### Tarea 1: Formularios de Login y Registro en el Frontend

Prompt:

‘Necesito hacer un sistema básico de login funcional para usuarios, compuesto por una interfaz de usuario en el frontend y una API en el backend para la autenticación. Los usuarios deben poder registrarse con un nombre de usuario/correo electrónico y contraseña, y luego iniciar sesión con esas credenciales.

Podrías comenzar creando los formularios de login y de registro de usuario?

Que el formulario de login sea un html responsivo con campos correo y contraseña obligatorios, un botón de enviar y un link a la página de registro. Se debe contar con una validación del lado del cliente en JavaScript de obligatoriedad de los campos.

Que el formulario de registro sea un html responsivo con campos correo y contraseña obligatorios y un botón de registrar. Se debe contar con una validación del lado del cliente en JavaScript de confirmación de contraseña.’

Rta:

‘No he podido crear los archivos HTML porque no tengo permisos para escribir en la raíz del sistema de archivos. Para continuar, por favor indícame en qué carpeta de tu proyecto deseas que coloque los archivos login.html y register.html, o si prefieres que te proporcione el contenido de los archivos para que los copies manualmente. ¿Cómo prefieres proceder?’

‘quiero que el proyecto lo generes en esa ruta: C:\Users\Barbara\_Allegretti\Documents\Curso\_IA\Unidad\_6\Sistema\_Login’

Ahí generó los 2 html en la carpeta que le definí. Y respondió lo siguiente:

‘¡Listo! Los archivos login.html y register.html han sido creados en la ruta que indicaste:

C:\Users\Barbara\_Allegretti\Documents\Curso\_IA\Unidad\_6\Sistema\_Login

Ambos formularios son responsivos y cuentan con validación del lado del cliente en JavaScript:

- El formulario de login valida que el correo y la contraseña sean obligatorios, y tiene un enlace a la página de registro.

- El formulario de registro valida que el correo, la contraseña y la confirmación de contraseña sean obligatorios, y que ambas contraseñas coincidan. Incluye un enlace para volver al login.

¿Quieres que continúe con la API backend para la autenticación o necesitas algún ajuste en los formularios?’

### Tarea 2: Configuración del Backend (Endpoints de Autenticación)

‘sí, quisiera que continuaras con la API backend para la autenticación. El backend lo quiero con Java 21 y Spring Boot. Habría que generar un endpoint simple para el registro de usuarios POST /api/register y otro para login POST /api/login, que reciban 'username' y 'password' desde un body en formato JSON. Para ambos endPoints, imprimir los datos recibidos en consola.’

Primero no podía generar por un problema con la versión de mvn, actualicé las variables de entorno para que tome una versión más nueva y luego repetí el prompt anterior.

Luego, tuve que pedirle que generara los archivos dentro de la ruta que había especificado, porque no sé dónde había generado las clases.

Rta: ‘¡Listo! Ahora tienes el backend de Spring Boot correctamente ubicado en:

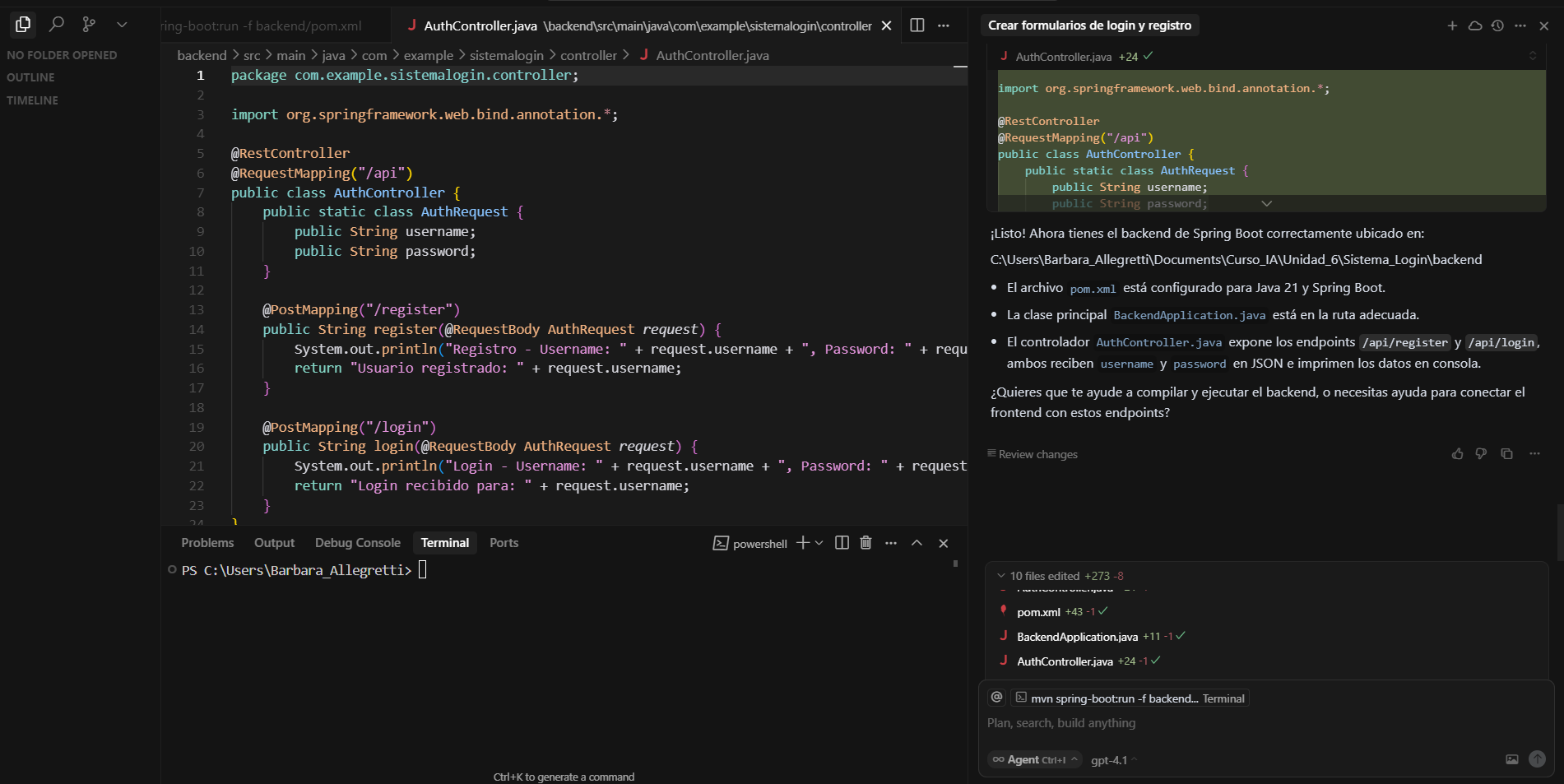
C:\Users\Barbara\_Allegretti\Documents\Curso\_IA\Unidad\_6\Sistema\_Login\backend

El archivo pom.xml está configurado para Java 21 y Spring Boot.

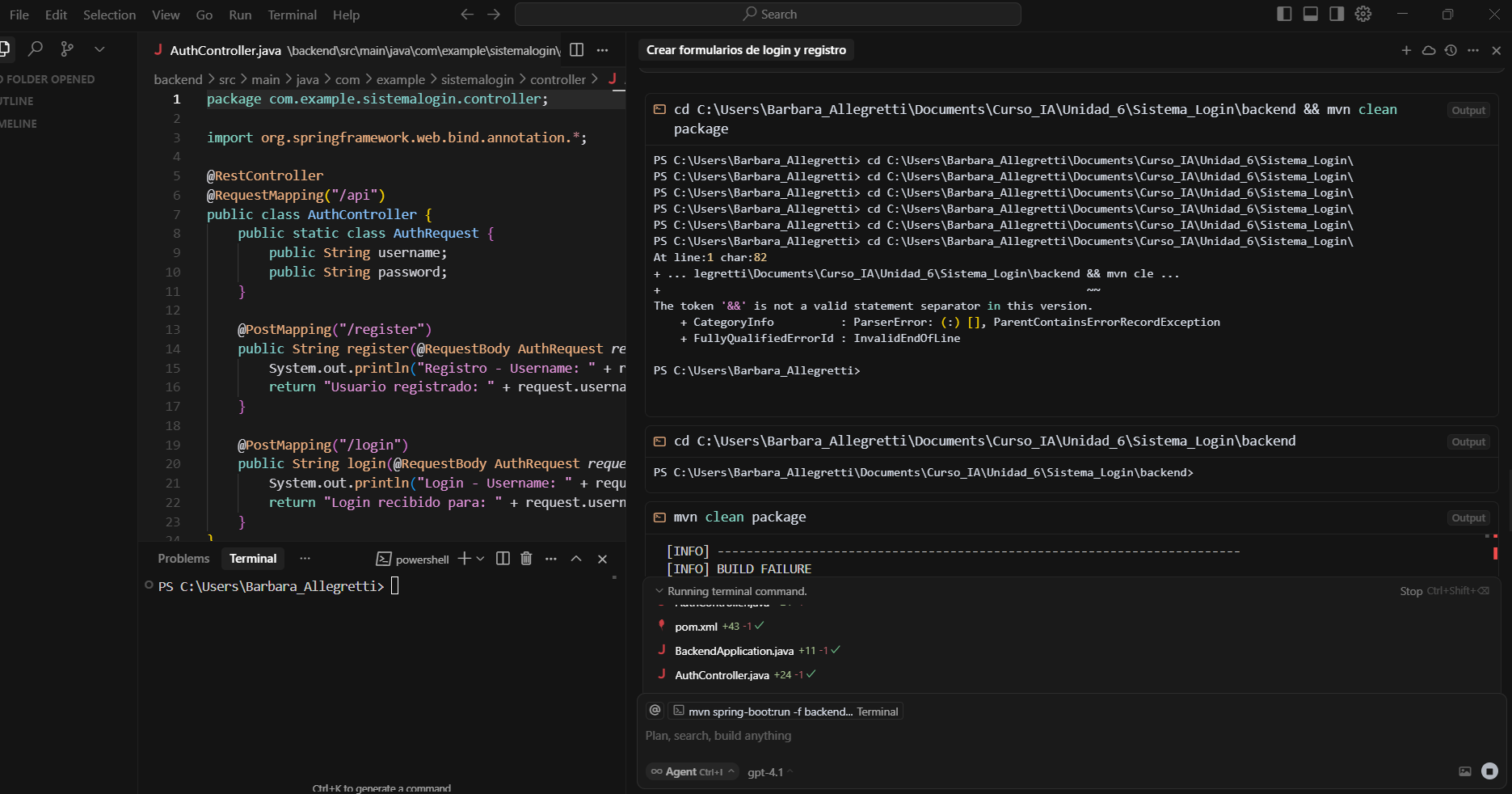
La clase principal BackendApplication.java está en la ruta adecuada.

El controlador AuthController.java expone los endpoints /api/register y /api/login, ambos reciben username y password en JSON e imprimen los datos en consola.

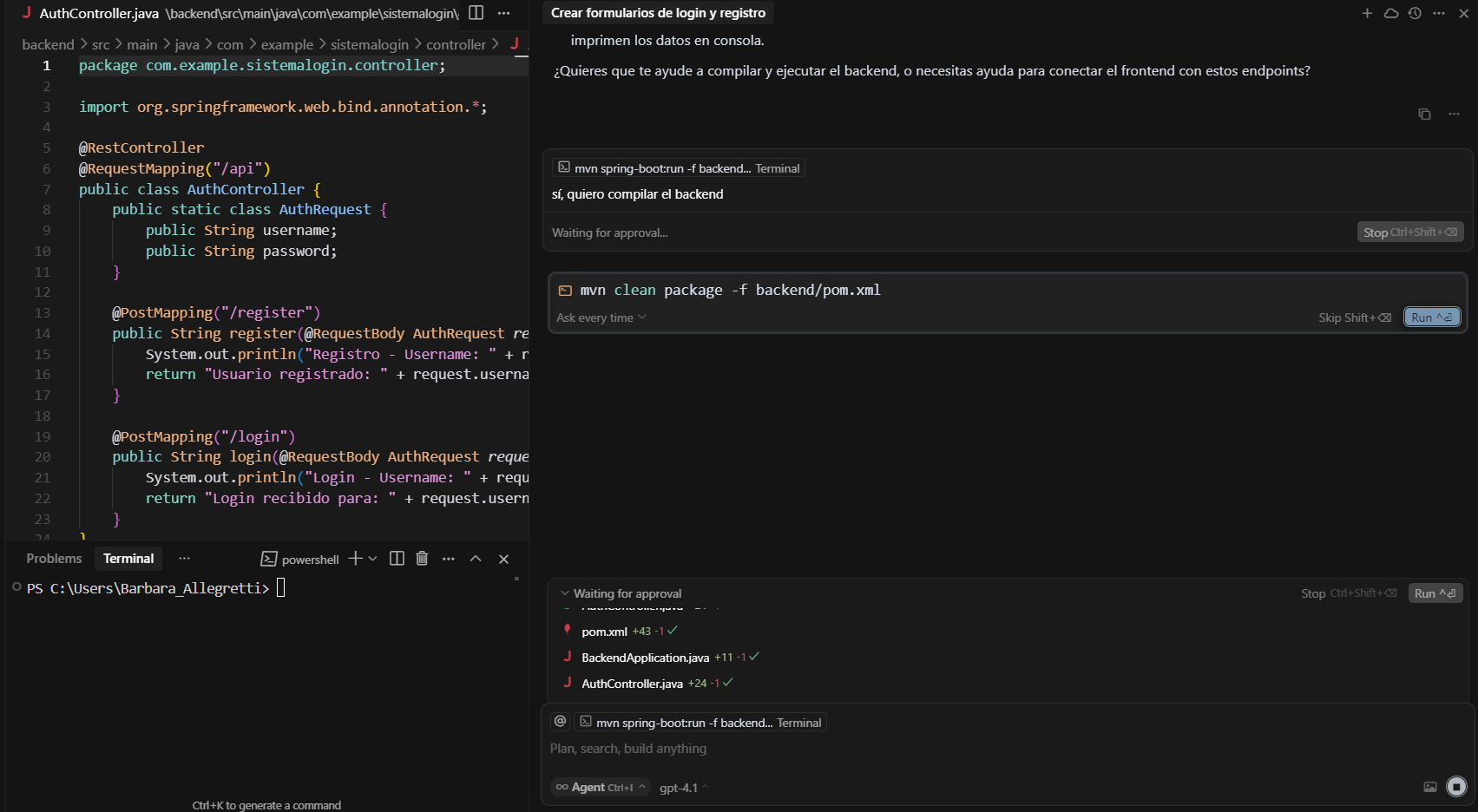
¿Quieres que te ayude a compilar y ejecutar el backend, o necesitas ayuda para conectar el frontend con estos endpoints?’

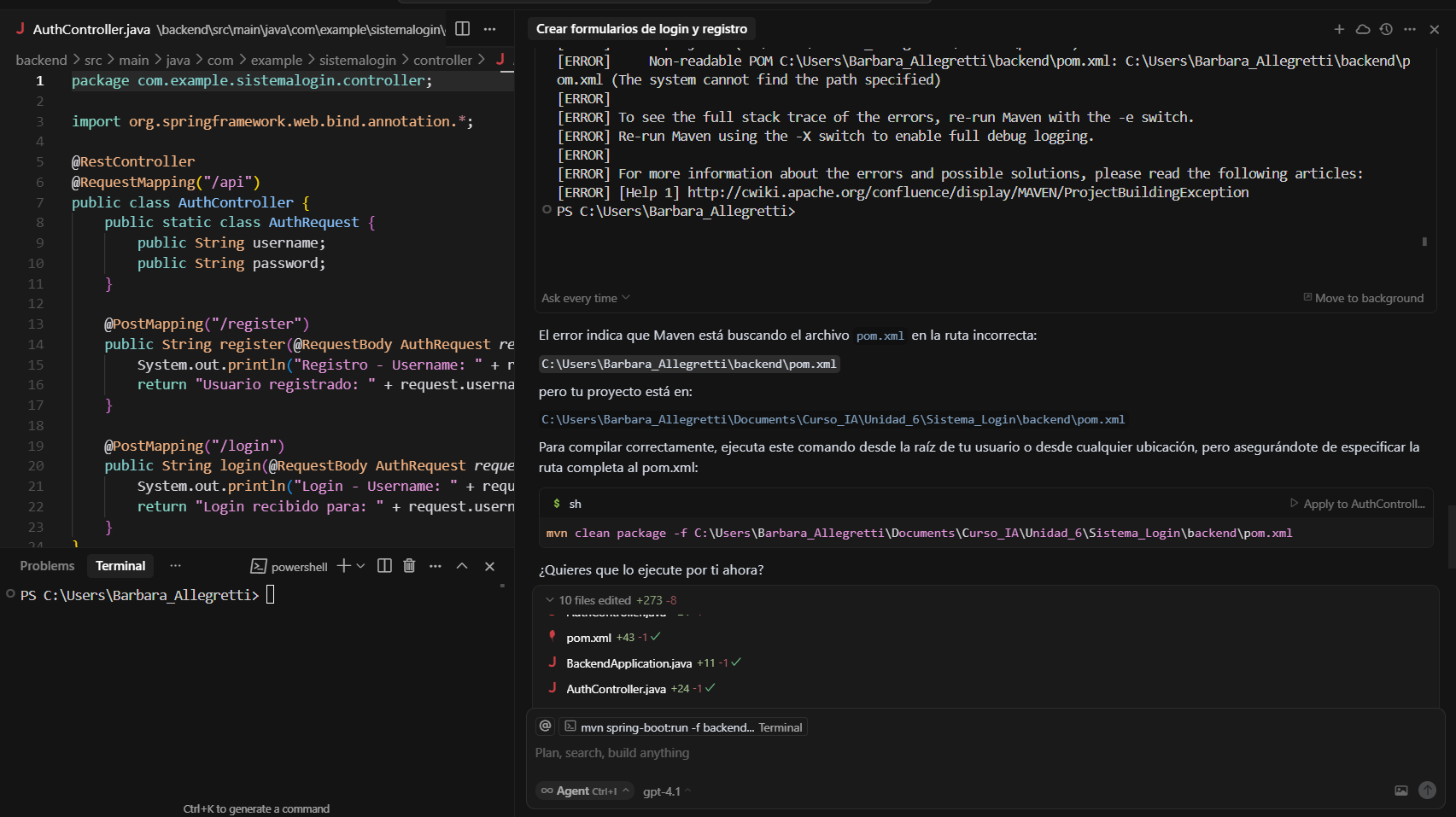


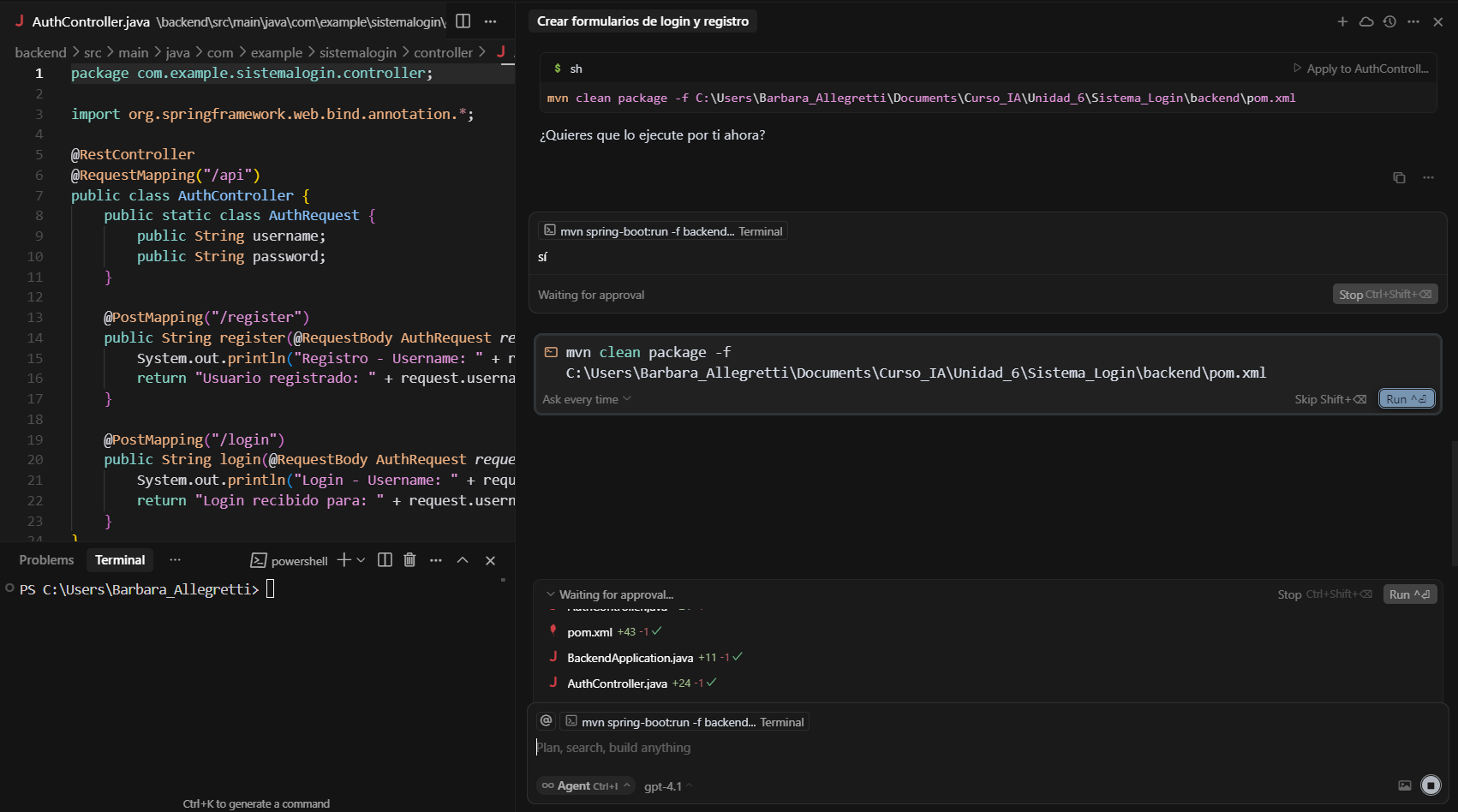
Le dije que sí quería compilar el backend:



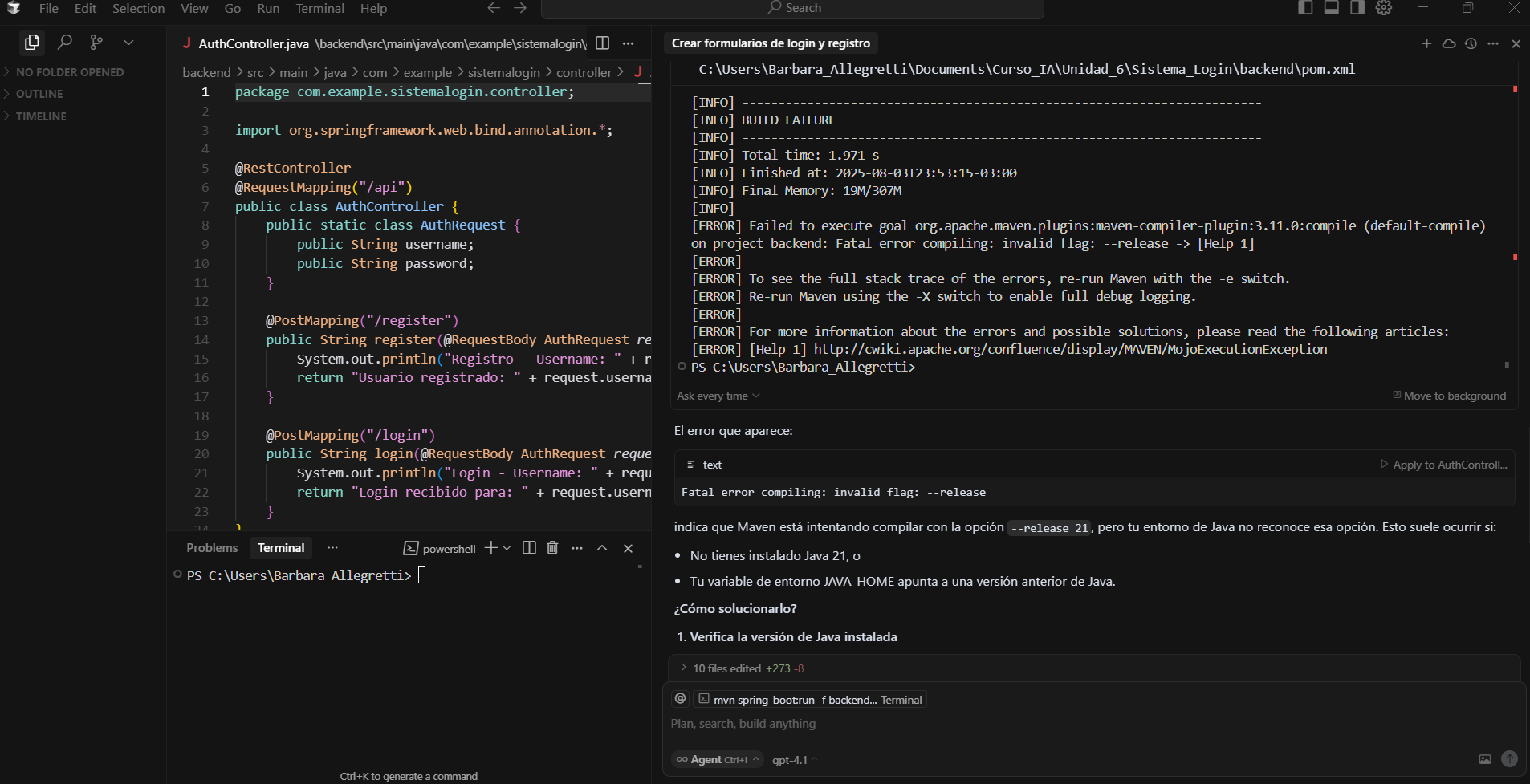
Tiró error al compilar. Le dí al stop y lo quise hacer de vuelta:

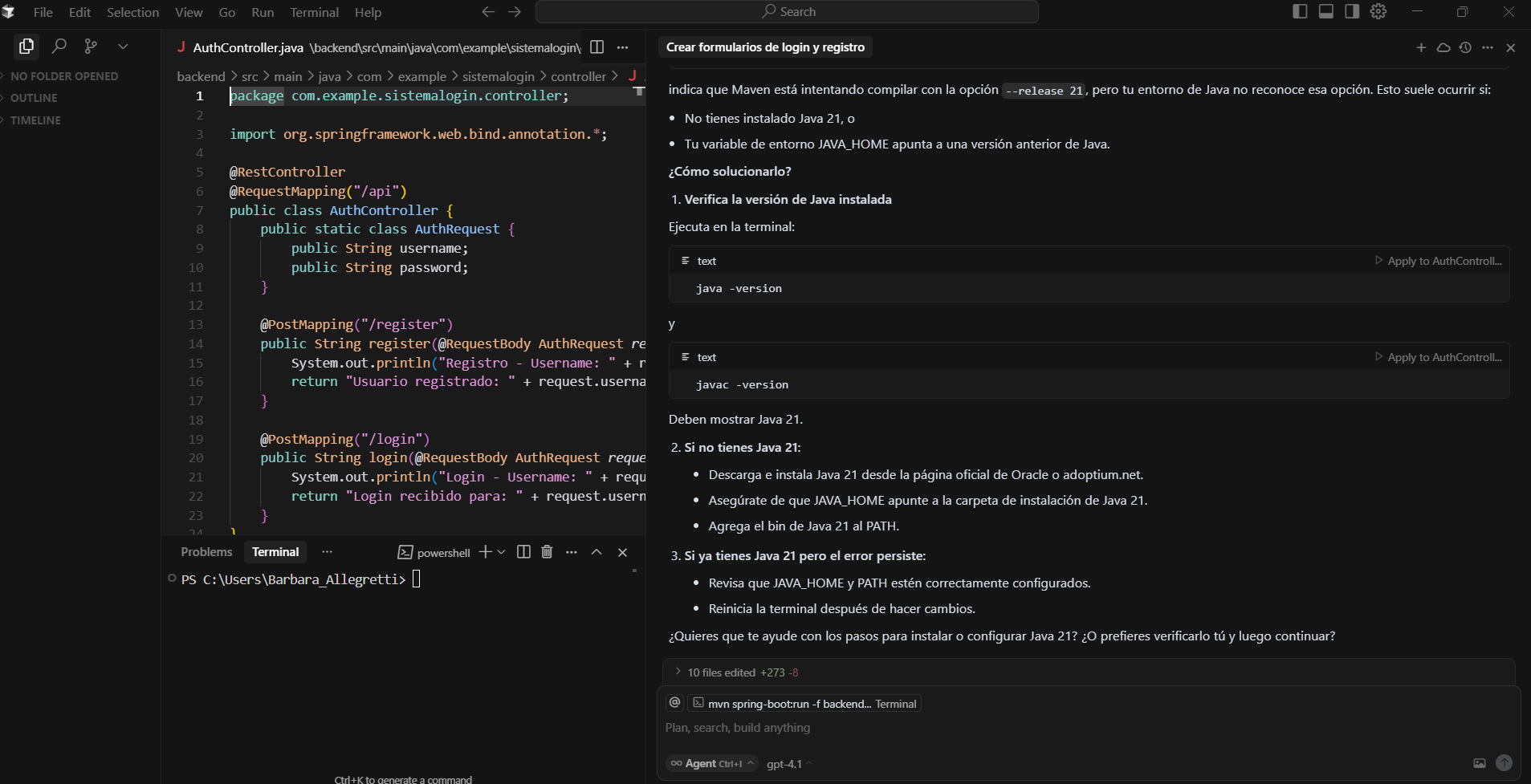




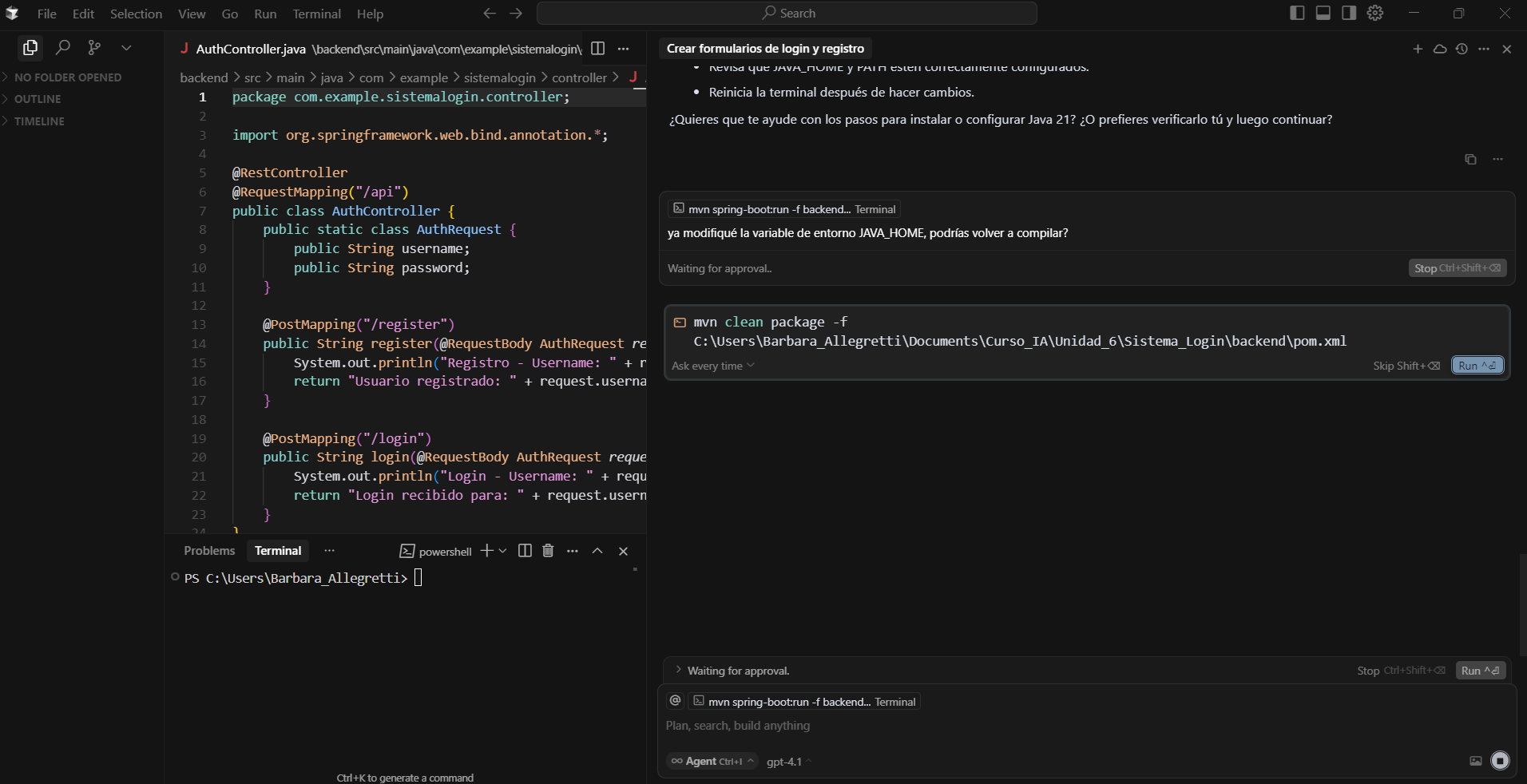


Tenía otra versión de java en las variables de entorno. Indicó cuál era el error:

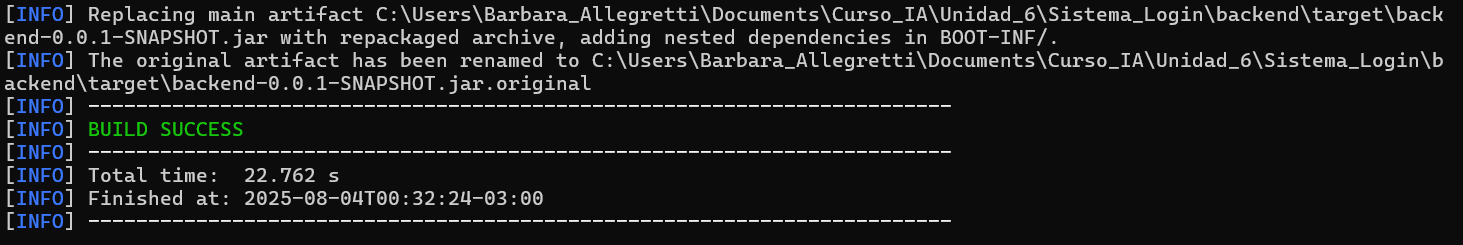




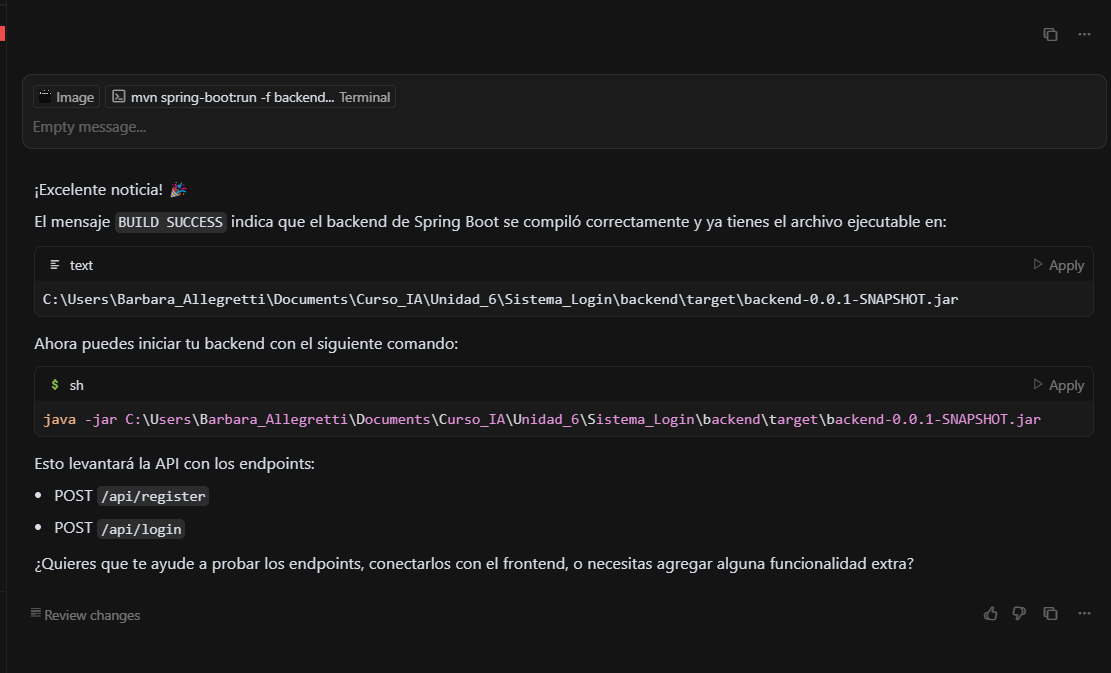
Modifiqué la variable de entorno JAVA\_HOME y le indiqué que volviera a compilar.



Tiraba errores de compilación, por mvn, me indicó que revisara las versiones de java, javac, si se estaba usando esa versión, qué versión de java estaba usando mvn. Luego que verifiqué todo, me indicó que compilara por consola:

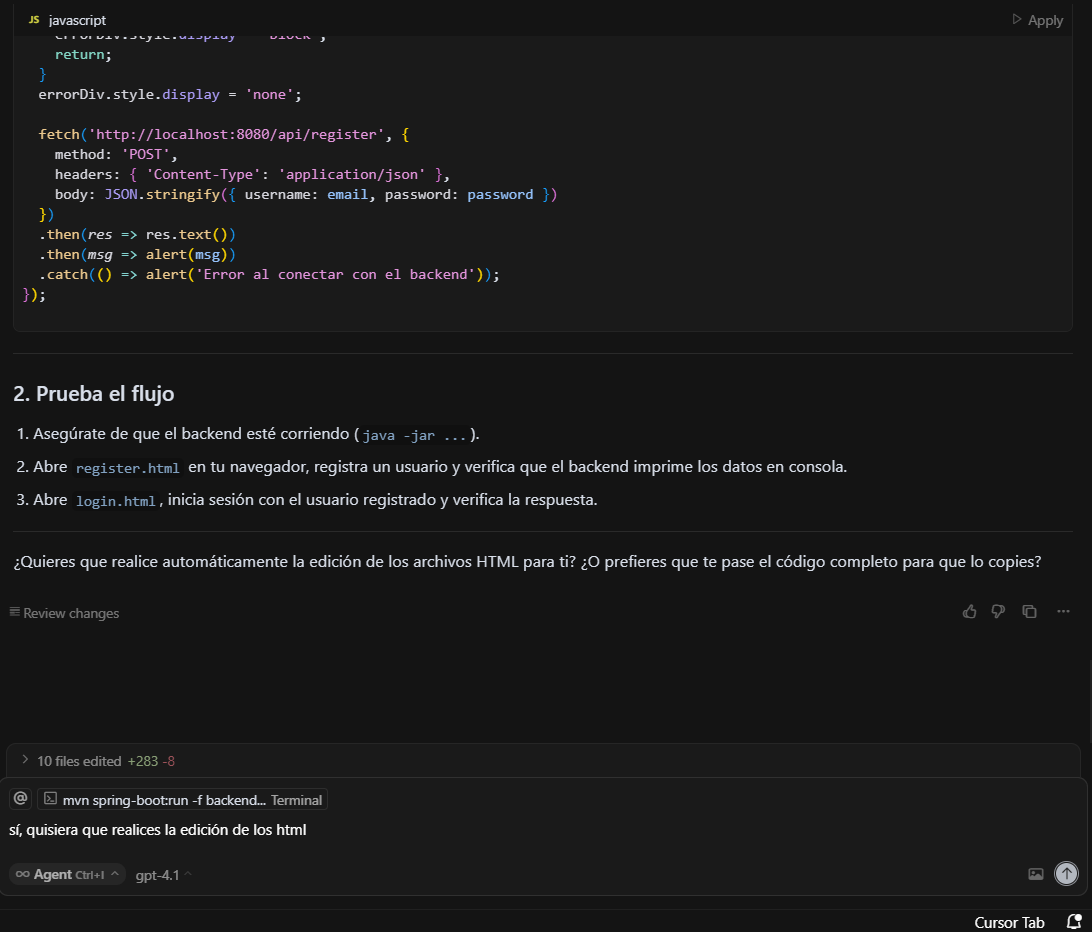


Le subí la imagen del resultado de la compilación:

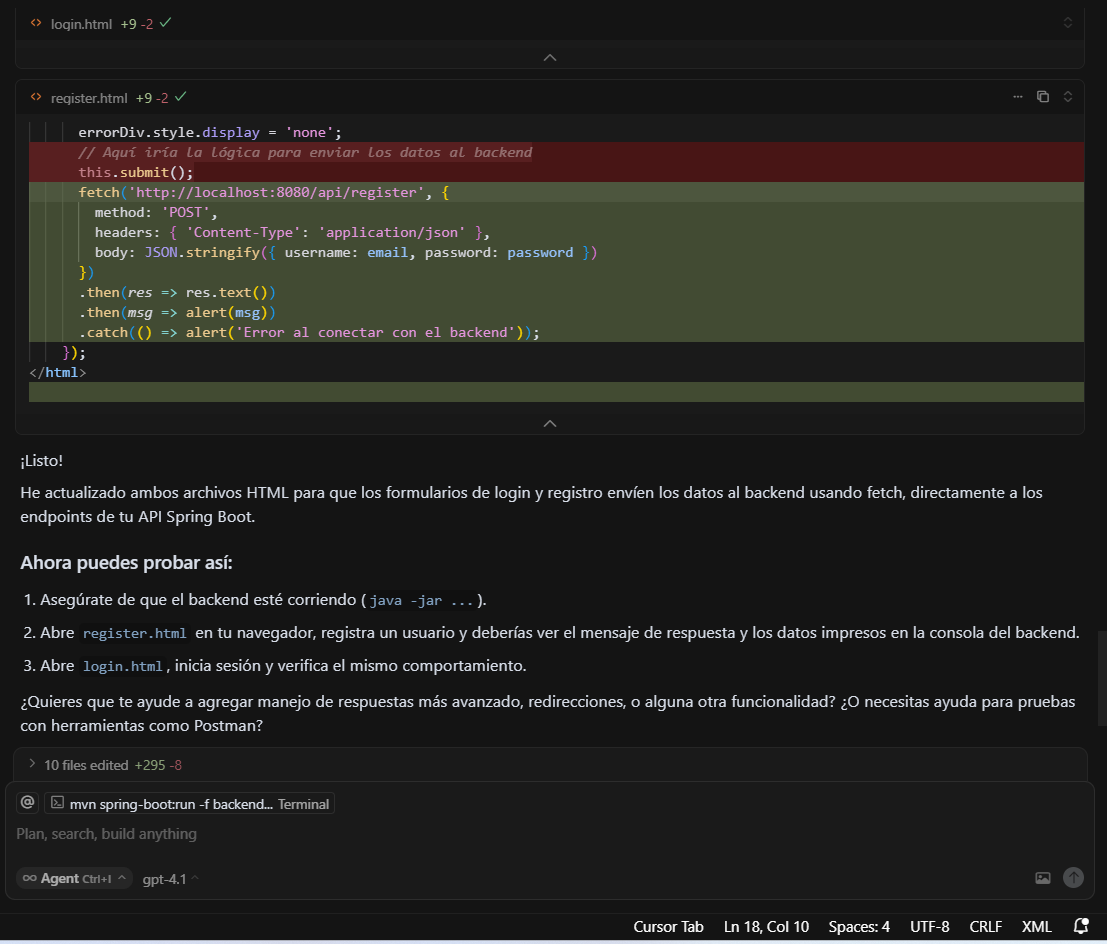


### Tarea 3: Integración Frontend-Backend

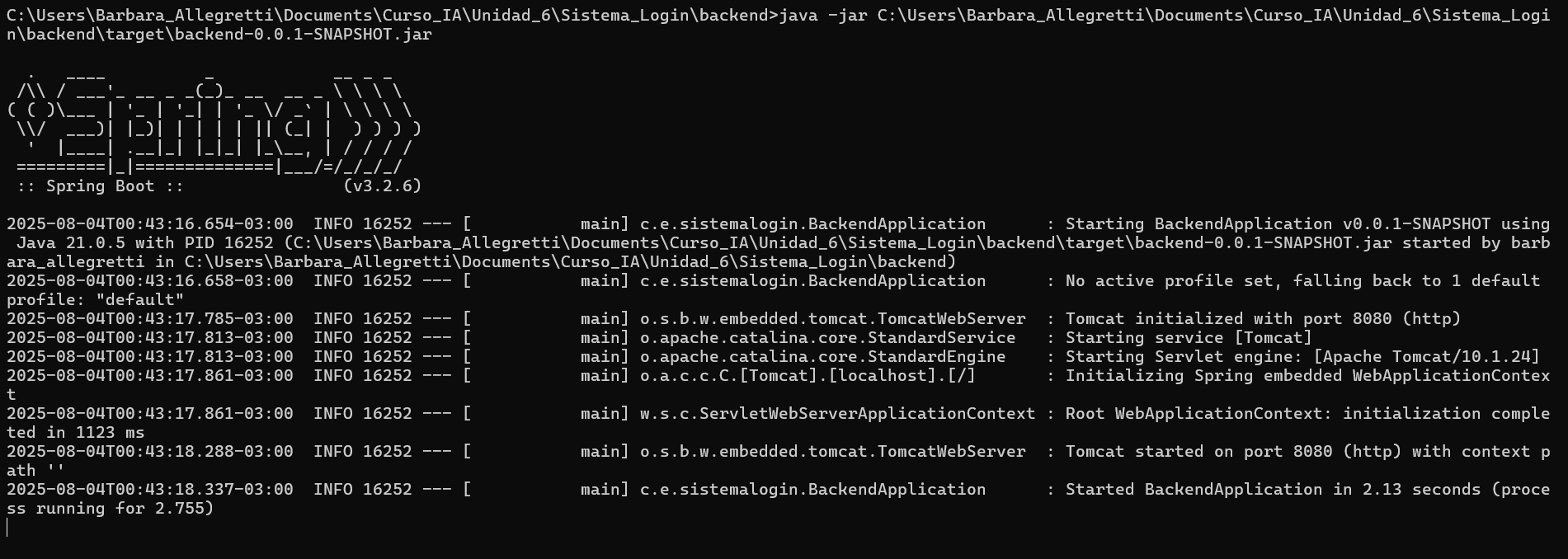
### sí, quisiera conectarlos con el frontend y probarlos



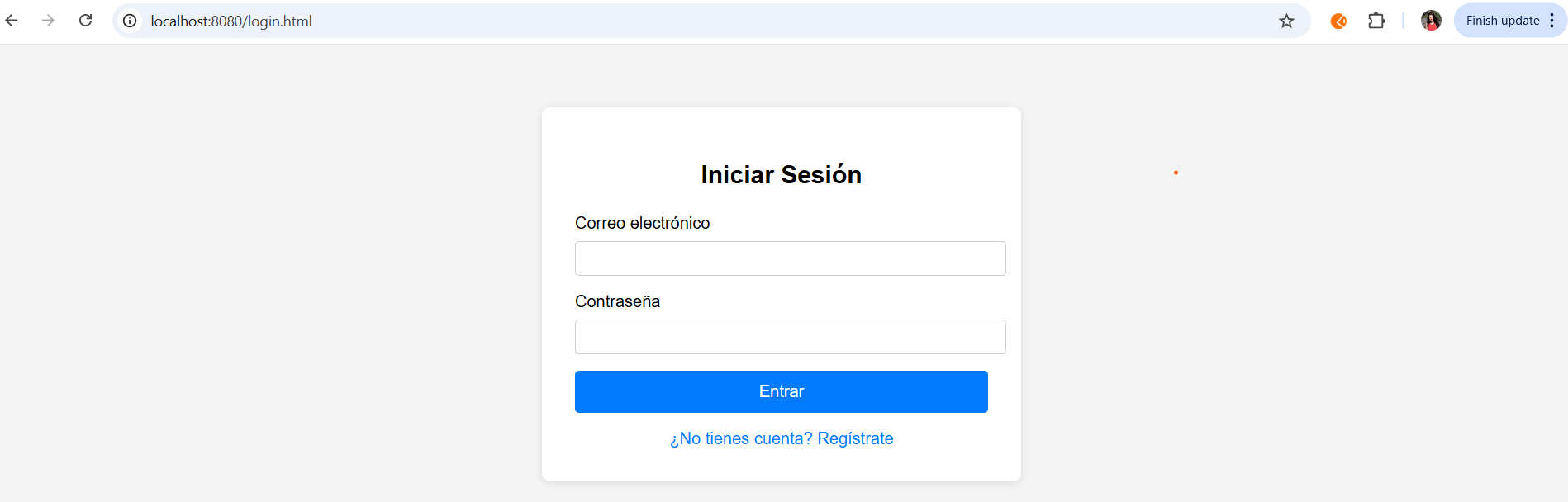
Modificó los html y acepté los cambios:

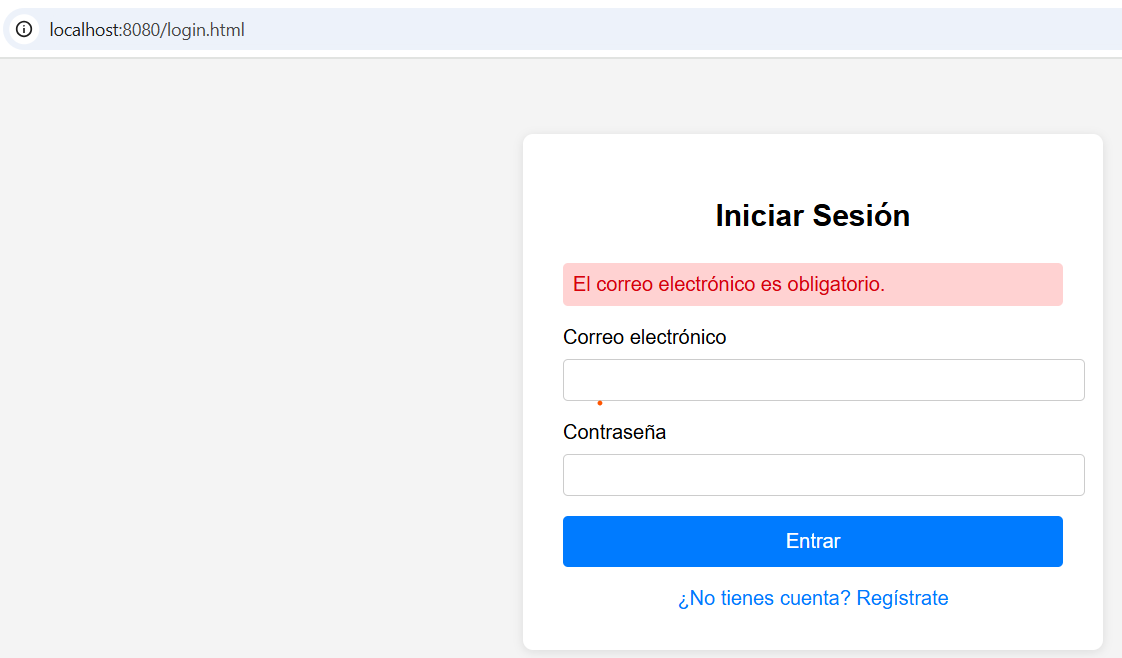


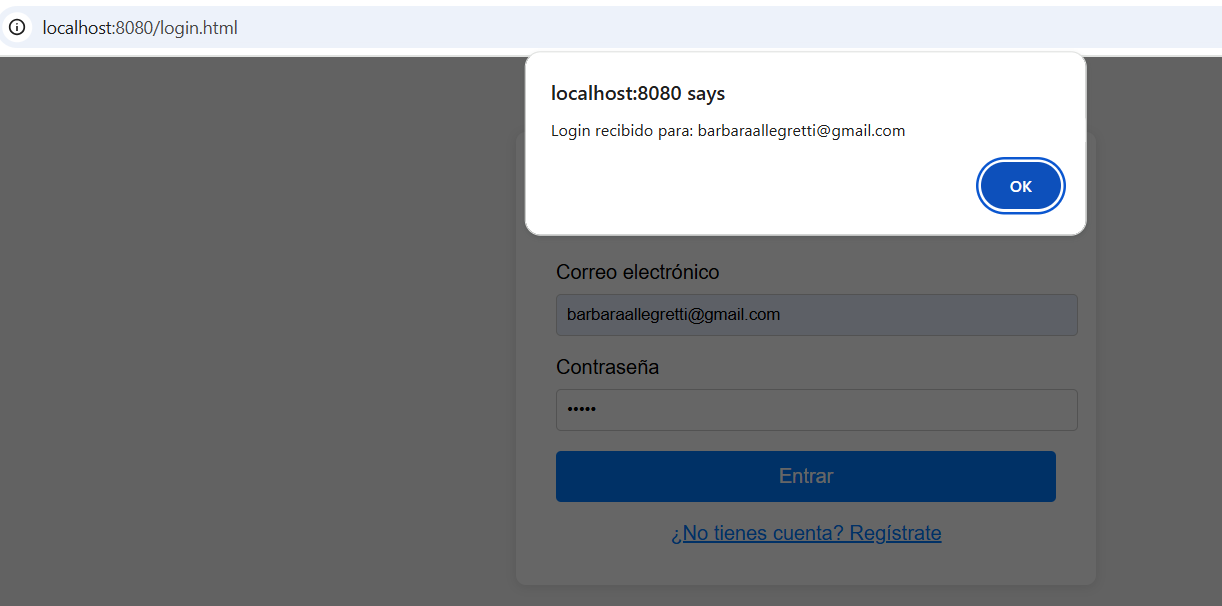
Levanté el backEnd:

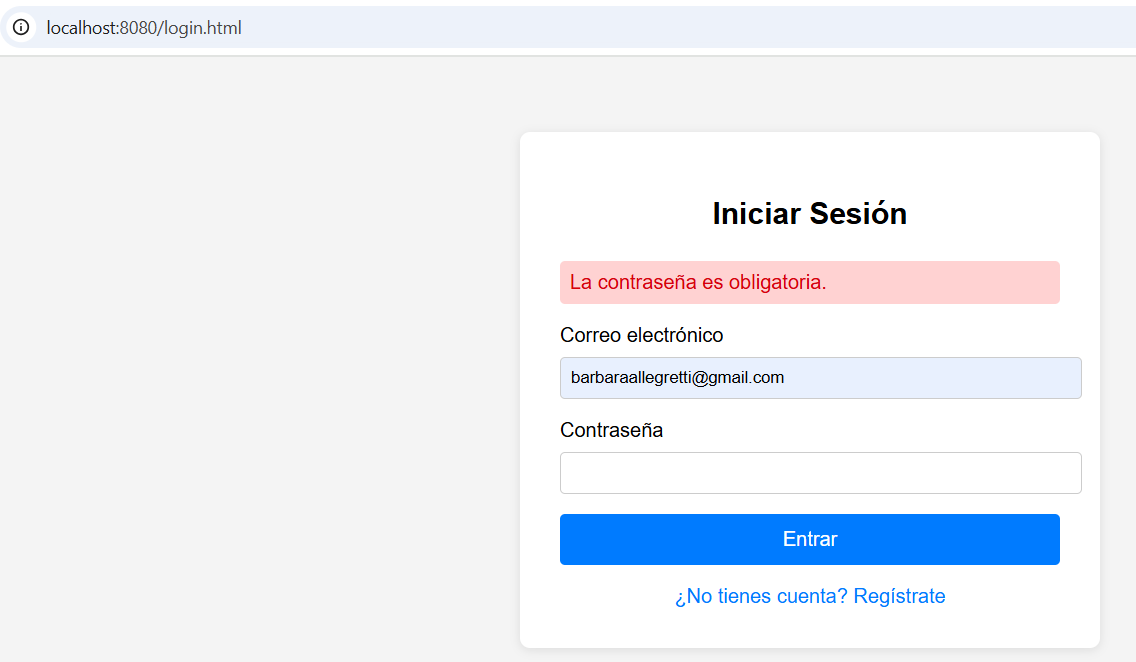


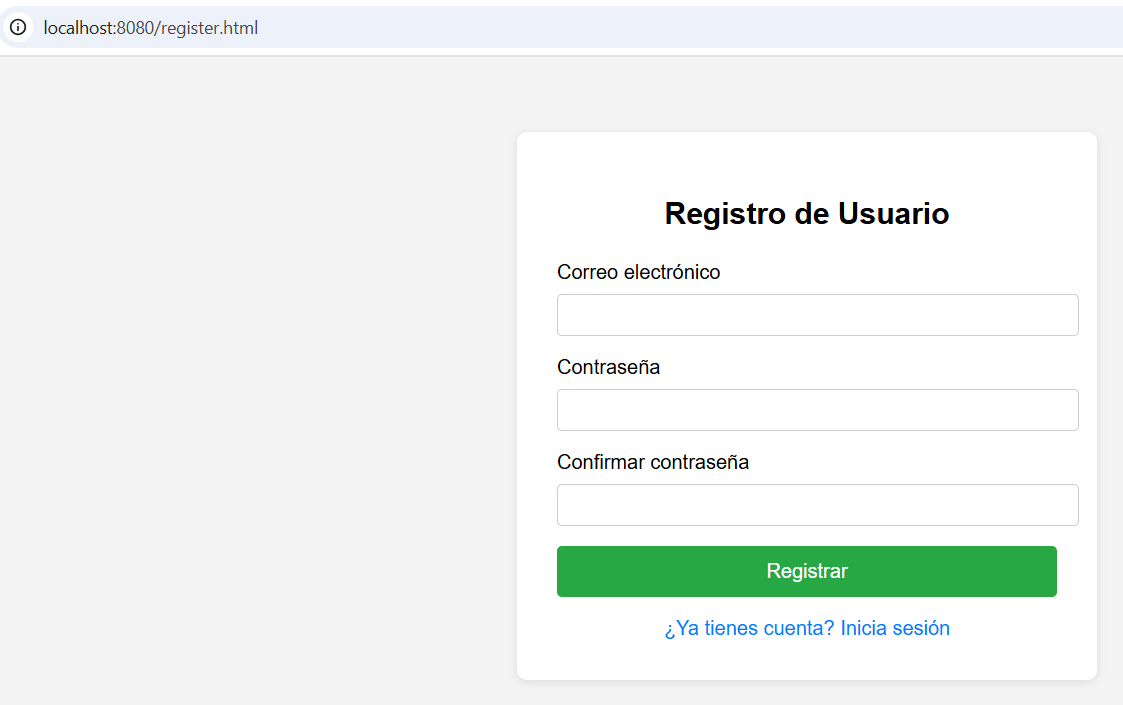
No levantaba en el browser al poner <http://localhost:8080/login.html> ya que necesitaba tener los html en carpeta static dentro de resource (Spring Boot, por convención, sirve archivos estáticos desde: src/main/resources/static/). Moví los archivos a mano, luego tiraba error porque faltaba el application.properties para que levantara los archivos de la carpeta static, lo generó. Tuve que compilar nuevamente por fuera, por consola.

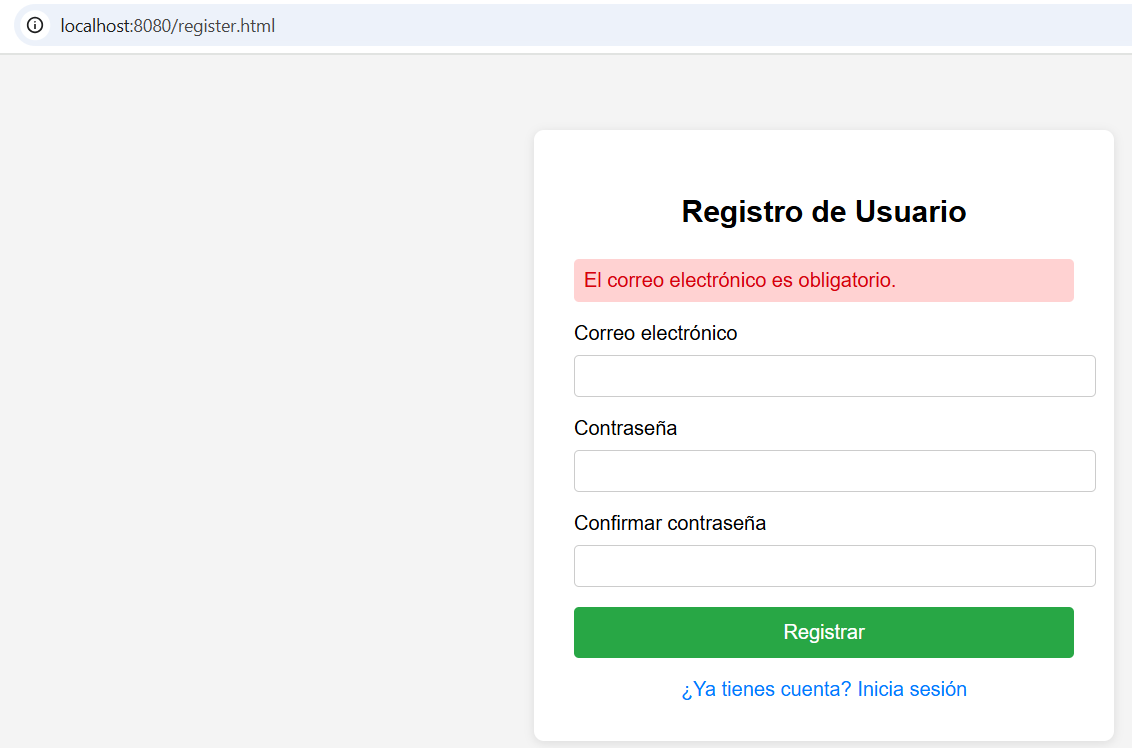


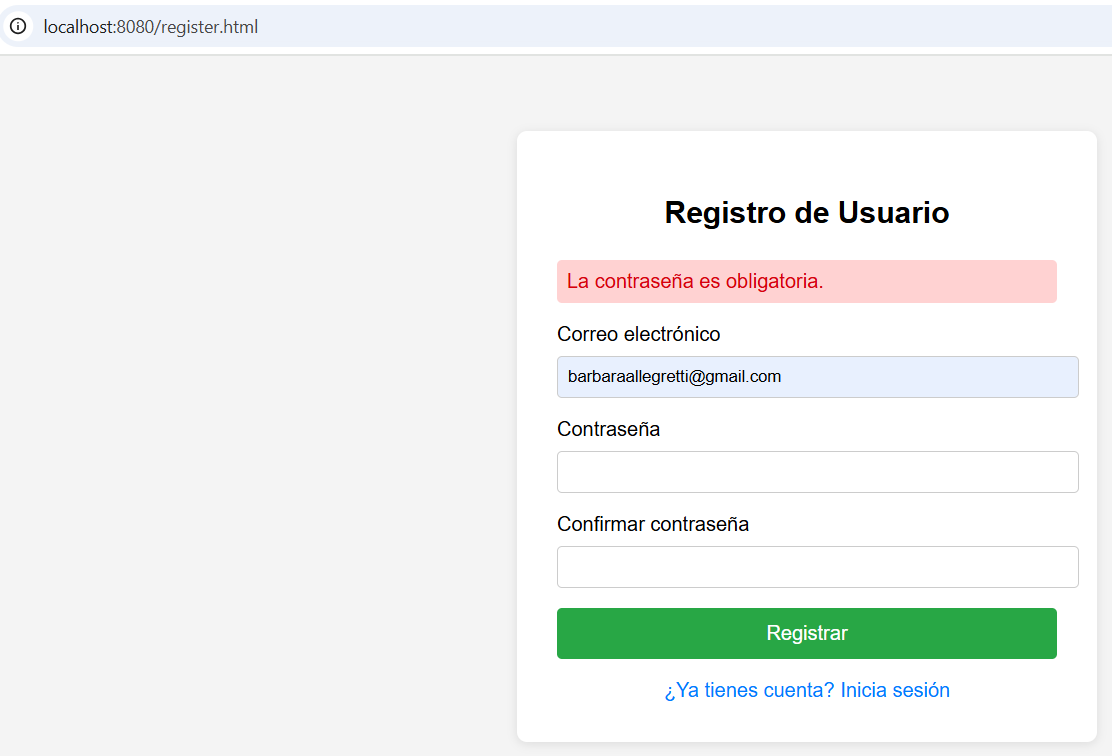


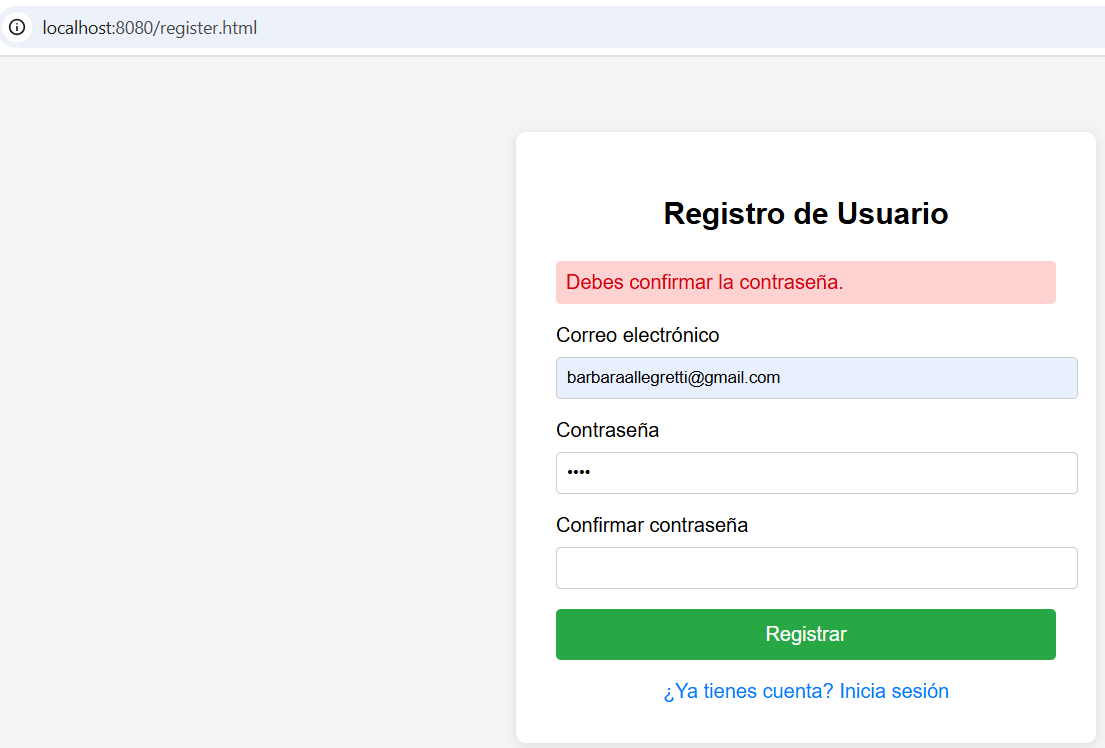


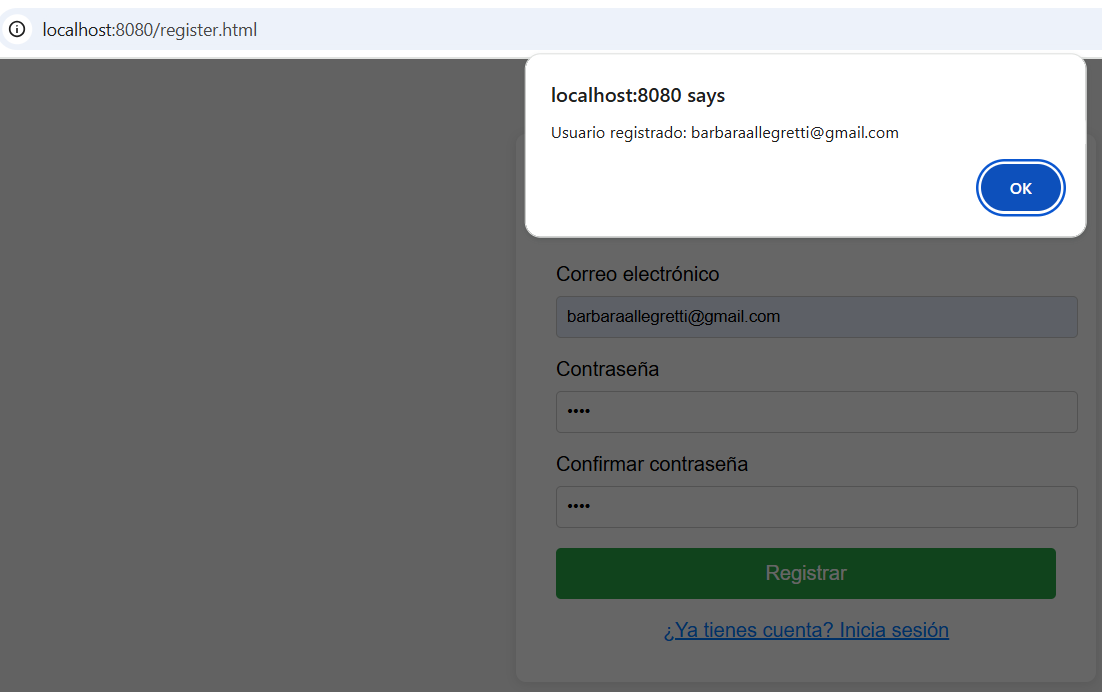








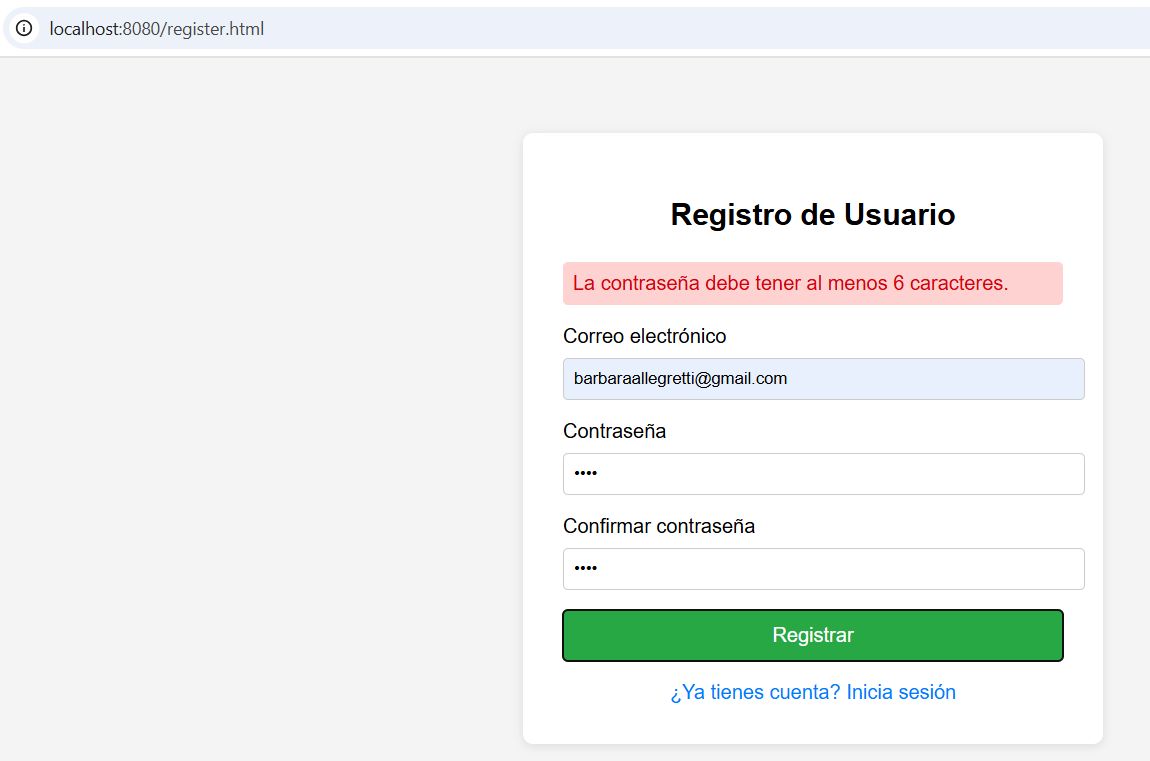


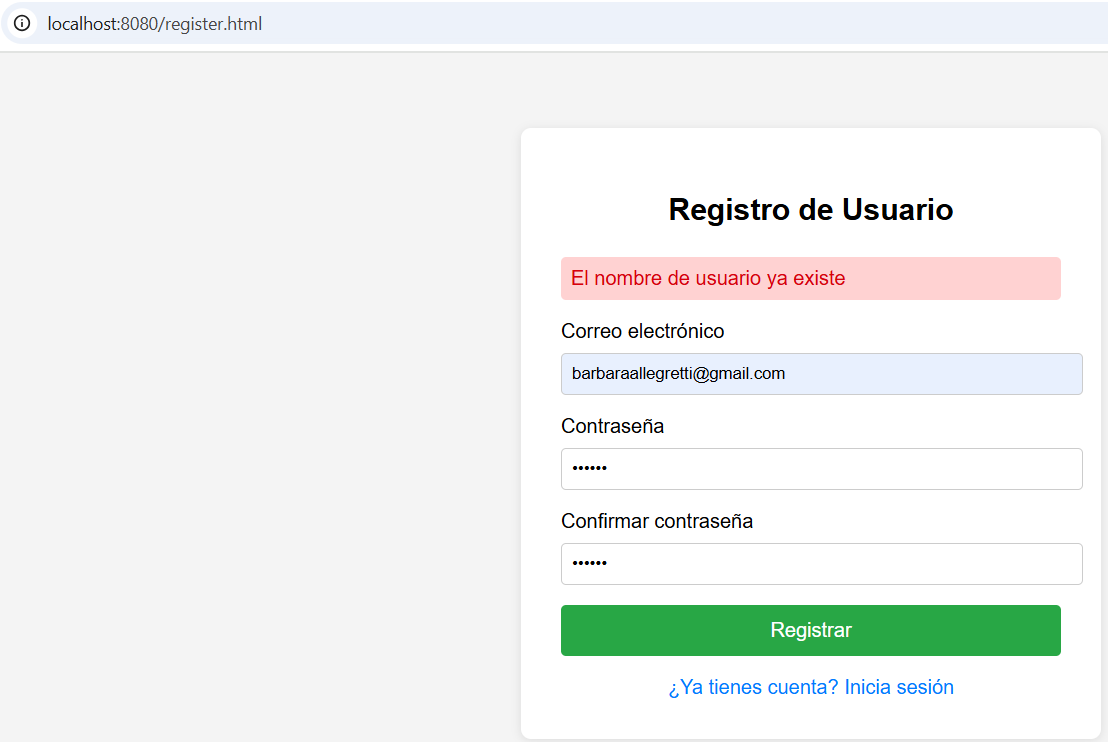


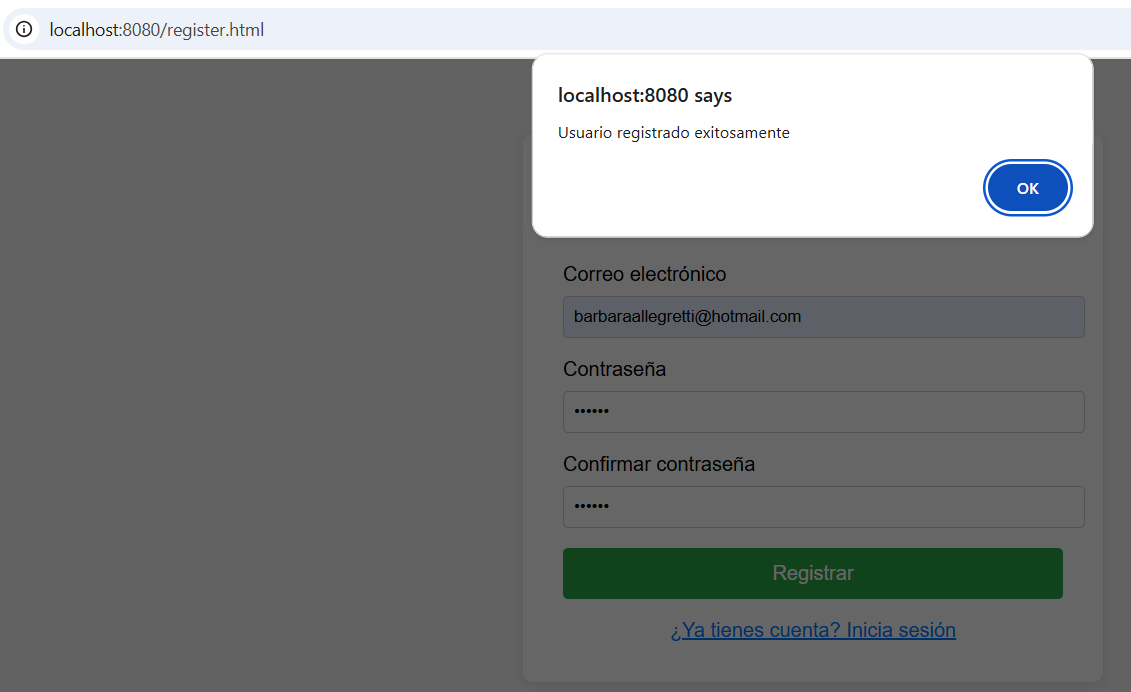
### Tarea 4: Implementar la Lógica de Autenticación y Manejo de Errores

Prompt: ‘necesitaría implementar hashing de contraseñas en el backend al registrar, y verificar contraseñas al iniciar sesión. También mejorar el frontend para mostrar mensajes de error amigables que provengan del backend. En una ruta para /api/register, podrías hashear la contraseña de forma segura usando bcrypt y para /api/login, comparar la contraseña enviada con el hash almacenado?’

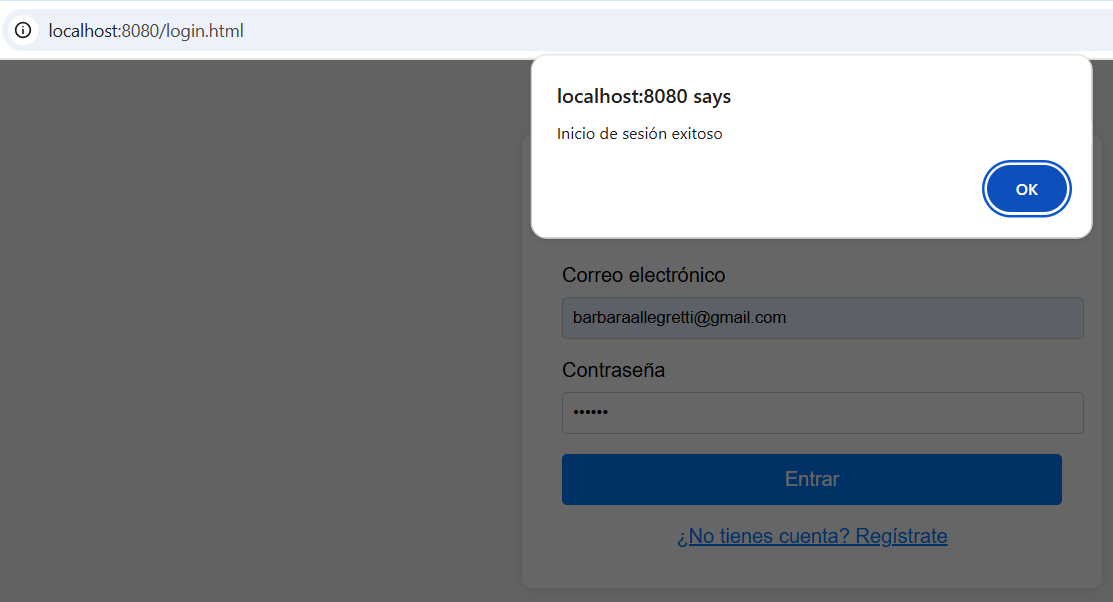
Modificó el pom para agregar la dependencia spring-security-crypto, creó la clase AuthService para hashear las contraseñas, modificó las clases y los html para mostrar los msjs retornados por el backEnd:











### Tarea 5: Agregar Tests Unitarios e Integración

Prompt: ‘necesitaría generar test unitarios y de integración utilizando jUnit. podrías generar tests con jUnit para el endpoint '/api/login'. Cubrí casos de login exitoso, contraseña incorrecta y usuario inexistente.? y también generar los tests para el endPoint '/api/register'?’

Generó tests unitarios y de integración. Modificó el pom. Luego, modificó nuevamente el pom ya que había un error con un annotation que no estaba disponible con la versión de Spring Boot utilizada.

