

TPE401 — Taller de Proyecto de Especialidad

Carrera: Técnico en Programación y Análisis de Sistemas

Duración: 90 horas académicas (3 clases/semana; \approx 60 horas reloj; 9–10 semanas)

Trabajo: Individual **Modalidad:** Presencial (laboratorio)

Evaluación: 50 % Prototipo + 50 % Informes (5 avances, 10 % c/u)

Obligatorio: GitHub (código, issues, tablero Kanban^a)

^aKanban: método visual de gestión de trabajo en columnas: *Por hacer* / *En progreso* / *Hecho*. Úsalo en GitHub Projects.

Lineamientos generales

Método de desarrollo (1 persona). Usa **Kanban personal**¹ en GitHub Projects + prácticas ligeras de Lean y XP^a.

Modelado. UML^b simplificado (Casos de Uso, Actividades) + MER^c + **DFD**^d (alternativa no-OO). Wireframes^e.

Software libre. Reflexiona sobre *La Catedral y el Bazar* (Eric S. Raymond) y elige licencia (MIT/GPL/Apache).

Definiciones tempranas (Avance 2). Deja por escrito: método, lenguaje/framework, SGBD y motivación.

Glosario mínimo. Objetivos SMART^f, CRUD^g, pitch^h.

^aLean/XP: enfoques ágiles para entregar valor rápido, refactorizar y probar temprano, evitando desperdicios.

^bUML: lenguaje de modelado para diagramas de *Casos de Uso*, *Clases*, *Actividades*, etc. Úsalo de forma simple, comunicativa.

^cMER: Modelo Entidad-Relación para diseñar la base de datos relacional.

^dDFD (Diagrama de Flujo de Datos): alternativa *no orientada a objetos* para representar procesos, almacenes y flujos de datos a diferentes niveles (0, 1, ...).

^eWireframe: bosquejo de pantallas para acordar interfaz antes de programar.

^fSMART: Específicos, Medibles, Alcanzables, Relevantes y con Tiempo definido.

^gCRUD: Crear, Leer, Actualizar y Borrar; operaciones básicas sobre datos.

^hPitch: presentación breve (5–7 min) para explicar problema, solución y valor del prototipo.

Cronograma por semanas

El taller dura 10 semanas (**90 horas académicas**). Cada semana son 9 horas académicas distribuidas en 3 clases. **Revisa y actualiza tu tablero Kanban en GitHub en cada clase.**

Semana	Avance
1	Repo creado en GitHub, README inicial, tablero Kanban, borrador de objetivos SMART.
2	Avance 1: Informe inicial (problema, objetivos SMART, innovación), reflexión Catedral y Bazar, licencia, diagrama procesos/DFD.
3	Requerimientos RF/RNF, MER preliminar, Casos de Uso, wireframes, Issues.
4	Avance 2: Informe análisis y diseño, MER final, Casos de Uso, wireframes, BD inicial en SGBD.
5	Desarrollo CRUD base, commits frecuentes, capturas de pantallas.
6	Avance 3: Informe desarrollo inicial, CRUD funcional con BD, prueba exploratoria.
7	Diseño del Plan de Pruebas (casos, entradas/salidas, criterios).
8	Avance 4: Ejecución del Plan de Pruebas, evidencias, validaciones, reportes.
9	Integración final, preparación del Resumen Ejecutivo y del pitch.
10	Avance 5: Informe final, dossier de modelos, demo y pitch, README final, tablero cerrado.

Hitos, momentos y entregables (por avance)

Avance 1 (Semana 2, 18h acumuladas)

Entregables: Informe de definición (problema, objetivos SMART³, innovación), reflexión *Catedral vs Bazar* + licencia, diagrama de procesos (Actividades/BPMN) o DFD.

GitHub: repo + README, tablero Kanban base, Issue inicial.

Avance 2 (Semana 4, 36h acumuladas)

Entregables: Informe de análisis y diseño (RF/RNF, **método**, **tecnologías**), MER, Casos de Uso, wireframes.

Software: BD inicial en SGBD. **GitHub:** Issues por RF y tareas priorizadas.

Avance 3 (Semana 6, 54h acumuladas)

Entregables: Informe de desarrollo inicial (módulos, capturas), **prueba exploratoria** (errores detectados).

Software: CRUD⁴ funcional con BD. **GitHub:** commits frecuentes; Issues de bugs.

Avance 4 (Semana 8, 72h acumuladas)

Entregables: Plan de Pruebas (casos, datos, salidas esperadas), ejecución con evidencias, mejoras aplicadas.

Modelos: Trazabilidad RF ↔ Casos de Prueba. **GitHub:** Issues cerrados.

Avance 5 (Semana 10, 90h acumuladas)

Entregables: Informe final (Resumen Ejecutivo), dossier de modelos (procesos/DFD, MER, UML), **demo** + **pitch**⁵.

GitHub: README final, tablero completo, Issues cerrados. *Extra opcional* (si necesitas subir notas de informes): Manual de usuario/técnico.

Criterios de evaluación

Criterios del Prototipo (50 %)

- **Funcionalidad CRUD** completa y estable.
- **Validaciones y reportes** (coherentes con requerimientos).
- **Calidad de código:** estructura, comentarios, claridad.
- **Pruebas** ejecutadas (Avance 4) y **correcciones** evidenciadas.
- **Innovación aplicada** (usabilidad, eficiencia, aportes simples de IA si procede).

Criterios de Informes (50 %)

- **Claridad y coherencia:** redacción, orden, referencias a Issues/commits.
- **Aplicación de malla:** análisis, diseño (MER/UML/DFD), pruebas.
- **Evidencias:** capturas, tablas (trazabilidad, plan de pruebas).
- **Puntualidad** y consistencia con el repositorio **GitHub**.

Guía de entregables por semanas

Cada semana debes entregar evidencia clara de tu trabajo en **GitHub** (commits, Issues, tablero). Esta tabla resume los productos mínimos.

Semana	Entregables clave
1	Repo creado, README, tablero Kanban; definición de problema y usuarios; borrador de objetivos SMART.
2	A1: Informe inicial, reflexión sobre software libre, licencia, diagrama de procesos/DFD.
3	Requerimientos RF/RNF; MER preliminar; Casos de Uso; wireframes; Issues creados.
4	A2: Informe de análisis y diseño, MER final, Casos de Uso, wireframes, BD inicial en SGBD.
5	Desarrollo CRUD base; commits frecuentes; capturas de pantallas de módulos.
6	A3: Informe de desarrollo inicial; CRUD funcional conectado a BD; registro de errores en Issues.
7	Diseño del Plan de Pruebas con tabla de casos y criterios.
8	A4: Ejecución del Plan de Pruebas, informe de resultados, validaciones, reportes; Issues cerrados.
9	Integración final; Resumen Ejecutivo redactado; preparación del pitch.
10	A5: Informe final, dossier de modelos, demo del prototipo y pitch; README y tablero final en GitHub.

Plantillas rápidas de pruebas (para A4)

ID	Requerimiento	Entrada	Salida esperada	Estado	
CP-01	RF-01 Crear usuario	Nombre, Email válido, Rol	“Creado” + registro en BD	OK	Falla
CP-02	RF-02 Login	Email + Password	Acceso correcto (rol)	OK	Falla
CP-03	RNF-01 Validación	Email sin @	Mensaje de error visible	OK	Falla

Fecha	Versión	Casos ejecutados	Errores detectados	Acciones/Commits	
2025-04-10	v0.5	CP-01, CP-02, CP-03	ERR-01, ERR-02	#12 login; email regex	fix #13