|  |  |  |
| --- | --- | --- |
| Asignatura: Programación III | Día y hora: Martes y Jueves de 6:30 a 8:10 p.m. | Grupo: 01 |
| Docente: Ing. Rodrigo Ortiz, MIR | **E-mail:** d.rortiz@ufg.edu.sv | **Aula:** B25-26 |
| Estudiante: |  | |

Apellidos

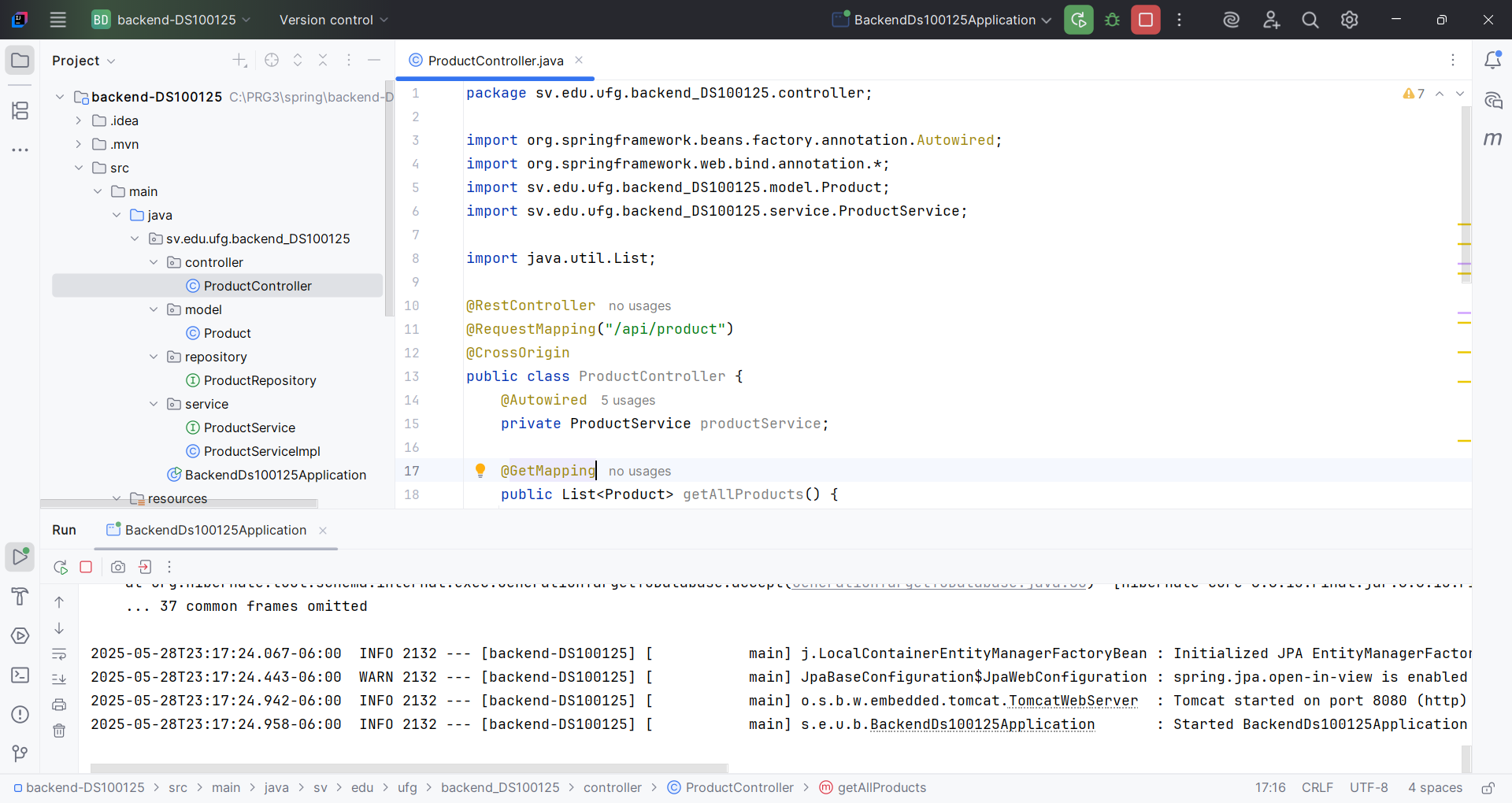
Nombres

|  |  |  |  |
| --- | --- | --- | --- |
| **Carrera:** | **Carné:** | **Fecha:** | **Calificación:** |

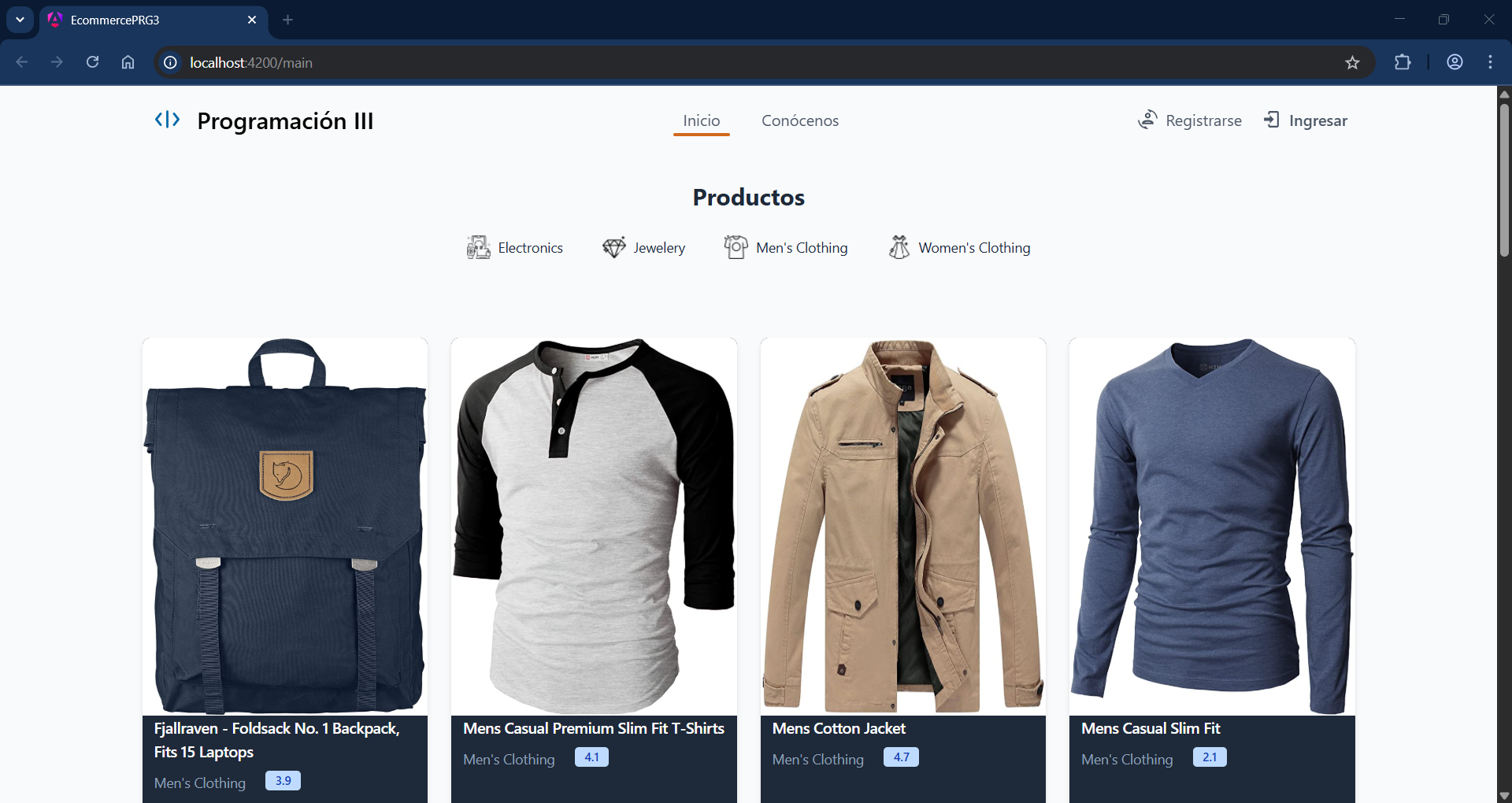
Realiza los pasos que ya conoces para restaurar la base de datos y para poner en funcionamiento tu API Rest (IntelliJ – Spring Boot) y tu Aplicación Ecommerce (Angular)

Tu laboratorio 4 consta de los siguientes requerimientos:

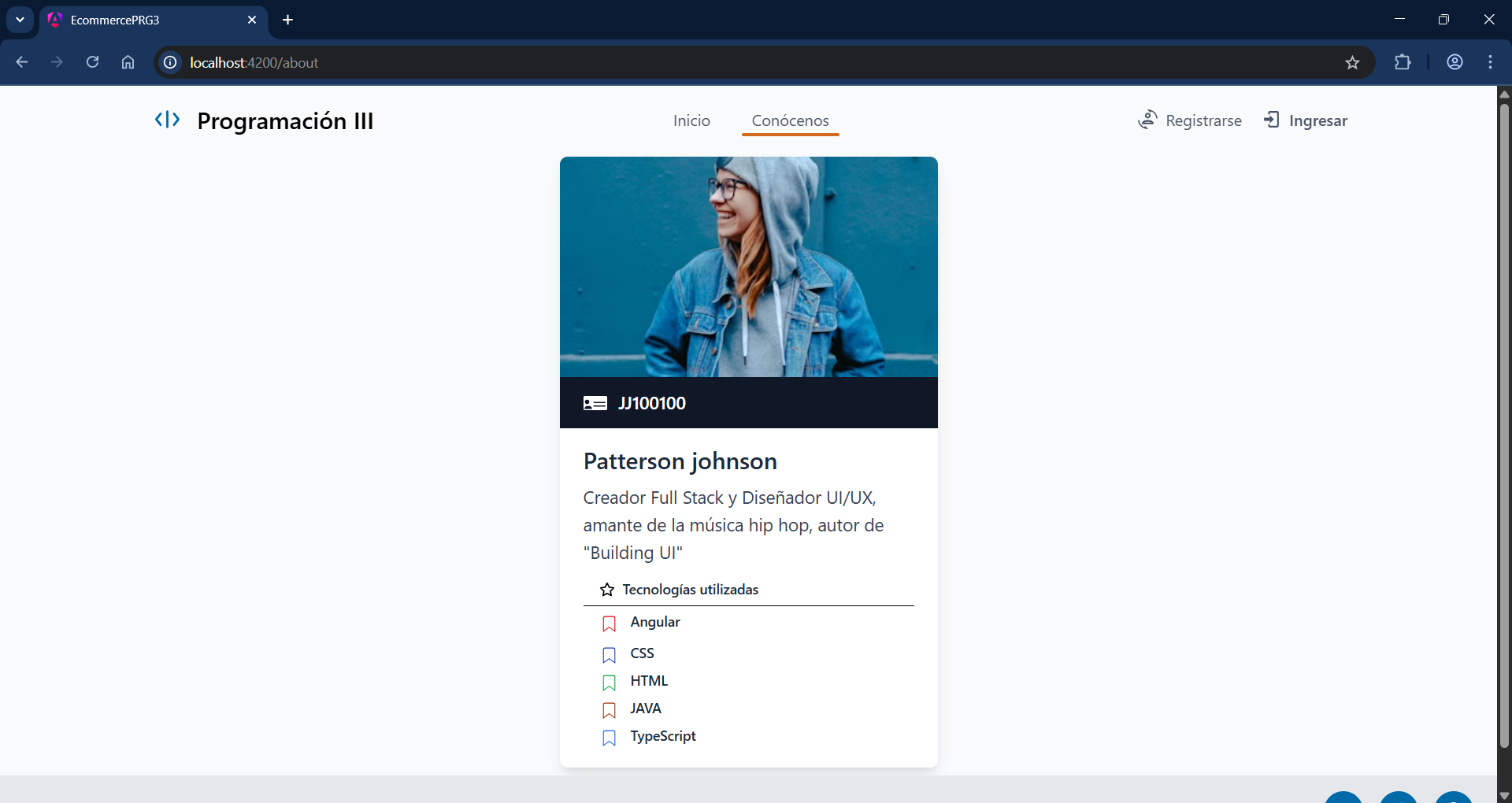
## API Rest con Spring Boot CRUD (**Funcional**) (20%)



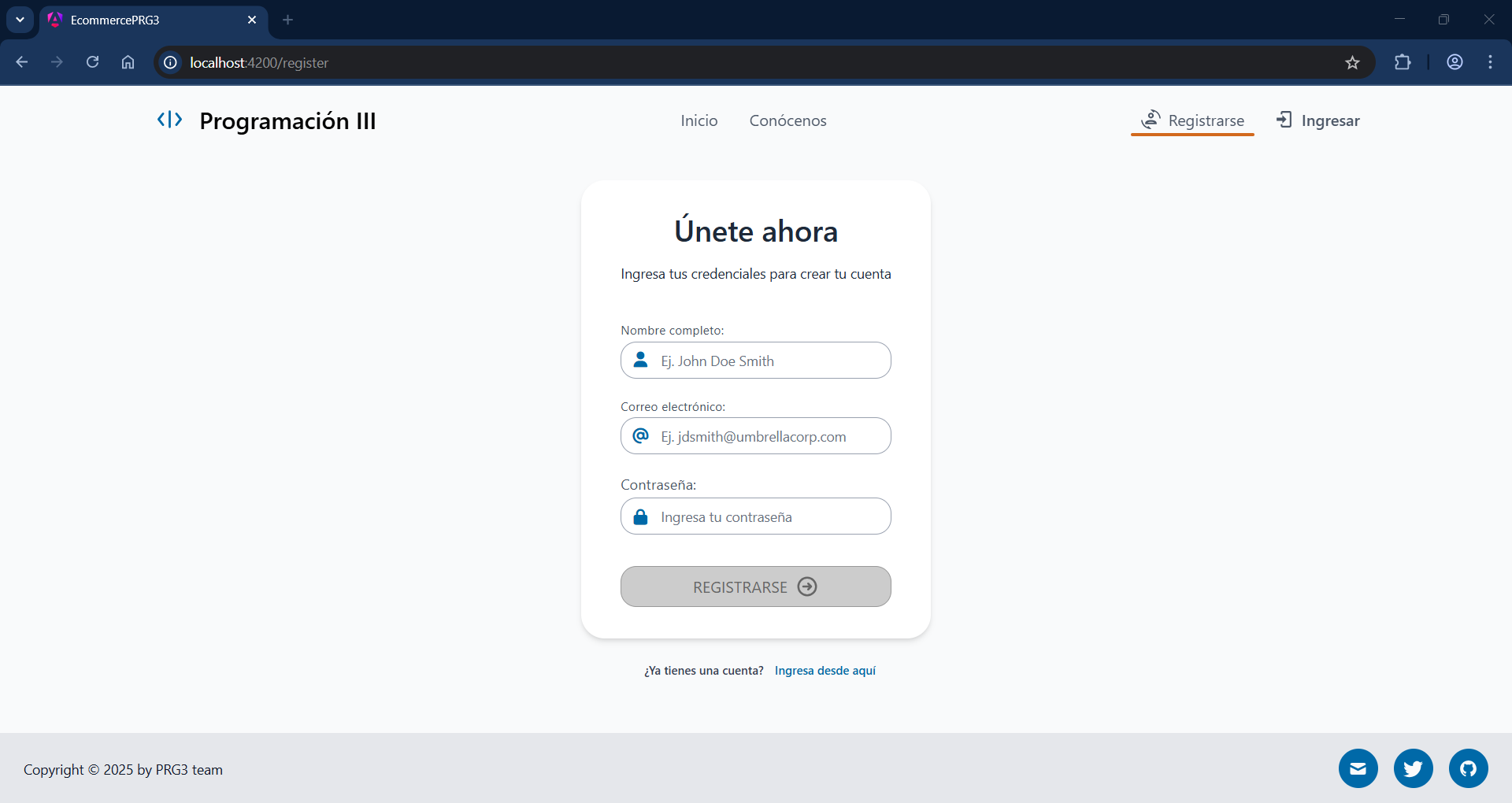
## Listado de productos cargados desde el API (5%)



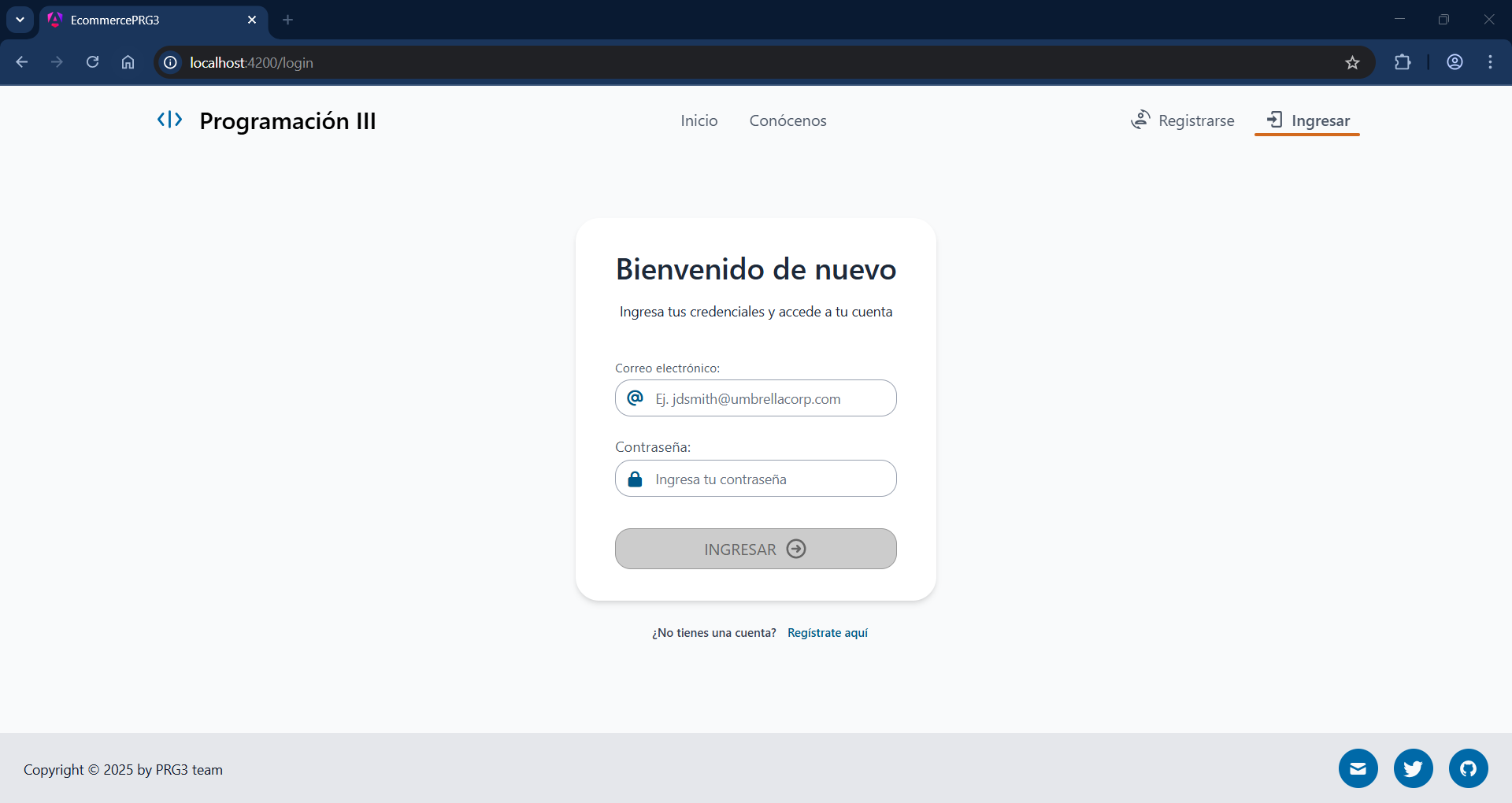
## Pantalla del autor de la aplicación (5%) (**Con tu información completa**)



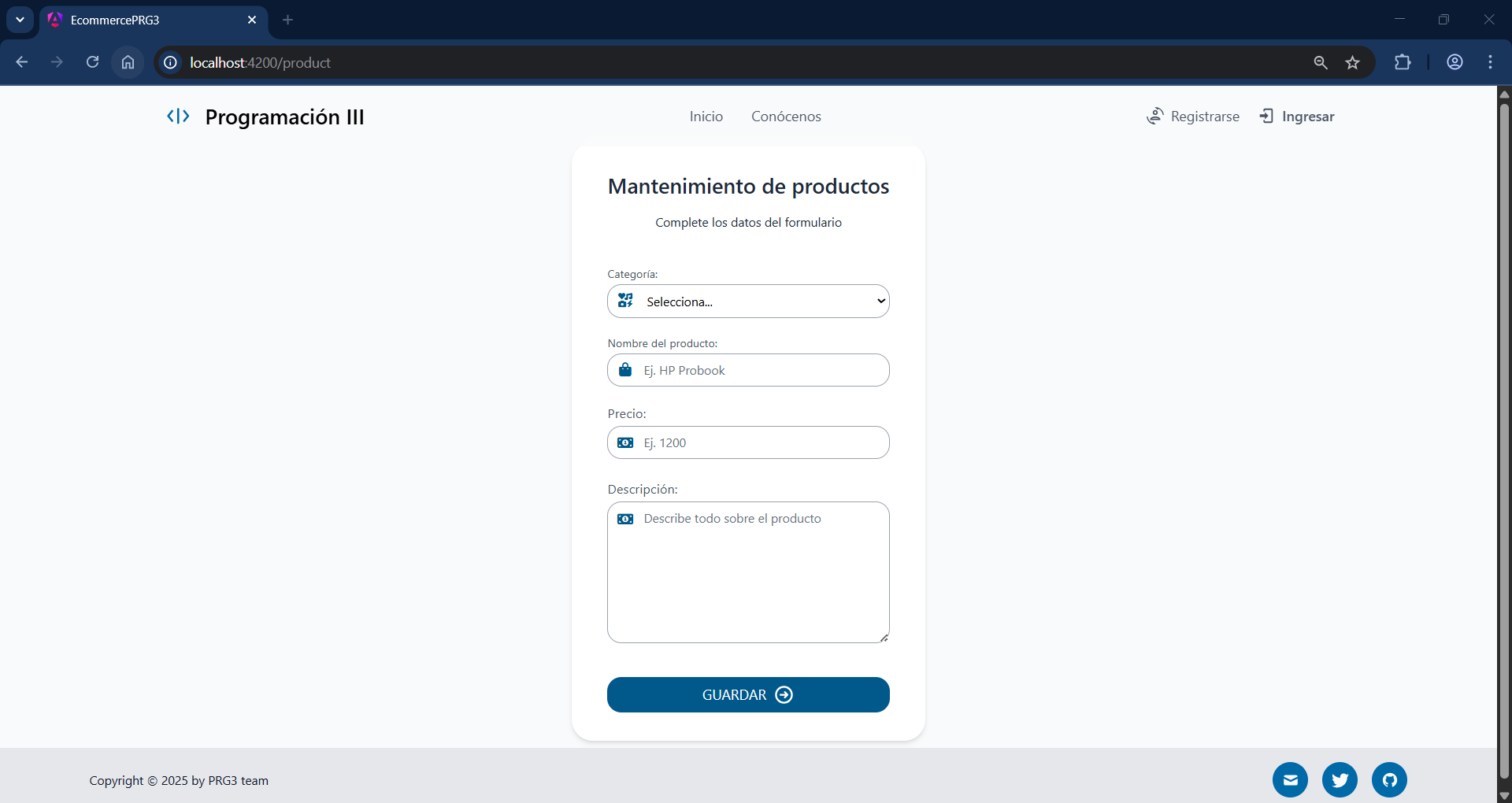
Diseño del Formulario de Registro (Prototipo) (5%)



Diseño del Formulario de Login (Prototipo) (5%)



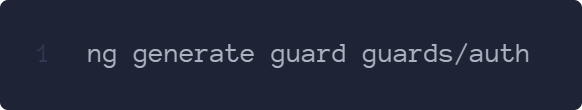
Mantenimiento de productos (Funcional) (20%)



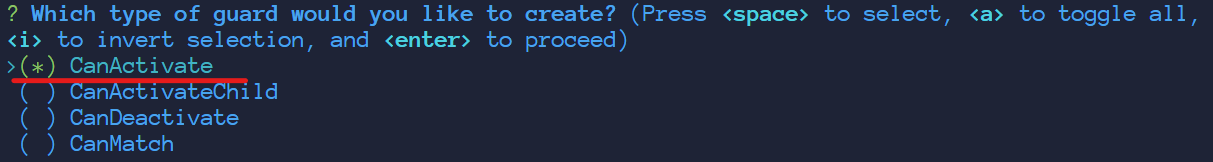
Autenticación con JWT (Login funcional) (20%)

En tu proyecto de Angular crea lo siguiente utilizando el generador de componentes:

* Un **servicio** llamado ***auth*** colocado en una carpeta llamada **services**
* Un **guard** de nombre ***auth*** colocado en una carpeta llamada **guards**

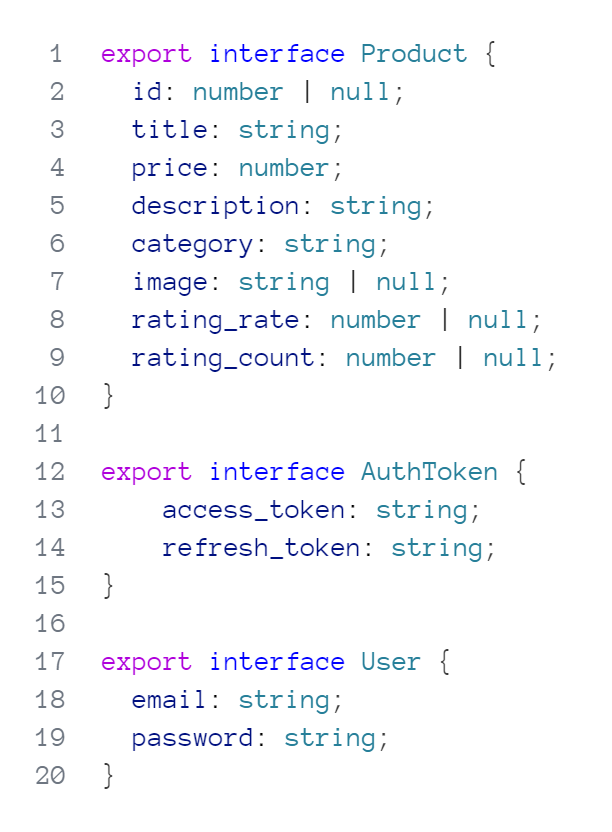


Cuando pregunte el tipo de guard a crear selecciona CanActivate



Configuración de las interfaces

Abre tu archivo **interfaces.ts** y agrega las interfaces para el token de autenticación y para las credenciales del usuario



Configuración de variable de entorno

Abre tu archivo **environment.ts** y agrega la propiedad authURL como se muestra a continuación:



Configuración del servicio de autenticación

Abre tu archivo **auth.service.ts** y agrégale el siguiente código:



Configuración del guard para proteger las rutas (10%)

Abre tu archivo **auth.guard.ts** y agrégale el siguiente código:



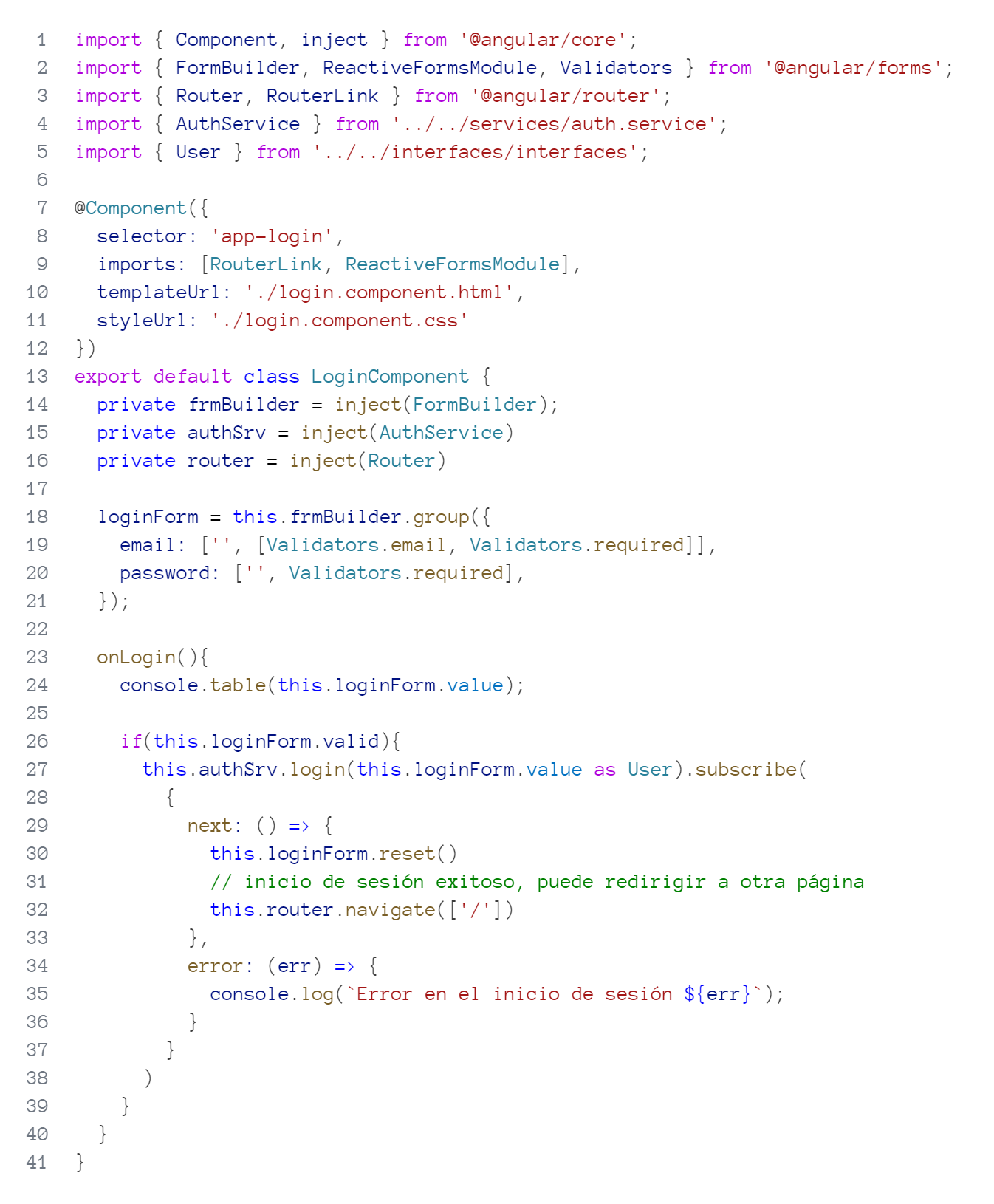
Configuración de las rutas agregando CanActivate

Abre tu archivo **app.routes.ts** y agrégale el siguiente código:



Configuración de la lógica para la página de login

Abre tu archivo **login.component.ts** y agrégale el siguiente código:



Configuración del componente principal

Abre tu archivo **app.component.ts** y agrégale el siguiente código:



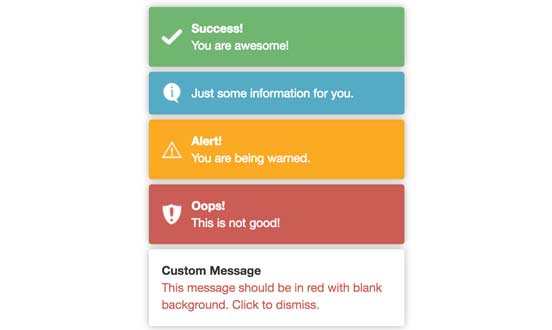
Ejecuta tu proyecto y comprueba su funcionamiento en el navegador.

|  |  |  |
| --- | --- | --- |
| Criterio | Ponderación | Obtenido |
| API Rest con Spring Boot CRUD (Funcional) | 20% |  |
| Listado de productos cargados desde el API | 5% |  |
| Pantalla del autor de la aplicación | 5% |  |
| Formulario de Registro (Prototipo) | 5% |  |
| Formulario de Login (Prototipo) | 5% |  |
| Mantenimiento de productos (Funcional) | 20% |  |
| Autenticación con JWT (Login funcional) | 20% |  |
| Guards en rutas de formulario de productos | 10% |  |
| Archivo de WORD en formato .doc o .docx | 10% |  |

## Retos comodines:

La realización de uno de estos retos adicionales pude otorgar **1 punto** en la calificación, si y solo si es necesario:

* Agrega alertas de tipo toast a los eventos:
  + Crear un nuevo producto
  + Actualizar un producto
  + Eliminar un producto



* Agrega un botón para poder cerrar la sesión en la barra de navegación, este botón solo debe mostrarse si el usuario este logueado.



* Para la entrega debes comprimir **todos** los archivos y carpetas, **exceptuando** la carpeta **node\_modules** y tampoco la carpeta **.angular**
* Recuerda agregar este archivo en formato **.docx o .doc** cualquier otro formato será penalizado con -1 punto.
* Todos los porcentajes de cada elemento de este laboratorio están sujetos a la **funcionalidad** del proyecto.