 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: <https://www.figma.com/design/7uuGFrBAiA3oblY7giKNL/Sin-t%C3%ADtulo?node-id=5-270&t=pgOzxSjk61uZXmHJ-1>

Tablero: <https://trello.com/b/JjWZRkV2/pidam2a-tecnoimagex-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 centrada principalmente en el desarrollo de guiones y preproducción, es muy similar a la anterior. Como punto negativo, es una aplicación de pago, aunque cuenta con un periodo de prueba gratuito.

	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

HU-001: Registro o inicio de sesión. <https://trello.com/c/kJQgHfmb>

HU-002: Crear un nuevo proyecto. <https://trello.com/c/HxpaQmR3>

HU-003: Editar información del proyecto. <https://trello.com/c/LzOvhpEs>

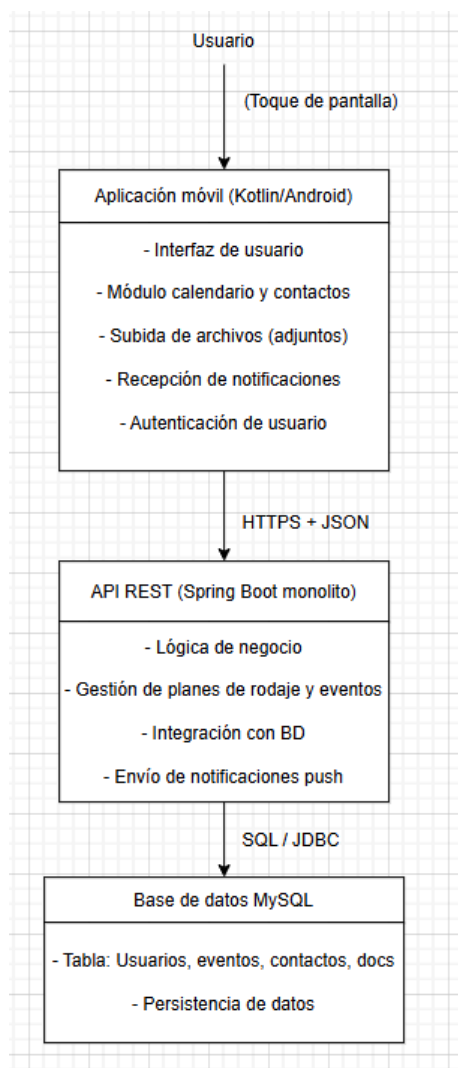
HU-004: Añadir evento al calendario. <https://trello.com/c/Kb2oBNjQ>

HU-005: Marcar un evento como cumplido. <https://trello.com/c/aFQ0mnhK>

HU-006: Recibir notificaciones. <https://trello.com/c/BkPI6ULQ>

## 5. Arquitectura

Diagrama:



## Decisiones de arquitectura:

### **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.

## 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.

Necesitaremos almacenar archivos con MySQL y subir adjuntos para 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.

Contaremos con datos de contactos que incluyen puesto, nombre, teléfono, correo electrónico y dirección. Es fundamental también tener en cuenta los eventos relacionados con la producción, fechas de rodajes y tareas completadas o pendientes dentro del calendario, y datos de registro de los usuarios.

Se utilizarán Kotlin y el Android SDK para la construcción de la aplicación móvil nativa y usaremos Spring Boot para desarrollar el backend, que gestiona la lógica de negocio y accesos a la base de datos.

Con Spring Data JPA tendremos un acceso seguro y estructurado a la base de datos MySQL.

Jackson será la librería encargada de la serialización y deserialización de datos entre objetos Java y estructuras JSON.

El sistema de persistencia para usuarios, proyectos, contactos, eventos y documentos se realizará en una base de datos MySQL.

## Riesgos técnicos y mitigación:

**Caída del servidor:** 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 saturar el servidor.

**Acceso no autorizado:** Un problema muy real y común es el acceso de un usuario malintencionado a los datos de los actores o a los documentos del plan de rodaje provocando problemas de privacidad por la filtración de los guiones o robo de información sensible.

La mejor forma de solucionarlo es diferenciar los roles de cada usuario (Actores, directores, productores) para que cada uno tenga un acceso específico a la aplicación.

## **6. Requisitos no funcionales (NFR)**

### **Rendimiento y capacidad:**

Como requisitos no funcionales debemos tener en cuenta cómo de rápido y eficiente debe ser el sistema y cuántos recursos puede manejar. También se relaciona con la escalabilidad, es decir, cómo responde el sistema al aumento de usuarios o datos.

#### **NFR-001. Tiempo de respuesta móvil.**

La aplicación debe responder en un máximo de 500 ms para operaciones comunes (cargar proyectos, mostrar eventos, acceder al calendario).

#### **NFR-002. Tiempo de respuesta API**

Las llamadas a la API REST deben responder en  $\leq 300$  ms en condiciones normales de tráfico.

#### **NFR-003. Capacidad de usuarios concurrentes**

El backend debe soportar al menos 50 usuarios concurrentes, valor adecuado para productoras pequeñas y usuarios académicos.

#### **NFR-004. Carga de archivos**

Los documentos adjuntos (guion literario, guion técnico y storyboard) deben subirse en menos de 3 segundos para archivos de hasta 10 MB.

#### **NFR-005. Throughput mínimo**

El sistema debe ser capaz de procesar 50 solicitudes por segundo sin degradación crítica.

## **Seguridad:**

Definimos cómo se protege la información y se controla el acceso al sistema.

### **NFR-006. Autenticación segura**

El acceso se realizará mediante autenticación basada en JWT con expiración automática.

### **NFR-007. Autorización por roles**

El sistema debe diferenciar permisos entre el usuario estándar y el administrador del sistema.

### **NFR-008. Cifrado en tránsito**

Toda la comunicación entre aplicación móvil y backend debe usar HTTPS + TLS 1.2 o superior.

### **NFR-009. Cifrado en reposo**

Las contraseñas deben almacenarse con hash BCrypt. Transformando con el algoritmo BCrypt la contraseña en un hash y permitiendo que la contraseña se almacene en la base de datos de forma segura.

### **NFR-010. Protección contra ataques**

El backend debe incluir protección contra fuerza bruta, inyección SQL (ya cubierta por JPA) y accesos no autorizados a documentos.

## **Accesibilidad y experiencia de usuario (UX):**

Definimos cómo los usuarios interactúan con la aplicación y garantizamos que todos puedan usarla de manera efectiva.

### **NFR-011. Compatibilidad mínima**

La aplicación debe funcionar en dispositivos Android.

### **NFR-012. Tiempo de aprendizaje**

El usuario debe poder comprender el uso básico de la aplicación en menos de 5 minutos.

### **NFR-013. Contraste y legibilidad**

Toda la interfaz debe cumplir al menos el estándar WCAG 2.1 AA de contraste.

#### **NFR-014. Consistencia visual**

Todos los módulos deben seguir un diseño coherente con el prototipo de Figma.

### **Calidad del código y mantenibilidad:**

Describimos cómo debe estar construido el código para que sea fácil de mantener, entender y extender.

#### **NFR-015. Estándar de codificación**

El proyecto móvil debe seguir las guías de estilo oficiales de Kotlin y Android Jetpack.

El backend debe seguir los estándares de Spring Boot y Java.

#### **NFR-016. Cobertura de pruebas**

Se requiere pruebas unitarias y pruebas de integración en el backend

### **Observabilidad**

Definimos la capacidad de supervisar y diagnosticar el sistema mientras está en funcionamiento.

#### **NFR-017. Logs estructurados**

El backend debe almacenar logs con nivel INFO, WARN y ERROR usando formato JSON para depuración.

#### **NFR-018. Seguimiento de errores**

Los fallos en la app móvil deben ser reportados mediante un sistema como Firebase Crashlytics.

### **Verificación de NFR**

Es importante saber cómo vamos a comprobar que se cumplen los NFR.

#### **NFR-019. Pruebas de rendimiento**

Se usarán pruebas de carga con JMeter o Gatling. Debe verificarse la latencia API  $\leq 300\text{ms}$  y 50 usuarios concurrentes sin errores.

## NFR-020. Pruebas de seguridad

Test de autenticación y expiración JWT, revisión contra inyección SQL y acceso indebido y validación de logs y registros de intentos fallidos.

## NFR-021. Accesibilidad

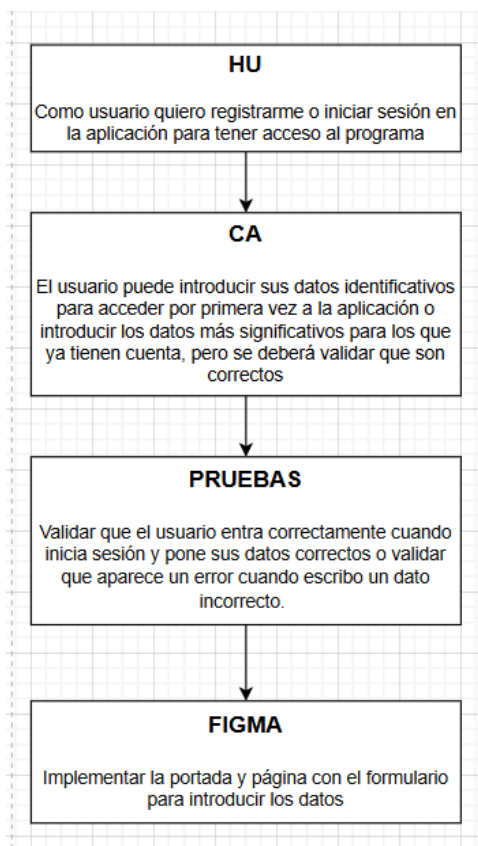
Validación del contraste con herramientas específicas para ello, prueba manual de TalkBack y revisión de tamaños mínimos táctiles (48 dp).

## NFR-022. Observabilidad

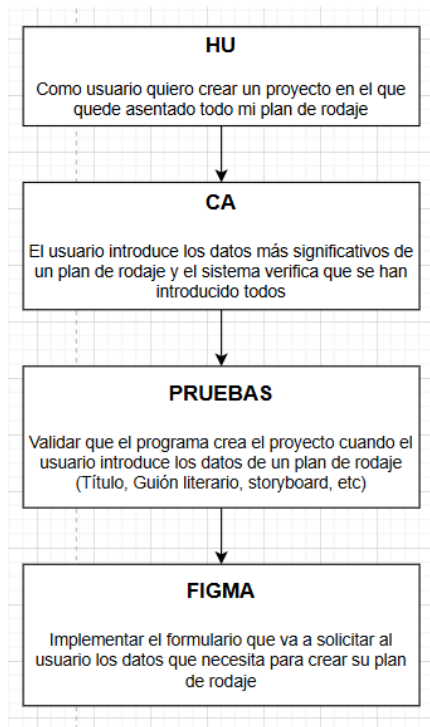
Verificación de que los logs se almacenan correctamente.

# 7. Trazabilidad HU

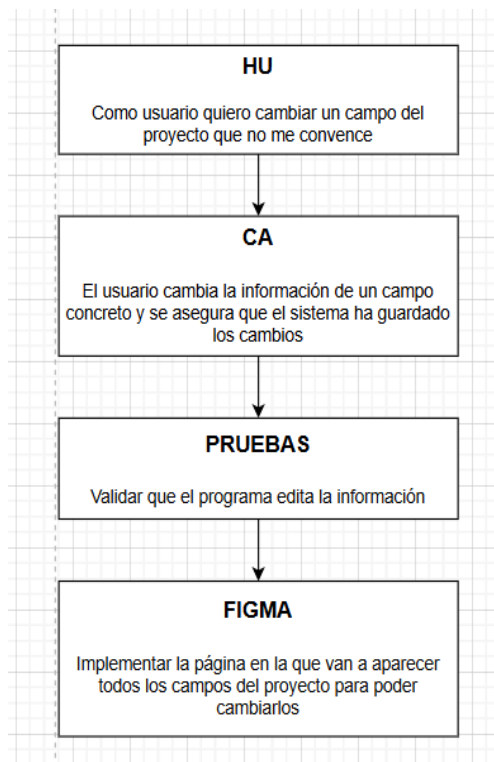
HU-001: Registro o inicio de sesión



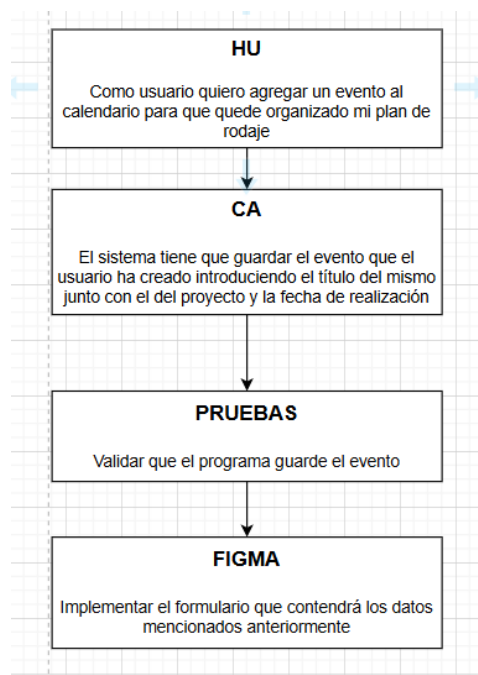
## HU-002: Crear un nuevo proyecto



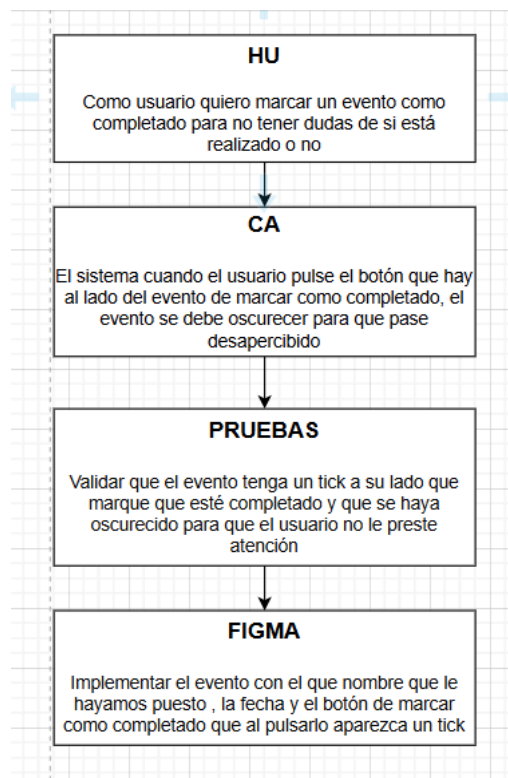
## HU-003: Editar información del proyecto



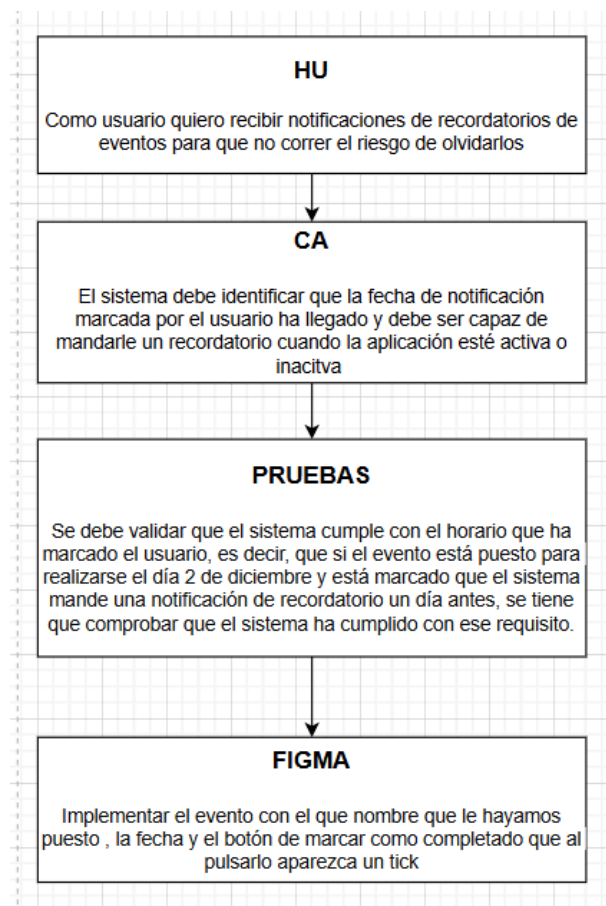
#### HU-004: Añadir evento al calendario



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



## HU-006: Recibir notificaciones



## 8. Prototipo Figma y Anexos

La guía de diseño la creamos gracias a los colores, la tipografía y el tipo de diseño que hemos elegido. Podemos decir que en nuestra aplicación tenemos dos colores principales, estos son el gris oscuro, casi negro, porque contrasta con el beige clarito del fondo y, por otro lado, como segundo color principal, el verde que también resalta con el fondo, además de ser el color que más sobresale en la paleta de colores, es por ello que lo destinamos a los botones. Como colores secundarios hemos elegido un verde más oscuro que usamos para destacar aquellos botones, textos o iconos interactivos y el blanco roto que permite diferenciar los formularios del fondo para que el usuario sepa perfectamente cuáles son rellenables y cuáles no.

#201E1F	#59514C	#F6F6F6	#F0EFEF	#D9D2CC	#668047	#455533
#B40000						

El nivel de contraste que existe entre el texto principal y el fondo es 14.44:1, comprobado a través de WebAIM Contrast Checker. Además, hemos mirado la paleta de color en el Página 5 de 7 Cristina Carrasco Guillén DISEÑO DE UNA APP DIGITAL simulador Stark y el color verde y los marrones y negros mantienen una clara diferencia, aunque para no depender exclusivamente del color hemos usado los botones y los iconos.

El proyecto cuenta con dos tipos de fuentes, una para los títulos y otra para los textos. Para la primera elegimos la fuente Raleway, por su aspecto sofisticado perfecta para destacar la información y además es muy legible para los textos cortos. Para los textos con el resto de información elegimos Nunito, una fuente redondeada que mejora la legibilidad y deja un aspecto muy acogedor. Ambas fuentes son muy sencillas, sin llamativos adornos para facilitar la lectura. El tamaño del título principal es 28px encontrando títulos H1, H2 y H3 y el texto con un máximo de 16px pudiendo encontrar texto con diferentes tamaños dependiendo de la necesidad de la pantalla o la información.

El estilo que le hemos dado a nuestra aplicación es un estilo minimalista y profesional, donde hemos cuidado de cada detalle para que los usuarios se sientan lo más cómodos posibles al usar la app.

El tamaño grande de los botones es muy apropiado para los dedos en el móvil, con las esquinas redondeadas, generando sombra para separarlos del fondo y así hacer que destaquen mejor, junto con un color llamativo claramente diferenciador del resto de elementos. Como toque especial añadimos la bellota caracterizadora de la tierra.

Los formularios por el contrario se han diseñado con un color más claro que el del fondo, con un tono casi blanco, fácil de reconocer para todos los usuarios, pues estos casi siempre nos lo encontramos en cualquier aplicación, tanto móvil como web, de un tono parecido. Añadimos también información en la parte superior de cada recuadro para indicar al usuario qué información debe añadir en él.

Cada uno de los recuadros del formulario se encuentra bien separado del otro para evitar confusión y errores. El layout mantiene un encuadre organizado y simétrico, organizado de forma vertical. Centramos los elementos principales para mejorar su lectura en movimiento, en definitiva, un diseño consistente que permite que la app sea más predecible y coherente. Pero este encuadre no sería posible sin un buen espaciado, tanto en la parte superior e inferior como en los laterales, generando márgenes de 24px, y por supuesto también dejamos aproximadamente unos 16px de separación entre los elementos importantes, títulos, cajas de formularios, etc.

A lo largo de todas las pantallas van a aparecer iconos reconocibles para el usuario con los que este pueda interactuar, un ejemplo es el icono del lápiz el cual, al verlo, el usuario va a entender inmediatamente que si pulsa sobre él va a poder escribir o modificar alguna información dentro de los formularios. Lo mismo ocurre con el icono de la papelería, de calendario, notificaciones, etc.

Hemos utilizado tanto variantes como componentes para generar un prototipo navegable lo más parecido a la aplicación real posible. Para hacer más rápido y práctico el diseño del

prototipo hemos creado diferentes variables (texto para los botones o para el color), en Figma.

## 9. Anexos

### Plan de pruebas

HU: Iniciar sesión o registrarse

Pruebas: Comprobar si el registro o el inicio de sesión es efectivo

Resultado: El inicio de sesión o registro en el figma debe ser efectivo y en el caso del inicio de sesión se debe simular que la contraseña es válida

Captura:

**iComencemos!**

↓

Email  
aaaa

Contraseña  
aaaa

Contraseña incorrecta

**Rellene el formulario** ↓

Nombre  
aaaaa

Apellidos  
aaaaa

Email  
aaaaa

Contraseña  
aaaaa

Registrarse ↗

HU: Crear un proyecto

Prueba: Comprobar que el proyecto se ha creado correctamente

Resultado: El figma debe simular la creación de un proyecto

Captura:

Nombre del proyecto  
Introduce datos

Adjuntar guión literario

Adjuntar guión técnico

Adjuntar Storyboard

Documentación

Casting y equipo

Desglose de guión

Guardar proyecto ↗

**Proyectos guardados**

La Siberia Estremeña

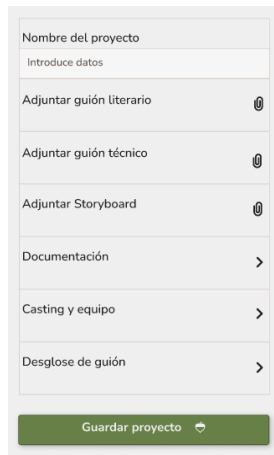
Las mil lunas

HU: Editar proyecto


Pruebas: Comprobar que se han guardado los cambios realizados


Resultado: El figma debe simular que ha cambiado la información del proyecto y para ello accedemos de nuevo al formulario cuando pulsemos el proyecto creado y cambiaremos el campo que queramos


Captura:



Nombre del proyecto  
Introduce datos

Adjuntar guión literario 


Adjuntar guión técnico 

Adjuntar Storyboard 

Documentación >

Casting y equipo >

Desglose de guión >

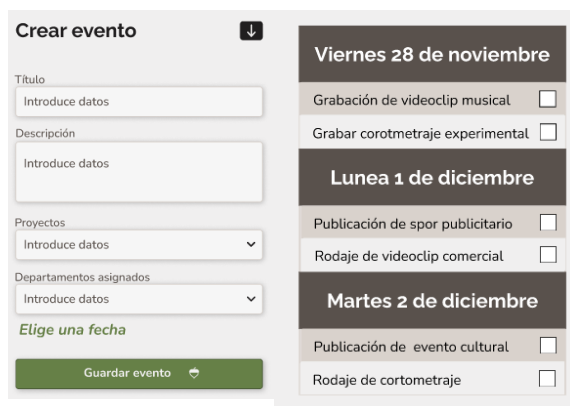
Guardar proyecto 


HU: Agregar evento

Prueba: Comprobar que se ha creado el evento

Resultado: El figma debe simular la creación del evento

Captura:



**Crear evento** 


Título  
Introduce datos

Descripción  
Introduce datos

Proyectos  
Introduce datos

Departamentos asignados  
Introduce datos

*Elige una fecha*

Guardar evento 

**Viernes 28 de noviembre**

Grabación de videoclip musical ☐

Grabar corotmetraje experimental ☐

**Lunea 1 de diciembre**

Publicación de spor publicitario ☐

Rodaje de videoclip comercial ☐

**Martes 2 de diciembre**

Publicación de evento cultural ☐

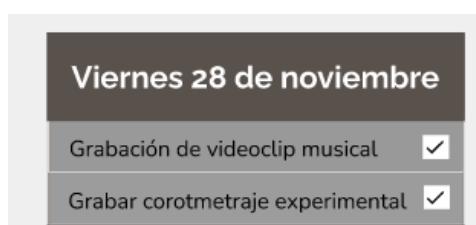
Rodaje de cortometraje ☐

HU: Marcar evento como completado

Pruebas: Comprobar que se ha marcado un evento como completado

Resultado: El figma debe simular lo que debe pasar cuando marquemos el evento como completado

Captura:



**Viernes 28 de noviembre**

Grabación de videoclip musical ☒

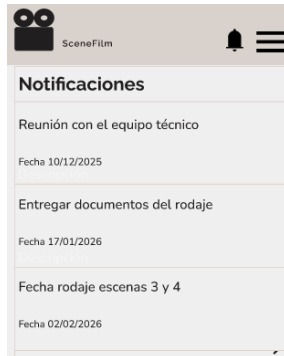
Grabar corotmetraje experimental ☒

HU: Recibir notificaciones

Pruebas: Comprobar si hay una notificación de recordatorio de eventos

Resultado: El figma debe simular lo que sucede cuando pulsamos en el icono de notificaciones

Captura:



## KPIs iniciales

### 1. Aceptación de las historias de usuario

Número de HU: 6

Validadas en el prototipo: 6

### 2. Defectos detectados

No implementar la página de eventos

No establecer un buen uso de colores

Número de defectos: 2

Defectos solucionados: 2

### 3. Lead Time

Tiempo empleado en las historias de usuario: 3 días

Tiempo empleado en el prototipo: 14 días

Tiempo empleado en la documentación: 7 días

## Enlaces

Git Hub: <https://github.com/AlejandroBs-06/ProyectoIntermodular>

Figma:

<https://www.figma.com/design/7uuGFrbAiA3oblY7giKNL/Sin-t%C3%ADtulo?node-id=5-270&t=KxYdSHdHiQCwZ30V-1>

Trello:

<https://trello.com/invite/b/6901e6f6846650931fa9d51e/ATT1c5bed0fae4d5298b9ce3b4ef4e66d0840058397A/pidam2a-tecnoimagex-scenefilm>

## Cambios relevantes

Cambio realizado	Fecha de realización
Creación de las historias de usuario	12/11/2025
Implementación de la HU en el trello	26/11/2025
Conjunto de los prototipos: Comparación de los prototipos y elección del mejor diseño para crear el prototipo global	6/12/2025