

GALITROCO - Sistema de Intercambio de Habilidades

Autor: Antonio Campos

Universidad: Universitat Oberta de Catalunya (UOC)

Asignatura: Trabajo Final de Máster

Fecha: Octubre 2025

Versión: (PEC2)

DESCRIPCIÓN DEL PROYECTO

GaliTroco es una plataforma web que permite a los usuarios de Galicia intercambiar habilidades y servicios sin necesidad de dinero. Un usuario puede ofrecer sus conocimientos (programación, idiomas, cocina, etc.) a cambio de aprender otras habilidades de la comunidad.

Características principales:

- ☒ Sistema de autenticación y registro de usuarios
 - ☒ Publicación de habilidades (ofertas y demandas)
 - ☒ Propuestas de intercambio entre usuarios
 - ☒ Sistema de valoraciones con estrellas (1-5)
 - ☒ Panel de administración para moderación
 - ☒ Sistema de reportes de contenido inapropiado
 - ☒ Notificaciones en tiempo real
 - ☒ Recuperación de contraseña por email
-

RESUMEN DE TECNOLOGÍAS UTILIZADAS

Backend

- **Lenguaje:** PHP 8.2
- **Servidor Web:** Apache 2.4 (integrado en Docker php:8.2-apache)
- **Base de Datos:** PostgreSQL 15 (Supabase)
- **Autenticación:** Sesiones PHP con tokens hexadecimales SHA-256
- **Email:** Brevo API (ex-Sendinblue) - 300 emails/día gratuitos
- **Deploy:** Render.com (Docker container)

Frontend

- **Framework:** Angular 19.2.0
 - **Lenguaje:** TypeScript 5.7.2
 - **UI:** Angular Material 19.2.19
 - **HTTP Client:** HttpClient con RxJS
 - **Deploy:** Render.com (Static Site)
-

Infraestructura

- **Repositorio:** GitHub (tonikampos/render-test-php)
- **CI/CD:** Auto-deploy desde GitHub main branch
- **Base de datos:** Supabase Cloud PostgreSQL

URLS DEL PROYECTO DESPLEGADO

Producción (Render.com) ☒ OPERATIVO

- **Backend API:** <https://render-test-php-1.onrender.com/api.php>
- **Frontend Angular:** <https://galitroco-frontend.onrender.com>
- **Base de Datos:** Supabase PostgreSQL 15 (cloud)

Estado: Ambos servicios desplegados en Render.com con auto-deploy desde GitHub.

IMPORTANTE ANTES DE EMPEZAR - LEER PRIMERO

 Limitación de Render Free Tier

ATENCIÓN EVALUADORES: El backend en Render.com (plan gratuito) entra en "modo sleep" después de 15 minutos de inactividad.

La **primera petición** después del sleep puede tardar **30-90 segundos** en responder.

Síntomas:

- Al abrir el frontend, aparece "Error al conectar con el servidor"
- Timeout o pantalla de carga infinita
- Error 502 Bad Gateway

☒ SOLUCIÓN (IMPORTANTE):

1. **Antes de probar el frontend**, abrir esta URL en una pestaña nueva:

```
https://render-test-php-1.onrender.com/api.php?resource=health
```

2. **Esperar 30-90 segundos** hasta ver esta respuesta JSON:

```
{
  "success": true,
  "data": {
    "status": "healthy",
    "timestamp": "2025-10-28...",
    "database": "connected"
  },
}
```

```
"message": "API funcionando correctamente"
}
```

3. **Ahora sí**, abrir el frontend:

```
https://galitroco-frontend.onrender.com
```

💡 Tras el primer "despertar", el backend responde normalmente (< 1 segundo) mientras esté activo.

☑ REQUISITOS PARA PROBAR LA APLICACIÓN

Navegador y Sistema

Requisito	Mínimo	Recomendado
Navegador	Chrome 90+, Firefox 88+, Edge 90+	Chrome/Edge 120+
Resolución	1024x768	1920x1080 o superior
JavaScript	Habilitado (requerido)	Habilitado
Cookies	Habilitadas (requerido)	Habilitadas
Conexión	2 Mbps	10+ Mbps
Bloqueadores	Desactivar para el sitio	Desactivar

Tiempos Estimados

- 🕒 **Cold start inicial:** 30-90 segundos (solo primera vez)
- 🕒 **Carga del frontend:** 3-5 segundos
- 🕒 **Prueba rápida:** 10-15 minutos
- 🕒 **Prueba completa:** 25-30 minutos

Testing local (Opcional)

- **Backend:** <http://localhost/probatfm/backend/api/>
- **Frontend:** <http://localhost:4200>
- **Base de Datos:** PostgreSQL local o Supabase (recomendado)

📦 REQUISITOS PREVIOS

Para ejecutar el backend:

- PHP 8.2 o superior
- Apache 2.4+ con mod_rewrite

- PostgreSQL 15+
- Composer (opcional, si se usan dependencias)
- Extensiones PHP: `pdo_pgsql`, `json`, `session`

Para ejecutar el frontend:

- Node.js 18+ (recomendado: 20.x)
- npm 9+ o 10+
- Angular CLI 19 (`npm install -g @angular/cli`)

Para la base de datos:

- PostgreSQL 15+
- Cliente SQL (psql, pgAdmin, DBeaver, etc.)

INSTRUCCIONES DE INSTALACIÓN

1 CLONAR O DESCOMPRIMIR EL PROYECTO

Con el ZIP:

```
unzip PEC2_pry_Campos_Antonio.zip
cd PEC2_pry_Campos_Antonio
```

2 CONFIGURAR LA BASE DE DATOS

Opción A: Usar Supabase (Recomendado para PEC2)

La aplicación ya está configurada para usar Supabase. Solo necesitas las credenciales correctas en `backend/config/database.php`.

Opción B: PostgreSQL Local

1. Crear base de datos:

```
CREATE DATABASE galitroco_tfm;
```

2. Ejecutar esquema:

```
psql -U postgres -d galitroco_tfm -f database/schema.sql
```

3. Cargar datos de prueba:

```
psql -U postgres -d galitroco_tfm -f database/seeds.sql
```

4. Configurar credenciales:

Editar `backend/config/database.php`:

```
return [  
    'host' => 'localhost',  
    'port' => '5432',  
    'dbname' => 'galitroco_tfm',  
    'user' => 'postgres',  
    'password' => 'tu_password'  
];
```

3 CONFIGURAR EL BACKEND

1. Configurar Apache Virtual Host (ejemplo):

```
<VirtualHost *:80>  
    ServerName galitroco.local  
    DocumentRoot "C:/xampp/htdocs/probatfm"  
  
    <Directory "C:/xampp/htdocs/probatfm">  
        Options Indexes FollowSymLinks  
        AllowOverride All  
        Require all granted  
    </Directory>  
</VirtualHost>
```

2. Actualizar archivo hosts (Windows: `C:\Windows\System32\drivers\etc\hosts`):

```
127.0.0.1 galitroco.local
```

3. Configurar variables de entorno:

Crear archivo `backend/config/.env` (opcional):

```
DB_HOST=localhost  
DB_PORT=5432  
DB_NAME=galitroco_tfm  
DB_USER=postgres  
DB_PASSWORD=tu_password
```

```
BREVO_API_KEY=tu_api_key_de_brevo  
FRONTEND_URL=http://localhost:4200
```

4. Verificar instalación:

Acceder a: <http://galitroco.local/backend/api/index.php?resource=health>

Debe mostrar:

```
{  
  "success": true,  
  "data": {  
    "status": "healthy",  
    "timestamp": "2025-10-28...",  
    "database": "connected"  
  },  
  "message": "API funcionando correctamente"  
}
```

4 CONFIGURAR EL FRONTEND

Opción A: Usar frontend desplegado en Render (RECOMENDADO)

El frontend ya está **desplegado y operativo** en Render.com. Solo necesitas acceder a:

```
https://galitroco-frontend.onrender.com
```

Configuración actual:

- ☒ Build automático desde GitHub (branch **main**)
- ☒ Conectado al backend de producción (<https://render-test-php-1.onrender.com>)
- ☒ CORS configurado correctamente
- ☒ Variables de entorno de producción

Opción B: Ejecutar frontend localmente (OPCIONAL - para desarrollo)

1. Instalar dependencias:

```
cd frontend  
npm install
```

2. Configurar URL del backend:

El frontend local está configurado para usar el backend de **producción** (Render).

Editar `frontend/src/environments/environment.ts` si quieres usar backend local:

```
export const environment = {  
  production: false,  
  // Backend en Render (por defecto)  
  apiUrl: 'https://render-test-php-1.onrender.com/api.php'  
  
  // O backend local (si lo tienes corriendo):  
  // apiUrl: 'http://galitroco.local/backend/api/api.php'  
};
```

3. Iniciar servidor de desarrollo:

```
npm start
```

El frontend local estará disponible en: `http://localhost:4200`

 **Más detalles:** Ver `frontend/README_FRONTEND.md`

4.1 Desplegar frontend en Render (YA CONFIGURADO)

Si necesitas redespugar o configurar desde cero:

1. Crear nuevo Static Site en Render:

- Build Command: `cd frontend && npm install && npm run build:prod`
- Publish Directory: `frontend/dist/frontend/browser`
- Auto-Deploy: Yes (desde GitHub `main` branch)

2. Variables de entorno en Render:

- No se requieren variables de entorno
- La URL del backend está en `environment.prod.ts`

3. Verificar archivo `environment.prod.ts`:

```
export const environment = {  
  production: true,  
  apiUrl: 'https://render-test-php-1.onrender.com/api.php'  
};
```

Estado: ☒ Frontend desplegado y operativo en Render

TESTING Y VALIDACIÓN

Testing del Backend

Se ha realizado testing exhaustivo de 25 endpoints en producción (Render.com) con 92% de éxito (23/25 tests pasados).

Documento de evidencias: Ver [documentacion_tecnica/TESTING_Y_ENDPOINTS_TFM.md](#)

Credenciales de prueba:

Administrador:

- Email: [admin@galitroco.com](#)
- Password: [Pass123456](#)
- Rol: [administrador](#)

Usuario Demo:

- Email: [demo@galitroco.com](#)
- Password: [Pass123456](#)
- Rol: [usuario](#)

Usuario Test:

- Email: [test@galitroco.com](#)
- Password: [Pass123456](#)
- Rol: [usuario](#)

Testing del Frontend

Testing en Producción (RECOMENDADO):

El frontend está **desplegado en Render** e integrado con el backend de producción.

1. **Acceder a:** <https://galitroco-frontend.onrender.com>

2. **Probar flujo completo:**

- Registro de usuario
- Login
- Crear habilidad
- Proponer intercambio
- Aceptar/Rechazar propuesta
- Completar intercambio
- Valorar usuario

Testing Local (OPCIONAL):

Si prefieres probar localmente:

1. Iniciar frontend: [npm start](#) (en carpeta [/frontend](#))
2. Abrir navegador en: <http://localhost:4200>
3. El frontend local se conecta automáticamente al backend de Render

ESTADO DEL PROYECTO (PEC2)

Backend: ☒ **92% OPERATIVO** (23/25 endpoints OK)

- ☒ 25 endpoints implementados y testeados en producción
- ☒ 23/25 tests pasados (92% éxito)
- ☒ 2 bugs críticos corregidos (transacciones ACID, router)
- ☒ 0 bugs pendientes críticos
- ☒ Autenticación con Sesiones PHP + tokens hexadecimales SHA-256
- ☒ Sistema de email funcional (Brevo API - 300 emails/día)
- ☒ Desplegado en Render.com con auto-deploy desde GitHub
- ☒ Documentación técnica completa (25+ páginas)

Frontend: ☒ **50% IMPLEMENTADO Y DESPLEGADO** (12/16 tests OK)

Funcionalidades COMPLETADAS (- ☒ Autenticación completa (login, registro, logout) - ☒ Listado de habilidades con filtros y búsqueda - ☒ Detalle de habilidades - ☒ Creación de habilidades - ☒ Visualización de intercambios propios - ☒ Proponer intercambio (con dialog) - ☒ Sistema de valoraciones (con dialog de estrellas) - ☒ Perfiles públicos y privados - ☒ Panel de administración (reportes) - ☒ Guards de seguridad (auth, admin) - ☒ Angular Material Design completo - ☒ **Desplegado en Render.com como Static Site**

Funcionalidades PENDIENTES (☐):

- ⌚ Edición de habilidades propias
- ⌚ Eliminación de habilidades
- ⌚ Botones aceptar/rechazar intercambios
- ⌚ Botón completar intercambio

Documentación: Ver [TESTING_FRONTEND_MANUAL.md](#) para detalles completos (12/16 tests = 75%)

Base de Datos: ☒ **100% OPERATIVA**

- ☒ Esquema completo con 10 tablas
- ☒ Relaciones e integridad referencial
- ☒ Índices y constraints
- ☒ Seeds con datos de prueba
- ☒ Migración a Supabase exitosa

📁 ESTRUCTURA DEL PROYECTO

```
galitroco/
├── backend/                                # API HTTP en PHP
│   ├── api/                              # Endpoints de la API
│   │   ├── index.php                    # Router principal (endpoint único)
│   │   ├── auth.php                    # Autenticación y recuperación password
│   │   ├── habilidades.php              # CRUD habilidades
│   │   ├── intercambios.php             # Sistema de intercambios
│   │   ├── valoraciones.php             # Sistema de valoraciones
│   │   ├── reportes.php                 # Reportes y moderación
│   │   ├── mensajes.php                 # Sistema de mensajería
│   │   └── ...
│   ├── config/                          # Configuración
│   │   ├── database.php                 # Conexión PostgreSQL (Supabase)
│   │   ├── cors.php                    # CORS y configuración cookies
│   │   └── .env                         # Variables de entorno (no en repo)
│   ├── models/                          # Modelos de datos (clases PHP)
│   ├── utils/                           # Utilidades
│   │   ├── Response.php                 # Respuestas JSON estandarizadas
│   │   ├── Auth.php                    # Gestión de sesiones y tokens
│   │   └── EmailService.php             # Envío de emails (Brevo API)
│   └── Dockerfile                       # Contenedor Docker para Render
├── frontend/                            # Aplicación Angular
│   ├── src/
│   │   ├── app/
│   │   │   ├── core/                   # Servicios y guards
│   │   │   ├── features/               # Componentes por módulo
│   │   │   ├── shared/                 # Componentes compartidos
│   │   │   └── layout/                 # Header, footer, etc.
│   │   └── environments/               # Configuración de entornos
│   ├── package.json
│   └── angular.json
├── database/                            # Scripts SQL
│   ├── schema.sql                      # Esquema completo
│   ├── seeds.sql                       # Datos de prueba
│   └── incremental_*.sql                # Migraciones
├── documentacion_tecnica/               # Documentación adicional
│   ├── TESTING_Y_ENDPOINTS_TFM.md
│   ├── ARQUITECTURA_DEPLOY.md
│   └── ...
└── README.md                           # Este archivo
```

🔒 SEGURIDAD

Medidas implementadas:

- ☒ Contraseñas hasheadas con bcrypt (cost 12)
- ☒ Prepared statements en todas las queries (prevención SQL Injection)
- ☒ Validación de entrada en todos los endpoints (sanitización)
- ☒ CORS configurado específicamente para dominios permitidos
- ☒ HTTPS en producción (certificados SSL de Render)
- ☒ Tokens de sesión hexadecimales SHA-256 (64 caracteres)
- ☒ Sesiones PHP con cookies SameSite=None; Secure
- ☒ Guards de autenticación y autorización en frontend (canActivate)
- ☒ Middleware de protección en backend (verificación de sesión)
- ☒ Rate limiting implícito (Render free tier)
- ☒ Variables de entorno para credenciales sensibles

LICENCIAS Y RECURSOS DE TERCEROS

Backend

- **PHP:** Licencia PHP License 3.01
- **PostgreSQL:** Licencia PostgreSQL License (similar a MIT)
- **Brevo:** API comercial (plan gratuito: 300 emails/día)

Frontend

- **Angular:** Licencia MIT
- **Angular Material:** Licencia MIT
- **RxJS:** Licencia Apache 2.0
- **TypeScript:** Licencia Apache 2.0

Servicios Cloud

- **Render.com:** Servicio comercial (plan gratuito)
- **Supabase:** Open Source (PostgreSQL) + servicios cloud

 **Lista completa:** Ver [LICENCIAS_TERCEROS.md](#)

PROBLEMAS CONOCIDOS Y SOLUCIONES

1. 🕒 Cold Start en Render (MUY COMÚN)

Síntomas:

- Primera petición tarda 30-90 segundos
- Error 502 Bad Gateway
- Frontend muestra "Error al conectar con servidor"
- Pantalla de carga infinita

Causa: Render free tier pone el backend en sleep tras 15 minutos de inactividad.

☑ **Solución:**

1. Abrir primero: <https://render-test-php-1.onrender.com/api.php?resource=health>
 2. Esperar respuesta JSON (30-90 segundos)
 3. Ahora abrir el frontend
 4. Tras "despertar", funciona normalmente (< 1 segundo)
-

2. 🍪 Cookies bloqueadas o no se guardan

Síntomas:

- No persiste login tras refrescar página
- Siempre pide autenticación
- Error 401 Unauthorized en peticiones autenticadas

Causa: Navegador bloquea cookies de terceros o navegación privada.

☑ **Solución:**

- **Chrome/Edge:** Settings → Privacy → Allow all cookies (temporalmente)
 - **Firefox:** Settings → Privacy → Standard mode
 - **NO usar modo incógnito/privado** (bloquea cookies cross-site)
 - Verificar que cookies están habilitadas en navegador
-

3. 🚫 Error CORS "blocked by CORS policy"

Síntomas:

- Error en consola: [Access to fetch at '...' has been blocked by CORS policy](#)
- Peticiones fallan desde frontend local

Causa: Backend no reconoce el dominio de origen.

☑ **Solución:**

- Si usas frontend local (localhost:4200): Ya está configurado en [backend/config/cors.php](#)
 - Si usas otro puerto: Añadir a whitelist en CORS config
 - **Render production:** Ya configurado para [galitroco-frontend.onrender.com](#)
-

4. 📧 Email de recuperación no llega

Síntomas:

- Solicitar recuperación de contraseña → no llega email

Causa: Brevo API puede tardar o email en spam.

✓ **Solución:**

1. **Revisar carpeta SPAM** (muy común)
 2. Esperar 1-2 minutos (Brevo puede tardar)
 3. Verificar que el email existe en base de datos
 4. Límite: 300 emails/día en plan gratuito
-

5. 🗄 Base de datos con datos antiguos

Síntomas:

- Usuarios o habilidades que ya eliminaste siguen apareciendo
- Contadores incorrectos

Causa: Base de datos de prueba no reseteada.

✓ **Solución:**

- Usar las 3 cuentas de prueba predefinidas (admin, demo, test)
 - Si necesitas reset: Contactar con autor (no hay acceso directo a Supabase)
-

6. 🌀 Frontend muestra página en blanco

Síntomas:

- Pantalla blanca después de cargar
- No hay errores visibles

Causa: JavaScript deshabilitado o error crítico de carga.

✓ **Solución:**

1. Abrir **DevTools** (F12) → Console
 2. Ver errores en consola
 3. Verificar JavaScript habilitado
 4. Probar en navegador diferente
 5. Limpiar caché (Ctrl+Shift+R)
-

7. 🐢 Aplicación muy lenta

Síntomas:

- Todas las operaciones tardan mucho
- Timeout frecuentes

Causa: Limitaciones del plan gratuito de Render.

✓ **Solución:**

- **Normal:** Render free tier tiene CPU compartida y limitada
 - Primera petición siempre lenta (cold start)
 - Después mejora significativamente
 - Esperar pacientemente (no es un bug, es limitación del hosting)
-

8. ✕ Error 404 en rutas del frontend

Síntomas:

- Refrescar página en [/habilidades](#) → Error 404
- Links directos no funcionan

Causa: Render Static Site necesita configuración para SPA.

☑ Solución:

- Ya configurado con [render.yaml](#) y rewrites
 - Si persiste: Siempre navegar desde la home <https://galitroco-frontend.onrender.com>
 - Usar navegación interna (no F5 en subrutas)
-

9. 🚫 No puedo acceder al panel de admin

Síntomas:

- Acceder a [/admin](#) → Redirige a login
- Mensaje "No tienes permisos"

Causa: Usuario no tiene rol [administrador](#).

☑ Solución:

- Usar cuenta: [admin@galitroco.com](#) / [Pass123456](#)
 - Solo este usuario tiene rol de administrador
 - No se pueden crear admins desde el frontend (solo en BD)
-

☎ SOPORTE Y CONTACTO

Autor: Antonio Manuel Campos Gerpe

Email UOC: acamposg@uoc.edu (*verificar email correcto*)

GitHub: <https://github.com/tonikampos/render-test-php>

Proyecto: Trabajo Final de Máster - UOC

Consultor/Tutor: (*Nombre del tutor asignado*)

⚠ **Nota para evaluadores:** Si encuentran problemas técnicos al probar la aplicación:

1. Revisar primero la sección "🔗 **PROBLEMAS CONOCIDOS Y SOLUCIONES**"
 2. El 90% de problemas son cold start de Render (esperar 60 segundos)
 3. Para consultas urgentes, contactar por email institucional
-

GUÍA DE PRUEBAS PARA EVALUADORES


Esta sección proporciona instrucciones paso a paso para probar todas las funcionalidades de GaliTroco.

Opción 1: Pruebas en Producción (RECOMENDADO - Sin instalación)

Acceso directo:

- **Frontend:** <https://galitroco-frontend.onrender.com>
- **Backend API:** <https://render-test-php-1.onrender.com/api.php>

Ventajas:

- ☒ No requiere instalación local
- ☒ Base de datos con datos de muestra
- ☒ Configuración completa y operativa
-  Tiempo de prueba: 15-20 minutos

Credenciales de Prueba

Utiliza estos usuarios para probar diferentes escenarios:

Administrador (panel admin)

```
Email:      admin@galitroco.com
Password:   Pass123456
Rol:        Administrador
```

Usuario Demo (para intercambios)

```
Email:      demo@galitroco.com
Password:   Pass123456
Rol:        Usuario normal
```

Usuario Test (para intercambios)

```
Email:      test@galitroco.com
Password:   Pass123456
Rol:        Usuario normal
```

⚠ RECORDATORIO ANTES DE EMPEZAR TESTING

1. Despertar el backend primero (obligatorio):

```
https://render-test-php-1.onrender.com/api.php?resource=health
```

Esperar hasta ver respuesta JSON (puede tardar 30-90 segundos la primera vez).

2. Verificar que está "despierto":

Debe mostrar:

```
{
  "success": true,
  "data": {
    "status": "healthy",
    "timestamp": "...",
    "database": "connected"
  },
  "message": "API funcionando correctamente"
}
```

3. Ahora sí, abrir el frontend:

```
https://galitroco-frontend.onrender.com
```

📋 Escenario 1: Registro y Autenticación (3 minutos)

1.1 Probar registro de nuevo usuario

1. Ir a: <https://galitroco-frontend.onrender.com/registro>
2. Completar formulario:
 - Nombre de usuario: **nuevo_usuario**
 - Email: **nuevo@test.com**
 - Contraseña: **Test123456**
 - Ubicación: **A Coruña**
3. Click en **"Registrarse"**
4. **Resultado esperado:** Redirección automática al login

1.2 Probar login

1. Ir a: <https://galitroco-frontend.onrender.com/login>
2. Ingresar credenciales:
 - Email: **demo@galitroco.com**

- Password: **Pass123456**
- 3. Click en **"Iniciar sesión"**
- 4. **Resultado esperado:** Redirección al dashboard con mensaje de bienvenida

1.3 Verificar persistencia de sesión

1. Cerrar navegador completamente
2. Abrir navegador y volver a: **<https://galitroco-frontend.onrender.com>**
3. **Resultado esperado:** Usuario sigue autenticado (sesión persiste)

1.4 Probar logout

1. Click en icono de usuario (esquina superior derecha)
2. Click en **"Cerrar sesión"**
3. **Resultado esperado:** Redirección a página de login

Escenario 2: Gestión de Habilidades (5 minutos)

2.1 Crear habilidad de tipo "Oferta"

1. Login como **demo@galitroco.com**
2. Ir a: **"Mis Habilidades"** → **"Nueva Habilidad"**
3. Completar formulario:
 - Tipo: **Oferta**
 - Categoría: **Tecnología e Informática**
 - Título: **Clases de Python para principiantes**
 - Descripción: **Enseño programación en Python desde cero**
 - Duración estimada: **60 minutos**
4. Click en **"Publicar"**
5. **Resultado esperado:** Habilidad visible en "Mis Habilidades"

2.2 Crear habilidad de tipo "Demanda"

1. Click en **"Nueva Habilidad"** nuevamente
2. Completar formulario:
 - Tipo: **Demanda**
 - Categoría: **Clases y Formación**
 - Título: **Busco clases de inglés conversacional**
 - Descripción: **Necesito mejorar mi inglés hablado**
 - Duración estimada: **90 minutos**
3. Click en **"Publicar"**
4. **Resultado esperado:** Ambas habilidades visibles en el listado

2.3 Editar habilidad

1. En "Mis Habilidades", click en **icono de editar** (lápiz)

2. Modificar el título o descripción
3. Click en **"Guardar cambios"**
4. **Resultado esperado:** Cambios reflejados inmediatamente

2.4 Pausar/Activar habilidad

1. Click en botón **"Pausar"** de una habilidad
 2. **Resultado esperado:** Estado cambia a "Pausada" (no visible en búsquedas públicas)
 3. Click en **"Activar"** nuevamente
 4. **Resultado esperado:** Estado vuelve a "Activa"
-

Escenario 3: Sistema de Intercambios Completo (7 minutos)

3.1 Proponer intercambio (Usuario Demo)

1. Login como `demo@galitroco.com`
2. Ir a **"Explorar Habilidades"**
3. Buscar habilidades de otros usuarios
4. Click en una habilidad que te interese
5. Click en botón **"Proponer Intercambio"**
6. Seleccionar tu habilidad a ofrecer (de las que creaste)
7. Escribir mensaje: `Hola, me interesa tu habilidad. ¿Podemos intercambiar?`
8. Click en **"Enviar Propuesta"**
9. **Resultado esperado:**
 - Mensaje de confirmación
 - Propuesta visible en "Mis Intercambios" → "Enviados"
 - Notificación enviada al otro usuario

3.2 Aceptar propuesta (Usuario Test)

1. **Cerrar sesión** de Usuario Demo
2. Login como `test@galitroco.com`
3. Ir a **"Mis Intercambios"** → **"Recibidos"**
4. Ver la propuesta recibida
5. Click en **"Aceptar"**
6. **Resultado esperado:**
 - Estado cambia a "Aceptado"
 - Se crea conversación automática
 - Notificación enviada al Usuario Demo

3.3 Enviar mensajes en conversación

1. Desde "Intercambios Aceptados", click en **"Ver conversación"**
2. Escribir mensaje: `Perfecto, ¿cuándo empezamos?`
3. Enviar mensaje
4. **Resultado esperado:** Mensaje visible en el chat

3.4 Completar intercambio

1. Click en botón "**Marcar como Completado**"
2. Confirmar acción
3. **Resultado esperado:**
 - Estado cambia a "Completado"
 - Aparece opción para valorar

3.5 Valorar usuario

1. Click en "**Valorar Usuario**"
 2. Seleccionar puntuación (1-5 estrellas): 5 estrellas
 3. Escribir comentario: Excelente experiencia, muy profesional
 4. Click en "**Enviar Valoración**"
 5. **Resultado esperado:**
 - Valoración registrada
 - Puntuación promedio del usuario actualizada
 6. **Cambiar a Usuario Demo** y repetir valoración (ambos deben valorarse)
-

Escenario 4: Búsqueda y Filtros (3 minutos)

4.1 Buscar por categoría

1. Ir a "**Explorar Habilidades**"
2. Seleccionar categoría: Tecnología e Informática
3. **Resultado esperado:** Solo habilidades de esa categoría

4.2 Buscar por palabra clave

1. En el buscador, escribir: inglés
2. **Resultado esperado:** Habilidades que contengan "inglés" en título o descripción

4.3 Filtrar por tipo

1. Seleccionar filtro: Solo ofertas
2. **Resultado esperado:** Solo habilidades de tipo "Oferta"

4.4 Filtrar por ubicación

1. Seleccionar: A Coruña
 2. **Resultado esperado:** Solo usuarios de A Coruña
-

Escenario 5: Panel de Administración (3 minutos)

5.1 Acceso al panel admin

1. **Cerrar sesión** de usuario normal
2. Login como `admin@galitroco.com` / `Pass123456`
3. Ir a "**Panel de Administración**"
4. **Resultado esperado:** Acceso permitido (solo admins)

5.2 Ver estadísticas

1. En dashboard admin, verificar widgets:
 - Total de usuarios registrados
 - Total de habilidades publicadas
 - Intercambios completados
 - Valoración promedio de la plataforma
2. **Resultado esperado:** Estadísticas en tiempo real

5.3 Gestionar reportes (si hay)

1. Ir a sección "**Reportes**"
2. Ver reportes pendientes
3. Click en un reporte para ver detalles
4. Marcar como "**Revisado**" o "**Resuelto**"
5. **Resultado esperado:** Estado del reporte actualizado

5.4 Gestionar usuarios

1. Ir a sección "**Usuarios**"
2. Ver listado completo de usuarios
3. Buscar usuario específico
4. **Opciones disponibles:**
 - Ver perfil completo
 - Desactivar/Activar cuenta
 - Ver historial de intercambios
5. **Resultado esperado:** Acciones de moderación funcionan

Escenario 6: Recuperación de Contraseña (2 minutos)

6.1 Solicitar recuperación

1. **Cerrar sesión** de cualquier usuario
2. En página de login, click en "**¿Olvidaste tu contraseña?**"
3. Ingresar email: `demo@galitroco.com`
4. Click en "**Enviar enlace de recuperación**"
5. **Resultado esperado:**
 - Mensaje de confirmación
 - Email enviado con token de recuperación (vía Brevo API)

6.2 Verificar token (opcional)

1. Revisar consola de Brevo (si tienes acceso)
 2. **Resultado esperado:** Email entregado exitosamente
-

☒ Checklist de Verificación Final

Marca cada funcionalidad después de probarla:

Autenticación:

- ☐ Registro de nuevo usuario funciona
- ☐ Login con credenciales correctas
- ☐ Sesión persiste después de cerrar navegador
- ☐ Logout funciona correctamente
- ☐ Recuperación de contraseña envía email

Habilidades:

- ☐ Crear habilidad tipo "Oferta"
- ☐ Crear habilidad tipo "Demanda"
- ☐ Editar habilidad existente
- ☐ Pausar/Activar habilidad
- ☐ Eliminar habilidad
- ☐ Ver habilidades propias
- ☐ Explorar habilidades de otros usuarios

Intercambios:

- ☐ Proponer intercambio funciona
- ☐ Receptor recibe notificación
- ☐ Aceptar propuesta cambia estado
- ☐ Rechazar propuesta funciona
- ☐ Marcar como completado funciona
- ☐ Ver historial de intercambios

Valoraciones:

- ☐ Sistema de estrellas (1-5) funciona
- ☐ Comentario se guarda correctamente
- ☐ Valoración promedio se calcula
- ☐ Valoraciones visibles en perfil

Mensajería:

- ☐ Conversación se crea automáticamente
- ☐ Enviar mensaje funciona
- ☐ Mensajes se muestran en orden cronológico
- ☐ Notificación de nuevo mensaje

Búsqueda:

- ☐ Filtrar por categoría funciona
- ☐ Buscar por palabra clave
- ☐ Filtrar por tipo (oferta/demanda)
- ☐ Filtrar por ubicación

Panel Admin:

- ☐ Solo admins acceden al panel
- ☐ Estadísticas se muestran correctamente
- ☐ Gestión de reportes funciona
- ☐ Gestión de usuarios funciona

Troubleshooting Común

Problema: "No se puede conectar al servidor"

Solución:

- Verificar que la URL sea correcta: <https://galitroco-frontend.onrender.com>
- Render puede tardar 30-60 segundos en "despertar" el servicio si estuvo inactivo

Problema: "Error al crear habilidad"

Solución:

- Verificar que todos los campos obligatorios estén completos
- El título debe tener al menos 10 caracteres
- La descripción debe tener al menos 20 caracteres

Problema: "No puedo proponer intercambio"

Solución:

- Asegúrate de tener al menos 1 habilidad propia publicada
- No puedes proponer intercambio con tus propias habilidades
- La habilidad del otro usuario debe estar "Activa"

Problema: "No recibo email de recuperación"

Solución:

- Revisar carpeta de spam
- El email puede tardar 1-2 minutos en llegar
- Verificar que el email esté registrado en la base de datos

Endpoints API para Testing Avanzado

Si deseas probar la API directamente (con Postman, curl, etc.):

Base URL: <https://render-test-php-1.onrender.com/api.php>

Autenticación

```
POST /api.php?resource=auth&action=login
Body: {"email": "demo@galitroco.com", "password": "Pass123456"}
```

Habilidades

```
GET /api.php?resource=habilidades
GET /api.php?resource=habilidades&id=1
POST /api.php?resource=habilidades
```


Intercambios

```
GET /api.php?resource=intercambios
POST /api.php?resource=intercambios
PUT /api.php?resource=intercambios&id=1&action=acceptar
```

 **Documentación completa de API:** Ver archivo [TESTING_Y_ENDPOINTS_TFM.md](#)

Tiempo Total de Pruebas

Tipo de Prueba	Tiempo Estimado	Incluye
Prueba Express	5-10 minutos	Login + explorar habilidades + proponer intercambio
Prueba Rápida	10-15 minutos	Funcionalidades básicas (escenarios 1-2)
Prueba Completa	25-30 minutos	Todos los escenarios (1-6) + checklist
Prueba Exhaustiva	45-60 minutos	Todo + testing manual API + verificar documentación

 **Importante:** Añadir 1-2 minutos extra al primer acceso por cold start de Render.

REFERENCIAS Y DOCUMENTACIÓN ADICIONAL

Documentación técnica incluida:

- [backend/README_BACKEND.md](#) - Guía detallada del backend
- [frontend/README_FRONTEND.md](#) - Guía detallada del frontend
- [database/README_DATABASE.md](#) - Guía de la base de datos
- [TESTING_Y_ENDPOINTS_TFM.md](#) - Testing exhaustivo de la API

- [ARQUITECTURA_DEPLOY.md](#) - Arquitectura de despliegue
- [GUIA_RECUPERACION_PASSWORD.md](#) - Sistema de recuperación de contraseña

Documentación externa:

- [Angular Docs](#)
- [PHP Manual](#)
- [PostgreSQL Docs](#)
- [Angular Material](#)
- [Render Docs](#)
- [Supabase Docs](#)



HISTORIAL DE VERSIONES

v1.1 - PEC2 Final (28 Octubre 2025)

- ☒ Backend completo con 25 endpoints (23 operativos = 92%)
- ☒ Frontend Angular con 50% funcionalidades core (12/16 tests OK)
- ☒ Sistema de intercambios funcional (proponer + listar)
- ☒ Sistema de valoraciones (dialog implementado)
- ☒ Panel de administración operativo
- ☒ Testing exhaustivo en producción documentado
- ☒ Documentación técnica completa (25+ páginas)
- ☒ Ambos desplegados en Render.com con auto-deploy
- ☒ Sistema de autenticación con sesiones PHP + tokens hexadecimales

v1.0 - PEC2 Inicial (23 Octubre 2025)

- ☒ Primera versión funcional desplegada
- ☒ Backend operativo con endpoints principales
- ☒ Frontend básico con autenticación
- ☒ Base de datos en Supabase configurada

v0.5 - PEC1 (Septiembre 2025)

- ☒ Planificación inicial
- ☒ Análisis de requisitos
- ☒ Diseño de base de datos
- ☒ Wireframes de interfaces



LICENCIA DEL PROYECTO

Este proyecto es un Trabajo Final de Máster para la UOC con fines académicos.

Código fuente: Propiedad de Antonio Campos

Uso: Exclusivamente educativo

Redistribución: No permitida sin autorización



CAPTURAS DE PANTALLA Y RECURSOS VISUALES

Nota: Para ver capturas de pantalla de la aplicación en funcionamiento, consultar:

- [docs/screenshots/](#) (si existe en el proyecto)
- Memoria PEC2 (documento Word con imágenes incluidas)
- O probar directamente la aplicación en: <https://galitroco-frontend.onrender.com>

Recomendación para evaluadores: La mejor forma de evaluar es **probando la aplicación real en producción** siguiendo la guía de pruebas anterior.



CONSIDERACIONES ACADÉMICAS

Alcance del Proyecto (PEC2)

Este documento y la aplicación representan el estado del TFM en la **PEC2 (Octubre 2025)**.

Progreso actual:

- ☒ Backend: 92% funcional (23/25 endpoints operativos)
- ☒ Frontend: 50% funcional (12/16 tests completados - funcionalidades core)
- ☒ Base de datos: 100% diseñada e implementada
- ☒ Despliegue: 100% operativo en Render.com
- ☒ Documentación: Completa y exhaustiva

Próximos pasos (PEC3 - Diciembre 2025):

- ⌚ Completar funcionalidades pendientes del frontend (50% restante)
- ⌚ Implementar sistema de notificaciones en tiempo real
- ⌚ Mejorar UX/UI basado en feedback
- ⌚ Testing exhaustivo de usuarios
- ⌚ Corrección de bugs menores
- ⌚ Optimización de rendimiento
- ⌚ Memoria final completa

Limitaciones Conocidas (Plan Gratuito)

- ⚠ **Render Free Tier:** Cold start de 30-90 segundos tras inactividad
- ⚠ **Supabase Free:** Límite de 500 MB base de datos
- ⚠ **Brevo Free:** 300 emails/día máximo
- ⚠ **Sin dominio propio:** URLs técnicas (.onrender.com)

Estas limitaciones **NO afectan la funcionalidad** para propósitos académicos y de evaluación.



GLOSARIO DE TÉRMINOS

Para facilitar la evaluación, definimos términos técnicos clave:

- **Cold Start:** Tiempo de arranque del servidor tras inactividad (limitación de hosting gratuito)
 - **CORS:** Cross-Origin Resource Sharing - Política de seguridad para peticiones entre dominios
 - **Sesión PHP:** Mecanismo de autenticación server-side con cookies
 - **Token Hexadecimal:** Cadena de 64 caracteres (SHA-256) para identificar sesiones
 - **Endpoint:** URL específica de la API que realiza una operación (ej: crear habilidad)
 - **Guard:** Protección en Angular que verifica permisos antes de acceder a rutas
 - **Supabase:** Servicio cloud de PostgreSQL (alternativa open source a Firebase)
 - **Render:** Plataforma de hosting con despliegue automático desde GitHub
 - **Brevo:** Servicio de email transaccional (ex-Sendinblue)
-

Última actualización: 28 de octubre de 2025

Versión del documento: 1.1 (PEC2 - Revisión Final)

Estado: ☒ Listo para entrega y evaluación PEC2

Próxima revisión: Diciembre 2025 (PEC3 Final)