



UT 1. Idea inicial y planteamiento del proyecto

Proyecto Final de Grado

Edición Online - Rev. 2025082001

Docente: DIANA PADILLA FREIXINET

Alumno: CARLOS JIMÉNEZ VITÓN

2ºDAM

Centro: Linkia FP

Módulo: Proyecto Final de Grado

Contenido

1. Roles del equipo	5
1.1 Reparto de funciones.	5
1.2 Justificación del reparto de roles.....	5
2. Introducción y descripción de la idea inicial	6
2.1 Contexto y necesidad detectada.	6
2.2 Solución propuesta.....	7
2.3 Relevancia de la idea.....	8
2.4 Resumen ejecutivo.....	9
3. Proceso de creatividad e innovación	9
3.1 Técnicas aplicadas.....	9
3.2 Ejemplos de ideas iniciales generadas.	11
3.3 Justificación de la propuesta seleccionada.	11
4. Análisis del entorno y diagnóstico inicial	12
• Matriz DAFO.....	12
• Análisis PESTEL	12
• Benchmarking.	¡Error! Marcador no definido.
5. Definición de objetivos y fases del proyecto	14
• Objetivo general.....	14
• Al menos 3 objetivos SMART.....	14
• Fases del proyecto.....	14
• Indicadores de logro.....	15
6. Planificación de recursos y calendario	16
• Recursos humanos.	16
• Recursos materiales y tecnológicos.....	16
• Recursos financieros.....	16
7. Protección de la idea	17
• Elementos a proteger.....	17
• Tipo de protección seleccionado	17

• Justificación de la elección.	 Error! Marcador no definido.	17
8. Modelo de negocio		17
• Desarrollo.....		17
9. Conclusiones		18
• Síntesis de aprendizajes del grupo.		18
• Principales dificultades encontradas.....		19
10. Bibliografía		19
• Referencias.....		19

1. Roles del equipo

1.1 Reparto de funciones.

Aunque este Trabajo Final se desarrolla de manera individual, he decidido estructurar el proyecto siguiendo la misma dinámica que tendría un equipo profesional. Esta organización permite mantener un flujo de trabajo ordenado, evitar pérdidas de tiempo y asegurar que todas las fases del proyecto (planificación, diseño, desarrollo, investigación y documentación) siguen un criterio coherente.

1.2 Justificación del reparto de roles.

- **Coordinación del proyecto:** En ausencia de un equipo real, asumo el rol de coordinador definiendo:
 - La organización del repositorio GitHub, ramas y control de versiones.
 - La coherencia entre las diferentes partes del proyecto:
 - Aplicación móvil **AppVidriera** en React Native
 - API **ApiConexSql** en .NET Core
 - Base de datos SQL Server
 - Documentación del TFG
- **Área creativa:** Como responsable creativo aplico técnicas de diseño y generación de ideas propias de un entorno profesional:
 - Creación de wireframes y diseño preliminar de pantallas.
 - Definición de la identidad visual de la app (paleta de colores, iconos, estilo general).
 - Análisis de referentes de aplicaciones reales del sector industrial y de gestión de tareas.
 - Pruebas para simplificar la navegación con React Navigation.

- **Investigación y análisis:** Esta parte del trabajo ha sido importante debido a las tecnologías empleadas:

- Estudio de la **Google Calendar API**.

- Revisión de documentación oficial de **React Native**, .NET.

- Análisis de métodos para conectar la app móvil con la API propia y con Google.

- Evaluación de diferentes estructuras de base de datos y modelos.

- Comparación de alternativas.

- **Redacción y diseño del documento:** Como responsable de la documentación:

- Estructuro todo el TFG siguiendo un índice.

- Redacto las secciones técnicas de forma clara para que cualquier lector pueda entender cómo funciona la aplicación, la API y la integración con Google Calendar.

- Reviso y maqueto el documento final en Word.

2. Introducción y descripción de la idea inicial

2.1 Contexto y necesidad detectada.

En la industria del vidrio, especialmente en entornos como el de Vidriera Arandina, la gestión de tareas, mantenimientos, incidencias y organización diaria suele realizarse con herramientas dispersas: notas en papel, mensajes de WhatsApp, calendarios personales, hojas de cálculo o simples recordatorios manuales. Este sistema fragmentado provoca pérdida de información, duplicidades, falta de seguimiento y dificultades para coordinar trabajos entre operarios y responsables.

Además, en los procesos industriales es crucial llevar un control claro de fechas, revisiones periódicas, mantenimientos preventivos y tareas programadas. Sin una herramienta centralizada, resulta complicado tener una visión global que permita anticiparse a problemas, mejorar la productividad y optimizar el flujo de trabajo.

A nivel personal, detecté esta necesidad tras varios años de experiencia directa en el mantenimiento industrial y en la organización de actividades dentro de este sector. La falta de una aplicación que conecte el trabajo diario con un calendario estructurado motivó la creación de una solución que sea accesible, intuitiva y moderna.

2.2 Solución propuesta.

La solución planteada es **AppVidriera**, una aplicación móvil desarrollada en **React Native** que se integra con:

Una API propia construida en **.NET (ApiConexSql)**

Una base de datos **SQL Server**

A **Google Calendar API** para el análisis, visualización y gestión de eventos

El objetivo es centralizar y organizar la información relacionada con tareas industriales, eventos, mantenimientos y actividades relevantes, todo desde una única plataforma móvil.

La app permite:

- Consultar eventos programados del Google Calendar corporativo.
- Visualizar mantenimientos preventivos, tareas e incidencias.
- Registrar información organizada en la base de datos.
- Unificar datos procedentes tanto del sistema interno como del calendario de Google.
- Acceder fácilmente desde cualquier dispositivo móvil mediante.

Con esta arquitectura, **AppVidriera** combina la flexibilidad de un sistema propio con la potencia de las herramientas ya existentes (como Google Calendar), evitando duplicar sistemas y aprovechando integraciones reales de uso diario.

2.3 Relevancia de la idea.

La relevancia de esta propuesta surge por varios motivos:

- Responde a una necesidad real del sector industrial

Es una herramienta diseñada específicamente para mejorar el control operativo en entornos donde el flujo de trabajo es intenso y la coordinación es esencial.

- Profesionaliza la gestión de tareas y mantenimientos

Permite que la organización deje de depender de recordatorios manuales o de información dispersa, dando lugar a un sistema más fiable y eficiente.

- Facilita la digitalización

Muchas empresas del sector aún no están plenamente digitalizadas. Una app como AppVidriera supone un paso directo hacia procesos más modernos, organizados y accesibles.

- Aprovecha tecnologías actuales y estándares abiertos

La integración con Google Calendar facilita la adopción inmediata, ya que las herramientas de Google forman parte del uso diario en muchas industrias.

- Permite ampliar la solución en el futuro

La arquitectura modular (App móvil + API + BD + Google Calendar) está pensada para crecer:

- Nuevos módulos de mantenimiento
- Gestión de operarios
- Control de inventario
- Notificaciones push
- Panel web.

2.4 Resumen ejecutivo.

AppVidriera es una aplicación móvil diseñada para optimizar la gestión de tareas, mantenimientos y eventos en el entorno industrial de la vidriera. El proyecto integra una app desarrollada en React Native con una API propia en .NET y una base de datos SQL Server. Además, se conecta con la Google Calendar API para centralizar, sincronizar y visualizar eventos de forma unificada.

La solución surge ante la necesidad de ordenar información dispersa, mejorar la coordinación operativa y disponer de un sistema accesible desde cualquier dispositivo móvil. Permite consultar, registrar y analizar actividades, facilitando el control del día a día y reduciendo errores derivados de la gestión manual. Su arquitectura modular está pensada para crecer con nuevas funcionalidades según las necesidades reales del entorno industrial.

AppVidriera representa un paso firme hacia la digitalización de procesos industriales, sustituyendo métodos tradicionales basados en papel o comunicaciones informales por un sistema estructurado, centralizado y tecnológicamente avanzado.

Aunque nace orientada al sector del vidrio, su planteamiento es plenamente ampliable y adaptable al departamento de mantenimiento de cualquier empresa, independientemente del sector, permitiendo aplicar la misma metodología de organización, planificación y control en múltiples ámbitos profesionales.

3. Proceso de creatividad e innovación

3.1 Técnicas aplicadas.

3.1.1 Brainstorming orientado (lluvia de ideas guiada en tutorías)

Durante las reuniones con la profesora se generaron diversas ideas relacionadas con:

Cómo integrar la API de Google Calendar de forma útil para un entorno industrial.

Qué estructura y contenidos debía incluir la pantalla Home para ser realmente funcional para un operario.

Qué funcionalidades aportarían valor añadido para garantizar la sostenibilidad del proyecto a medio y largo plazo.

Qué módulos podrían incorporarse en versiones futuras (inventario, incidencias, mantenimientos preventivos...).

Aunque no se realizó un brainstorming grupal clásico, sí se aplicó el principio fundamental, que es generar ideas libremente, sin juzgarlas de inmediato, para después evaluarlas y seleccionar las más útiles.

3.1.2 Técnica SCAMPER aplicada a la idea inicial

Durante el desarrollo del proyecto fui aplicando SCAMPER de manera práctica para explorar variaciones y mejoras:

- Sustituir:

Registros manuales por formularios digitales.

Notificaciones internas por integración con Google Calendar.

- Adaptar:

Transformar una interfaz industrial tradicional en un diseño móvil moderno e intuitivo.

Adaptar el sistema para poder escalarlo a cualquier empresa, no solo al sector del vidrio.

- Modificar:

Reorganizar el Home para mostrar accesos directos.

- Proponer nuevos usos:

Utilizar la app para programar turnos o eventos internos de cualquier departamento.

- Eliminar:

Funcionalidades demasiado avanzadas para esta fase, como notificaciones push complejas.

3.2 Ejemplos de ideas iniciales generadas.

Crear un calendario propio gestionado desde SQL Server.

Uno de los análisis creativos y técnicos más relevantes del proyecto fue determinar si AppVidriera debía incluir un calendario propio desarrollado desde cero o si resultaba más adecuado usar directamente Google Calendar mediante su API oficial.

Durante las tutorías con la profesora valoramos ambas opciones:

- Calendario propio:
- Google Calendar vía API

3.3 Justificación de la propuesta seleccionada.

Tras evaluar la complejidad, el valor añadido y el tiempo disponible, la conclusión fue clara: usar Google Calendar era la opción más eficiente, potente y sostenible.

- Calendario propio:

Mayor complejidad al desarrollar repeticiones, avisos, sincronización, recordatorios, días laborables...

- Google Calendar vía API

Sistema robusto, estable y ampliamente usado.

Funciones avanzadas ya resueltas (eventos periódicos, recordatorios, sincronización entre dispositivos).

Evita reinventar algo que ya funciona perfectamente.

Reduce la carga de desarrollo y aumenta la sostenibilidad del proyecto.

Se integra de manera natural con los procedimientos de trabajo reales de una empresa.

4. Análisis del entorno y diagnóstico inicial

4.1 Matriz DAFO.

Fortalezas:

- Integración directa con Google Calendar, herramienta robusta y ampliamente extendida.
- Arquitectura modular (App + API + BD) que permite crecimiento y futuras ampliaciones.
- Experiencia real en el sector industrial y de mantenimiento, lo que aporta enfoque práctico y ajustado a necesidades reales.

Debilidades:

- Proyecto desarrollado por una sola persona ya que las horas disponibles limitan la complejidad.
- Dependencia de la API de Google.
- Versión inicial sin funciones avanzadas (inventario, notificaciones push).

Oportunidades:

- Las empresas industriales están en pleno proceso de digitalización y automatización.
- Falta de apps específicas y económicas para mantenimiento accesibles desde el móvil en pymes.

Amenazas:

- Existencia de software comercial de mantenimiento (GMAO) más completo, aunque caro.
- Dependencia tecnológica del ecosistema de Google.
- Competencia creciente en apps móviles de productividad y organización.

4.2 Análisis PESTEL.

P – Político

- Impulso europeo a la digitalización industrial.
- Políticas de apoyo a la eficiencia en procesos operativos dentro de empresas.

E – Económico

- Muchas pymes no pueden asumir el coste elevado de software especializado (GMAO).
- Las soluciones internas o personalizadas suelen ser más rentables a largo plazo.

S – Social

- Necesidad de mejorar la comunicación interna en entornos industriales.

T – Tecnológico

- Google ofrece APIs estables y documentadas que facilitan la integración.
- Expansión del desarrollo multiplataforma (React Native) que reduce costes.

E – Ecológico

- Sustituir documentos en papel por herramientas digitales reduce residuos.
- Mejor organización del mantenimiento puede evitar averías que generen desperdicio energético.

L – Legal

- Las apps internas pueden controlar mejor el almacenamiento de información sensible.

[4.3 Benchmarking.](#)

Para evaluar el posicionamiento de AppVidriera se ha analizado la herramienta **GMAO** similar y presente en el mercado industrial Software de mantenimiento muy utilizado internacionalmente. Permite crear órdenes de trabajo, controlar inventarios, registrar incidencias y visualizar mantenimientos preventivos.

Ventaja competitiva principal de AppVidriera:

AppVidriera destaca por su simplicidad, Economía y conexión nativa con Google Calendar, lo que hace que su adopción sea inmediata y cercana al trabajo real de los operarios.

5. Definición de objetivos y fases del proyecto

[5.1 Objetivo general.](#)

Desarrollar una aplicación móvil integrada con Google Calendar que permita organizar, visualizar y gestionar tareas y mantenimientos en entornos industriales, mejorando la coordinación, la eficiencia y la digitalización de procesos.

[5.2 Objetivos SMART.](#)

- Implementar la integración entre AppVidriera, la API .NET con la BBDD y Google Calendar.
- Diseñar e implementar una interfaz móvil en React Native para que los usuarios puedan acceder a las funciones principales

[5.3 Fases del proyecto](#)

El proyecto se ha organizado siguiendo un enfoque clásico dividido en cinco fases: Inicio, Planificación, Ejecución, Seguimiento y Cierre.

-Fase de Inicio

- Identificación de la necesidad en el entorno industrial.
- Realización del documento idea inicial y planteamiento del proyecto.
- Primera toma de contacto con tecnologías clave (React Native, API .NET, SQL Server, Google Calendar).

- Fase de Planificación

- Creación de la estructura del proyecto móvil, API, BD y conexión con Google Calendar.
- Preparación del repositorio y estructura del proyecto.

- Fase de Ejecución

- Desarrollo de la aplicación móvil con React Native.
- Implementación de la API en .NET y modelo de datos en SQL Server.
- Integración de Google Calendar a través de su API oficial.
- Pruebas funcionales y ajustes en base a resultados.

- Fase de Seguimiento

- Reuniones periódicas de revisión en tutorías con la profesora.
- Validación de funciones de la App

- Fase de Cierre

- Documentación final del TFG.
- Pruebas finales de funcionamiento.
- Preparación de la presentación final.

5.4 Indicadores de logro.

- Reducción del número de clics necesarios para acceder a funciones clave
- Reducción del tiempo medio para marcar una tarea como completada

6. Planificación de recursos y calendario

6.1 Recursos humanos.

Siendo un proyecto individual, todos los roles los desempeño yo con gran apoyo externo en la toma de decisiones. Las tutorías con la profesora me ayudaron sobre todo en las fases de investigación y diseño, ya que me permitió aclarar dudas y ver qué opciones eran más viables.

6.2 Recursos materiales y tecnológicos.

Recursos materiales:

- Ordenador personal
- Móvil Android
- Conexión a internet

Herramientas de software:

- React Native
- Visual Studio Code
- .Net para la API
- SQL Server Management Studio
- Potsman
- Google Cloud para API Calendar
- Git Hub
- Microsoft Office
- Draw IO

6.3 Recursos financieros.

El coste real ha sido prácticamente cero, porque casi todo el software que utilice es gratuito.

Internet → 50€

7. Protección de la idea

7.1 Elementos a proteger.

El nombre del proyecto “AppVidriera”

El diseño de las pantallas

El código fuente de la aplicación

7.2 Tipo de protección seleccionado.

Copyright (derechos de autor) para el código y la documentación.

Marca registrada para el nombre del proyecto

8. Modelo de negocio (Business Model Canvas)

8.1 Desarrollo de los 9 bloques.

8.1.1 Segmentos de clientes

-Empresas industriales.

-Pymes.

-Departamentos de mantenimiento de cualquier sector.

-Autónomos. Para gestionar su calendario de trabajo.

8.1.2 Propuesta de valor.

- App ligera, sencilla y accesible desde cualquier móvil.

- Coste muy bajo frente a soluciones comerciales complejas.

- Escalable y adaptable a diferentes empresas.

8.1.3 Canales

- Publicación en Google Play.

- Contacto directo con empresas industriales de la zona.

- Presentación en centros de formación o ciclos de FP.

8.1.4 Relación con clientes

- Atención personalizada para empresas.

- Prueba gratuita en caso de querer comercializarla.

8.1.5 Fuentes de ingresos

- Licencia mensual o anual.
- Personalizaciones.
- Instalación y configuración inicial.
- Mantenimiento.

8.1.6 Recursos clave

- Código de la aplicación.
- API .NET y BBDD SQL.
- Credenciales Google.
- Herramientas de control de versiones.

8.1.7 Actividades Clave

- Desarrollo y mejora continua de la aplicación.
- Control de errores y soporte a usuario
- Integración de nuevos módulos

8.1.8 Alianzas clave

- Google

8.1.9 Estructura de costes

- Hosting del servidor y BBDD
- API Google calendar a niveles superiores
- Coste del dominio web
- Publicidad y difusión

9. Conclusiones

9.1 Síntesis de aprendizajes del grupo.

Aunque he realizado el proyecto de forma individual, he aprendido muchas cosas tanto a nivel técnico como organizativo. Una de las principales lecciones ha sido entender cómo unir diferentes tecnologías en un mismo proyecto: la app en React Native, la API en .NET , la base de datos en SQL Server y la integración con Google Calendar. Antes de empezar pensaba que sería más

sencillo, pero poco a poco he ido viendo que cada decisión afecta al resto del sistema.

También he aprendido a planificar mejor. Las tutorías con mi profesora me ayudaron a centrarme cuando me iba por caminos demasiado complejos y a entender que es mejor avanzar paso a paso. Otra cosa importante que he aprendido es que, aunque haya muchas formas de resolver un problema, lo mejor suele ser elegir la opción más práctica y sostenible (como usar el calendario de Google en lugar de crear uno desde cero).

Por último, he mejorado mi forma de documentar y justificar decisiones técnicas, algo que no siempre hacemos cuando programamos, pero que he descubierto que es algo muy útil.

9.2 Principales dificultades encontradas.

Uno de los retos más importantes que me he encontrado ha sido entender bien como funciona la API de Google calendar para poderla aprovechar al máximo en mi app.

Organizar el tiempo para poder compaginar trabajo con el desarrollo del proyecto también ha sido algo complicado de gestionar. Sobre todo porque no siempre avanza a la velocidad que uno espera.

10. Bibliografía

Referencias en formato APA 7^a edición (mínimo 5)

Google. (2024). Google Calendar API Documentation. Google Developers.

<https://developers.google.com/calendar>

Meta. (2024). React Native Documentation. Meta Open Source.

<https://reactnative.dev/docs>

Microsoft. (2024). ASP.NET Core Documentation. Microsoft Learn.

<https://learn.microsoft.com/aspnet/core>

OpenAI. (2024). ChatGPT (versión GPT-5.1) [Modelo de lenguaje de inteligencia artificial].

<https://chat.openai.com>

Ministerio de Industria, Comercio y Turismo. (2024). Industria Conectada 4.0: Programa de digitalización para empresas. Gobierno de España.

<https://www.industriaconectada40.gob.es>

Instituto Nacional de Ciberseguridad. (2024). Guía de ciberseguridad para pymes. INCIBE.

<https://www.incibe.es>

AENOR. (2022). UNE-EN 13306: Mantenimiento – Terminología de mantenimiento. Asociación Española de Normalización.

<https://www.aenor.com>

Cámara de Comercio de España. (2023). Informe sobre digitalización de pymes en España. Cámara de España.

<https://www.camara.es>

Infoautónomos. (2023). Guía de software de gestión para pymes y autónomos. Infoautónomos.

<https://www.infoautonomos.com>

Fracttal One Software GMAO

Fracttal. (2024). Fracttal One: Software GMAO en la nube.

<https://www.fracttal.com/es/>