INFORME DETALLADO DE LOS LTIMOS CAMBIOS DE LOGIN

OBJETIVO GENERAL
Implementar un flujo de login **muy bsico** que permita al frontend validar usuarios ya existentes en el backend sin introducir capas adicionales de seguridad, sesiones ni tokens. El foco fue ofrecer un punto de integracin mnimo para las pruebas manuales del formulario.

CAMBIOS EN EL BACKEND (MODULO ‘USER-SOLUTION‘)
1. **DTO de entrada (‘LoginRequest‘)**
   - Ubicacin: ‘user-solution/src/main/java/edu/uoc/epcsd/user/application/rest/request/LoginRequest.java‘
   - Campos: ‘email‘ y ‘password‘, ambos marcados con ‘@NotBlank‘ para rechazar peticiones vacas a nivel de validacin.
   - Motivo: disponer de un contrato explcito y sencillo que refleje los datos mnimos que enva el frontend.

2. **DTO de salida (‘LoginResponse‘)**
   - Ubicacin: ‘user-solution/src/main/java/edu/uoc/epcsd/user/application/rest/response/LoginResponse.java‘
   - Campos devueltos: ‘id‘, ‘fullName‘, ‘email‘, ‘phoneNumber‘.
   - Motivo: regresar un perfil ligero del usuario autenticado sin exponer la contrasea ni datos innecesarios.

3. **Servicio de autenticacin (‘UserService‘ / ‘UserServiceImpl‘)**
   - Se aadi el mtodo ‘Optional<User> authenticate(String email, String password)‘ en la interfaz y su implementacin.
   - Lgica: buscar por email y filtrar si la contrasea coincide exactamente con la almacenada en memoria.
   - Motivo: reutilizar el repositorio existente para comprobar credenciales sin aadir cifrado ni persistencia adicional, acorde al requerimiento de sencillez.

4. **Endpoint REST ‘/users/login‘**
   - Ubicacin: ‘user-solution/src/main/java/edu/uoc/epcsd/user/application/rest/UserRESTController.java‘.
   - Comportamiento: recibe ‘LoginRequest‘, delega en ‘authenticate‘ y devuelve ‘200 OK‘ con ‘LoginResponse‘ o ‘401 Unauthorized‘ si no coincide.
   - Motivo: exponer un punto de entrada claro para el formulario del frontend, manteniendo la semntica HTTP estndar de xito o credenciales invlidas.

CAMBIOS EN EL FRONTEND (MODULO ‘FRONTEND-SOLUTION‘)
1. **Cliente de usuario (‘src/services/user/api.js‘)**
   - Configura un cliente Axios apuntando a ‘process.env.VUE_USER_API_URL‘ (por defecto ‘http://localhost:18080‘).
   - Expone ‘login(credentials)‘ que enva un POST a ‘/users/login‘ con email y contrasea.
   - Motivo: centralizar la llamada y permitir reconfigurar la URL del backend sin tocar componentes.

2. **Componente de login (‘src/components/UserLogin.vue‘)**
   - Formulario con campos controlados ‘email‘ y ‘password‘, estilos de Bootstrap y mensajes dinmicos.

- Lgica: invoca ‘api.login‘, muestra mensaje de bienvenida con el ‘fullName‘ retornado o un mensaje de error si recibe ‘401‘.
- Motivo: ofrecer una interfaz mnima para probar el endpoint y visualizar rpidamente si las credenciales son vlidas.

3. **Integracin en la aplicacin (‘src/App.vue‘)**
- Se importa y renderiza ‘<UserLogin />‘ junto al formulario de productos existente.
- Motivo: disponer del login en la pantalla principal para facilitar la validacin manual sin navegacin adicional.

FLUJO COMPLETO DE AUTENTICACIN
1. El usuario rellena email y contrasea en ‘UserLogin.vue‘.
2. El componente llama a ‘api.login‘, que enva la peticin al backend de usuarios.
3. ‘UserRESTController‘ recibe la peticin, valida campos y delega en ‘UserService.authenticate‘.
4. ‘UserServiceImpl‘ busca el usuario por email y comprueba si la contrasea coincide exactamente.
5. Si coincide, se responde con ‘LoginResponse‘ (id, nombre, email, telfono); si no, se responde ‘401‘ sin cuerpo.
6. El frontend muestra un mensaje verde de bienvenida o rojo de credenciales incorrectas segn el resultado.

SUPUESTOS Y LIMITACIONES CONSCIENTES
- No se implement cifrado de contraseas ni tokens; las contraseas se comparan en claro porque se pidi un login "sin complicaciones".
- No hay sesiones ni almacenamiento de estado en frontend; solo se muestra el resultado de la autenticacin.
- El catlogo de usuarios proviene del repositorio en memoria ya presente en el proyecto; no se aadi creacin ni semillado extra.

RAZONES DETRS DE LAS DECISIONES
- **Simplicidad operativa:** al reutilizar los repositorios existentes y evitar dependencias nuevas, el endpoint funciona con la configuracin actual del proyecto.
- **Contratos explcitos:** los DTOs de entrada/salida documentan qu espera y qu ofrece el login, facilitando pruebas y depuracin.
- **Retroalimentacin clara en UI:** el componente muestra mensajes directos (xito/error) para validar manualmente sin herramientas externas.

PRUEBAS REALIZADAS
- Se intent ejecutar ‘mvn -q -f user-solution/pom.xml test‘, pero fall por un error 403 al descargar el parent POM desde Maven Central (limitacin del entorno).
- Las pruebas manuales se basan en llamadas desde el formulario al endpoint; no se aadieron tests automatizados especficos para el login.

PRXIMOS PASOS SUGERIDOS
- Sustituir la comparacin de contraseas en claro por hashing (e.g., BCrypt) y almacenamiento cifrado.
- Emitir tokens o sesiones para mantener al usuario autenticado en posteriores peticiones.
- Aadir validaciones de formato de email y reglas de contrasea ms estrictas en frontend y backend.

- Incorporar pruebas unitarias para ‘authenticate‘ y pruebas de integracin para el endpoint ‘/users/login‘.