



Comprehensive Development Plan: Internship Program Management Application

Overall Objective: To develop the Minimum Viable Product (MVP) of a web application for managing internship programs, focusing on the Dynamic Activity Checklist, documents, and monitoring.

Frontend Technology Stack: React.js

Team: 4 Interns

TOTAL DURATION: 3 WEEKS (15 Working Days)

Introduction and Problem Statement

The current management of internship programs, particularly those supported by European initiatives like Erasmus+, relies heavily on manual processes, disparate documents (PDFs, CVs, evaluation sheets), and fragmented communication channels between the host organization (Yudaya S.L.), the sending partner (Mechatronics Technical School N° 1, Poland), and the interns.

Problem: This manual system introduces risk of error, lack of real-time transparency regarding intern progress (tasks and commitment), and delays in final certification (Europass Mobility).

Solution: This project aims to build a centralized web application to **digitize and automate** the entire internship lifecycle. The application will serve as a single source of truth for documentation, schedules, and real-time activity tracking via a dynamic checklist, ensuring clear monitoring and timely program closure.

I. PROJECT OVERVIEW TIMELINE (3 Weeks)

The project is divided into three concurrent and high-intensity phases.

Module	Week 1	Week 2	Week 3
Backend Core (Intern 1)	Models and Security (CRUD APIs)	Weekly Monitoring APIs	Integration Tests with Frontend
Backend Logic (Intern 2)	Checklist and Document Models	Checklist APIs and Schedule Module	Optimization and API Documentation

Frontend UI (Intern 3)	Wireframes, React Setup, and Authentication Views	HR Dashboard and Intern Profile	KEY Operational Checklist View
QA/Support (Intern 4)	DevOps/Git Setup, Test Cases	Continuous API Testing (B1/B2)	Compliance Testing (Functions) and Deployment

II. FUNCTIONAL DEVELOPMENT OF THE MVP (EXTENDED DETAIL)

The project must meet the needs of three main user roles:

A. Functionalities per User Role

Role	Key Views	Required Functionalities
PARTICIPANT (Intern)	Login, My Profile, Activity Checklist , Document Upload, Schedule Module.	* Task Tracking: View list of assigned tasks (Section 2 of the Agreement) and mark them as In Progress or Completed . * Documentation: Upload CV and Final Internship Report. * Communication: View contact details for supervisors. * Schedule: Register daily check-in and check-out times.
SUPERVISOR (Company/Sending Tutor)	Login, List of Assigned Participants, Intern Profile.	* Weekly Monitoring: Conduct the "Evaluation of the participant's internship review at least once a week" (SEGUIMIENTO_SEMANAL table). * View Checklist: Monitor the intern's task progress in real-time. *

		<p>Documents: Download CVs and Final Reports. *</p> <p>Final Approval: Mark the internship program as 'Completed and Approved' using the tutor_final_approval field once the internship is finished.</p>
HR / Coordination	Login, General Dashboard, Reports Module, Program Administration.	<p>* Master Management: Create new PROGRAMA_PRACTICAS, add ACTIVIDAD (tasks) to those programs, and assign Supervisors. *</p> <p>Reports: View the general status of all interns (e.g., <i>Internships ending this month</i>). * Final Documents: Issue and register the Internship Certificate (Europass Mobility).</p>

B. Detailed Checklist Module Flow (KEY)

This is the central module that must be fully operational by the end of Week 3.

1. Backend (Intern 2)

1. **Initial Load:** The complete list of the 23 "Functions to be performed" (Section 2) is loaded into the ACTIVIDAD table.
2. **Automatic Assignment:** When a new PARTICIPANTE is created and assigned a PROGRAMA_PRACTICAS, the Backend must automatically generate 23 records in the PARTICIPANTE_ACTIVIDAD table for that intern, with the status field initialized to "Pending".
3. **Read API:** Create a secure endpoint (GET /api/intern/{id}/checklist) that returns the task name, description, and its status.
4. **Update API:** Create an endpoint (POST /api/intern/checklist/{id_relation}/update) that allows the intern to update the status (e.g., from "Pending" to "Completed").

2. Frontend (Intern 3) - Using React.js

1. **Setup:** Use a modern React component structure (Functional Components and Hooks).
2. **View:** A single clean and responsive view for the intern with filters by activity type (Technical, Safety, Soft Skills). Use useState to manage task status.
3. **Interaction:** Each task must have an associated checkbox that, when checked, calls the **Update API** (2.B.1.4) to change the status to "Completed".
4. **Tutor Visualization:** The Tutor accesses a "read-only" view of the same Checklist to verify progress.

III. TASK AND ROLE DISTRIBUTION (WEEKLY ADJUSTMENT)

The distribution remains the same, but tasks must be completed in one-week blocks.

Week 1: Setup and Structure (Database and Authentication)

Role	Priority Tasks (100% Completion)	Key Intern
Backend Core	Master Models (Participant, Supervisor, Program). Implementation of Authentication and basic CRUD APIs.	Intern 1
Backend Logic	Task Models (ACTIVITY, PARTICIPANTE_ACTIVIDAD) and Document Models. Initial implementation of the secure file upload system.	Intern 2
Frontend UI	React project setup (Vite/CRA), Wireframe Design, and Development of Login/Registration Views (React Components).	Intern 3
QA/Soporte	Git Configuration and creation of Test Cases based on the 23 "Functions	Intern 4

	to be performed" from the agreement.	
--	--------------------------------------	--

Week 2: Business Logic and View Connection

Role	Priority Tasks (100% Completion)	Key Intern
Backend Core	Weekly Monitoring APIs (SEGUIMIENTO_SEMANAL). Security and authorization adjustments.	Intern 1
Backend Logic	Checklist APIs (GET/UPDATE) and Schedule Module. Implementation of Automatic Task Assignment.	Intern 2
Frontend UI	Development of React Components for the HR Dashboard and the Intern Profile view. Initial API connection (Data Fetching from Intern 1).	Intern 3
QA/Soporte	Continuous API Testing (Intern 1 and 2). Ensure that the Checklist logic functions in the backend.	Intern 4

Week 3: Checklist, Final Tests, and Deployment

Role	Priority Tasks (100% Completion)	Key Intern
Backend Core	Frontend Integration Tests. Code optimization.	Intern 1

Backend Logic	Final API Documentation. Support for QA tests.	Intern 2
Frontend UI	Implementation and testing of the Checklist View (KEY) using React State/Hooks and the Document Interface.	Intern 3
QA/Soporte	Compliance Testing (UAT). Final Documentation. Deployment to Production.	Intern 4

IV. DATABASE DESIGN (Unchanged)

The database schema remains the same, as it is the minimum logical structure to support the required MVP functionalities.

Master Tables (Core Entities)

Table	Key Fields (Attributes)	Type	Notes
PARTICIPANTE	id (PK), full_name, training_sector, sending_partner, program_id (FK), tutor_final_approval	Various / BOOL	Intern data and final tutor sign-off.
SUPERVISOR	id (PK), full_name, email, role_type (Host/Sending)	Various	Tutor data.
PROGRAMA_PRACTICAS	id (PK), program_name, start_date, end_date, host_org_id (FK)	Various	Framework agreement data.

ACTIVIDAD	id (PK), program_id (FK), function_descriptio n, activity_type (Technical/Safety/S oft)	Various	Tasks from Section 2 of the agreement (the basis for the checklist).
DOCUMENTO	id (PK), participant_id (FK), doc_type (CV/Report/etc.), file_path, upload_date	Various	Secure file storage.

Relationship Tables (Dynamic Checklist and Monitoring)

Table	Key Fields (Attributes)	Type	Notes
PARTICIPANTE_ACTIVIDAD	id (PK), participant_id (FK), activity_id (FK)	Various	KEY for the Checklist.
	status	ENUM	(Pending, In Progress, Completed).
	completion_date	DATE	
SEGUIMIENTO_SEMANAL	id (PK), participant_id (FK), tutor_id (FK)	Various	Weekly tutor evaluation (Section 5).
	week_num	INT	
	tutor_evaluation	TEXT	

V. APPLICATION WORKFLOW (FLUJO DE TRABAJO DE LA APLICACIÓN)

This section details the complete lifecycle of an internship within the application,

demonstrating the interaction between the three user roles.

1. HR/Coordinator (Start of the Program)

- **Setup:** HR creates a new PROGRAMA_PRACTICAS and defines the master list of ACTIVIDAD (tasks) for that program (Backend Core/Logic: Interns 1 & 2).
- **Assignment:** HR registers a new PARTICIPANTE (Intern) and links them to the Program and their SUPERVISOR (Tutor).
- **Automatic Task Generation (KEY STEP):** The Backend automatically generates the PARTICIPANTE_ACTIVIDAD records (the checklist) for the new intern.

2. Participant (Daily/Weekly Activity)

- **Login:** The intern logs in and accesses their **Activity Checklist** (Frontend: Intern 3).
- **Task Tracking:** As tasks from the "Functions to be performed" list are completed, the intern marks the respective items on the checklist as '**Completed**', sending an update to the Backend (Intern 2 API).
- **Scheduling:** The intern registers their daily check-in and check-out times in the Schedule Module.
- **Documentation:** The intern uploads their CV and, later, their Final Report.

3. Supervisor (Weekly Monitoring)

- **Monitoring:** The Tutor logs in and views the Intern's profile, specifically the **Activity Checklist**, to see the status updated in real-time.
- **Weekly Evaluation:** The Tutor fills out the weekly SEGUIMIENTO_SEMANAL evaluation, rating the intern's progress and providing feedback (Backend Core: Intern 1 API).
- **Documentation Review:** The Tutor downloads and reviews the intern's Final Report.

4. Program Closure (Final Approval)

- **Final Vetting:** Once all tasks are marked as 'Completed' and the Tutor has reviewed the final documents and attendance records.
- **Tutor Sign-Off:** The Supervisor updates the PARTICIPANTE record by setting the **tutor_final_approval** field to **True**. This action finalizes the internship in the system.
- **HR Action:** HR uses the reports module to issue the final Certificate of Practices (Europass Mobility) based on the completed checklist and final approval.

VI. DATABASE DESIGN (Unchanged)

The database schema remains the same, as it is the minimum logical structure to support the required MVP functionalities.

Master Tables (Core Entities)

Table	Key Fields	Type	Notes

	(Attributes)		
PARTICIPANTE	id (PK), full_name, training_sector, sending_partner, program_id (FK), tutor_final_approval	Various / BOOL	Intern data and final tutor sign-off.
SUPERVISOR	id (PK), full_name, email, role_type (Host/Sending)	Various	Tutor data.
PROGRAMA_PRACTICAS	id (PK), program_name, start_date, end_date, host_org_id (FK)	Various	Framework agreement data.
ACTIVIDAD	id (PK), program_id (FK), function_description, activity_type (Technical/Safety/Soft)	Various	Tasks from Section 2 of the agreement (the basis for the checklist).
DOCUMENTO	id (PK), participant_id (FK), doc_type (CV/Report/etc.), file_path, upload_date	Various	Secure file storage.

Relationship Tables (Dynamic Checklist and Monitoring)

Table	Key Fields (Attributes)	Type	Notes
PARTICIPANTE_ACTIVIDAD	id (PK), participant_id (FK), activity_id (FK)	Various	KEY for the Checklist.

	status	ENUM	(Pending, In Progress, Completed).
	completion_date	DATE	
SEGUIMIENTO_SEMANAL	id (PK), participant_id (FK), tutor_id (FK)	Various	Weekly tutor evaluation (Section 5).
	week_num	INT	
	tutor_evaluation	TEXT	