

GUÍA PARA EL ALUMNADO

Hackatón FP RepoProjects (2º DAM + 2º ASIR)

Duración: 4 horas (descanso 30 min)

Stack obligatorio: Angular + NodeJS (Express + TypeScript) + MySQL

Entorno: VS Code + IA integrada (Codeium, Copilot, Codex u otra)

1. Objetivo del hackatón

El objetivo es desarrollar un MVP funcional de un repositorio de proyectos de final de ciclo. El foco del taller es aprender a trabajar con herramientas de IA para generar código, tomando el rol de analistas y arquitectos que diseñan y validan.

2. Arquitectura técnica

Frontend: Angular + Angular Material + Reactive Forms + HttpClient + Interceptor.

Backend: NodeJS + Express + TypeScript + JWT + middleware de roles.

Base de datos: MySQL con scripts SQL reproducibles.

Control de versiones: GitHub obligatorio (repositorio por equipo).

3. Herramientas necesarias

Obligatorio: Visual Studio Code, NodeJS (LTS), Angular CLI, MySQL Server, MySQL Workbench o phpMyAdmin, Git.

IA integrada en VS Code (a elección del equipo): Codeium/Windsurf Plugin, GitHub Copilot (si disponible), Codex u otras extensiones.

4. Concepto clave: nuevo rol profesional

En este taller el rol principal no es escribir código manualmente, sino diseñar la solución, definir contratos API, redactar prompts efectivos, validar el resultado y corregir errores. La IA es un asistente que genera código, pero el equipo es responsable de su calidad y funcionamiento.

5. Requisitos funcionales

La aplicación debe permitir gestionar un repositorio de proyectos de final de ciclo.

Roles: ALUMNO, DOCENTE, ADMIN.

6. Entidad principal: Proyecto

Campos mínimos: titulo, descripcion, ciclo (DAM/ASIR), curso_academico, tecnologias, repositorio_url, demo_url (opcional), estado, autor, fecha_creacion.

7. Estados del proyecto

Estados: BORRADOR, PENDIENTE, PUBLICADO, RECHAZADO.

8. Flujo mínimo obligatorio

1) ALUMNO inicia sesión. 2) Crea proyecto (BORRADOR). 3) Envía a revisión (PENDIENTE). 4) DOCENTE inicia sesión. 5) Publica proyecto (PUBLICADO). 6) Listado público muestra proyectos PUBLICADOS.

9. Pantallas mínimas esperadas

Login (email/password).

Listado público de proyectos (solo PUBLICADOS) con filtros.

Mis proyectos (ALUMNO): crear, listar propios, enviar a revisión.

Pendientes (DOCENTE/ADMIN): listar pendientes y publicar.

10. Filtros mínimos del listado público

Filtro por ciclo (DAM/ASIR), filtro por texto (titulo/descripcion/tecnologias) y filtro por curso_academico.

11. Requisitos técnicos obligatorios

Backend: Express + TypeScript, API REST /api, JWT y control por roles.

DB: carpeta /db con schema.sql y seed.sql.

Frontend: Angular Material, reactive forms, interceptor para JWT.

12. Entregables mínimos del repositorio

README.md con instrucciones de instalación y ejecución.

db/schema.sql y db/seed.sql.

backend funcional y frontend funcional.

13. Modo de trabajo recomendado con IA

Paso 1: diseñar antes de pedir código. Paso 2: escribir prompts concretos indicando stack, archivos, endpoints y formatos JSON. Paso 3: verificar con pruebas. Paso 4: iterar corrigiendo errores.

14. Ejecución local mínima

Backend: cd backend, npm install, npm run dev.

Frontend: cd frontend, npm install, ng serve -o.

DB: ejecutar schema.sql y seed.sql en Workbench/phpMyAdmin.

15. Configuración de github

Utilizaremos GitHub Classroom que tiene como ventaja crear 1 tarea y Github genera 1 repo por equipo.

En el siguiente enlace se podrán unir a github classroom:

<https://classroom.github.com/classrooms/261791337-pro2p-dam-asir-hackaton-2026-classroom-8ec359>

1. Entren con su cuenta GitHub.
2. Verán una pantalla tipo:
 - a. “Join an existing team” o “Create a new team”
3. Un alumno del equipo pulsa:
 - a. **Create a new team**
4. Le pone nombre:
 - a. Equipo-01
5. Los demás seleccionan ese equipo y se unen.

IMPORTANTE:

- Solo **1 alumno por equipo** crea el team.
- Nombre obligatorio: Equipo-01, Equipo-02, etc.
- Cada equipo debe tener **exactamente 5 miembros** (en tu caso 25 alumnos).

Una vez asignados los miembros a cada equipo:

Una vez aceptado, GitHub Classroom creará el repositorio del equipo.

1. Entrar en el repositorio del equipo.
2. Verificar que contiene las carpetas:
 - *frontend/*
 - *backend/*
 - *db/*
 - *Docs/*

Abrir terminal:

```
git clone <URL_DEL_REPO>
```

```
cd <nombre_repo>
```

```
code .
```

Donde la URL DEL REPO es: <https://github.com/PRO2P-DAM-ASIR-HACKATON-2026/template-repoprojects>

Reparto de roles (minuto 10-12)

Cada equipo asigna roles mínimos:

- Analista/Arquitecto (coordina prompts y decisiones)
- Backend lead
- Frontend lead
- DB/Infra lead (ASIR preferente)
- Integrador/QA

Abrir el backlog y repartir tareas

En el repositorio abrir:

docs/backlog.md

1. Leer el backlog completo.
2. Dividir el trabajo por sprint.
3. Cada alumno se asigna una tarea concreta.
4. El analista registra decisiones en un fichero (si existe docs/decisiones.md o en el README).

16. Flujo obligatorio de trabajo – Normas de equipo

Norma 1 — No se programa “a lo loco”

Antes de pedir código a la IA, deben definir:

- endpoint
- estructura de tabla
- JSON request/response
- rol necesario

Norma 2 — Prompts cortos por fases

Prohibido pedir:

“Hazme toda la app completa”.

Se trabaja en fases:

1. DB + seed
2. Backend auth
3. Frontend login
4. Proyectos públicos
5. Mis proyectos
6. Moderación

Norma 3 — Integración continua

Cada cambio debe terminar en:

- commit
- push

Frecuencia recomendada:

- cada 20–30 minutos

Norma 4 — Estructura fija del repositorio

No cambiar nombres base:

- frontend/
- backend/
- db/
- docs/

Checklist técnico mínimo (lo primero que deben verificar)

Verificar Node y Angular

En terminal:

node -v

npm -v

ng version

