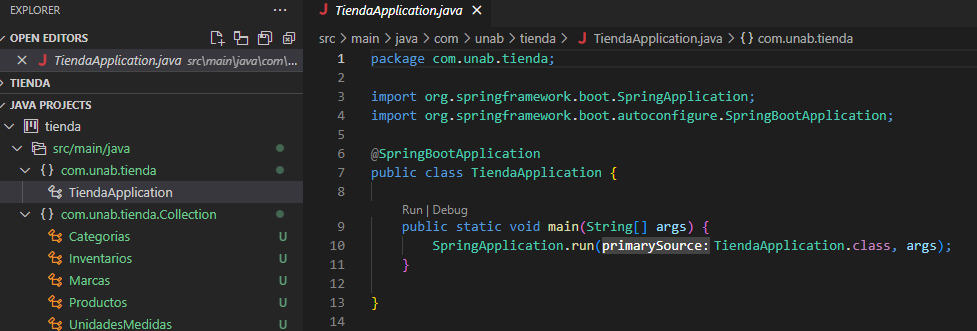
|  |
| --- |
| **SPRINT 2: Desarrollo del Backend** |

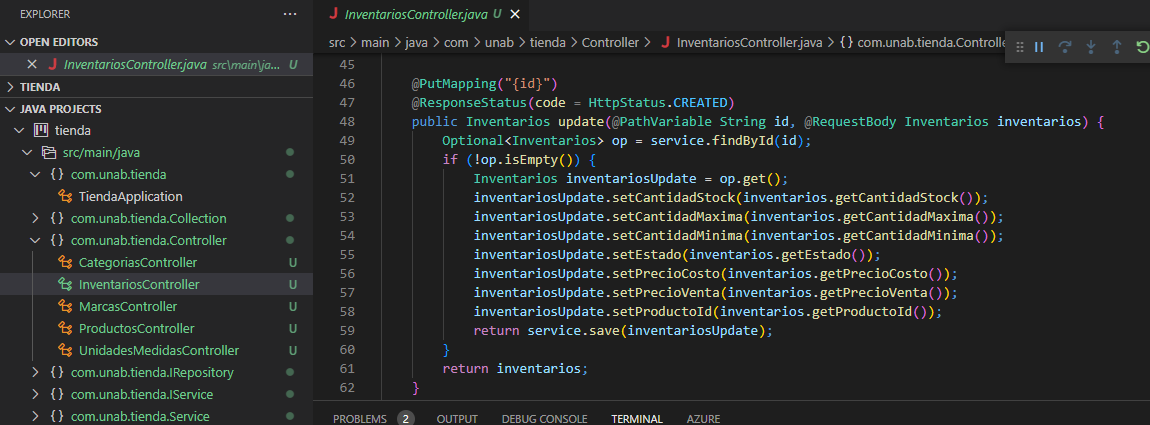
|  |  |
| --- | --- |
| **Identificación Proyecto** | |
| Nombre Proyecto: | Software para la gestión de una tienda de barrio |
| Número Equipo: | N.A |
| **Integrantes del equipo** | |
| Rol  (Líder-Desarrollador – Cliente) | Nombre |
| Líder | Robinson Daniel Cruz Castro |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Stakeholder | Jesús Ariel González Bonilla |

|  |
| --- |
| **Evidencia construcción del Backend** |

Creación en Spring Boot (se descargó el proyecto Spring Boot con las dependencias necesarias desde Spring Initializr y se trabajó desde VS code como Java Project)



Construcción Collection, Controller, Repository, Service (siguiendo la guía trabajada en clase se realizó desde cero cada uno de los módulos para cada estructura en el esquema de datos (categorías, **inventarios**, marcas, productos, unidades medidas)



La API se probó desde el localhost, desplegándola desde el archivo TiendaApplication.java

A screenshot of a computer

Description automatically generated with medium confidence

Se desplegó con éxito y se conecto a la base de datos en mongo

Text

Description automatically generated

Mongo

Graphical user interface, text, application

Description automatically generated

Application.properties

Text

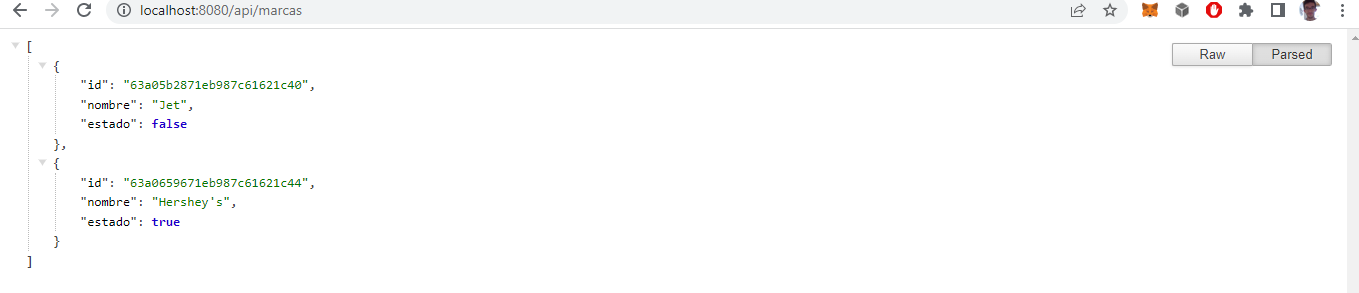
Description automatically generated

Terminal

A screenshot of a computer

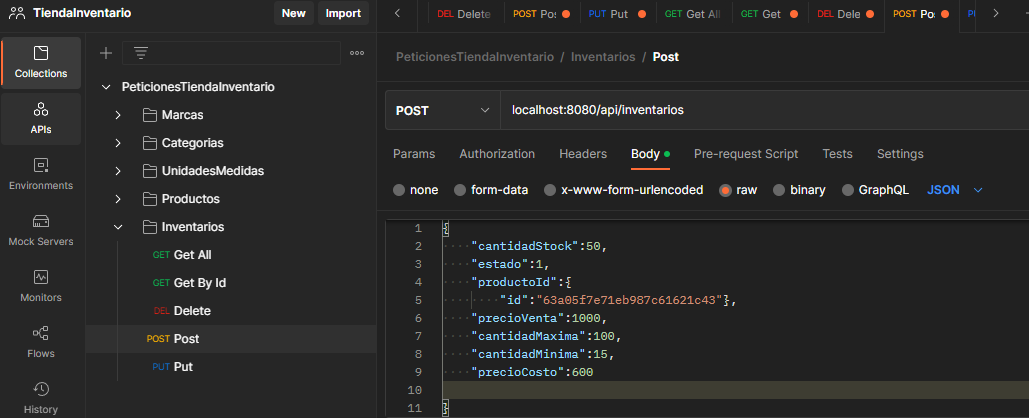
Description automatically generated with medium confidence

LocalHost:8080



|  |
| --- |
| **Evidencias de los “endpoint” con el consumo de recursos del API REST** |

Con el API desplegado en el LocalHost:8080 se procede a crear el workspace de peticiones desde el navegador con Postman, definiendo cada una de las peticiones mapeadas en la API (getAll, getById, post, put, delete) para cada una de las estructuras.



Y se probó su correcto funcionamiento para cada una, siendo la Inventarios la que recopila el uso de todas (un inventario para un producto clasificado bajo una marca, categoría y unidad medida)

Graphical user interface, text

Description automatically generated

|  |
| --- |
| **Evidencia JIRA (Seguimiento del proyecto)** |

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

|  |
| --- |
| **Repositorio de Código GitLab o GitHub** |

URL del repositorio: <https://github.com/rcruzca/C4WEB-TiendaInventario.git>

Graphical user interface, text, application, email

Description automatically generated

|  |
| --- |
| **Evidencias de las Reuniones de Equipo** |

\*Debido a la falta de tiempo y que cada miembro parecía querer hacer las cosas por su propio lado a sus tiempos, decidí presentar este sprint individualmente.