**Plan de Pruebas — Sistema Portafolio**

**Versión**: mayo 2025  
**Ámbito**: Funcionalidades completas del backend + frontend (proyectos, imágenes, repos, comentarios, autenticación).

**1  Entorno**

|  |  |  |
| --- | --- | --- |
| **Componente** | **URL local** | **Descripción** |
| **Frontend (Vite)** | http://localhost:5173 | UI React 19 |
| **Backend (Express)** | http://localhost:3000 | API REST + archivos |
| **DB** | PostgreSQL 14 | Esquema con tablas \*login, proyectos, |
| proyecto\_\* puente, comentarios |  |  |
| **Uploads** | /uploads/{user}/{proyecto}/… | Sistema de ficheros |

**2  Pruebas Frontend (manual/UI)**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Caso** | **Pasos** | **Resultado esperado** |
| **F‑01** | Crear comentario | Iniciar sesión → /proyectos/43 → escribir texto → **Publicar** | Comentario aparece en primera posición con usuario y fecha actual. |
| **F‑02** | Eliminar comentario propio | Con comentario F‑01 visible → **×** → confirmar | Comentario desaparece sin recargar página. |
| **F‑03** | Cambio de enlace repo refleja archivos | Entrar como autor → Editar proyecto 43 → cambiar repositorio\_github → guardar → volver a vista pública | Lista **Archivos del repo** se vacía y, tras clonado, muestra ficheros del nuevo repo (≤60 s). |
| **F‑04** | Paginación en búsqueda | Buscar “React” → navegar páginas con **Siguiente/Anterior** | Resultados cambian y botón de página actual se marca **active**. |

**3  Pruebas Backend (Postman o curl)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Método / Ruta** | **Setup** | **Paso** | **Esperado** |
| **B‑01** | POST /proyectos | Token válido | Crear proyecto con URL repo → 201 | Carpeta /uploads/{uid}/{id}/repo creada y contiene .git. |
| **B‑02** | PUT /proyectos/43 | Proyecto con repo viejo | Cambiar repositorio\_github | • 200 “actualizado” |
| • Carpeta repo borrada y reclonada. |  |  |  |  |
| **B‑03** | DELETE /comentarios/:id | Crear comentario propio | Borrar con token ajeno | 403 “No autorizado”. |
| **B‑04** | GET /public/proyectos/43 | — | Llamar sin token | 200 JSON con keys imagenes, repo\_files, categorias, etc. |

**4  Pruebas de Seguridad**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Vectores** | **Paso de prueba** | **Resultado** |
| **S‑01 SQLi** | Endpoint **POST /proyectos/:id/comentarios** | Body { "contenido": "'; DROP TABLE login; --" } con token válido | 400 o 422; la tabla *login* sigue existiendo (verificar en psql); entrada no insertada. |
| **S‑02 SQLi** | **PUT /proyectos/43** campo titulo | Inyectar Test'); UPDATE proyectos SET publico=false; -- | Respuesta 400; ningún campo externo modificado. |
| **S‑03 Path Traversal** | **GET /uploads/../../etc/passwd** directamente por navegador | El servidor estático prohíbe salir de /uploads; responde 403 o 404. |  |

**5  Pruebas No Funcionales (spot‑check)**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Métrica** | **Herramienta** | **Criterio** |
| **NF‑01** | Tiempo clonado repo | Cronómetro | ≤ 60 s para repos < 50 MB. |
| **NF‑02** | Uso de conexión PG | pg\_stat\_activity | < 10 conexiones simultáneas bajo carga normal. |

**6  Procedimiento de Ejecución**

1. Migrar BD y arrancar backend npm start.
2. Arrancar frontend npm run dev.
3. Seguir tabla en orden, registrando **resultado**, **fecha** y **tester**.
4. Tickets JIRA se crean para cualquier fallo.

Fin del plan.