

ANÁLISIS Y ESTRUCTURA RECOMENDADA PARA ENTREGA PEC2

Fecha: 28 de octubre de 2025

Estudiante: Antonio Campos

Asignatura: Trabajo Final de Máster - UOC

Entrega: PEC2

ANÁLISIS DE REQUISITOS DE LA PEC2

☒ Documentos a entregar (según enunciado):

1. PROYECTO (Código + Instrucciones)

- ☒ Códigos fuente completos
- ☒ Instrucciones de uso
- ☒ Instrucciones de instalación

2. MEMORIA DE PROYECTO

Debe incluir:

- ☒ Continuación y mejoras de capítulos PEC1
- ☒ Diagramas y casos de uso
- ☒ Estudio de usabilidad
- ☒ Arquitectura de la aplicación
- ☒ Análisis de mercado
- ☒ Viabilidad

3. RECURSOS ADICIONALES

- ☒ Código propio
 - ☒ Código de terceros utilizado
 - ☒ Bases de datos (scripts SQL)
 - ☒ Licencias de terceros
-

FORMATO DE ENTREGA REQUERIDO

Según las normas de la PEC2:

Opción 1: Entrega separada

- **Memoria:** PEC2_mem_Campos_Antonio.pdf
 - **Proyecto:** PEC2_pry_Campos_Antonio.zip
-

Opción 2: Entrega unificada (RECOMENDADA) ☒

```
PEC2_Campos_Antonio.zip
├── documentacion/
│   └── PEC2_mem_Campos_Antonio.pdf
└── proyecto/
    └── [todo el código]
```

ESTRUCTURA RECOMENDADA COMO MIEMBRO DEL TRIBUNAL

ESTRUCTURA ÓPTIMA PARA EVALUACIÓN

```
PEC2_Campos_Antonio.zip
├── 1_DOCUMENTACION/
│   ├── PEC2_mem_Campos_Antonio.pdf          # MEMORIA PRINCIPAL
│   │   (OBLIGATORIO)
│   │   # Contenido:
│   │   # - Introducción y objetivos
│   │   # - Planificación temporal
│   │   # - Análisis de requisitos
│   │   # - Diagramas (casos de uso,
│   │   clases, secuencia)
│   │   # - Estudio de usabilidad
│   │   # - Arquitectura técnica
│   │   # - Análisis de mercado
│   │   # - Viabilidad económica
│   │   # - Conclusiones
│   ├── README_PROYECTO.pdf                # Guía técnica del proyecto
│   │   # (tu actual README_PEC2.pdf)
│   ├── INSTRUCCIONES_USO.pdf              # Guía para probar la
│   │   aplicación
│   │   # - URLs de acceso
│   │   # - Usuarios de prueba
│   │   # - Flujos de testing
│   │   # - Troubleshooting
│   ├── TESTING_BACKEND.pdf                # Documentación de testing
│   │   backend
│   │   ├── TESTING_FRONTEND.pdf           # Documentación de testing
│   │   │   frontend
│   │   ├── LICENCIAS_TERCEROS.pdf          # Licencias software
│   │   │   utilizado
│   │   └── diagramas/
│   │       # NUEVO - Imágenes para la
```

memoria		
	diagrama_casos_uso.png	
	diagrama_clases.png	
	diagrama_secuencia_intercambio.png	
	diagrama_base_datos.png	
	arquitectura_sistema.png	
	mockup_interfaz_1.png	
	mockup_interfaz_2.png	
	2_PROYECTO/	
	README.md	# Instrucciones técnicas
rápidas		
	api.php	# Proxy API (CRÍTICO)
	Dockerfile	# Config Docker Render
	render.yaml	# Config deploy Render
	.gitignore	# Exclusiones Git
	backend/	# BACKEND PHP
	api/	# Endpoints HTTP
	index.php	# Router principal
	auth.php	# Autenticación
	habilidades.php	# CRUD habilidades
	intercambios.php	# Sistema intercambios
	valoraciones.php	# Valoraciones
	conversaciones.php	# Mensajería
	reportes.php	# Reportes
	notificaciones.php	# Notificaciones
	usuarios.php	# Gestión usuarios
	categorias.php	# Categorías
	config/	# Configuración
	database.php	# Conexión BD
	cors.php	# CORS y cookies
	models/	# Modelos de datos
	Usuario.php	
	Habilidad.php	
	Intercambio.php	
	Valoracion.php	
	Conversacion.php	
	Mensaje.php	
	Notificacion.php	
	Reporte.php	
	utils/	# Utilidades
	Response.php	# Respuestas JSON
	Auth.php	# Sesiones y tokens
	EmailService.php	# Envío de emails
	.htaccess	# Config Apache
	README.md	# Doc backend

— frontend/	# FRONTEND ANGULAR
— src/	
— app/	
— core/	# Servicios centrales
— guards/	# Guards
— interceptors/	# Interceptores
— services/	# Servicios
— features/	# Módulos funcionales
— admin/	# Panel admin
— auth/	# Login/Registro
— habilidades/	# CRUD habilidades
— home/	# Home
— intercambios/	# Intercambios
— perfil/	# Perfil
— reportes/	# Reportes
— valoraciones/	# Valoraciones
— layout/	# Layout
— header/	
— footer/	
— main-layout/	
— shared/	# Compartidos
— components/	
— models/	
— app.component.ts	
— app.config.ts	
— app.routes.ts	
— environments/	# Configuración entornos
— environment.ts	# Desarrollo
— environment.prod.ts	# Producción
— index.html	
— main.ts	
— styles.scss	
— public/	# Assets públicos
— angular.json	
— package.json	
— tsconfig.json	
— README.md	
— database/	# SCRIPTS SQL
— schema.sql	# Esquema completo (12 tablas + 1 vista)
— seeds.sql	# Datos de ejemplo (3 usuarios + 6 habilidades)
— README_INSTALACION.md	# Instrucciones de instalación
BD	

CHECKLIST DE PREPARACIÓN DE LA ENTREGA

☒ 1. DOCUMENTACIÓN (Carpeta **1_DOCUMENTACION/**)

A. Memoria Principal (OBLIGATORIO)

PEC2_mem_Campos_Antonio.pdf

Contenido mínimo según requisitos:

- ☐ **Portada**
 - Título del proyecto
 - Autor (Antonio Campos)
 - Universidad (UOC)
 - Asignatura (TFM)
 - Fecha (Octubre 2025)
- ☐ **Índice**
- ☐ **1. Introducción** (continuación PEC1)
 - Contexto del proyecto
 - Motivación
 - Objetivos generales
- ☐ **2. Objetivos del proyecto** (mejora PEC1)
 - Objetivos principales
 - Objetivos secundarios
 - Alcance del proyecto
- ☐ **3. Planificación temporal** (revisión PEC1)
 - Cronograma PEC1 vs realidad
 - Ajustes realizados
 - Nueva planificación PEC2-PEC3
 - Diagrama de Gantt actualizado
- ☐ **4. Análisis de requisitos** (nuevo/ampliado)
 - Requisitos funcionales
 - Requisitos no funcionales
 - Priorización (MoSCoW)
- ☐ **5. Casos de uso** (NUEVO - requisito PEC2)
 - Diagrama general de casos de uso
 - Descripción detallada de casos principales:

- UC01: Registro de usuario
 - UC02: Login
 - UC03: Publicar habilidad
 - UC04: Buscar habilidades
 - UC05: Proponer intercambio
 - UC06: Aceptar/Rechazar intercambio
 - UC07: Valorar usuario
 - UC08: Administración (reportes)
- ☐ **6. Estudio de usabilidad** (NUEVO - requisito PEC2)
 - Principios de usabilidad aplicados
 - Diseño de interfaz (mockups/capturas)
 - Flujos de usuario
 - Accesibilidad
 - Responsive design
 - ☐ **7. Arquitectura de la aplicación** (NUEVO - requisito PEC2)
 - Arquitectura general (diagrama)
 - Stack tecnológico
 - Patrón de diseño (MVC, capas)
 - Diagrama de clases (backend)
 - Diagrama de componentes (frontend)
 - Diagrama de secuencia (flujo de intercambio)
 - Diagrama de base de datos (ERD)
 - ☐ **8. Análisis de mercado** (NUEVO - requisito PEC2)
 - Estudio de competidores
 - Análisis DAFO
 - Propuesta de valor diferencial
 - Público objetivo
 - ☐ **9. Viabilidad** (NUEVO - requisito PEC2)
 - Viabilidad técnica
 - Viabilidad económica (costes estimados)
 - Viabilidad temporal
 - Riesgos identificados
 - ☐ **10. Implementación técnica**
 - Tecnologías utilizadas
 - Decisiones de diseño
 - Problemas encontrados y soluciones
 - ☐ **11. Testing y validación**
 - Estrategia de testing

- Casos de prueba ejecutados
- Resultados obtenidos
- ☐ **12. Conclusiones**
 - Objetivos cumplidos
 - Lecciones aprendidas
 - Trabajo futuro (PEC3)
- ☐ **13. Referencias y bibliografía**
- ☐ **14. Anexos**
 - Capturas de pantalla
 - Código relevante
 - Licencias de terceros

B. Documentación técnica complementaria

- ☐ **README_PROYECTO.pdf** (tu actual README_PEC2.pdf)
 - Instrucciones de instalación local
 - Instrucciones de uso del sistema
 - Credenciales de prueba
 - Troubleshooting
- ☐ **TESTING_BACKEND.pdf** (tu actual TESTING_Y_ENDPOINTS_TFM.pdf)
- ☐ **TESTING_FRONTEND.pdf** (tu actual TESTING_FRONTEND_MANUAL.pdf)
- ☐ **LICENCIAS_TERCEROS.pdf**

C. Carpeta de diagramas (RECOMENDADO)

- ☐ Diagrama de casos de uso
- ☐ Diagrama de clases (backend)
- ☐ Diagrama de secuencia (flujo de intercambio)
- ☐ Diagrama de base de datos (ERD)
- ☐ Diagrama de arquitectura del sistema
- ☐ Mockups de interfaz

☒ 2. PROYECTO (Carpeta **2_PROYECTO/**)

Estructura del código

- ☐ ☒ Carpeta **backend/** completa
- ☐ ☒ Carpeta **frontend/** completa
- ☐ ☒ Carpeta **database/** con scripts SQL
- ☐ ☒ Archivo **api.php** en raíz (crítico)

- ☐ ☒ Archivo **README.md** en raíz con instrucciones rápidas
- ☐ ☒ Archivos de configuración (Dockerfile, render.yaml)
- ☐ ⚠ **NO incluir** carpeta **.git/** (demasiado pesada)
- ☐ ⚠ **NO incluir** carpeta **node_modules/** (demasiado pesada)
- ☐ ⚠ **NO incluir** carpeta **old/** (archivos internos)
- ☐ ⚠ **NO incluir** archivos de configuración local (.env, config.local.php)

Archivos esenciales a verificar:

Backend:

- ☐ 10 endpoints funcionales en **backend/api/**
- ☐ 8 modelos en **backend/models/**
- ☐ Configuración de BD en **backend/config/database.php**
- ☐ Utilidades (Response, Auth, EmailService)

Frontend:

- ☐ Código fuente en **frontend/src/**
- ☐ Configuración Angular (angular.json, package.json)
- ☐ Environments (environment.ts, environment.prod.ts)

Database:

- ☐ **schema.sql** - Esquema completo (12 tablas + 1 vista)
- ☐ **seeds.sql** - Datos de ejemplo
- ☐ Instrucciones de instalación

CRITERIOS DE VALORACIÓN (ANÁLISIS)

1. Seguimiento y trabajo en equipo (20%)

Estrategia para maximizar puntos:

- ☒ Has documentado toda la comunicación en README_PEC2.md
- ☒ Has seguido las recomendaciones del tutor (no JWT, API HTTP no REST)
- ☒ Has aplicado las correcciones sugeridas
- 💡 **Añade en la memoria:** Sección sobre iteraciones con el tutor y cambios aplicados


2. Realización y cumplimiento de la planificación (20%)

Puntos clave:

- ⚠ Debes documentar en la memoria:
 - Planificación inicial (PEC1)
 - Desviaciones encontradas
 - Replanificación aplicada
 - Justificación de cambios
- 💡 **Recomendación:** Crear tabla comparativa "Planificado vs Realizado"

3. Calidad del producto (30%)

Tu situación actual:

- ☒ Backend completo y funcional (10 endpoints)
- ☒ Frontend 50% implementado (12/16 tests OK según tu doc)
- ☒ Base de datos completa (12 tablas + 1 vista)
- ☒ Despliegue en producción funcional
- ☒ Testing documentado
-  **Mejora para PEC2:** Documenta arquitectura y decisiones técnicas

4. Calidad de la memoria (30%)

Checklist de calidad:

- ☐ Portada profesional
- ☐ Índice completo con numeración
- ☐ Redacción académica (no coloquial)
- ☐ Sin faltas ortográficas
- ☐ Uso correcto de tablas e imágenes
- ☐ Referencias bibliográficas en formato APA
- ☐ Numeración de páginas
- ☐ Uso de diagramas UML correctos
- ☐ Capturas de pantalla con buena resolución
- ☐ Conclusiones bien argumentadas

RECOMENDACIONES ESPECÍFICAS PARA LA MEMORIA

Diagramas que DEBES incluir (requisito PEC2):

1. Diagrama de Casos de Uso

```
Actor: Usuario
- Login
- Registrarse
- Publicar habilidad
- Buscar habilidades
- Proponer intercambio
- Valorar usuario
```

```
Actor: Administrador
- Gestionar reportes
- Moderar contenido
```

2. Diagrama de Clases (Backend)

Usuario
Habilidad
Intercambio
Valoracion
Conversacion
Mensaje
Notificacion
Reporte

3. Diagrama de Secuencia: "Proponer Intercambio"

Usuario1 → Frontend → Backend → BD
1. Selecciona habilidad
2. POST /intercambios
3. Valida sesión
4. Crea intercambio
5. Envía notificación a Usuario2
6. Retorna confirmación

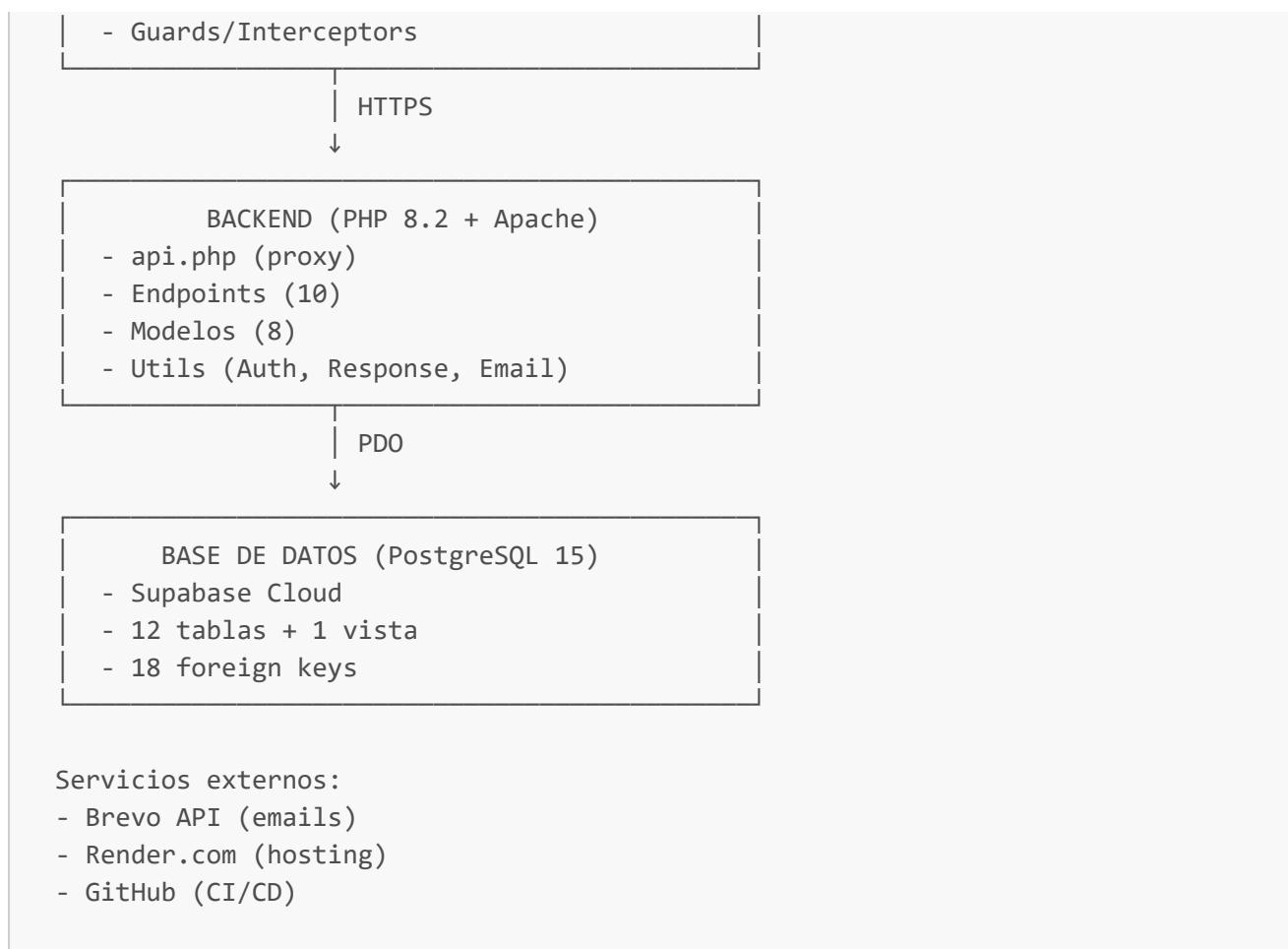
4. Diagrama de Base de Datos (ERD)

12 tablas:
- usuarios (PK: id)
- habilidades (FK: usuario_id, categoria_id)
- intercambios (FK: habilidad_ofrecida_id, habilidad_solicitada_id, proponente_id, receptor_id)
- valoraciones (FK: evaluador_id, evaluado_id, intercambio_id)
- conversaciones (FK: intercambio_id)
- mensajes (FK: conversacion_id, emisor_id)
- notificaciones (FK: usuario_id)
- reportes (FK: reportador_id, revisor_id)
- categorias_habilidades (PK: id)
- password_resets (FK indirecto: email)
- sesiones (FK: usuario_id)
- participantes_conversacion (FK: conversacion_id, usuario_id)

+ 1 vista: estadisticas_usuarios

5. Diagrama de Arquitectura

```
graph LR
    subgraph Frontend [FRONTEND (Angular 19)]
        direction TB
        C[Components]
        S[Services]
    end
```



TAREAS PENDIENTES PARA COMPLETAR LA ENTREGA

CRÍTICAS (Obligatorias para PEC2)

1. ☐ Crear la memoria principal [PEC2_mem_Campos_Antonio.pdf](#)

- Todos los capítulos requeridos
- Diagramas de casos de uso
- Estudio de usabilidad
- Arquitectura técnica
- Análisis de mercado
- Viabilidad

2. ☐ Crear diagramas UML

- Casos de uso
- Clases
- Secuencia
- Base de datos (ERD)

3. ☐ Renombrar documentos existentes

- [README_PEC2.pdf](#) → [README_PROYECTO.pdf](#)
- [TESTING_Y_ENDPOINTS_TFM.pdf](#) → [TESTING_BACKEND.pdf](#)

- TESTING_FRONTEND_MANUAL.pdf → TESTING_FRONTEND.pdf

🌀 IMPORTANTES (Recomendadas)

4. ☐ **Crear README_INSTALACION.md** en carpeta `database/`
 - Instrucciones paso a paso
 - Requisitos previos
 - Comandos exactos
5. ☐ **Crear .gitignore para la entrega**
 - Excluir `.git/`
 - Excluir `node_modules/`
 - Excluir `old/`
 - Excluir archivos de configuración local
6. ☐ **Verificar que NO se incluyan:**
 - Archivos de configuración con credenciales
 - Carpeta `.git/` (muy pesada)
 - Carpeta `node_modules/` (muy pesada)
 - Carpeta `old/` (archivos internos)

🌀 OPCIONALES (Mejoran la presentación)

7. ☐ Crear capturas de pantalla de alta calidad
8. ☐ Crear mockups de diseño
9. ☐ Crear vídeo demo (3-5 minutos)
10. ☐ Crear presentación PowerPoint de resumen

📦 COMANDO FINAL PARA GENERAR EL ZIP

Paso 1: Limpiar archivos innecesarios

```
# Desde la raíz del proyecto
cd d:\xamppUOC\htdocs\probatfm

# Eliminar carpetas pesadas/innecesarias
Remove-Item -Recurse -Force frontend\node_modules -ErrorAction SilentlyContinue
Remove-Item -Recurse -Force frontend\angular -ErrorAction SilentlyContinue
Remove-Item -Recurse -Force .git -ErrorAction SilentlyContinue
Remove-Item -Recurse -Force old -ErrorAction SilentlyContinue
```

Paso 2: Crear estructura de entrega

```
# Crear carpeta temporal de entrega
$entrega = "D:\PEC2_Campos_Antonio"
New-Item -ItemType Directory -Path $entrega -Force
New-Item -ItemType Directory -Path "$entrega\1_DOCUMENTACION" -Force
New-Item -ItemType Directory -Path "$entrega\2_PROYECTO" -Force

# Copiar documentación
Copy-Item "d:\xampUOC\htdocs\probatfm\PEC2_mem_Campos_Antonio.pdf"
"$entrega\1_DOCUMENTACION\"
Copy-Item "d:\xampUOC\htdocs\probatfm\README_PEC2.pdf"
"$entrega\1_DOCUMENTACION\README_PROYECTO.pdf"
Copy-Item "d:\xampUOC\htdocs\probatfm\TESTING_Y_ENDPOINTS_TFM.pdf"
"$entrega\1_DOCUMENTACION\TESTING_BACKEND.pdf"
Copy-Item "d:\xampUOC\htdocs\probatfm\TESTING_FRONTEND_MANUAL.pdf"
"$entrega\1_DOCUMENTACION\TESTING_FRONTEND.pdf"
Copy-Item "d:\xampUOC\htdocs\probatfm\LICENCIAS_TERCEROS.pdf"
"$entrega\1_DOCUMENTACION\"

# Copiar proyecto (sin .git, node_modules, old)
Copy-Item "d:\xampUOC\htdocs\probatfm\*" "$entrega\2_PROYECTO\" -Recurse -
Exclude @('.git','node_modules','.angular','old')

# Crear ZIP
Compress-Archive -Path "$entrega\*" -DestinationPath
"D:\PEC2_Campos_Antonio.zip" -Force

Write-Host "☑ Entrega creada en: D:\PEC2_Campos_Antonio.zip"
```

☑ CHECKLIST FINAL ANTES DE ENTREGAR

Verificación de contenido:

- ☐ El ZIP se llama `PEC2_Campos_Antonio.zip`
- ☐ Dentro hay 2 carpetas: `1_DOCUMENTACION` y `2_PROYECTO`
- ☐ La memoria se llama `PEC2_mem_Campos_Antonio.pdf`
- ☐ La memoria tiene TODOS los capítulos requeridos
- ☐ Hay diagramas de casos de uso, clases, secuencia y BD
- ☐ El código fuente está completo (backend, frontend, database)
- ☐ El archivo `api.php` está en la raíz del proyecto
- ☐ NO hay carpeta `.git/` (reduce 200+ MB)
- ☐ NO hay carpeta `node_modules/` (reduce 500+ MB)
- ☐ NO hay carpeta `old/` (archivos internos)
- ☐ El tamaño total del ZIP es < 50 MB
- ☐ Has probado descomprimir el ZIP y verificar que todo está

Verificación de calidad:

- ☐ La memoria tiene buena redacción (sin coloquialismos)

- ☐ No hay faltas de ortografía
 - ☐ Las imágenes tienen buena resolución
 - ☐ Los diagramas UML son correctos
 - ☐ Las tablas están bien formateadas
 - ☐ Hay referencias bibliográficas en formato APA
 - ☐ El índice tiene numeración de páginas
 - ☐ Las conclusiones están bien argumentadas
-

RESUMEN EJECUTIVO






Lo que TIENES:

- ☒ Proyecto funcional desplegado
- ☒ Backend completo (10 endpoints)
- ☒ Frontend 50% implementado
- ☒ Base de datos completa
- ☒ Testing documentado
- ☒ Documentación técnica (README_PEC2.pdf)

Lo que FALTA para PEC2:

- ☒ Memoria académica formal (PEC2_mem_Campos_Antonio.pdf)
- ☒ Diagramas UML (casos de uso, clases, secuencia, ERD)
- ☒ Estudio de usabilidad
- ☒ Análisis de mercado
- ☒ Viabilidad económica
- ☒ Reorganizar en estructura de entrega (2 carpetas)

Prioridad de tareas:

1.  **Crear la memoria principal** (requisito obligatorio)
 2.  **Crear diagramas UML** (requisito obligatorio PEC2)
 3.  **Reorganizar archivos** en estructura 1_DOCUMENTACION + 2_PROYECTO
 4.  **Limpiar archivos** innecesarios (.git, node_modules, old)
 5.  **Generar ZIP final** con nomenclatura correcta
-

Tiempo estimado para completar: 8-12 horas de trabajo

Fecha límite PEC2: 5 de noviembre de 2025 (8 días restantes)

- ☒ **ESTRUCTURA VALIDADA Y LISTA PARA SER IMPLEMENTADA**