



📋 ENUNCIAT PRACTICA INTERMODULAR DWES-DWEC-PI SCENA MVP

📅 **Període d'avaluació:** 2 de febrer de 2026 - 22 de febrer de 2026

⌚ **Durada:** 20 dies naturals (~14 dies hàbils)

🎓 **Estudis:** Cicle Formatiu de Grau Superior - Desenvolupament d'Aplicacions Web (DAW)

👤 **Tutor:** Xavier Sastre

🏫 **Centre:** CIFP Francesc de Borja Moll

📋 TAULA DE CONTINGUTS

1. [CONTEXT](#)
2. [MEMBRES DE L'EQUIP](#)
3. [OBJECTIUS D'APRENENTATGE](#)
4. [FUNCIONALITATS REQUERIDES \(MVP\)](#)
5. [FORA D'ABAST](#)
6. [STACK TECNOLÒGIC](#)
7. [MODEL DE DADES](#)
8. [PLANIFICACIÓ TEMPORAL](#)
9. [CRITERIS D'AVALUACIÓ \(Total: 100 punts\)](#)
10. [RESUM DE TASQUES OBLIGATÒRIES](#)
11. [ENTREGABLES](#)
12. [INSTRUCCIONS D'ENTREGA I DEFENSA](#)
13. [EXTRES OPCIONALS \(Bonus: +10%\)](#)
14. [FAQ \(Pregutes Freqüents\)](#)
15. [CONSELLS FINALS](#)

AVÍS SOBRE L'ús D'INTEL·LIGÈNCIA ARTIFICIAL (IA)

Està **estrictament prohibit** l'ús de codi directament extret de motors d'IA (com ChatGPT, GitHub Copilot, Claude, etc.) per a la realització d'aquesta pràctica.

- L'objectiu d'aquest projecte és avaluar les **teves competències reals** de programació.
 - Qualsevol fragment de codi que l'alumne no sàpiga explicar detalladament o que no coincideixi amb el nivell i les tècniques treballades a classe es considerarà un intent de frau.
 - **La detecció d'ús fraudulent d'IA suposarà el suspens immediat (nota 0) del projecte.**
-

CONTEXT

Scena és una aplicació web progressiva (PWA) concebuda per revolucionar el descobriment d'esdeveniments i locals d'oci a Mallorca. El projecte complet inclou funcionalitats avançades com:

- Visualització de mapa de calor generat per interacció d'usuaris
- Tres rols d'usuari diferenciats (Usuari bàsic, Administrador, Promotora)
- Sistema de check-in amb geolocalització en temps real
- Anàlisi de tendències i recomanacions personalitzades
- Integració amb APIs de geolocalització i mapes interactius complexos

El projecte original està dissenyat per ser desenvolupat en aproximadament **12 setmanes** amb un equip de desenvolupadors especialitzats (backend, frontend, UI/UX).

Justificació de la Reducció d'Abast

Aquesta evaluació **NO pretén replicar el projecte complet**, sinó desenvolupar un **MVP (Minimum Viable Product)** funcional que:

- Demostri competències core en desenvolupament web servidor (DWES)
- Demostri competències core en desenvolupament web client (DWEC)
- Sigui viable en **20 dies** de desenvolupament en equip
- Mantingui l'essència del projecte: gestió i descobriment d'esdeveniments geolocalitzats

S'han eliminat funcionalitats complexes (mapa de calor, rols avançats, temps real) i se substitueixen per alternatives més senzilles que compleixen els objectius pedagògics.

MEMBRES DE L'EQUIP

- **Luka Francos Morales**
- **Martí Jiménez Lliteres**
- **Mateo López Lorenz**

⚠ ADVERTIMENT: Perquè el treball sigui considerat vàlid, **és obligatori que hi hagi commits repartits de forma equitativa** entre tots els membres de l'equip. Si no es detecta una activitat clara de tots els components al repositori de Git, el projecte no es considerarà apte per a la seva avaluació.

OBJECTIUS D'APRENENTATGE

DWES (Desenvolupament Web Servidor)

- **OA-DWES-1:** Desenvolupar una API REST completa amb almenys 8 endpoints funcionals seguint principis RESTful
- **OA-DWES-2:** Implementar un sistema d'autenticació amb JWT (registre, login, protecció d'endpoints)
- **OA-DWES-3:** Dissenyar i implementar un model de dades relacional amb almenys 3 entitats relacionades (1:N i N:M)
- **OA-DWES-4:** Aplicar validacions server-side en formularis i operacions CRUD
- **OA-DWES-5:** Gestionar la persistència de dades utilitzant JPA/Hibernate o equivalent (Spring Data, Sequelize, etc.)
- **OA-DWES-6:** Documentar endpoints API amb especificacions clares (mètode HTTP, ruta, payload, resposta)

DWEC (Desenvolupament Web Client)

- **OA-DWEC-1:** Desenvolupar una Single Page Application (SPA) amb almenys 5 vistes/components diferenciatos
 - **OA-DWEC-2:** Consumir l'API REST desenvolupada mitjançant peticions asíncrones (fetch/axios/HttpClient)
 - **OA-DWEC-3:** Implementar gestió d'estat centralitzada (Vuex, Redux, Services amb RxJS, Context API, etc.)
 - **OA-DWEC-4:** Validar formularis client-side abans d'enviar dades al servidor
 - **OA-DWEC-5:** Dissenyar interfícies responsives aplicant frameworks CSS (Bootstrap, Tailwind, Material, etc.)
 - **OA-DWEC-6:** Integrar una biblioteca de mapes (Leaflet, Google Maps) per visualitzar esdeveniments geolocalitzats
-

FUNCIONALITATS REQUERIDES (MVP)

1. Gestió d'Usuaris (Prioritat: ALTA)

Backend (DWES):

- **POST /api/auth/register:** Registre d'usuaris amb email, password, nom
 - Validació: email únic, password mínim 6 caràcters
 - Encriptació de password (bcrypt o similar)
- **POST /api/auth/login:** Login que retorna JWT token

- Validació de credencials
- Generació de token amb expiració (24h recomanat)
- **GET /api/users/me:** Obtenir perfil de l'usuari autenticat
 - Endpoint protegit (requereix token JWT)

Models de dades:

users

id: BIGINT PRIMARY KEY AUTO_INCREMENT

email: VARCHAR(255) UNIQUE NOT NULL

password: VARCHAR(255) NOT NULL

name: VARCHAR(100) NOT NULL

created_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Frontend (DWEC):

- **Component Registre:** Formulari amb camps email, password, confirmació password, nom
 - Validació: emails vàlids, passwords coincidents, camps obligatoris
- **Component Login:** Formulari email + password
 - Emmagatzematge del token en localStorage/sessionStorage
 - Redirecció a pàgina principal després de login
- **Guard/Middleware:** Protecció de rutes privades (redirigir a login si no hi ha token)
- **Component Navbar:** Mostrar nom d'usuari quan està logat, botó logout

Criteris d'Acceptació:

- Un usuari pot registrar-se amb email vàlid i rep confirmació visual
- Un usuari pot fer login i el sistema emmagatzema el token
- Les rutes privades només són accessibles amb token vàlid
- El password s'emmagatzema encriptat a la base de dades (verifiable)
- El logout elimina el token i redirigeix a login

2. Gestió d'Esdeveniments (CRUD) (Prioritat: ALTA)

Backend (DWES):

- **GET /api/events:** Llistar tots els esdeveniments (públic)
 - Retornar dades ordenades per data d'inici (pròxims primer)
 - Incloure informació de l'usuari creador
- **GET /api/events/{id}:** Obtenir detalls d'un esdeveniment (públic)
- **POST /api/events:** Crear esdeveniment (protegit)
 - Validacions: nom obligatori, data inici < data fi, coordenades vàlides
 - Associar esdeveniment amb usuari autenticat

- **PUT /api/events/{id}**: Editar esdeveniment (protegit, només creador)
- **DELETE /api/events/{id}**: Eliminar esdeveniment (protegit, només creador)

Models de dades:

events

id: BIGINT PRIMARY KEY AUTO_INCREMENT

name: VARCHAR(255) NOT NULL

description: TEXT

category: VARCHAR(50) (música, esport, cultura, altres)

start_date: DATETIME NOT NULL

end_date: DATETIME NOT NULL

latitude: DECIMAL(10,8) NOT NULL

longitude: DECIMAL(11,8) NOT NULL

address: VARCHAR(255)

user_id: BIGINT FOREIGN KEY -> users(id) ON DELETE CASCADE

created_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Frontend (DWEC):

- **Component Llista Esdeveniments**: Mostra esdeveniments en targetes/llista
 - Mostrar: nom, categoria, data, adreça
 - Botó "Veure detalls" per cada esdeveniment
- **Component Details Esdeveniment**: Visualització completa d'1 esdeveniment
 - Mostrar mapa amb pin a la ubicació
 - Si l'usuari és el creador: botons Editar/Eliminar
- **Component Formulari Esdeveniment**: Per crear/editar
 - Camps: nom, descripció, categoria (select), dates (inputs type="datetime-local"), coordenades, adreça
 - Validació: tots els camps obligatoris, data fi posterior a data inici
 - **Bonus**: Selector de coordenades clicant en mapa
- **Servei HTTP**: Mètodes per fer totes les peticions CRUD

Criteris d'Acceptació:

- Es mostren tots els esdeveniments ordenats cronològicament
- Un usuari logat pot crear un esdeveniment i aquest es persisteix a BD
- Un usuari només pot editar/eliminar els seus propis esdeveniments
- Les dates són validades tant al client com al servidor

- Les coordenades es visualitzen correctament en un mapa
-

3. Visualització en Mapa Interactiu (Prioritat: ALTA)

Backend (DWES):

- Reutilitzar **GET /api/events** amb tots els esdeveniments
- Opcional: **GET /api/events?bounds=latMin,latMax,lngMin,lngMax** per filtrar per àrea visible del mapa

Frontend (DWEC):

- **Component Mapa:** Integració de Leaflet o Google Maps
 - Mostrar un pin/marker per cada esdeveniment
 - Popup al clicar pin amb: nom esdeveniment, categoria, data
 - Link "Veure més" que redirigeix a detalls de l'esdeveniment
- **Centrat inicial:** Coordenades de Mallorca (39.695°N, 3.018°E) amb zoom adequat
- **Responsive:** El mapa s'adapta a diferents mides de pantalla

Criteris d'Acceptació:

- El mapa es carrega amb pins de tots els esdeveniments actius
 - Els pins són clicables i mostren informació bàsica
 - La navegació pel mapa és fluida (zoom, desplaçament)
 - El mapa és responsive i es veu correctament en mòbil
-

4. Sistema de Favorits ("M'agrada") (Prioritat: MITJANA)

Backend (DWES):

- **POST /api/events/{id}/like:** Afegir esdeveniment a favorits (protegit)
 - Validació: usuari no pot afegir 2 vegades el mateix esdeveniment
- **DELETE /api/events/{id}/like:** Eliminar de favorits (protegit)
- **GET /api/users/me/likes:** Obtenir esdeveniments favorits de l'usuari autenticat

Models de dades:

user_likes

id: BIGINT PRIMARY KEY AUTO_INCREMENT

user_id: BIGINT NOT NULL

event_id: BIGINT NOT NULL

created_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP

FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE

FOREIGN KEY (event_id) REFERENCES events(id) ON DELETE CASCADE

UNIQUE INDEX idx_user_event (user_id, event_id)

Frontend (DWEC):

- **Component Details Esdeveniment:** Botó "❤️ M'agrada" / "💔 Ja no m'agrada"
 - Canvia d'estat visual segons si l'usuari ja l'ha afegit
- **Vista Els Meus Favorits:** Llista d'esdeveniments marcats com a favorits
 - Reutilitza component de llista esdeveniments
- **Actualització d'estat:** Al clicar el botó, actualitza l'estat local i fa petició

Criteris d'Acceptació:

- Un usuari logat pot marcar/desmarcar esdeveniments com a favorits
- El botó mostra visualment si ja està afegit a favorits
- La llista de favorits només mostra esdeveniments marcats per l'usuari
- Les dades persisteixen a la base de dades (relació N:M)
- No es poden afegir duplicats (constraint de BD funciona)

5. Filtratge d'Esdeveniments (Prioritat: MITJANA)

Backend (DWES):

- **GET /api/events?category={categoria}:** Filtrar per categoria
- **GET /api/events?date={YYYY-MM-DD}:** Filtrar esdeveniments que tinguin lloc en una data específica
- **Opcional GET /api/events?search={text}:** Cerca per nom o descripció

Frontend (DWEC):

- **Component Filtres:** Barra/formulari amb:
 - Select de categoria (Totes, Música, Esport, Cultura, Altres)
 - Input de data (type="date")
 - Input de cerca (opcional)
 - Botó "Filtrar" o aplicació automàtica (onChange)
- **Integració amb Llista i mapa:** Els filtres actualitzen tant la llista com els pins del mapa
- **Reset:** Botó per netejar tots els filtres

Criteris d'Acceptació:

- Els filtres funcionen correctament i retornen resultats esperats
- Es poden combinar múltiples filtres (categoria + data)
- La llista i el mapa s'actualitzen quan s'apliquen filtres

- Hi ha feedback visual quand no hi ha resultats
-

6. Disseny Responsive i UX (Prioritat: MITJANA)

Frontend (DWEC):

- **Framework CSS:** Utilitzar Bootstrap, Tailwind, Material UI o similar
- **Navbar responsive:** Hamburger menu en mòbil
- **Targetes d'esdeveniments:** Disseny atractiu amb imatge placeholder/categoría
- **Formularis:** Estils coherents, missatges d'error clars
- **Estats de càrrega:** Spinners/loaders durant peticions a l'API
- **Toasts/Alerts:** Notificacions d'èxit/error en operacions (crear esdeveniment, login, etc.)

Criteris d'Acceptació:

- L'aplicació és usable en dispositius de 320px a 1920px d'ample
 - Els formularis mostren errors de validació de manera clara
 - Hi ha feedback visual durant operacions asíncrones
 - La paleta de colors i tipografia són coherents
 - La navegació és intuïtiva sense necessitat de documentació
-

✗ FORA D'ABAST

Aquestes funcionalitats del projecte original **NO** són requerides per l'avaluació:

1. **Mapa de calor:** Requereix processament de dades en temps real.
 2. **Rol d'Administrador i Promotora:** Només usuaris bàsics que poden crear els seus propis esdeveniments.
 3. **Sistema de check-in amb geolocalització:** Escurçat a sistema de favorits.
 4. **Upload d'imatges:** S'utilitzen placeholders o icones de categoria.
 5. **Sistema de notificacions:** No requerit.
 6. **Tests automatitzats:** Extra opcional (bonus).
-

💻 STACK TECNOLÒGIC

Capa	Tecnologia Recomanada	Alternatives Vàlides	Obligatori
Frontend Framework	Angular + TypeScript	React, Vue 3, Svelte	<input checked="" type="checkbox"/> Sí
Estils	Bootstrap 5	Tailwind, Material UI, Bulma	<input checked="" type="checkbox"/> Sí

Capa	Tecnologia Recomanada	Alternatives Vàlides	Obligatori
Backend Framework	Spring Boot + Java	Node.js + Express, Django, FastAPI	<input checked="" type="checkbox"/> Sí
Base de Dades	PostgreSQL	MySQL, MariaDB, SQLite	<input checked="" type="checkbox"/> Sí
Autenticació	JWT	Sessions + cookies	<input checked="" type="checkbox"/> Sí
Mapes	Leaflet	Google Maps API, Mapbox	<input checked="" type="checkbox"/> Sí

MODEL DE DADES

Esquema Relacional Mínim

users (id, email, password, name, created_at)

events (id, name, description, category, start_date, end_date, latitude, longitude, address, user_id, created_at)

user_likes (id, user_id, event_id, created_at)

PLANIFICACIÓ TEMPORAL

Fase	Dies	Tasques Específiques
Setup	2	Repo, entorns backend/frontend, BD
Auth	4	Backend auth, Frontend auth components
Events	6	Backend CRUD, Frontend CRUD components
Mapa	2	Integració Leaflet, Pins, Popups
Favorites	2	Like system (back/front)
Polish	4	Filtres, UX/UI, Responsive, Docs

CRITERIS D'avaluació (Total: 100 punts)

- **DWES - Backend (45 pts):** API REST, Auth, Persistència, Validacions.
 - **DWEC - Frontend (45 pts):** SPA, Consum API, Estat, UX/UI, Mapes.
 - **General (10 pts):** Integració, Documentació, Git.
-

RESUM DE TASQUES OBLIGATÒRIES

Tasques específiques del projecte

- **Seguretat:** Implementar autenticació segura mitjançant JWT (registre i login).
- **Esdeveniments:** Desenvolupar el sistema complet de creació, llistat i gestió d'esdeveniments.
- **Registres:** Gestionar les inscripcions/likes dels usuaris a esdeveniments específics.
- **Mapa:** Integrar un mapa interactiu per a la localització visual dels esdeveniments.

Punts comuns obligatoris (tots els projectes)

- **Control de versions:** Més de 20 commits amb missatges descriptius.
 - **Base de dades:** Scripts SQL de creació d'esquema i dades inicials (seed/data).
 - **Documentació:** README professional amb instruccions d'instal·lació i credencials de prova.
 - **API:** Documentació dels endpoints de l'API (Postman, Swagger o Markdown).
-

ENTREGABLES

1. **Repositori Git Públic** amb històric de commits.
 2. **README.md Complet** amb instal·lació, tecnologies i credencials.
 3. **Scripts de Base de Dades** (schema.sql).
 4. **Documentació d'API** (API.md).
-

INSTRUCCIONS D'ENTREGA I DEFENSA

L'entrega es realitzarà exclusivament a través de la tasca corresponent a **Google Classroom**. Per considerar el projecte com a entregat, cal aportar:

1. Enllaç al Repositori

- URL del repositori (GitHub/GitLab) amb accés públic o compartit amb el tutor.
- El repositori ha de contenir el fitxer **README .md** amb tota la documentació requerida.

2. Vídeo de Defensa Tècnica (Obligatori)

Dada la prohibició de l'ús d'IAs generatives per a la creació de codi, cada alumne (o equip) ha d'adjuntar un **vídeo explicatiu** que serveixi com a defensa del projecte i verificació d'autoria.

- **Durada:** 3 - 5 minuts.
- **Format:** Enllaç a una plataforma de vídeo (Loom, YouTube ocult, Google Drive, etc.).
- **Contingut mínim del vídeo:**
 - **Identificació:** Una breu introducció de l'alumne/a.
 - **Demo del MVP:** Mostra el flux principal de l'aplicació (login -> funcionalitat core -> resultat).
 - **Explicació tècnica (Backend):** Obre l'editor de codi i explica com has implementat una part crítica de la lògica.
 - **Explicació tècnica (Frontend):** Explica com es gestiona l'estat en un component i com es realitza la comunicació asíncrona amb el servidor.
- **Objectiu:** L'alumne ha de demostrar que **entén i sap explicar cada línia de codi** que ha entregat.

[!IMPORTANT] **Verificació d'Autoria i Coneixement:** En l'exercici de les atribucions acadèmiques que li corresponen, el professor té la potestat de declarar aquest projecte com a **no apte** en cas d'evidències de manca d'autoria o si es manifesta una manca de coneixement consolidat del codi lliurat. Atès que aquest projecte representa la prova final de consolidació de les **principals competències dels mòduls implicats (DWES i DWEC)**, la seva invalidació comportarà la impossibilitat de superar l'avaluació contínua del curs. En aquest cas, per superar el mòdul, l'alumne haurà de demostrar l'assoliment dels **Resultats d'Aprendentatge (RA)** prescrits pel currículum oficial de la Formació Professional en l'examen teòric i pràctic de la 1a convocatòria oficial.

✨ EXTRES OPCIONALS (Bonus: +10%)

- Deploy en producció (+3 pts)
- Tests automatitzats (+3 pts)
- Swagger docs (+2 pts)
- Dark Mode (+1 pt)

❓ FAQ (Preguntes Freqüents)

Consulta el detall al document original si tens dubtes sobre l'stack o les funcionalitats.

CONSELLS FINALS

- Prioritza funcionalitats ALTA.
 - Fes commits diaris.
 - No deixis el README pel darrer dia.
-

Data de publicació: 2 de febrer de 2026

Versió del document: 3.1

Autor: Xavier Sastre i Flexas