Imagen que contiene Forma

Descripción generada automáticamenteTexto

Descripción generada automáticamente con confianza baja

Proyecto de desarrollo de aplicaciones multiplataforma

GESTIÓN DE CONSULTAS MÉDICAS

Ciclo Formativo de Grado Superior

**Desarrollo de aplicaciones multiplataforma (IFCS02)**

Curso 2024-2025

Autor/a/es:

**Maximiliano Benjamín Ramos Aguirre**

Tutor/a:

**Federico Banda Sierra**

**Una captura de pantalla de un celular con la imagen de una caricatura

Descripción generada automáticamente con confianza baja**

Departamento de Informática y Comunicaciones

**IES Luis Vives**

Resumen

**Gestión de Consultas Médicas** es un proyecto orientado a facilitar la gestión de citas médicas en centros de salud mediante una solución multiplataforma. Se compone de una aplicación móvil para pacientes (desarrollada en Kotlin para Android), un portal web para la gestión de citas por parte del personal sanitario (en React.js) y una API RESTful (en ASP.NET Core) que conecta ambos entornos. Este sistema permite reservar, modificar y cancelar consultas médicas, así como gestionar usuarios y agendas médicas. El objetivo es mejorar la experiencia del paciente, optimizar los recursos del centro médico y contribuir a la digitalización del sector sanitario.

Abstract

**Medical Appointment Management** is a cross-platform project aimed at improving appointment scheduling in healthcare centers. It includes an Android mobile app for patients (developed in Kotlin), a web portal for healthcare staff (developed in React.js), and a RESTful API (built with ASP.NET Core) that integrates both platforms. The system allows users to book, modify, and cancel appointments, and enables doctors to manage their schedules. The project seeks to enhance patient experience, optimize resources, and support the digital transformation of the healthcare industry.

Índice

[1. INTRODUCCIÓN 3](#_Toc178519360)

[1.1. OBJETIVO 3](#_Toc178519361)

[1.2. ALCANCE 3](#_Toc178519362)

[1.3. JUSTIFICACIÓN 3](#_Toc178519363)

[2. IMPLEMENTACIÓN 4](#_Toc178519364)

[2.1. ANÁLISIS DE LA APLICACIÓN 4](#_Toc178519365)

[2.1.1. Requisitos funcionales de la aplicación 4](#_Toc178519366)

[2.1.2. Análisis y selección de las tecnologías 4](#_Toc178519367)

[2.1.3. Planificación de la realización del proyecto 4](#_Toc178519368)

[2.1.4. Otros 5](#_Toc178519369)

[2.2. DISEÑO 6](#_Toc178519370)

[2.3. IMPLEMENTACIÓN 6](#_Toc178519371)

[2.4. IMPLANTACIÓN 6](#_Toc178519372)

[2.5. DOCUMENTACIÓN 7](#_Toc178519373)

[3. CONCLUSIONES 7](#_Toc178519374)

[3.1. RESULTADOS Y DISCUSIÓN 7](#_Toc178519375)

[3.2. TRABAJO FUTURO (OPCIONAL) 7](#_Toc178519376)

[4. BIBLIOGRAFÍA 7](#_Toc178519377)

[ANEXOS 7](#_Toc178519378)

# INTRODUCCIÓN

En la actualidad, los centros de salud deben enfrentarse a innumerables problemáticas con la gestión de eficiencia de citas al médico, la organización de la agenda de los médicos y la atención oportuna de sus pacientes. Las soluciones más tradicionales, las que dependen de procesos manuales o de herramientas que están desconectadas, acaban generando un importante número de retrasos, errores administrativos y una experiencia poco favorable para los pacientes y para el personal médico.

Este proyecto, que se ha desarrollado en el contexto del Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Multiplataforma (DAM), tiene como finalidad la generación de un sistema multiplataforma para la gestión de la consulta médica que permita modernizar todos estos procesos a partir de tecnologías actuales. La solución propuesta está caracterizada por:

Una API RESTful, creada en ASP.NET Core, que es el motor del sistema y que expone funcionalidades seguras y escalables para todos los soportes.

Una aplicación móvil para los pacientes, desarrollada a partir de Kotlin, con la que los usuarios pueden visualizar fácilmente sus citas con médicos desde su terminal Android.

Un sitio web para médicos y gestores, desarrollado en React.js, el cual condiciona la organización de la agenda médica, el manejo de usuarios y la supervisión administrativa general. La finalidad de este proceso simultáneo de desarrollo de estas tres plataformas es el de aportar una solución digital total, plenamente disponible y eficiente, que ofrezca la mejor experiencia de uso de los pacientes y facilite la experiencia de uso de los profesionales de la medicina. El proyecto ha sido diseñado y desarrollado por elementos de trabajo a través de una metodología iterativa, que ha ido adaptándose a las necesidades que se han ido detectando durante el desarrollo de este.

## OBJETIVO

El objetivo principal que sustenta la operación de este proyecto está cimentado en el diseño y desarrollo de una solución digital de tipo multiplataforma que gestione de forma adecuada las distintas consultas médicas de un centro de salud determinado, de forma que se facilite la gestión e interacción que puedan llevar a cabo paciente-salud, mediante la dotación de herramientas tecnológicas de última generación.

A través del desarrollo del sistema que se posteriormente se va a describir dentro del marco de este proyecto se persiguen los siguientes objetivos específicos:

**Objetivos Técnicos**

* El objetivo principal del proyecto es el siguiente: Crear una API RESTful, robusta y segura, con ASP.NET Core en el backend y Entity Framework Core como herramienta de acceso a datos.
* Crear una aplicación móvil Android, desarrollada en Kotlin, que permita a los pacientes consultar y gestionar sus citas médicas.
* Crear un portal web responsive en React.js y TypeScript que permita a los doctores y administradores gestionar el sistema desde cualquier dispositivo con acceso a Internet.
* Crear una estructura de base de datos relacional eficiente y escalable con SQL Server.
* Asegurar la sincronización de las tres plataformas establecidas en el API común.

**Objetivos de Aprendizaje**

* Profundizar en el uso de frameworks modernos tanto en el backend (ASP.NET Core) como en el frontend (React.js, Kotlin).
* Consolidar conocimientos sobre desarrollo de aplicaciones móviles y su conexión con APIs externas.
* Aprender a estructurar un proyecto real aplicando principios de arquitectura limpia y buenas prácticas de desarrollo.
* Utilizar herramientas de planificación como diagramas de Gantt y metodologías de desarrollo iterativo.
* Aplicar técnicas de autenticación y autorización seguras (por ejemplo, JWT) para proteger el acceso a los datos del sistema.

**Objetivos Funcionales**

* Mejorar la organización interna de un centro médico a través de la digitalización de procesos.
* Reducir los tiempos de espera y la carga administrativa mediante una gestión automatizada de citas.
* Facilitar a los pacientes el acceso a sus consultas desde cualquier lugar y en cualquier momento.
* Proporcionar a los doctores una herramienta intuitiva para visualizar y gestionar su agenda.
* Ofrecer a los administradores una plataforma integral para el control y supervisión del sistema completo.

## ALCANCE

El proyecto se delimitará en las siguientes áreas y funcionalidades, considerando un desarrollo que sea alcanzable en un período de 3 meses:

**Funcionalidades Básicas (Obligatorias)**

**Backend (API RESTful en ASP.NET Core):**

* + Gestión de usuarios (pacientes, médicos, administradores).
  + CRUD de citas médicas y gestión de horarios.
  + Autenticación y autorización (mediante JWT).
  + Conexión a una base de datos para el almacenamiento seguro de datos.

**Aplicación Móvil (Android con Kotlin):**

* + Registro e inicio de sesión para pacientes.
  + Visualización de disponibilidad de citas.

**Portal Web de Administración:**

* + Panel de control para la gestión de consultas y agendas médicas.
  + Módulo de administración para registrar y actualizar datos de pacientes y profesionales.
  + Funcionalidades CRUD para servicios y citas.
  + Creación de archivos PDF de citas.

## JUSTIFICACIÓN

En el contexto actual, muchos sistemas de salud aún presentan importantes limitaciones en la gestión digital de citas médicas, lo que se traduce en largos tiempos de espera, sobrecarga administrativa y deficiencias en la comunicación entre pacientes y profesionales sanitarios. Esta situación afecta negativamente tanto a la calidad del servicio como a la experiencia del usuario.

Este proyecto surge como respuesta a dicha problemática, proponiendo una solución ágil, moderna e intuitiva que permita:

* Mejorar la calidad y eficiencia del servicio médico.
* Reducir la carga de trabajo administrativo mediante la automatización de procesos.
* Ofrecer a pacientes y doctores una experiencia digital más fluida y accesible.
* Impulsar la transformación digital en el ámbito sanitario, especialmente en centros de salud de menor escala o sin sistemas informatizados.

# IMPLEMENTACIÓN

## ANÁLISIS DE LA APLICACIÓN

La aplicación permitirá a los pacientes gestionar sus citas de forma sencilla y a los profesionales médicos visualizar y organizar sus agendas.

### Requisitos funcionales de la aplicación

 Registro e inicio de sesión de usuarios.

 Visualización de disponibilidad y horarios.

 Reserva, modificación y cancelación de citas.

 Gestión de usuarios, citas y agendas desde el portal web.

### Análisis y selección de las tecnologías

Se han valorado varias tecnologías y se han elegido las siguientes:

* **Backend:** ASP.NET Core (robustez, integración con Entity Framework Core, buen soporte para APIs REST).
* **Base de datos:** PostgreSQL (compatibilidad y fiabilidad).
* **Frontend Web:** React.js (popularidad, rendimiento, componentes reutilizables).
* **App Móvil:** Kotlin (idioma oficial Android, moderno, seguro).
* **Despliegue:** Vercel(Portal Web) y Render(Api y BBDD).

### Planificación de la realización del proyecto

**Fase 1: Análisis y Diseño (Semanas 1-2)**

* **Semana 1 a Semana 2:**
  + Recolección de requisitos y definición del alcance.
  + Diseño de la arquitectura del sistema (diagramas UML, modelo entidad-relación para la base de datos).
  + Especificación de la interfaz de usuario para la aplicación móvil y el portal web.

**Fase 2: Desarrollo del Backend (Semanas 3-5)**

* **Semana 3 a Semana 5:**
  + Implementación del API RESTful en ASP.NET Core.
  + Configuración y conexión de la base de datos.
  + Desarrollo de funcionalidades de autenticación y autorización.
  + Pruebas iniciales utilizando herramientas como Postman.

**Fase 3: Desarrollo de la Aplicación Móvil (Semanas 4-7)**

* **Semana 4 a Semana 7:**
  + Creación de la interfaz de usuario (uso de Fragments y RecyclerView).
  + Integración de la aplicación móvil con el API para la gestión de citas.
  + Implementación de funcionalidades de registro, reserva y notificaciones.

**Fase 4: Desarrollo del Portal Web de Administración (Semanas 7-10)**

* **Semana 7 a Semana 10:**
  + Desarrollo del panel de control para la gestión de consultas médicas.
  + Implementación de funcionalidades CRUD y generación de informes.
  + Realización de pruebas integradas entre el portal web, la aplicación móvil y el API.

**Fase 5: Pruebas Finales, Optimización y Documentación (Semanas 11-12)**

* **Semana 11 a Semana 12:**
  + Ejecución de pruebas de integración y usabilidad.
  + Optimización del rendimiento y corrección de errores.
  + Redacción de la documentación final y preparación del informe del proyecto.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.

### Otros

# Presupuesto del Proyecto

Este presupuesto estima los costes necesarios para el desarrollo completo del proyecto durante un período de 12 semanas, considerando recursos humanos, infraestructura tecnológica y otros gastos relacionados.

## Recursos Humanos

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Horas estimadas | Tarifa por hora (€) | Coste (€) |
| Análisis y diseño | 40 | 25 | 1,000 |
| Desarrollo Backend | 80 | 25 | 2,000 |
| Desarrollo Frontend Web | 60 | 25 | 1,500 |
| Desarrollo App Móvil | 60 | 25 | 1,500 |
| Testing y corrección | 30 | 20 | 600 |
| Documentación y entrega | 20 | 20 | 400 |
| Total Recursos Humanos | 290 |  | 7,000 |

## 2. Infraestructura y Servicios

|  |  |  |  |
| --- | --- | --- | --- |
| Concepto | Coste mensual (€) | Meses | Coste total (€) |
| Hosting backend (Render) | 10 | 3 | 30 |
| Hosting web (Vercel) | 0 | 3 | 0 |
| Base de datos PostgreSQL | 15 | 3 | 45 |
| Servicio email SMTP | 5 | 3 | 15 |
| Licencias software | 0 | 0 | 0 |
| Total Infraestructura |  |  | 90 |

## 3. Hardware y Otros Gastos

|  |  |
| --- | --- |
| Concepto | Coste (€) |
| Amortización PC de desarrollo | 125 |
| Materiales y papelería | 50 |
| Total Hardware y Otros | 175 |

## 4. Resumen Final del Presupuesto

|  |  |
| --- | --- |
| Categoría | Coste (€) |
| Recursos Humanos | 7,000 |
| Infraestructura y Servicios | 90 |
| Hardware y Otros | 175 |
| TOTAL GENERAL | 7,265 |

## Notas

• Las tarifas por hora se han ajustado a niveles de un desarrollador junior o becario.  
• Se han utilizado servicios con planes gratuitos o de bajo coste para minimizar gastos.  
• El coste humano está valorado para dar una idea profesional, aunque en un TFG es trabajo personal no remunerado.  
• Este presupuesto es una aproximación para un desarrollo profesional realista en el tiempo estipulado.

## DISEÑO

También tendrá sus propios apartados. A modo de ejemplo podremos servirnos de:

* Prototipado: Diseño de pantallas (no el resultado final), bocetos, Mockups

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

* Diagramas E/R o modelo de datos
* Diagramas de clase genérico (Atributos y métodos principales)
* Casos de uso (El desarrollo de cada caso de uso en anexos)
* Diagramas de flujo de datos: Pueden complementar aquellos casos de uso
* complejos.
* Base de datos. Estructura de tablas a utilizar, soporte lógico y físico.
* Arquitectura: Tecnologías utilizadas en cada parte del proyecto

## IMPLEMENTACIÓN

Se desarrollará una API RESTful en ASP.NET Core con controladores para cada entidad.

La app móvil se conectará al backend mediante llamadas HTTP usando Retrofit.

El portal web utilizará Axios o Fetch para la comunicación con la API.

## IMPLANTACIÓN

El despliegue inicial se realizará localmente y, posteriormente, en servidores de Vercel (Portal Web) y Render (Api Service + Base de datos).  
El APK será instalado directamente en dispositivos Android de prueba.

## DOCUMENTACIÓN

Se entregará:

* Documentación del proyecto que incluirá (estructura, API endpoints, esquemas de BD).

# CONCLUSIONES

Este proyecto ha permitido aplicar de forma real los conocimientos adquiridos a lo largo del ciclo formativo, desarrollando una solución útil para un problema real. Se ha trabajado en equipo con planificación, adaptando los conocimientos técnicos a necesidades concretas y desarrollando habilidades en backend, frontend y despliegue en la nube.

## RESULTADOS Y DISCUSIÓN

A lo largo del desarrollo se han superado diversos retos técnicos, especialmente en la sincronización de agendas y gestión de estados entre frontend y backend. La planificación inicial se ha cumplido en su mayor parte, con pequeños ajustes derivados de la complejidad real en fases intermedias.

## TRABAJO FUTURO (OPCIONAL)

Hasta donde se ha llegado y posibles ampliaciones del proyecto (continuación del alcance).

# BIBLIOGRAFÍA

 Microsoft Docs – ASP.NET Core: <https://learn.microsoft.com/aspnet/core>

 Android Developers – Kotlin: <https://developer.android.com/kotlin>

 React Official Website: <https://react.dev>

 PostgreSQL Documentation: <https://www.postgresql.org/docs/>

 Azure Architecture Center: <https://learn.microsoft.com/azure/architecture>

 Figma Design Tool: <https://www.figma.com>

# ANEXOS

Detalle del código y otra documentación que consideréis oportuna.

I. Diagrama de clases  
II. Modelo E/R  
III. Prototipos de interfaz  
IV. Código fuente relevante