



ESTRUCTURA DEL PROYECTO - ENTREGA PEC2

Fecha: 28 de octubre de 2025

Proyecto: GaliTroco - Plataforma de Intercambio de Habilidades

Estudiante: Antonio Campos

ESTRUCTURA ACTUAL

```
probatfm/
├── api.php
└── backend/api/
    ├── README_PEC2.md
    ├── README_PEC2.pdf
    ├── README.md
    ├── TESTING_Y_ENDPOINTS_TFM.md
    ├── TESTING_Y_ENDPOINTS_TFM.pdf
    ├── TESTING_FRONTEND_MANUAL.md
    ├── TESTING_FRONTEND_MANUAL.pdf
    ├── LICENCIAS_TERCEROS.md
    ├── LICENCIAS_TERCEROS.pdf
    ├── Dockerfile
    ├── render.yaml
    ├── .dockerignore
    ├── .gitignore
    └── backend/tribunal)
        ├── api/
        │   ├── index.php
        │   └── auth.php
        └── password
            ├── habilidades.php
            ├── intercambios.php
            ├── valoraciones.php
            ├── conversaciones.php
            ├── reportes.php
            ├── notificaciones.php
            └── usuarios.php
```

Raíz del proyecto

⚠ CRÍTICO - Proxy hacia

📄 Documento principal para el

📄 Versión PDF del documento

📄 README general del proyecto

🛠 Testing backend + endpoints

📄 Versión PDF

🛠 Testing frontend manual

📄 Versión PDF

📄 Licencias de software

📄 Versión PDF

🐳 Configuración Docker para

🚀 Configuración de despliegue

🚫 Exclusiones para Docker

🚫 Exclusiones para Git

☑ CARPETA BACKEND (para el

Endpoints HTTP

Router principal (endpoint único)

Autenticación y recuperación

CRUD habilidades

Sistema de intercambios

Sistema de valoraciones

Mensajería

Sistema de reportes

Notificaciones

Gestión de usuarios

└─ categorias.php	# Categorías de habilidades
└─ config/	# Configuración
└─ database.php	# Conexión PostgreSQL (Supabase)
└─ cors.php	# CORS y cookies
└─ models/	# Modelos de datos (clases PHP)
└─ Usuario.php	
└─ Habilidad.php	
└─ Intercambio.php	
└─ Valoracion.php	
└─ Conversacion.php	
└─ Mensaje.php	
└─ Notificacion.php	
└─ Reporte.php	
└─ utils/	# Utilidades compartidas
└─ Response.php	# Respuestas JSON estandarizadas
└─ Auth.php	# Gestión de sesiones y tokens
└─ EmailService.php	# Envío de emails (Brevo API)
└─ .htaccess	# Configuración Apache
└─ info.php	# Información PHP (phpinfo)
└─ README.md	# Documentación del backend
└─ frontend/tribunal)	# <input checked="" type="checkbox"/> CARPETA FRONTEND (para el
└─ src/	# Código fuente Angular
└─ app/	# Aplicación Angular
└─ core/	# Servicios centrales y guards
└─ guards/	# Guards (auth, admin)
└─ interceptors/	# Interceptores HTTP
└─ services/	# Servicios (auth, api, etc.)
└─ features/	# Módulos funcionales
└─ admin/	# Panel administración
└─ auth/	# Login, registro, recuperación
└─ habilidades/	# CRUD habilidades
└─ home/	# Página principal
└─ intercambios/	# Gestión intercambios
└─ perfil/	# Perfil de usuario
└─ reportes/	# Sistema de reportes
└─ valoraciones/	# Valoraciones
└─ layout/	# Componentes de layout
└─ header/	# Barra superior
└─ footer/	# Pie de página
└─ main-layout/	# Layout principal
└─ shared/	# Componentes compartidos
└─ components/	# Componentes reutilizables
└─ models/	# Interfaces TypeScript

├── app.component.ts	# Componente raíz
├── app.config.ts	# Configuración Angular
├── app.routes.ts	# Rutas de la aplicación
├── environments/	# Configuración de entornos
│ ├── environment.ts	# Desarrollo
│ └── environment.prod.ts	# Producción
├── index.html	# HTML principal
├── main.ts	# Entry point
└── styles.scss	# Estilos globales
├── public/	# Assets públicos
│ ├── _headers	# Configuración headers Render
│ └── _redirects	# Redirecciones Render
├── angular.json	# Configuración Angular CLI
├── package.json	# Dependencias npm
├── tsconfig.json	# Configuración TypeScript
└── README.md	# Documentación del frontend
├── database/	# <input checked="" type="checkbox"/> Scripts SQL
│ ├── schema.sql	# Esquema completo (12 tablas)
│ └── seeds.sql	# Datos de ejemplo (3 usuarios + 6
habilidades)	
├── old/	# <input checked="" type="checkbox"/> Archivos no entregables
│ └── index.php	# Página de prueba (contador +
email)	
├── ANALISIS_FRONTEND_Y_PLAN.md	# Análisis interno
├── ANALISIS_SEGURIDAD_RESET.md	# Análisis de seguridad
├── ANALISIS_TABLAS_VISTAS.md	# Análisis de BD
├── ARQUITECTURA_DEPLOY.md	# Documentación de deploy
├── ESTADO_IMPLEMENTACION_FUNCIONALIDADES.md	
├── GUIA_PREPARAR_ENTREGA_PEC2.md	# Guía interna
├── INFRAESTRUCTURA_TECNOLOGICA_TFM.md	
├── README_INSTALACION_BD.md	# Ya no necesario
├── README_RESET.md	# Ya no necesario
├── extract_schema_supabase.php	# Script extracción BD
├── generate_password_hash.php	# Utilidad hashing
└── reset_database.php	# Script reset BD

⚠ ARCHIVOS CRÍTICOS

🔴 IMPRESCINDIBLES

1. **api.php** (raíz)

- **Función:** Proxy que redirige todas las peticiones al backend
- **Importancia:** **CRÍTICA** - Sin este archivo la aplicación NO funciona

- **Motivo:** El frontend Angular llama a `/api.php?resource=...` en todas las peticiones
- **Eliminar:** ✗ NO - Es el punto de entrada único de la API

2. `backend/` (carpeta completa)

- **Función:** Backend PHP con API HTTP
- **Importancia:** **CRÍTICA** - Lógica de negocio completa
- **Contenido:** 10 endpoints + 8 modelos + utilidades + configuración
- **Eliminar:** ✗ NO

3. `frontend/` (carpeta completa)

- **Función:** Aplicación Angular 19
- **Importancia:** **CRÍTICA** - Interfaz de usuario
- **Contenido:** 16 componentes + 12 servicios + guards + interceptores
- **Eliminar:** ✗ NO

4. `database/schema.sql` y `database/seeds.sql`

- **Función:** Instalación de base de datos PostgreSQL
- **Importancia:** **CRÍTICA** - Necesarios para montaje local
- **Contenido:** Esquema completo + datos de ejemplo
- **Eliminar:** ✗ NO

🕒 RECOMENDABLES

5. `README_PEC2.md/pdf`

- **Función:** Documento principal para el tribunal
- **Importancia:** **MUY ALTA** - Guía de evaluación
- **Contenido:** Instrucciones de uso, testing, arquitectura
- **Eliminar:** ✗ NO

6. Documentos de testing (MD + PDF)

- `TESTING_Y_ENDPOINTS_TFM.md/pdf`
- `TESTING_FRONTEND_MANUAL.md/pdf`
- **Función:** Documentación de pruebas realizadas
- **Importancia:** **ALTA** - Demostración de testing
- **Eliminar:** ✗ NO

7. `LICENCIAS_TERCEROS.md/pdf`

- **Función:** Listado de licencias de software usado
- **Importancia:** **MEDIA** - Transparencia académica
- **Eliminar:** ✗ NO

8. README.md (raíz)

- **Función:** README general del proyecto
 - **Importancia:** **MEDIA** - Información general
 - **Diferencia con README_PEC2:** Más técnico, menos académico
 - **Eliminar:** ⚠ **OPCIONAL** - Si confunde, puedes eliminarlo
-

🌀 OPCIONALES (Archivos técnicos)

9. Dockerfile

- **Función:** Configuración del contenedor Docker
- **Importancia:** **BAJA** - Solo para deploy en Render
- **Eliminar:** ⚠ **OPCIONAL** - Útil si el tribunal quiere ver cómo se despliega

10. render.yaml

- **Función:** Configuración de despliegue en Render
- **Importancia:** **BAJA** - Solo para deploy
- **Eliminar:** ⚠ **OPCIONAL**

11. .dockerignore y .gitignore

- **Función:** Exclusiones para Docker/Git
 - **Importancia:** **BAJA** - Configuración técnica
 - **Eliminar:** ⚠ **OPCIONAL**
-

📁 CARPETA **old/**

☑ Archivos correctamente movidos:

1. **index.php** - Página de prueba (contador de visitas + email Resend)

- **No forma parte de GaliTroco**
- ☑ **BIEN MOVIDO**

2. **Documentos de análisis interno:**

- ANALISIS_FRONTEND_Y_PLAN.md
- ANALISIS_SEGURIDAD_RESET.md
- ANALISIS_TABLAS_VISTAS.md
- ARQUITECTURA_DEPLOY.md
- ESTADO_IMPLEMENTACION_FUNCIONALIDADES.md
- GUIA_PREPARAR_ENTREGA_PEC2.md
- INFRAESTRUCTURA_TECNOLOGICA_TFM.md
- ☑ **BIEN MOVIDOS** - Son documentos de desarrollo interno

3. Scripts de utilidades:

- `extract_schema_supabase.php` - Extracción de esquema desde Supabase
- `generate_password_hash.php` - Generador de hashes bcrypt
- `reset_database.php` - Reset de BD local
- ☒ **BIEN MOVIDOS** - Son herramientas de desarrollo

4. Documentación de instalación obsoleta:


- `README_INSTALACION_BD.md`
- `README_RESET.md`
- ☒ **BIEN MOVIDOS** - Ya no necesarios (info en `README_PEC2.md`)

☒ VALIDACIÓN FINAL

Checklist de Entrega:

- ☒ **Backend completo** en `/backend/`
- ☒ **Frontend completo** en `/frontend/`
- ☒ **Scripts SQL** en `/database/` (schema.sql + seeds.sql)
- ☒ **api.php** en raíz (proxy crítico)
- ☒ **README_PEC2.md** (documento principal)
- ☒ **Documentación de testing** (backend + frontend)
- ☒ **Licencias** documentadas
- ☒ **Archivos técnicos** de deploy (Dockerfile, render.yaml)
- ☒ **Archivos no entregables** movidos a `/old/`

Resumen de Archivos:

Carpeta/Archivo	Estado	Observaciones
<code>api.php</code>	<input checked="" type="checkbox"/> Correcto	CRÍTICO - Proxy de backend
<code>backend/</code>	<input checked="" type="checkbox"/> Correcto	Carpeta para el tribunal
<code>frontend/</code>	<input checked="" type="checkbox"/> Correcto	Carpeta para el tribunal
<code>database/</code>	<input checked="" type="checkbox"/> Correcto	Solo 2 archivos SQL (perfecto)
<code>README_PEC2.md/pdf</code>	<input checked="" type="checkbox"/> Correcto	Documento principal
Testing MD/PDF	<input checked="" type="checkbox"/> Correcto	Documentación completa
<code>LICENCIAS_TERCEROS.md/pdf</code>	<input checked="" type="checkbox"/> Correcto	Transparencia
<code>old/</code>	<input checked="" type="checkbox"/> Correcto	Archivos no entregables
<code>index.php</code>	<input checked="" type="checkbox"/> Movido a old/	Ya no en raíz
<code>README.md</code>	 Revisar	¿Duplicado de README_PEC2?

RECOMENDACIÓN FINAL

☒ Tu estructura es **EXCELENTE**

La reorganización que hiciste es **casi perfecta**:

1. ☒ **Backend** y **Frontend** en carpetas separadas (claro para el tribunal)
2. ☒ **Database** con solo 2 archivos SQL (schema.sql + seeds.sql)
3. ☒ **Documentación** completa en la raíz (fácil de encontrar)
4. ☒ **Archivos técnicos** de deploy presentes (Dockerfile, render.yaml)
5. ☒ **Carpeta old/** con archivos no entregables (orden)

Única sugerencia:

Revisar **README.md**:

- Si es igual o muy similar a **README_PEC2.md** → **Eliminar** para evitar confusión
- Si contiene información complementaria → **Mantener** y renombrar a **README_GENERAL.md**

CONTACTO

Antonio Campos

Email: kampos@gmail.com

Proyecto: GaliTroco TFM

Fecha entrega: 5 de noviembre de 2025

☒ ESTRUCTURA VALIDADA Y LISTA PARA ENTREGA PEC2