

 SceneFilm	Equipo: Alejandro Bolaños Sánchez Cristina Carrasco Guillén	Tutor: María Mercedes Martínez Fragoso
--	---	---

# PROYECTO INTERMODULAR

Nombre del proyecto: SceneFilm

Ciclo Formativo: Diseño de Aplicaciones Multiplataforma (DAM)

Grupo 2ºA

Fecha: 3/11/2025

Versión: 1

Enlaces:

- Repositorio: <https://github.com/AlejandroBs-06/ProyectoIntermodular>
- Figma:
- Tablero: <https://trello.com/b/JjWZRkV2/pidam2a-tecnologías-scenefilm>

## 2. Resumen ejecutivo

El problema que hemos detectado y que ha provocado que hayamos creado esta aplicación es que actualmente los productores audiovisuales, cuando tienen intención de desarrollar un nuevo proyecto tienen que utilizar múltiples ventanas en sus dispositivos.

Esas ventanas contienen la información de los actores, el número de escenas o los guiones lo que deriva a una mala organización. También hay casos de productores que utilizan aplicaciones de pago para tratar de tener mejor organizada toda la información dicha anteriormente y esta aplicación contará con una descarga gratuita para los usuarios y tratará de solucionar dichos problemas.

La solución que hemos propuesto para resolver este problema consiste en crear una aplicación gratuita, que permita a los productores audiovisuales tener un acceso instantáneo a su plan de rodaje y que sea funcional para dispositivos móviles y portátiles

Las producciones en Extremadura suelen venir de la mano de productoras pequeñas que no cuentan con mucho presupuesto para invertir en mejoras tecnológicas accesibles que les ayuden a acelerar el trabajo durante la fase de preproducción. Organizar un rodaje es una tarea muy compleja que en la mayor parte del tiempo se requiere mucha información y documentación que debe estar accesible en tiempo y forma.

El objetivo del producto es ofrecer de manera gratuita una solución a las múltiples tareas que realizan las productoras audiovisuales ante el desafío de organizar una producción audiovisual, a través de diferentes software, o de lo contrario tener que pagar una suscripción a una aplicación que resuelva sus necesidades.

El proyecto cubrirá la digitalización en la fase de postproducción, incluyendo el nombre del proyecto, permitiendo adjuntar documentos de gestión y administración, el guión literario, guión técnico y storyboard además de añadir los contactos de participantes en el rodaje y el desglose completo del plan de rodaje con un calendario.

No se incluirán herramientas como el diseño de la puesta en escena, visionado de planos y escenas, edición de video o invitación a proyectos grupales (posible para una versión súper).

### 3. Justificación

La aplicación SceneFilm ofrece beneficios de uso brindando al usuario la facilidad para acceder a todos los elementos que componen un plan de rodaje, de tal forma que tenga una mejor organización de su proyecto. Podrá controlar fechas de rodajes y eventos, acceder a una base de datos que contendrá el contacto de cada participante de la producción y comprobar guiones y storyboards.

Nuestra aplicación ofrece también algunos beneficios técnicos con la implementación de un calendario que muestre los eventos que se tienen que cumplir en un día y que se puedan marcar como completados, y se enviará alertas los días y las horas programadas por el usuario.

Un punto importante que pensamos en el momento de desarrollar esta app fue el beneficio educativo que tiene para la sociedad. Esta es la razón por la que consideramos esta aplicación como una herramienta que fomenta la organización en los estudiantes de comunicación audiovisual, cine, producción o cualquier formación relacionada con la creación de contenido audiovisual, debido a la facilidad de acceso para estos estudiantes a una aplicación gratuita que ofrece información de todos los puntos necesarios para desarrollar un plan de rodaje de forma clara y precisa.

### 3.2. Integración con la industria extremeña (sectores, clústeres, casos de uso)

Los sectores en los que está basada la aplicación son los siguientes:

- Sector audiovisual: Debido a que facilita la creación de planes de rodajes
- Educación: Debido a que esta aplicación puede ser utilizada en centros universitarios por estudiantes de comunicación audiovisual.

Los grupos de empresas que puedan solicitar esta aplicación son las siguientes:

- Universidad de Extremadura(UEX): Esta aplicación puede ser muy útil en este centro para la carrera de comunicación audiovisual ya que está basada en todos los elementos que contiene un plan de rodaje.
- Extremadura audiovisual: Esta aplicación puede ser muy solicitada por esta empresa debido a que su objetivo es impulsar el sector audiovisual a lo más alto y la aplicación que trata de coordinar y controlar proyectos audiovisuales les puede ser de mucha ayuda

Los casos de uso que puede tener esta aplicación son los siguientes:

- Crear proyecto: Consiste en que el usuario cuando quiera crear un proyecto tenga que introducir todos los campos que sean necesarios para el desarrollo de este. Deberá introducir el nombre, adjuntar los guiones, los números de escenas o planos y poner información sobre toda la gente que colabora: Actores, directores o cámaras.
- Anotar eventos: Consiste en que el usuario pueda agregar un evento en el calendario relacionado con su plan de rodaje, por ejemplo: El día 2 de noviembre se tienen que grabar 5 escenas, pues lo que se hará es viajar al calendario y poner el día 2 de noviembre:'Grabación de escenas'
- Editar información: Consiste en que el usuario pueda editar la información que ha puesto previamente para desarrollar un proyecto. Lo que tendrá que hacer es pinchar sobre su proyecto creado y acceder a la ventana que necesite cambiar, por ejemplo: N°Escenas. Ya desde ahí podrá cambiar el número puesto previamente al nuevo.

### 3.3. Análisis de productos similares (benchmark breve: 3–5 referencias)

La idea de realizar este proyecto viene dada por la falta de oferta donde, si bien podemos encontrar numerosas aplicaciones destinadas a productos audiovisuales, no se centran en el departamento de producción que gestiona el plan de rodaje.

Hemos investigado algunas de estas aplicaciones que pueden acercarse más a la idea de producto que hemos diseñado:

La primera aplicación que hemos analizado es FilmTouch que funciona como un gestor de contactos profesional, donde los usuarios pueden buscar rápidamente los profesionales con los que ha trabajado en un proyecto audiovisual y ponerse en contacto con ellos. Esta aplicación permite leer documentos en PDF y extraer automáticamente nombres, cargos y datos de contacto de las personas que aparecen.

Yamdu es una aplicación que a nuestro parecer se asemeja más a nuestra idea de proyecto, la cual permite también la importación de documentos, gestión de equipos de trabajo, además dispone de un calendario. Sin embargo su curva de aprendizaje puede ser algo elevada para productoras pequeñas que solo requieren funciones básicas y contiene algunas funcionalidades solo accesibles si dispones de una versión Pro. Además es una aplicación basada en la nube por lo que la información que se importe puede estar menos segura.

Por último investigamos la aplicación Celtx que al igual que la anterior también está basada en la nube, por lo que algunos usuarios se quejan sobre problemas de estabilidad y sincronización. Esta aplicación cubre desde la edición del guión hasta el plan de rodaje, sin embargo, como punto negativo vemos que la página está en inglés y no se puede cambiar.

	Enfoque y objetivo	Funcionalidades principales	Ventajas destacadas	Limitaciones a tener en cuenta
FilmScene	Agenda, preproducción	Base de datos, documentación, breakdowns, calendario, comunicación	Cubre necesidades básica de producción	Limitada para producciones especializadas
FilmTouch	Gestión agenda de contactos	Importa PDFs y extrae nombres y contactos. Base de datos personal.	Acceso rápido a contactos	No se centra en un equipo de producción

Yamdu	Gestión de equipos, fases de rodaje y datos producción	Importa guiones, crea breakdowns, calendario, call sheets, comunicación de equipo y versión móvil.	Centralizar datos y coordinar equipos	Curva de aprendizaje elevada y costes altos
Celtx	Desarrollo de guiones y preproducción	Editor de guiones, storyboard, breakdowns, presupuestos, colaboración en la nube	Abarca desde la preproducción y parte de la producción	Algunas funciones requieren suscripción o planes premium

### 3.4. Participación en ODS (mapa ODS ↔ funcionalidad/impacto)

El uso de esta aplicación puede contribuir a algunos ODS que la Comunidad Autónoma de Extremadura tiene previstos alcanzar:

**ODS4 - Educación de calidad:** Esta aplicación está destinada para cualquier producción audiovisual, y su uso gratuito la hace ideal para estudiantes de cualquier formación relacionada con la creación de contenido audiovisual, por lo que fomenta una educación de calidad.

**ODS8 - Trabajo decente y crecimiento económico:** El objetivo es crear una aplicación que cubra las necesidades de las productoras audiovisuales en la etapa de preproducción, que sea accesible para aquellas productoras de la comunidad con pocos trabajadores y pocos recursos, ayudando a crear un trabajo decente y fomentando su crecimiento económico.

**ODS9 - Industria, innovación e infraestructuras:** Impulsamos la infraestructura tecnológica en la industria audiovisual extremeña, favoreciendo la innovación en la organización y planificación de rodajes audiovisuales y así modernizar el sector en la región, favoreciendo a pequeñas productoras de la región, con pocos beneficios para invertir en nuevos recursos, una organización virtual que ayude a reducir el tiempo de trabajo y sus costes.

## 4. Historias de usuario

- 4.1. Convenciones de numeración y formato (p. ej., HU-001, HU-002...)
- 4.2. Ejemplos de historias (3–5) con Criterios de Aceptación (Given/When/Then)

- 4.3. Backlog priorizado (MoSCoW/Kano o similar) - Must/Should/Could → Imprescindible/Importante/Conveniente
- 4.4. Trazabilidad HU ↔ CA ↔ Pruebas ↔ Figma

HU-001: Registro o inicio de sesión

Como usuario quiero registrarme o iniciar sesión en la aplicación para que pueda acceder al programa.

Prioridad: Imprescindible

Criterios de aceptación:

- Dado que he pulsado el botón de registro en la portada.
- Cuando introduzco mis datos identificativos.
- Entonces el sistema crea una cuenta nueva y lleva al usuario a la página principal.

HU: Como usuario quiero registrarme o iniciar sesión en la aplicación para que pueda acceder al programa.

↓

CA: El usuario puede introducir sus datos identificativos para acceder por primera vez a la aplicación o introducir los datos más significativos para los que ya tienen cuenta, pero se deberá validar que son correctos.

↓

Pruebas: Validar que el usuario entra correctamente cuando inicia sesión y pone sus datos correctos o validar que aparece un error cuando escribo un dato incorrecto.

↓

Figma: Implementar la portada y página con el formulario para introducir los datos

HU-002: Crear nuevo proyecto

Como usuario quiero crear un proyecto en la página de inicio para guardar todo mi plan de rodaje

Prioridad: Imprescindible

Criterios de aceptación:

- Dado que he pulsado el botón de nuevo proyecto en la página principal .
- Cuando introduzco los datos que necesita el proyecto(Nombre,nº escenas,nº planos,etc).
- Entonces el sistema crea un proyecto nuevo y me lo muestra en la página principal.

HU: Como usuario quiero crear un proyecto en que quede asentado todo mi plan de rodaje

↓

CA: El usuario introduce los datos más significativos de un plan de rodaje y se asegura que los ha introducido todos

↓

Pruebas: Validar que el programa crea el proyecto cuando el usuario introduce los datos de un plan de rodaje(Título, NºEscenas, NºPlanos, etc) y validar que aparezca un mensaje de error cuando se deja un campo en blanco.

↓

Figma: Implementar el formulario que va a solicitar al usuario los datos que necesita para crear su plan de rodaje

#### HU-003: Editar información del proyecto

Como usuario quiero cambiar la información del proyecto para que se adapte a mis objetivos

Prioridad: Importante

Criterios de aceptación:

- Dado que he pulsado el botón de editar que aparece en un proyecto creado.
- Cuando edito el campo que quiero del proyecto, por ejemplo: El número de escenas y pulse guardar
- Entonces el sistema edita el proyecto y guarda los cambios.

HU: Como usuario quiero cambiar un campo del proyecto que no me convence

↓

CA: El usuario cambia la información de un campo concreto y se asegura que el sistema ha guardado los cambios

↓

Pruebas: Validar que el programa ha editado la información

↓

Figma: Implementar la página en la que van a aparecer los datos del proyecto cuando lo creamos la primera vez y el botón de guardar cambios.

#### HU-004: Añadir evento al calendario

Como usuario quiero añadir un evento al calendario para que tenga organizado gran parte del proyecto

Prioridad: Imprescindible

Criterios de aceptación:

- Dado que he pulsado en el botón de eventos
- Cuando introduzco el nombre del evento que tengo que realizar y el día que quiero hacerlo.
- Entonces el sistema crea el evento y me lo muestra junto con el día que he puesto

HU: Como usuario quiero agregar un evento al calendario para que quede organizado mi plan de rodaje

↓

CA: El sistema tiene que guardar el evento que ha introducido el usuario con los datos que ha introducido y mandar una notificación recordatorio para no se olvide de su realización

↓

Pruebas: Validar que el programa guarda el evento con la fecha marcada por el usuario

↓

Figma: Implementar el formulario que va a solicitar al usuario el nombre del evento, la fecha en la que quiere realizarlo y en qué momento quiere recibir una notificación de recordatorio, si un día, tres días o una semana antes

#### HU-005: Marcar evento como cumplido

Como usuario quiero marcar un evento del calendario como cumplido para no tener dudas de si lo he realizado o no

Prioridad: Importante

Criterios de aceptación:

- Dado que he accedido al evento que quiero
- Cuando pulso un botón que hay al lado del evento llamado 'marcar como completado'
- Entonces el sistema hace que ese evento se vea muy poco para no prestarle atención y se borrará cuando pasen días de completarlo

HU: Como usuario quiero marcar un evento como completado para no tener dudas de si está completado o no

↓

CA: El sistema, cuando el usuario pulse el botón que hay al lado del evento de marcar como completado, debe ser capaz de oscurecerlo para que pase desapercibido y a los días se borre

↓

Pruebas: Validar que el evento tenga un tick a su lado que marque que esté como completado y que este se haya oscurecido para que el usuario no le preste atención

↓

Figma: Implementar el evento con el nombre que le hayamos puesto y la fecha y a su lado va a tener un botón que cuando lo pulsamos se pondrá con un tick

#### HU-006: Recibir notificaciones

Como usuario quiero recibir notificaciones de que debo realizar un evento para que no se me olvide

Prioridad: Importante

Criterios de aceptación:

- Dado que cuando esté creando un evento
- Cuando esté introduciendo los datos, aparecerá un apartado en el que el programa me dirá cuando quiero recibir una notificación de recordatorio, puedo seleccionar que la quiero un día, tres días o una semana antes de la realización del mismo
- Entonces el sistema, cuando llega la fecha marcada, me mandará una notificación

HU: Como usuario quiero recibir notificaciones de recordatorios de eventos para que no corra el riesgo de olvidarlos

↓

CA: El sistema debe identificar que la fecha de notificación marcada por el usuario ha llegado y debe ser capaz de mandarle un recordatorio cuando la aplicación esté activa o inactiva

↓

Pruebas: Se debe validar que el sistema cumple con el horario que ha marcado el usuario, es decir, que si el evento está puesto para realizarse el día 2 de diciembre y está marcado que el sistema mande una notificación de recordatorio un día antes, se tiene que comprobar que el sistema ha cumplido con ese requisito.

↓

Figma: Implementaremos dentro del formulario en el que creamos el evento un menú con diferentes opciones en el que saldrán los momentos en los que el usuario quiere recibir la notificación.

## 5. Arquitectura

### 5.1. Diagrama (C1/C2: contexto y contenedores)

Usuario

| (Toque pantalla)



Aplicación Móvil (Kotlin / Android)

- Interfaz de usuario
- Módulo calendario y contactos
- Subida de archivos (adjuntos)
- Recepción de notificaciones
- Autenticación de usuario

| HTTPS + JSON



API REST (Spring Boot monolito)

- Lógica de negocio
- Gestión de planes de rodaje y eventos

- Integración con BD
- Envío de notificaciones push

SQL/JDBC

Base de datos MySQL

- Tabla: usuarios, eventos, contactos, docs
- Persistencia de datos

## 5.2. Decisiones de arquitectura (ADR) y numeración (p. ej., ADR-001, ADR-002)

5.2.1. ADR-001: Título (contexto, decisión, alternativas, consecuencias)

5.2.2. ADR-002: Título (... )

ADR-001: Elección de Arquitectura Cliente-Servidor con API REST.

Necesitamos un backend seguro, escalable y mantenible puesto que la aplicación necesita acceder a datos actualizados sobre los planes de rodaje, contactos, documentos y eventos, además de que algunas funciones como autenticación, sincronización, almacenamiento y notificaciones necesitan centralizarse.

Esta decisión viene determinada para adoptar una arquitectura cliente-servidor basada en una API REST que funciona como acceso a la aplicación móvil. La aplicación se comunica a través de HTTPS-JSON con el backend construido sobre Spring Boot.

Como alternativa tendríamos la arquitectura basada en GraphQL que permite que el cliente especifique exactamente qué datos necesita y reduce el número de llamadas al servidor, pero incrementa la complejidad del backend y la curva de aprendizaje del equipo es mayor.

Con ello centralizamos la lógica de negocio, obtenemos una mayor facilidad en el mantenimiento y versionado de la API, y creamos una independencia entre el cliente y el servidor. Como contra requiere conectividad para la mayoría de las funcionalidades y necesita gestionar compatibilidad de versiones API - App.

ADR-002: Aplicación Nativa Android con Kotlin.

Los usuarios se conectarán a la aplicación desde dispositivos Android para acceder a funciones nativas de calendario, almacenamiento, notificaciones push y manejo de adjuntos.

Necesitamos implementar la aplicación móvil como un cliente nativo Android que desarrollaremos en kotlin.

En lugar de Kotlin sería una alternativa Flutter que permite compilar para Android y iOS desde un único código base, pero tiene peor integración con APIs nativas de Android y mayor dependencia de plugin externos.

Con ello conseguiremos mejor rendimiento, experiencia de usuario, integración con APIs nativas como el calendario o las notificaciones, aunque no cubren plataforma IOS y depende del ecosistema Android para su distribución.

ADR-003: La aplicación requiere almacenar datos como el contacto de los usuarios, sus notificaciones, etc.

Para ello vamos a utilizar MySQL como base de datos relacional a la cual nos conectaremos a través de JDBC y la biblioteca de acceso a datos de Spring (Spring Data JPA).

MongoDB sería una alternativa para almacenar datos en forma de documentos flexibles pero usa relaciones complejas y las transacciones son más limitadas que en SQL.

Con esto conseguimos un modelo relacional adecuado, ya que MySQL es ampliamente soportado y escalable verticalmente y tiene una integración nativa con Spring, aunque sería complejo si decidimos cambiar a un sistema NoSQL y requiere ajustes de rendimiento para operaciones con muchos adjuntos.

### 5.3. Integraciones, datos y dependencias

Necesitamos integrar la API REST (Spring Boot) para que la app móvil acceda al backend de manera segura y estructurada.

Firebase Cloud Messaging va a permitir el envío de notificaciones push sobre eventos o recordatorios de rodaje.

Para sincronizar los eventos con el calendario del dispositivo usaremos el Calendario nativo de Android.

El almacenamiento de archivos con MySQL y subida de adjuntos nos permitirá guardar documentos del guión, storyboard y otra documentación necesaria para el rodaje.

Los datos que guardaremos para el plan de rodaje será el nombre del proyecto, número de escenas, número de planos, desgloses y fechas de rodaje.

#### 5.4. Riesgos técnicos y mitigación

- Caída del servidor que surge debido a que se han conectado múltiples usuarios a la app provocando que se ralentice o colapse lo cual afecta a los usuarios debido a que no podrían acceder a sus planes de rodaje o eventos.

La mejor forma de solucionarlo es optimizando la base de datos realizando consultas eficientes y creando índices en campos clave

También se puede mitigar evitando que un usuario haga demasiadas solicitudes en poco tiempo y sature el servidor