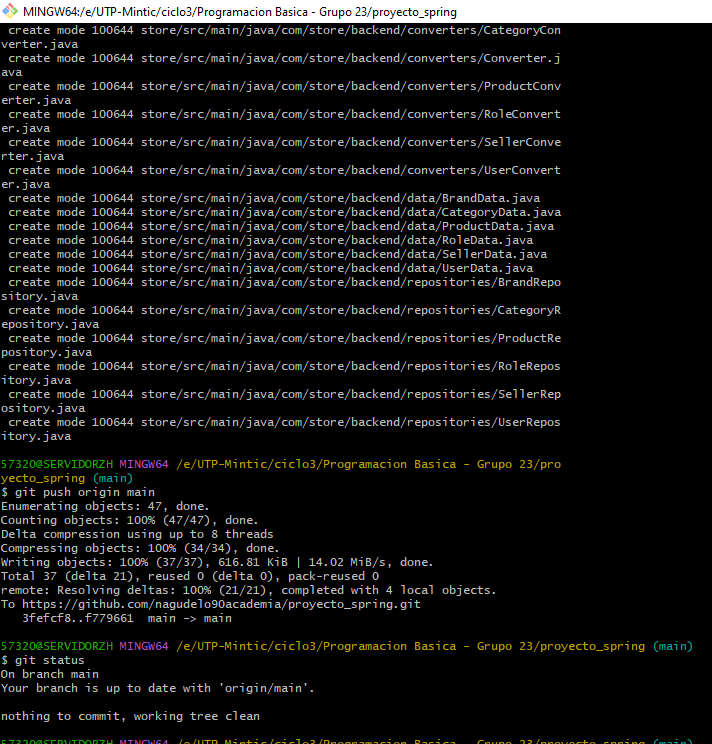
**BrandRepository** (Marca) de productos.



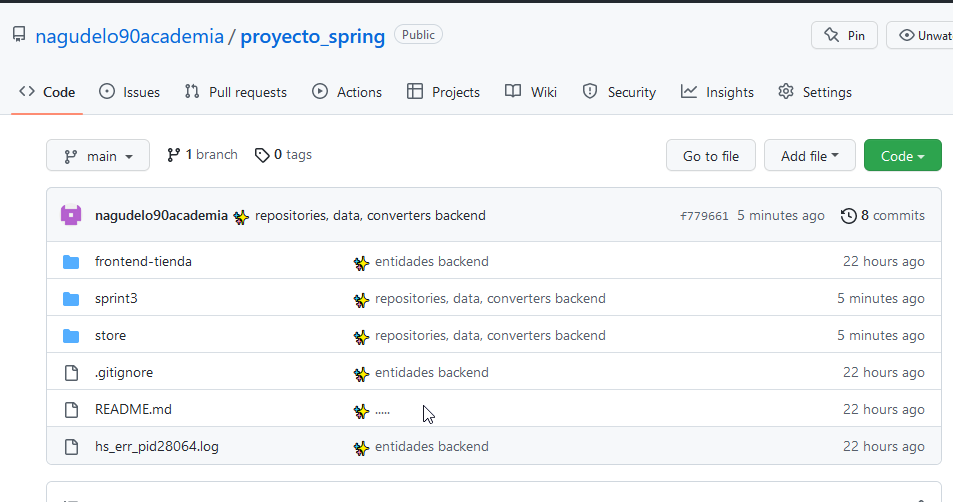
**ProductRepository**

* Creacion de los ---data--- para los DTO / VO de todas las entidades.
* 
* Creacion de los ---converters---para proteger las entidades.
* 
* Creacion de los ---services---
* 
* Creacion de los ---controllers---

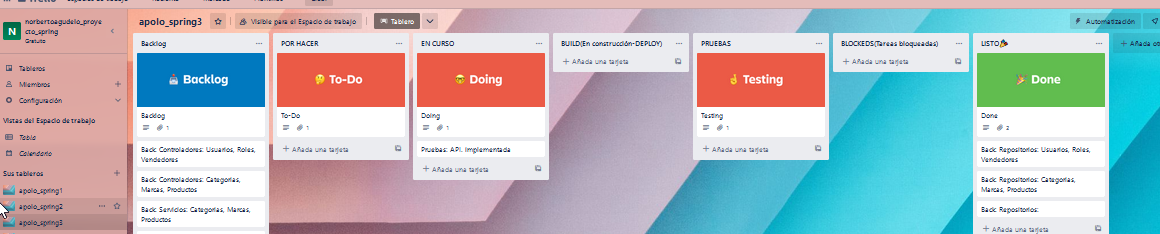


* Se actualiza el repo:
* 

<https://github.com/nagudelo90academia/proyecto_spring>



Sprint3 en trello: <https://trello.com/invite/b/yY3Kf7GB/5ba06fd70439231b3192cc64ee44ff49/apolospring3>



v. Informe de retrospectiva

**Formulario de reunión retrospectiva**

| **¿Qué salió bien en la iteración? (aciertos)** | **¿Qué no salió bien en la iteración? (errores)** | **¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)** |
| --- | --- | --- |
| * La creación de los repositories, data, converters. Y el inicio de los services y controllers.   Basados en las clases sincronicas explicación | * Los enlaces de relaciones Muchos a muchos en los data y converters supuso algunos inconvenientes.. | * Terminar los servicios y controllers. Manejo pleno de las pruebas de la api con funcionalidad completa |

vi. Historias de usuario a desarrollar en el sprint 4. (Trello)

<https://trello.com/invite/b/PQL3d2M3/f7ba9027be3babf91c9642565354777f/apolospring4>

* Implementación de Total de controladores, repositorios y servicios para la persistencia.
* Pruebas de esta implementación si es posible en nube.