

README_TEST_FRONT - Pruebas completas (Frontend + Backend + BD)

1. Objetivo

Este manual guía al estudiante a **probar todo el software** paso a paso:

- Frontend funcionando
- Backend funcionando
- Endpoints funcionando
- Validación en base de datos (PostgreSQL) en cada cambio

Cada paso incluye **qué hacer**, **qué esperar** y **cómo validar en la BD con SQL**.

2. Requisitos previos

- PostgreSQL instalado y corriendo
 - Python y Node.js instalados
 - Base de datos creada (ejemplo: **universidad**)
-

3. Configuración mínima

3.1 Backend (**.env** en la raíz)

```
APP_ENV=development
APP_DATABASE_URL=postgresql+psycopg://usuario:password@localhost:5432/universidad
APP_JWT_SECRET=change_me
APP_CORS_ORIGINS=["http://localhost:3000"]
```

3.2 Frontend (**frontend/.env**)

```
VITE_API_BASE_URL=http://127.0.0.1:8000
```

4. Paso 0 - Crear usuario Administrador (BD)

Objetivo: tener un admin para iniciar sesión.

Ejecutar en PostgreSQL:

```
CREATE EXTENSION IF NOT EXISTS pgcrypto;

INSERT INTO roles (name, description)
SELECT 'Administrador', 'Rol administrador'
WHERE NOT EXISTS (SELECT 1 FROM roles WHERE name = 'Administrador');
```

```
INSERT INTO roles (name, description)
SELECT 'Docente', 'Rol docente'
WHERE NOT EXISTS (SELECT 1 FROM roles WHERE name = 'Docente');

INSERT INTO roles (name, description)
SELECT 'Estudiante', 'Rol estudiante'
WHERE NOT EXISTS (SELECT 1 FROM roles WHERE name = 'Estudiante');

INSERT INTO users (email, full_name, hashed_password, is_active)
SELECT 'admin@ud.edu', 'Admin', crypt('AdminPass1234', gen_salt('bf')), true
WHERE NOT EXISTS (SELECT 1 FROM users WHERE email = 'admin@ud.edu');

INSERT INTO user_roles (user_id, role_id)
SELECT u.id, r.id
FROM users u, roles r
WHERE u.email = 'admin@ud.edu' AND r.name = 'Administrador'
AND NOT EXISTS (
  SELECT 1 FROM user_roles ur
  WHERE ur.user_id = u.id AND ur.role_id = r.id
);
```

Validación BD:

```
SELECT id, email, is_active FROM users WHERE email = 'admin@ud.edu';
SELECT r.name FROM roles r
JOIN user_roles ur ON ur.role_id = r.id
JOIN users u ON u.id = ur.user_id
WHERE u.email = 'admin@ud.edu';
```

5. Paso 1 - Levantar backend

```
cd backend
python -m venv .venv
.venv\Scripts\activate
pip install -r requirements.txt
uvicorn app.main:app --reload
```

Validación: abre <http://127.0.0.1:8000/docs>.

6. Paso 2 - Levantar frontend

```
cd frontend
npm install
npm run dev
```

Validación: abre <http://localhost:3000>.

7. Paso 3 - Login Admin (Frontend)

1. Entra a <http://localhost:3000/login>
2. Usuario: [admin@ud.edu](#)
3. Clave: [AdminPass1234](#)

Validación: redirige a [/admin](#).

8. Paso 4 - Crear Docente y Estudiante (Frontend + BD)

8.1 Crear Docente

En el panel Admin → Usuarios:

- Email: [docente@ud.edu](#)
- Nombre: [Docente Uno](#)
- Password: [DocentePass1234](#)
- Rol: Docente

Validación en BD:

```
SELECT id, email, is_active FROM users WHERE email = 'docente@ud.edu';
SELECT r.name FROM roles r
JOIN user_roles ur ON ur.role_id = r.id
JOIN users u ON u.id = ur.user_id
WHERE u.email = 'docente@ud.edu';
```

8.2 Crear Estudiante

En el mismo formulario:

- Email: [estudiante@ud.edu](#)
- Nombre: [Estudiante Uno](#)
- Password: [EstudiantePass1234](#)
- Rol: Estudiante

Validación en BD:

```
SELECT id, email, is_active FROM users WHERE email = 'estudiante@ud.edu';
SELECT r.name FROM roles r
JOIN user_roles ur ON ur.role_id = r.id
JOIN users u ON u.id = ur.user_id
WHERE u.email = 'estudiante@ud.edu';
```

9. Paso 5 - Crear Materia (Frontend + BD)

Panel Admin → Materias:

- Código: **MAT101**
- Nombre: **Cálculo I**
- Créditos: **4**

Validación BD:

```
SELECT * FROM subjects ORDER BY id DESC LIMIT 1;
```

10. Paso 6 - Crear Periodo académico (Frontend + BD)

Panel Admin → Periodos:

- Código: **2026-1**
- Nombre: **Periodo 2026-1**
- Inicio: **2026-01-01**
- Fin: **2026-06-30**

Validación BD:

```
SELECT * FROM periods ORDER BY id DESC LIMIT 1;
```

11. Paso 7 - Crear Inscripción (Frontend + BD)

Panel Admin → Inscripciones:

- Estudiante: **Estudiante Uno**
- Materia: **Cálculo I**
- Periodo: **Periodo 2026-1**

Validación BD:

```
SELECT * FROM enrollments ORDER BY id DESC LIMIT 1;
```

12. Paso 8 - Registrar Calificación (Frontend + BD)

Panel Admin → Calificaciones:

- Inscripción: **#1** (o la última creada)
- Nota: **95.5**
- Notas: **Parcial 1**

Validación BD:

```
SELECT * FROM grades ORDER BY id DESC LIMIT 1;
```

13. Paso 9 - Ver panel Estudiante (Frontend + BD)

1. Cierra sesión
2. Inicia sesión como **estudiante@ud.edu**

Revisar en:

- Materias (**/student/subjects**)
- Inscripciones (**/student/enrollments**)
- Calificaciones (**/student/grades**)

Validación BD (opcional):

```
SELECT id FROM users WHERE email = 'estudiante@ud.edu';  
SELECT * FROM enrollments WHERE user_id = <ID_ESTUDIANTE>;  
SELECT * FROM grades WHERE enrollment_id IN (  
    SELECT id FROM enrollments WHERE user_id = <ID_ESTUDIANTE>  
);
```

14. Paso 10 - Ver panel Docente (Frontend + BD)

1. Cierra sesión
2. Inicia sesión como **docente@ud.edu**

Revisar en:

- Calificaciones (**/teacher/grades**)

Validación BD:

```
SELECT * FROM grades ORDER BY id DESC LIMIT 5;
```

15. Paso 11 - Activar/Desactivar usuario (Frontend + BD)

En Admin → Usuarios:

- Desactiva al docente
- Luego vuelve a activarlo

Validación BD:

```
SELECT email, is_active FROM users
WHERE email IN ('docente@ud.edu', 'estudiante@ud.edu');
```

16. Paso 12 - Actualizar Materia (Frontend + BD)

Panel Admin → Materias:

- En "Actualizar materia", usa el ID creado.
- Cambia el nombre a **Cálculo I (Actualizado)** y créditos a **5**.

Validación BD:

```
SELECT id, code, name, credits FROM subjects WHERE id = <ID_MATERIA>;
```

17. Paso 13 - Desactivar Materia (Frontend + BD)

En el listado, pulsa **Desactivar** sobre la materia.

Validación BD:

```
SELECT id, is_active FROM subjects WHERE id = <ID_MATERIA>;
```

18. Paso 14 - Actualizar Periodo (Frontend + BD)

Panel Admin → Periodos:

- Cambia el nombre a **Periodo 2026-1 (Actualizado)**
- Cambia fecha fin a **2026-07-15**

Validación BD:

```
SELECT id, name, start_date, end_date FROM periods WHERE id = <ID_PERIODO>;
```

19. Paso 15 - Desactivar Periodo (Frontend + BD)

En el listado de periodos, pulsa **Desactivar**.

Validación BD:

```
SELECT id, is_active FROM periods WHERE id = <ID_PERIODO>;
```

20. Paso 16 - Cancelar Inscripción (Frontend + BD)

En el listado de inscripciones, pulsa **Cancelar**.

Validación BD:

```
SELECT id, is_active FROM enrollments WHERE id = <ID_INSCRIPCION>;
```

21. Paso 17 - Actualizar Calificación (Frontend + BD)

Panel Admin → Calificaciones:

- ID: <ID_CALIFICACION>
- Nota: 98.0
- Notas: Actualización

Validación BD:

```
SELECT id, value, notes FROM grades WHERE id = <ID_CALIFICACION>;
```

22. Paso 18 - Logout y sesión expirada (Frontend)

1. En cualquier panel, pulsa **Cerrar sesión**.
2. Intenta entrar a `/admin` directamente.

Resultado esperado: redirección a `/login`.

23. Paso 19 - Login fallido (Frontend + BD)

En `/login` usa contraseña incorrecta.

Resultado esperado: mensaje de credenciales inválidas.

Validación BD (opcional):

```
SELECT email, is_active FROM users WHERE email = 'admin@ud.edu';
```

24. Paso 20 - Restricciones por rol (Frontend)

1. Inicia sesión como **Estudiante**.
2. Intenta entrar manualmente a `/admin`.

Resultado esperado: redirección a `/denied`.

25. Paso 21 - Propiedad de datos (Backend + BD)

Con el estudiante autenticado:

- En `/student/grades` solo debe ver sus calificaciones.

Validación BD:

```
SELECT id FROM users WHERE email = 'estudiante@ud.edu';
SELECT * FROM grades WHERE enrollment_id IN (
  SELECT id FROM enrollments WHERE user_id = <ID_ESTUDIANTE>
);
```

26. Paso 22 - Re-activar Materia y Periodo (Frontend + BD)

Repite los pasos de actualización para dejar:

- Materia activa
- Periodo activo

Validación BD:

```
SELECT id, is_active FROM subjects WHERE id = <ID_MATERIA>;
SELECT id, is_active FROM periods WHERE id = <ID_PERIODO>;
```

27. Paso 23 - Integridad de datos (BD)

Verifica relaciones entre tablas:

```
SELECT e.id, u.email, s.name, p.name
FROM enrollments e
JOIN users u ON u.id = e.user_id
JOIN subjects s ON s.id = e.subject_id
JOIN periods p ON p.id = e.period_id
ORDER BY e.id DESC LIMIT 5;
```

28. Paso 24 - Pruebas negativas: duplicados (Frontend + BD)

28.1 Usuario duplicado

Intenta crear un usuario con email `docente@ud.edu` otra vez.

Resultado esperado: error por email duplicado.

Validación BD:

```
SELECT COUNT(*) FROM users WHERE email = 'docente@ud.edu';
```

28.2 Materia duplicada (mismo código)

Crea una materia con código `MAT101` nuevamente.

Resultado esperado: error por conflicto de negocio.

Validación BD:

```
SELECT COUNT(*) FROM subjects WHERE code = 'MAT101';
```

29. Paso 25 - Pruebas negativas: validaciones (Frontend + BD)

29.1 Usuario con contraseña corta

Intenta crear usuario con password `123`.

Resultado esperado: error de validación.

Validación BD:

```
SELECT COUNT(*) FROM users WHERE email = 'nuevo@ud.edu';
```

29.2 Materia con créditos inválidos

Intenta crear materia con créditos `0`.

Resultado esperado: error de validación.

Validación BD:

```
SELECT COUNT(*) FROM subjects WHERE code = 'MAT999';
```

29.3 Periodo con fechas inválidas

Intenta crear periodo con fecha fin menor a inicio.

Resultado esperado: error de validación.

Validación BD:

```
SELECT COUNT(*) FROM periods WHERE code = '2026-0';
```

30. Paso 26 - Pruebas de permisos (Frontend)

30.1 Estudiante intenta crear materia

Con estudiante autenticado, intenta entrar a </admin/subjects>.

Resultado esperado: redirección a </denied>.

30.2 Docente intenta crear usuario

Con docente autenticado, intenta entrar a </admin/users>.

Resultado esperado: redirección a </denied>.

31. Paso 27 - Verificación de logout total (Frontend + BD)

1. Inicia sesión como admin.
2. Cierra sesión.
3. Intenta ir a </admin>.

Resultado esperado: redirección a </login>.

Validación BD (opcional):

```
SELECT id, email FROM users WHERE email = 'admin@ud.edu';
```

32. Paso 28 - Validación de consistencia (BD)

Verifica que los registros "inactivos" no desaparecen:

```
SELECT id, is_active FROM subjects WHERE id = <ID_MATERIA>;  
SELECT id, is_active FROM periods WHERE id = <ID_PERIODO>;  
SELECT id, is_active FROM enrollments WHERE id = <ID_INSCRIPCION>;
```

33. Paso 29 - Consulta de auditoría básica (BD)

Confirma cantidades finales:

```
SELECT COUNT(*) AS total_users FROM users;  
SELECT COUNT(*) AS total_subjects FROM subjects;  
SELECT COUNT(*) AS total_periods FROM periods;  
SELECT COUNT(*) AS total_enrollments FROM enrollments;  
SELECT COUNT(*) AS total_grades FROM grades;
```

34. Checklist final

- ☐ Backend funcionando
- ☐ Frontend funcionando
- ☐ Login Admin correcto
- ☐ Docente creado y visible en BD
- ☐ Estudiante creado y visible en BD
- ☐ Materia creada y visible en BD
- ☐ Materia actualizada y desactivada
- ☐ Periodo creado y visible en BD
- ☐ Periodo actualizado y desactivado
- ☐ Inscripción creada y visible en BD
- ☐ Inscripción cancelada y validada en BD
- ☐ Calificación creada y visible en BD
- ☐ Calificación actualizada y validada en BD
- ☐ Panel Estudiante muestra sus datos
- ☐ Panel Docente muestra calificaciones
- ☐ Activar/Desactivar usuario funciona
- ☐ Logout redirige a login
- ☐ Restricción por rol funciona
- ☐ Pruebas negativas (duplicados) cubiertas
- ☐ Pruebas negativas (validaciones) cubiertas
- ☐ Integridad y conteos finales en BD correctos