



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Sistema de control de asistencia para alumnos de la UPV

Trabajo Fin de Grado

Grado en Ingeniería de Sistemas de Telecomunicación, Sonido e
Imagen

AUTOR/A: Ferri Tomás, Daniel

Tutor/a: Tomás Gironés, Jesús

CURSO ACADÉMICO: 2023/2024

Resumen

El control de asistencia de los alumnos en las diferentes sesiones de teoría o prácticas se ha realizado tradicionalmente por medio de hojas de papel, donde los alumnos tenían que firmar. Con la aparición de la pandemia esta práctica dejó de ser recomendada para evitar contagios.

Ante este problema se planteó como alternativa la posibilidad de que los alumnos utilizaran su propio móvil para realizar esta tarea. El objetivo de este proyecto es desarrollo de un sistema automático que permita un registro de los alumnos que asisten a cada sesión tanto presencial como de forma remota. Como requisitos se exige una solución muy ágil y sencilla de utilizar que permita realizar el control de asistencia sin que el profesor pierda más de un par de minutos.

Palabras clave: aplicación web; control de asistencia; docencia online; herramienta docente

Abstract

The attendance control of students in different theory or practical sessions has traditionally been done through paper sheets, where students had to sign. With the emergence of the pandemic, this practice ceased to be recommended to prevent contagion.

Faced with this problem, the possibility was considered as an alternative for students to use their own mobile phones to carry out this task. The objective of this project is the development of an automatic system that allows for the registration of students attending each session, both in-person and remotely. The requirements include a very agile and easy-to-use solution that allows attendance control without the teacher losing more than a couple of minutes.

Keywords: web application; attendance control; online teaching; teaching tool

Índice

Portada	1
Resumen	2
Abstract	3
Índice	4
Índice de imágenes	6
Glosario	7
1. Introducción	9
1.1. Contexto y motivación	9
1.2. Objetivos y alcance	9
1.3. Estructura del documento	10
2. Estado del arte y tecnologías utilizadas	11
2.1. Sistemas de control de asistencia	11
2.1.1. Control de asistencia en entornos empresariales	11
2.1.2. Control de asistencia en entornos educativos	11
2.2. Tecnologías utilizadas	12
2.2.1. React	12
2.2.2. Firebase	14
2.2.3. Single Page Application	15
2.2.4. Bases de datos NoSQL	16
3. Diseño de la aplicación	18
3.1. Análisis de requerimientos	18
3.1.1. Requisitos funcionales	18
3.1.2. Requisitos no funcionales	18
3.2. Justificación de las tecnologías utilizadas	19
3.2.1. Backend	19
3.2.2. Frontend	20
3.3. Arquitectura de la aplicación	20
3.3.1. Backend	20
3.3.2. Frontend	20
3.4. Diseño de la interfaz de usuario	21
3.4.1. Sección pública	21
3.4.2. Sección privada	21
3.5. Diseño de la base de datos	27
4. Implementación	28
4.1. Herramientas de trabajo	28
4.2. Estructura de la aplicación	28
4.2.1. Ficheros de Configuración	28
4.2.2. Código fuente	29
4.3. Seguridad y autenticación	30
4.3.1. Autenticación	30
4.3.2. Autorización	30
4.3.3. Seguridad	32

5. Manuales	33
5.1. Manual de instalación	33
5.1.1. Instalación de requerimientos	33
5.1.2. Creación del proyecto de Firebase	33
5.1.3. Clonado del repositorio, configuración y despliegue	43
5.2. Manual de uso	45
5.2.1. Instrucciones de Gestión:	45
5.2.2. Instrucciones para el Profesor:	45
5.2.3. Instrucciones para el Alumno:	46
6. Resultados y discusión	47
6.1. Evaluación de los resultados	47
6.2. Comparación con soluciones existentes	47
7. Conclusiones y trabajos futuros	49
7.1. Conclusiones	49
7.2. Trabajos futuros	49
Referencias bibliográficas	51
Adjunto A: Repositorio git con el código fuente	53

Índice de imágenes

Lista de imágenes

- [Imagen 1. Gestión de grupos](#)
- [Imagen 2. Crear grupo](#)
- [Imagen 3. Gestión de un grupo individual](#)
- [Imagen 4. Gestión de alumnos en un grupo](#)
- [Imagen 5. Gestión de profesores en un grupo](#)
- [Imagen 6. Visualización de alumnos que han asistido a una sesión](#)
- [Imagen 7. Esquema de la base de datos](#)
- [Imagen 8. Crear un proyecto en Firebase](#)
- [Imagen 9. Nombrar un proyecto en Firebase](#)
- [Imagen 10. Configurando Google Analytics para el proyecto](#)
- [Imagen 11. Activando autenticación](#)
- [Imagen 12. Lista de métodos de autenticación](#)
- [Imagen 13. Activando autenticación por email y contraseña](#)
- [Imagen 14. Agregando un segundo método de autenticación](#)
- [Imagen 15. Activando autenticación mediante Google](#)
- [Imagen 16. Múltiples autenticaciones activadas](#)
- [Imagen 17. Activando Firebase Datastore](#)
- [Imagen 18. Eligiendo ubicación de la base de datos](#)
- [Imagen 19. Eligiendo reglas por defecto](#)
- [Imagen 20. Activando Hosting](#)
- [Imagen 21. Hosting paso 1](#)
- [Imagen 22. Hosting paso 3](#)
- [Imagen 23. Hosting paso 4](#)
- [Imagen 24. Abriendo configuración del proyecto](#)
- [Imagen 25. Configuración de Firebase](#)

Glosario

- **Firebase [3][4]**

Plataforma de desarrollo de aplicaciones móviles y web propiedad de Google. Proporciona una variedad de servicios en la nube, como una base de datos en tiempo real, autenticación de usuarios, alojamiento web, entre otros, facilitando el desarrollo eficiente y escalable de aplicaciones.

- **Git [5][6]**

Sistema de control de versiones distribuido ampliamente utilizado en el desarrollo de software. Permite el seguimiento de cambios en el código fuente, facilita la colaboración entre equipos de desarrollo y brinda herramientas para gestionar ramas, fusiones y versiones del proyecto

- **Github [7]**

Plataforma en línea que utiliza el sistema de control de versiones Git para alojar y gestionar repositorios de código fuente. Ofrece características colaborativas, como la gestión de problemas, solicitudes de extracción y seguimiento de cambios, facilitando la colaboración en proyectos de desarrollo de software

- **JavaScript [8]**

Lenguaje de programación de alto nivel, interpretado y orientado a objetos. Ampliamente utilizado para el desarrollo web, permite la creación de contenido dinámico e interactivo en las páginas web. Es un componente fundamental en el desarrollo del lado del cliente.

- **Node [9]**

Entorno de ejecución de JavaScript del lado del servidor. Basado en el motor V8 de Google Chrome, permite a los desarrolladores ejecutar código JavaScript en el servidor, lo que facilita la creación de aplicaciones web escalables y de alto rendimiento. Este entorno es ampliamente utilizado para el desarrollo de aplicaciones web y servidores

- **React [10][11]**

Librería de JavaScript utilizada para construir interfaces de usuario interactivas. Desarrollada por Facebook, se centra en la creación eficiente de componentes reutilizables que representan la interfaz de usuario de una aplicación, facilitando el desarrollo de aplicaciones de una sola página (SPA)

- **Typescript [12]**

Superset de JavaScript que añade tipado estático al lenguaje. Desarrollado por Microsoft, proporciona herramientas adicionales para estructurar y organizar el código, mejorando la capacidad de mantenimiento y la detección de errores durante el desarrollo.

- **Yarn [13]**

Gestor de paquetes para aplicaciones JavaScript. Creado por Facebook, se utiliza para gestionar dependencias y garantizar la consistencia en la instalación de paquetes.

Yarn es conocido por su velocidad y eficiencia en comparación con otros administradores de paquetes.

1. Introducción

1.1. Contexto y motivación

El control de asistencia es una actividad esencial en el ámbito educativo, ya que permite a los docentes llevar un registro de la participación de los alumnos en las clases y evaluar su compromiso con el proceso educativo [1]. Tradicionalmente, esta tarea se ha llevado a cabo mediante hojas de papel en las que los alumnos firmaban en su fila correspondiente para marcar su asistencia. Sin embargo, este método manual consume tiempo y recursos valiosos que podrían ser empleados en actividades más productivas.

La irrupción de la pandemia ocasionó una gran cantidad de cambios en nuestras vidas [2]. Como no podía ser de otra forma el mundo de la docencia tuvo que adaptarse a las nuevas restricciones, para tratar de reducir al mínimo el número de contagios.

La Universidad Politécnica de Valencia (UPV) y más concretamente la Escuela Politécnica Superior de Gandía (EPSG) siguió la siguiente estrategia. En una primera fase el 100% de las clases se trataron de impartir de forma online. En una segunda fase se retomaron las clases presenciales, pero tratando que aquellos alumnos que al ser positivos no pudieran asistir pudieran seguir las clases de forma online.

En este contexto fue necesario cambiar muchas de las rutinas a las que estábamos acostumbrados. Una de ellas era la necesidad de pasar lista tanto en las clase de teoría como en las prácticas. La forma más tradicional de pasar lista consistía en nombrar en voz alta a cada uno de los alumnos que de asistir a esa sesión. Esta alternativa tiene el inconveniente de ser muy lenta y pesada. Para minimizar estos inconvenientes habitualmente se utiliza un sistema alternativo. Consiste en pasar a los alumnos una hoja de papel con un listado de sus nombres. Los alumnos han de ir pasándose esta hoja y firmando junto a su nombre. Esta alternativa que era una de las más utilizadas en la EPSG dejó de ser posible durante la pandemia al aumentar la posibilidad de contagios entre los alumnos.

Con esta problemática resultaba necesario definir una alternativa para pasar lista durante las clases y prácticas que por una parte pudiera realizarse sin consumir excesivo tiempo y que además sea cien por cien seguro a la transmisión de COVID.

En este contexto, el presente Trabajo de Fin de Grado (TFG) en Ingeniería de Telecomunicaciones tiene como objetivo diseñar e implementar un sistema basado en una página web accesible desde dispositivos móviles, que permita a los alumnos marcar su presencia en el aula de forma rápida y eficiente, evitando el contacto físico con el material compartido.

1.2. Objetivos y alcance

El objetivo principal de este TFG es desarrollar un sistema de control de asistencia que facilite y agilice la tarea de pasar lista en entornos educativos, al mismo tiempo que reduce el riesgo de contagio de enfermedades. Para ello, se plantean los siguientes objetivos específicos:

- Realizar una revisión bibliográfica y del estado del arte sobre sistemas de control de asistencia y tecnologías relacionadas.
- Diseñar la arquitectura, la interfaz de usuario y la base de datos del sistema.
- Implementar la solución utilizando tecnologías y herramientas adecuadas, garantizando la seguridad y la autenticación de los docentes, aunque no se garantiza la autenticidad de los alumnos.
- Analizar y discutir las limitaciones y posibles mejoras del sistema, así como su efectividad en entornos educativos reales.

El alcance de este proyecto se limita al desarrollo de una solución basada en una página web accesible desde dispositivos móviles, sin la creación de aplicaciones móviles específicas para diferentes plataformas. Además, se centra en el contexto educativo de la universidad, aunque la solución propuesta podría ser aplicable a otros entornos con adaptaciones adecuadas.

1.3. Estructura del documento

El presente documento se organiza de la siguiente manera:

- Capítulo 2: Revisión bibliográfica y estado del arte, donde se analizan sistemas de control de asistencia existentes y las tecnologías empleadas.
- Capítulo 3: Diseño de la solución, incluyendo los requisitos, la arquitectura, la interfaz de usuario y la base de datos.
- Capítulo 4: Implementación, destacando las tecnologías y herramientas utilizadas, así como los aspectos relacionados con la seguridad y autenticación de los docentes.
- Capítulo 5: Manuales, incluyendo tanto manual de instalación como de uso
- Capítulo 6: Resultados y discusión, incluyendo la evaluación de la efectividad del sistema en entornos educativos reales, las limitaciones y posibles mejoras.
- Capítulo 7: Conclusiones y trabajos futuros, donde se resumen los principales hallazgos del proyecto y se proponen líneas de investigación futuras.

Por último, se incluyen las referencias bibliográficas empleadas en el desarrollo del TFG.

2. Estado del arte y tecnologías utilizadas

2.1. Sistemas de control de asistencia

El control de asistencia ha experimentado una transformación considerable con la incorporación de tecnologías digitales, marcando una evolución desde los tradicionales métodos manuales hacia soluciones automatizadas y basadas en la nube. Estos sistemas modernos no solo agilizan el proceso de registro de asistencia, sino que también aportan funcionalidades adicionales como análisis de datos, integraciones de comunicación, y mejoras en la seguridad y eficiencia operativa.

2.1.1. Control de asistencia en entornos empresariales

En el contexto empresarial, el control de asistencia se ha adaptado para soportar una variedad de necesidades organizacionales, desde la gestión de tiempo de los empleados hasta la optimización de la productividad y la supervisión remota.

- **Connecteam** [18]: Esta herramienta no se limita al registro de asistencia; ofrece una plataforma integral para la gestión de empleados, incluyendo la comunicación, la capacitación del equipo, y el seguimiento de tareas y horarios, lo que resulta en una mejora de la productividad general.
- **Hubstaff** [19]: Especialmente diseñado para equipos remotos o híbridos, esta aplicación ofrece seguimiento de ubicación GPS y reportes detallados para analizar y optimizar la utilización del tiempo y los recursos. Su enfoque en el "monitoreo de la productividad" permite identificar tareas ineficientes y analizar la productividad.

2.1.2. Control de asistencia en entornos educativos

En el sector educativo, las soluciones de control de asistencia se han diversificado para abordar los desafíos específicos de las instituciones educativas, incluyendo la gestión académica, la comunicación entre el personal docente, los estudiantes y sus familias, y la adaptación a diferentes etapas educativas.

- **Alexia** [20]: Permite a los docentes pasar lista desde dispositivos móviles o la web. Se extiende a actividades extraescolares, y los padres pueden acceder y justificar ausencias en tiempo real. Incluye un servicio de notificaciones para informar a los padres sobre las faltas de asistencia.
- **Dinantia** [21]: Esta plataforma de comunicación web y móvil facilita pasar lista desde cualquier dispositivo y enviar notificaciones automáticas a los padres sobre las ausencias de los estudiantes. Los profesores utilizan un módulo de asistencia que proporciona estadísticas detalladas de asistencia.
- **Esemtia** [22]: Del Grupo edebé, ofrece una solución integral para todas las etapas educativas, desde Infantil hasta Formación Profesional, abarcando gestión académica y administrativa, y una función de justificación de faltas.

- **Gestión Aula:** Parte del Grupo Anaya, es un entorno de gestión integral que permite a los centros educativos crear su propio portal de comunicación y gestión de faltas.
- **iEduca [23]:** Disponible en la nube y como aplicación para dispositivos iOS y Android, facilita a los profesores el control de asistencia y ofrece a las familias acceso a información detallada relacionada con sus hijos.
- **Additio App [24]:** Combina el control de asistencia con un cuaderno de notas digital. Disponible en la web y para Android e iOS, permite a los docentes planificar cursos, seguir notas y controlar la asistencia, incluso sin conexión a Internet.

Estas herramientas digitales representan un avance significativo en comparación con los métodos tradicionales. Ofrecen una mayor eficiencia, seguridad en la gestión de datos, y una mejor administración del tiempo, siendo fundamentales tanto en entornos empresariales como educativos. Su adaptabilidad y las funcionalidades específicas que proporcionan destacan la importancia de la tecnología en la mejora de los procesos de control de asistencia en diversas esferas.

2.2. Tecnologías utilizadas

2.2.1. React

React [11], conocido también como React.js o ReactJS, es una biblioteca de JavaScript de código abierto desarrollada por Facebook, destinada a la construcción de interfaces de usuario en aplicaciones web y móviles. Su diseño modular y la capacidad de crear aplicaciones web dinámicas lo han posicionado como una herramienta esencial en el desarrollo de front-end.

Origen y Evolución:

Creado originalmente por Jordan Walke en Facebook, React se lanzó en 2011 y desde entonces ha evolucionado, convirtiéndose en un pilar en el desarrollo de aplicaciones modernas. La adopción de React se aceleró debido a su eficiencia y flexibilidad, viéndose implementado en grandes proyectos como Instagram y WhatsApp.

Características Detalladas:

- **Componentes Reutilizables:** React se basa en la composición de componentes, que encapsulan su propia estructura, estilo y lógica. Estos componentes pueden ser reutilizados en diferentes partes de una aplicación, lo que aumenta la coherencia y disminuye la duplicación de código.
- **JSX (JavaScript XML):** JSX es una extensión de sintaxis para JavaScript que permite escribir estructuras de componentes de UI en un formato que se asemeja al HTML. JSX mejora la legibilidad del código y facilita la descripción visual de la interfaz de usuario.

- **Virtual DOM:** React utiliza un DOM virtual que es básicamente una réplica ligera del DOM real. Al ocurrir cambios en la interfaz de usuario, React realiza primero las modificaciones en el DOM virtual y luego, mediante un algoritmo de reconciliación, actualiza el DOM real de manera eficiente. Esto minimiza las operaciones costosas en el DOM y mejora el rendimiento de la aplicación.
- **Data Binding Unidireccional:** React sigue un flujo de datos unidireccional o "top-down", lo que facilita el seguimiento de cambios en el estado de la aplicación y mejora la estabilidad del código.
- **Hooks:** Introducidos en la versión 16.8, los Hooks permiten a los componentes funcionales tener un estado y otros rasgos React sin necesidad de escribir una clase. Esto simplifica el código y mejora la legibilidad y la reutilización de los componentes.
- **Context API:** Permite compartir valores como preferencias de usuario, temas, etc., entre componentes, sin tener que pasar explícitamente props a través de cada nivel del árbol de componentes.

Aplicaciones y Uso Práctico:

- **Desarrollo de Aplicaciones de Página Única (SPA):** React es ideal para crear SPAs eficientes y dinámicas, donde el contenido se carga dinámicamente sin necesidad de recargar la página.
- **Aplicaciones Móviles con React Native:** La extensión de React, React Native, permite el desarrollo de aplicaciones móviles nativas para iOS y Android utilizando el mismo paradigma de desarrollo de componentes de React.

Integración con Otras Tecnologías:

React se puede integrar con una amplia gama de APIs y bibliotecas, incluyendo sistemas de diseño de UI como Material UI, y plataformas de backend como Firebase. Esta interoperabilidad lo hace versátil y adecuado para una variedad de proyectos de desarrollo.

A pesar de la popularidad de React, su curva de aprendizaje y la complejidad de su ecosistema pueden representar un desafío significativo, especialmente para los principiantes. La comprensión profunda de conceptos fundamentales como el estado de los componentes, su ciclo de vida, y el manejo adecuado de efectos secundarios es esencial para el desarrollo eficiente con React. Estos aspectos, aunque cruciales, pueden ser abrumadores para los desarrolladores novatos debido a la riqueza y diversidad del ecosistema de React.

Además, en el contexto de aplicaciones de gran envergadura y complejidad, la gestión del rendimiento emerge como otro desafío notable. Este aspecto es particularmente crítico en lo que respecta a la optimización de la renderización y la gestión eficiente del estado dentro de la aplicación. A medida que las aplicaciones escalan en tamaño y funcionalidad, garantizar una experiencia de usuario fluida y eficiente requiere una atención

meticulosa a estos detalles de rendimiento, lo que puede complicar el desarrollo y la optimización de aplicaciones complejas con React.

React continúa evolucionando, con actualizaciones y mejoras que refuerzan su posición como una herramienta fundamental para el desarrollo moderno de interfaces de usuario. Su enfoque en componentes reutilizables, eficiencia en la manipulación del DOM y la capacidad de integrarse con diversas tecnologías y herramientas lo hacen ideal para una amplia gama de aplicaciones web y móviles.

2.2.2. Firebase

Firebase [3][4], una plataforma desarrollada por Google, se ha establecido como una herramienta esencial para el desarrollo de aplicaciones web y móviles. Su amplia gama de servicios y funciones la convierte en una opción atractiva para desarrolladores que buscan eficiencia, escalabilidad y facilidad de integración.

Firebase es un servicio en la nube del tipo BaaS, lo que significa "Backend as a Service". Esto implica que proporciona a los desarrolladores de aplicaciones web y móviles una forma de conectar sus aplicaciones a un backend en la nube con servicios integrados como almacenamiento de datos, autenticación de usuarios, notificaciones push, y más, sin necesidad de gestionar la infraestructura subyacente. En esencia, BaaS ofrece un conjunto de herramientas y servicios que simplifican el desarrollo de aplicaciones web y móviles, permitiendo a los desarrolladores centrarse más en la experiencia del usuario y menos en la gestión de servidores y bases de datos.

Historia y Evolución de Firebase:

Firebase comenzó como una base de datos en tiempo real y se ha expandido para incluir una variedad de servicios que facilitan el desarrollo de aplicaciones. Desde su adquisición por Google en 2014, Firebase ha evolucionado para convertirse en una plataforma integral de desarrollo de aplicaciones.

Servicios Clave de Firebase:

- **Base de Datos NoSQL:** Firebase proporciona dos soluciones de base de datos: Realtime Database y Cloud Firestore. Ambas ofrecen la capacidad de sincronizar datos en tiempo real entre los usuarios y el servidor. Se trata de soluciones NoSQL lo que proporciona un coste reducido y una gran escalabilidad.
- **Autenticación:** Firebase Auth ofrece una solución completa para la autenticación de usuarios, soportando una variedad de proveedores como Google, Facebook, Twitter, y más.
- **Hosting:** Proporciona soluciones de alojamiento web estático. Dada las características de la herramienta a desarrollar este tipo de hosting es adecuado.
- **Cloud Storage:** y almacenamiento de archivos en la nube, facilitando la gestión de contenido multimedia y recursos estáticos.
- **Funciones en la Nube y Triggers:** Permite escribir funciones personalizadas que se ejecutan en respuesta a eventos en Firestore, Realtime Database, y otros servicios de Firebase.

- **Analytics y Crashlytics:** Ofrece herramientas integradas para el análisis de datos y el seguimiento de errores, permitiendo a los desarrolladores entender mejor el comportamiento del usuario y mejorar la calidad de la aplicación.

Beneficios conseguidos:

- **Configuración y Manejo:** La integración de Firebase en aplicaciones web y móviles es relativamente sencilla. Proporciona SDKs y bibliotecas para una amplia gama de lenguajes de programación y plataformas.
- **Escalabilidad:** Firebase está diseñado para escalar automáticamente según las necesidades del proyecto, lo que lo hace ideal para aplicaciones en crecimiento.
- **Seguridad:** Ofrece reglas de seguridad configurables para proteger los datos y garantizar que solo los usuarios autorizados tengan acceso.

Al considerar el uso de Firebase, es importante tener en cuenta algunas limitaciones y consideraciones clave. Una de ellas es la dependencia de la plataforma: al optar por Firebase, los desarrolladores se encuentran vinculados a la infraestructura y las limitaciones impuestas por Google, lo que puede influir en aspectos de flexibilidad y control sobre la aplicación. Además, en lo que respecta a los costos y planes de precios, aunque Firebase ofrece un plan gratuito, hay que tener en cuenta que las aplicaciones a gran escala pueden enfrentarse a costos significativos a medida que aumentan el uso de recursos y la demanda de servicios avanzados. Estas consideraciones son cruciales al evaluar Firebase como una solución para proyectos de desarrollo de aplicaciones.

Firebase representa un pilar fundamental en el desarrollo de aplicaciones modernas, proporcionando un conjunto de herramientas que facilitan la creación, gestión y escalabilidad de aplicaciones. Su continua evolución y adaptabilidad lo hacen una opción sólida para desarrolladores que buscan eficiencia y versatilidad en sus proyectos.

2.2.3. Single Page Application

Las Single Page Applications (SPA) han revolucionado el desarrollo web con su enfoque interactivo y dinámico. Una SPA es una aplicación web que carga una sola página HTML y actualiza dinámicamente esa página a medida que el usuario interactúa con la aplicación. Este modelo de diseño web ofrece varias ventajas y desafíos clave.

- **Experiencia de Usuario Mejorada:** Las SPA proporcionan una experiencia de usuario fluida y rápida similar a las aplicaciones de escritorio. Al eliminar la necesidad de recargar toda la página para cada acción, las SPA mejoran la interactividad y reducen el tiempo de espera para los usuarios.
- **Rendimiento Mejorado:** En las SPA, la mayoría de los recursos necesarios (HTML, CSS, Scripts) se cargan una sola vez, durante la visita inicial de la página. Solo se cargan datos específicos en respuesta a las acciones del usuario. Esto significa que menos datos viajan por la red, lo que se traduce en tiempos de carga más rápidos y menos uso de ancho de banda.

- **Desarrollo Simplificado y Mantenimiento:** Las SPA permiten a los desarrolladores escribir menos código y mantenerlo de manera más eficiente. Al separar el backend del frontend, las SPA facilitan el trabajo en paralelo y mejoran la eficiencia del desarrollo.
- **Uso de JavaScript Moderno:** Las SPA suelen depender en gran medida de JavaScript y frameworks modernos como Angular, Vue.js o React. Estos frameworks ofrecen un conjunto de herramientas potentes para crear interfaces de usuario interactivas y dinámicas.
- **SEO y Accesibilidad:** Uno de los desafíos de las SPA es la optimización para motores de búsqueda (SEO). Debido a que su contenido se carga dinámicamente, los motores de búsqueda pueden tener dificultades para indexar el contenido de una SPA. Sin embargo, técnicas como el Server Side Rendering (SSR) y herramientas como el prerendering ayudan a superar este problema.
- **Gestión de Estado:** La gestión del estado de la aplicación es un aspecto crucial en las SPA. El estado debe ser mantenido de manera coherente a medida que el usuario navega a través de la aplicación, lo que puede resultar complejo en aplicaciones de gran tamaño.
- **Seguridad:** Las SPA enfrentan desafíos específicos de seguridad, como la vulnerabilidad a Cross-Site Scripting (XSS). Es crucial adoptar prácticas de desarrollo seguro para mitigar estos riesgos.
- **Dependencia de JavaScript:** Las SPA dependen completamente de JavaScript para cargar y mostrar contenido, lo que puede ser un problema si los usuarios tienen JavaScript deshabilitado en sus navegadores.

2.2.4. Bases de datos NoSQL

Las bases de datos NoSQL, una alternativa a los sistemas de bases de datos relacionales tradicionales, se han popularizado en el mundo del desarrollo debido a su flexibilidad, escalabilidad y rendimiento en el manejo de grandes volúmenes de datos y estructuras de datos no convencionales.

- **Flexibilidad de Esquemas:** A diferencia de las bases de datos relacionales, las NoSQL no requieren un esquema fijo. Esto significa que pueden manejar una variedad de formatos de datos, lo que es ideal para aplicaciones que manejan diferentes tipos de datos o que evolucionan rápidamente.
- **Escalabilidad Horizontal:** Las bases de datos NoSQL están diseñadas para escalar horizontalmente, distribuyendo datos a través de múltiples servidores. Esto las hace ideales para aplicaciones en la nube y aplicaciones que experimentan cargas de trabajo variables o en rápido crecimiento.

- **Diferentes Tipos de Bases de Datos NoSQL:** Existen varios tipos de bases de datos NoSQL, cada una optimizada para diferentes casos de uso:
 - **Bases de Datos de Documentos:** Como MongoDB o Couchbase, son ideales para almacenar y recuperar documentos JSON o XML.
 - **Bases de Datos Clave-Valor:** Como Redis o DynamoDB, son excelentes para almacenar y recuperar pares de clave-valor, útiles para cachés y sesiones de usuario.
 - **Bases de Datos de Columnas Anchas:** Como Cassandra o HBase, son adecuadas para analizar grandes volúmenes de datos.
 - **Bases de Datos de Grafos:** Como Neo4j o Amazon Neptune, están diseñadas para manejar datos relacionales complejos, como redes sociales o sistemas de recomendación.
- **Rendimiento Óptimo para Big Data:** Las bases de datos NoSQL son particularmente adecuadas para Big Data y aplicaciones en tiempo real. Su capacidad para manejar grandes cantidades de datos distribuidos y su flexibilidad para manejar variados tipos de datos las hacen ideales para análisis de datos a gran escala y procesamiento de transacciones de alto volumen.
- **Desafíos de Consistencia y Transacciones:** Mientras que las bases de datos relacionales usan transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) para garantizar la integridad de los datos, muchas bases de datos NoSQL utilizan un modelo de consistencia eventual. Esto puede ser un desafío en aplicaciones que requieren garantías fuertes de consistencia de datos.
- **Complejidad en Consultas y Análisis:** Las consultas en bases de datos NoSQL pueden ser más complejas que en las bases de datos relacionales, especialmente para operaciones de agregación y análisis de datos avanzados. Esto puede requerir una curva de aprendizaje más pronunciada para los desarrolladores acostumbrados a SQL.
- **Consideraciones de Seguridad:** La seguridad es un aspecto crucial en las bases de datos NoSQL. Dado que muchos de estos sistemas fueron diseñados con un enfoque en la escalabilidad y el rendimiento, la seguridad puede no haber sido una prioridad desde el principio, requiriendo una atención especial durante la implementación.

En resumen, las SPA y las bases de datos NoSQL representan tendencias significativas en el desarrollo de aplicaciones web y móviles modernas. Ofrecen ventajas distintas en términos de experiencia del usuario, rendimiento, flexibilidad y escalabilidad, pero también presentan desafíos únicos que deben ser abordados cuidadosamente para aprovechar al máximo sus capacidades.

3. Diseño de la aplicación

3.1. Análisis de requerimientos

En general, un control de asistencia básico solo requiere de dos funciones, poder marcar la asistencia, y poder revisar posteriormente las asistencias. Sin embargo, el proyecto realizado implementa unos cuantos requerimientos más para mejorar la experiencia de usuario, tanto de alumnos como de profesores.

3.1.1. Requisitos funcionales

El propio sistema correspondiente será utilizado por dos tipos de usuarios diferentes, y por lo tanto, tendrá una lista de requerimientos diferente para cada tipo de usuario.

En la sección de profesorado, será necesario organizar los informes de asistencia, para lo cual se divide de la siguiente forma:

- Cada profesor tendrá su propio usuario, a través del cual podrá acceder a sus datos, pero no a los de los demás usuarios
- Los profesores podrán crear cursos, los cuales normalmente se corresponden a las asignaturas para las cuales quieren revisar la asistencia.
- Cada curso tendrá múltiples sesiones, que normalmente corresponderá a clases de teoría o sesiones de práctica de laboratorio.
- Cada curso tendrá una lista de alumnos, que pueden asistir o no a cada una de las sesiones.

En la sección de alumnos, el único objetivo que tenemos es que sea lo más sencillo para ellos marcar la asistencia, ya que todo el tiempo marcando la asistencia es tiempo que se pierde de la sesión docente.

3.1.2. Requisitos no funcionales

Usabilidad:

- El sistema deberá contar con una interfaz intuitiva y fácil de usar para garantizar la accesibilidad a usuarios de diversos niveles de habilidad tecnológica.
- La interfaz de usuario deberá ser responsiva, adaptándose a diferentes dispositivos y tamaños de pantalla.
- El sistema deberá ser compatible con navegadores web comunes, como Chrome, Firefox y Safari, para asegurar una experiencia consistente para los usuarios.

Rendimiento:

- El sistema deberá tener tiempos de respuesta rápidos para garantizar la eficiencia en el registro de asistencia.

- La capacidad del sistema deberá ser escalable para adaptarse al crecimiento del número de usuarios y clases.

Seguridad:

- Se implementarán medidas de seguridad robustas para proteger la integridad de los datos, incluyendo cifrado de datos en tránsito y en reposo.
- El acceso al sistema estará restringido mediante autenticación segura.

Fiabilidad:

- El sistema deberá ser altamente disponible, minimizando el tiempo de inactividad planificado y no planificado.

Compatibilidad:

- El sistema será compatible con sistemas operativos actuales, tanto en entornos de escritorio como móviles.

3.2. Justificación de las tecnologías utilizadas

3.2.1. Backend

Para satisfacer los requisitos de rendimiento, seguridad y fiabilidad, se ha optado por utilizar un backend gestionado externamente. Podemos considerar diferentes alternativas. Mostramos una lista con las más destacables:

- **Amazon Web Services (AWS):** Ofrece servicios escalables en la nube con una amplia gama de herramientas para el desarrollo de aplicaciones.
- **Microsoft Azure:** Plataforma en la nube con servicios integrados para la gestión de aplicaciones, con énfasis en la seguridad y la interoperabilidad.
- **Google Cloud Platform (GCP):** Similar a Firebase (también de Google), proporciona una variedad de servicios en la nube para el almacenamiento y la gestión de aplicaciones.
- **Heroku:** Una plataforma enfocada en la facilidad de uso y la simplicidad, ideal para proyectos más pequeños o startups.
- **Back4App:** Basada en Parse, es una opción para aquellos que buscan una solución de backend simple y de código abierto.
- **DigitalOcean:** Ofrece infraestructura en la nube con una configuración sencilla, ideal para desarrolladores que prefieren un control más directo.

Cada una de estas plataformas tiene sus propias fortalezas y puede ser más adecuada dependiendo de las necesidades específicas del proyecto.

La elección final ha sido Firebase de Google. Responde a los requisitos de la aplicación y proporciona una plataforma robusta y segura sin necesidad de una gestión intensiva por nuestra parte. Además de su simplicidad y facilidad de configuración, Firebase se destaca por sus costos reducidos, lo que lo hace ideal para proyectos que buscan eficiencia en el gasto. Su infraestructura escalable y su amplia gama de servicios integrados

facilitan la implementación y el mantenimiento del sistema, al tiempo que aseguran la disponibilidad y la protección de los datos. Esta combinación de beneficios hace de Firebase una solución óptima para el desarrollo de aplicaciones modernas y eficientes.

En el capítulo de evaluación se analiza la disponibilidad y el coste de la aplicación tras el periodo de prueba. Adelantamos que los resultados son muy positivos.

3.2.2. Frontend

El frontend influye más en los requisitos de usabilidad y compatibilidad. Por compatibilidad, elegimos utilizar un frontend web, y teniendo en cuenta que la diferencias de funcionalidades entre los tres grandes framerowks (Angular, React y Vue) no son demasiado grandes, elegimos React al tener más experiencia con él. Otra alternativa propuesta por el tutor fué usar Flutter, dado que obtiene aplicaciones multiplataforma. Fue descartada por la falta de experiencia en este entorno.

3.3. Arquitectura de la aplicación

3.3.1. Backend

En la sección de backend, al utilizar Firebase, nos conduce a utilizar una arquitectura Backend as a Service. Esta se caracteriza por que el código de la aplicación será ejecutado por el cliente. El backend se encarga de mantener las bases de datos, autenticar a los usuarios y almacenar ficheros. Por lo tanto, no tenemos muchas opciones de cómo configurar la arquitectura. La única opción real que tenemos es la estructura de la base de datos.

Como base de datos, elegimos Firestore Database, tanto por su base de datos en tiempo real, como por la estructura de colecciones dentro de colecciones, que junto al motor de reglas nos permite gestionar la autorización de forma total.

Firestore Database incluye un potente motor de reglas para gestionar la autorización de quien tiene acceso a qué acciones en cada recurso. Con este motor de reglas, restringimos el acceso a toda la sección privada únicamente a aquellos usuarios que tengan acceso.

Aparte de la base de datos, también se han activado en firebase las funcionalidades de autenticación (para gestionar las cuentas de usuario de los profesores) y de hosting (para servir el código del frontend).

3.3.2. Frontend

El frontend está programado utilizando typescript, con React como framework y material-ui como librería de componentes de interfaz de usuario.

El frontend está separado en dos secciones, una privada para los profesores, y una pública para los alumnos.

La sección privada, servida dentro de la subruta de /teacher es solo accesible si el usuario está autenticado. En caso de que no lo esté, se le pedirá que se autentifique.

La sección pública contiene únicamente los cursos que se han creado, y únicamente si tienen una sesión activa. Cualquier persona que acceda puede interactuar aquí sin tener

que autenticarse, para simplificar fricciones que puedan ralentizar a los alumnos y simplificar el proceso.

3.4. Diseño de la interfaz de usuario

La interfaz de usuario está diseñada utilizando el paradigma de mobile-first, priorizando que se visualice correctamente en dispositivos móviles. Además, está diseñada para ser lo más minimalista posible, reduciendo botones o funciones que pudiesen confundir a los usuarios.

3.4.1. Sección pública

La sección pública empieza en la página principal, que contiene la lista de sesiones activas en el momento en el que se ha accedido a la página. Haciendo click en una de estas sesiones, se accede a la sesión específica. Desde aquí, el alumno puede seleccionar su propio nombre y marcarse como que ha asistido a la clase.

3.4.2. Sección privada

La sección privada es algo más compleja, pero sigue un viaje de usuario bastante sencillo;

Primero, el profesor se tiene que autenticar en la página de login, utilizando una cuenta de google, o una cuenta de usuario y contraseña. Toda la gestión de cuentas (almacenamiento de credenciales, reseteo de contraseña...) está gestionada por Firebase.

A continuación, el profesor tiene acceso a su lista de grupos. Desde aquí, puede acceder a un grupo que ya exista, o crear uno nuevo.

Inicio / Teacher LOGOUT

Gestión de grupos

Grupos abiertos

Grupos cerrados

2A_proyecto_P1

2A_proyecto_P2

CREAR NUEVO GRUPO

INSTRUCCIONES DE ESTA PÁGINA:
Un grupo abierto es el que tiene una sesión abierta. Es decir, que permite a los alumnos registrarse. Recuerda cerrar las sesiones una vez pasado el tiempo suficiente para que los alumnos se apunten.
Pulsa sobre uno de los grupos para pasar lista o gestionarlo.
Pulsa sobre AÑADIR GRUPO para crear un nuevo grupo.

INSTRUCCIONES GENERALES DE USOS:

Imagen 1. Gestión de grupos

Inicio / Teacher / create LOGOUT

Crear Grupo

Nombre:
|

AÑADIR

INSTRUCCIONES:
Introduce el nombre del grupo y pulsa en AÑADIR.
Se recomienda usar como nombre de grupo un formato similar a "1A_física_P1". Donde se indica el curso, el cuatrimestre, la asignatura, el tipo (P-prácticas y T-teoría) y el número de grupo si hay más de unos.
Por ejemplo, una asignatura con un grupo de teoría y dos grupos de prácticas tendría que crear los grupos: 1A_física_T, 1A_física_P1 y 1A_física_P2.

Imagen 2. Crear grupo

Desde un grupo individual, el profesor puede ver la lista de sesiones pasadas, o crear una nueva. Además, puede acceder a dos secciones más:

Inicio / Teacher / 2A_proyecto_P1 LOGOUT

Gestión del grupo 2A_proyecto_P1

Sesiones abiertas

Sesiones cerradas

- 02_13-09-21_15:00
- 03_27-09-21_15:00
- 04_08-10-21_09:00
- 05_11-10-21_15:00
- 06_18-10-21_15:00
- 07_22-10-21_09:00
- 08_29-10-21_09:00
- 09_08-11-21_15:00
- 15_15-01-21_09:00
- 16_08-11-21_15:00

11_11-12-23_00:00 AÑADIR SESIÓN

HERRAMIENTAS:

- AÑADIR NUEVOS ESTUDIANTES AL GRUPO
- PERMITIR QUE OTROS PROFESORES GESTIONEN EL GRUPO

INSTRUCCIONES:

Una sesión abierta es la que permite a los alumnos registrarse. Recuerda cerrar las sesiones una vez pasado el tiempo suficiente para que los alumnos se apunten.

Pulsa sobre una de las sesiones para pasar lista o gestionarla.

Para crear una nueva sesión, introduce su nombre en el cuadro de texto y pulsa en "AÑADIR SESION". Se propone un nombre con formato por NÚM.SESIÓN_FECHA_HORA. Pero puedes cambiarlo y usar los nombres de sesiones que prefieras.

Con los botones de herramienta puedes añadir alumnos al grupo o permitir que otros profesores tengan acceso a este grupo.

Imagen 3. Gestión de un grupo individual

- Gestionar la lista de alumnos

Inicio / Teacher / 2A_proyecto_P1 / addstudents LOGOUT

Gestión de alumnos en 2A_proyecto_P1:

Alumnos actuales: (19)

Meseguer Munguía, Daniel

Noguera Belloch, Alejandro Claudio

Pamies Villagordo, Laura

Sandalinas García, Ignacio Miguel

Torres Cruz, Claudia

Tur Rubio, Adrián

Vañó Francés, Àngel

Villanueva Villalon, Ivan

Añadir alumnos:

AÑADIR

INSTRUCCIONES:
Si quieres añadir nuevos alumnos al grupo, escribe sus nombres en el cuadro de texto (uno por línea) y pulsa en AÑADIR.

Para añadir un grupo de Poliformat:

- Entra en Poliformat / asignatura / Calificaciones / Importar/Exportar / Exportación personalizada, selecciona solo "Nombre de estudiante" e indica el "grupo"
- Abre la hoja de cálculo generada y selecciona la primera columna desde la fila 2.
- Pega la selección en el cuadro de texto.

Imagen 4. Gestión de alumnos en un grupo

- Gestionar la lista de profesores

Inicio / Teacher / 2A_proyecto_P1 / addteachers LOGOUT

Gestión de acceso a 2A_proyecto_P1:

Profesores con permiso:

- Pepe Marín
- Daniel Ferri
- Enrique
- Jesús Tomás
- Stella

Profesores sin permiso:

- Asun Perez
- Anna Vidal
- Almanzor Sapena
- macasu

INSTRUCCIONES:
Pulsa sobre un profesor sin permiso para darle permiso. Pulsa sobre uno con permiso para quitárselo.

Imagen 5. Gestión de profesores en un grupo

Desde una sesión individual, el profesor puede activarla (o desactivarla), ver la lista de alumnos (tanto que han asistido como que no) y modificar el estado de los alumnos.

Inicio / Teacher / 2A_proyecto_P1 / 16_08-11-21_15:00 LOGOUT

ABRIR SESIÓN

Alumnos que asisten presencialmente (16)

Larrosa Quesada, Jorge	
Marín-Roig Zozaya, Jorge	
Meseguer Munguía, Daniel	
Noguera Belloch, Alejandro Claudio	
Pamies Villagordo, Laura	
Sandalinas García, Ignacio Miguel	
Torres Cruz, Claudia	
Tur Rubio, Adrián	

Alumnos que asisten online (1)

García Espert, Víctor	
-----------------------	--

Alumnos que no asisten (2)

Vañó Francés, Àngel	+
Villanueva Villalon, Ivan	+

INSTRUCCIONES:
Pulsa en ABRIR SESIÓN para que los alumnos puedan apuntarse a esta sesión. Recuerda cerrar la sesión ("CERRAR SESIÓN") una vez pasado el tiempo suficiente para que los alumnos se apunten.
Los alumnos han de usar el link, <https://servido.web> (actualmente <https://pasar-lista-75130.web.app/>), seleccionar la sesión que acabas de abrir y seleccionar su nombre.
Si algún alumno no dispone de navegador, puedes apuntarlo tú, pulsando en el botón "+". Si algún alumno accede de forma online, él no puede apuntarse a la sesión. Has de apuntarlo tú, pulsando en el botón de la casita. Si algún alumno se apunta por error, usa el botón de la papelera.
IMPORTANTE: Insistir en que los alumnos bajo ningún concepto pueden apuntarse a una sesión si no están físicamente en el aula.

Imagen 6. Visualización de alumnos que han asistido a una sesión

3.5. Diseño de la base de datos

Al utilizar Firestore Database, podemos utilizar colecciones como campos dentro de otras colecciones, lo cual es muy similar a IDs ajenas en una base de datos SQL estándar. Con esto, tenemos dos colecciones en la raíz,

La primera colección contiene los profesores, que identifican los diferentes usuarios del sistema, almacenan información básica de ellos (únicamente el nombre), y tienen la lista de cursos a las que tienen acceso.

La segunda colección contiene los cursos, que contienen la clase en curso actual (si la hubiese), la lista de alumnos del curso en una subcolección (únicamente se guarda el nombre) y la lista de sesiones del curso, la cual es una colección con la fecha y la lista de alumnos que asistieron.

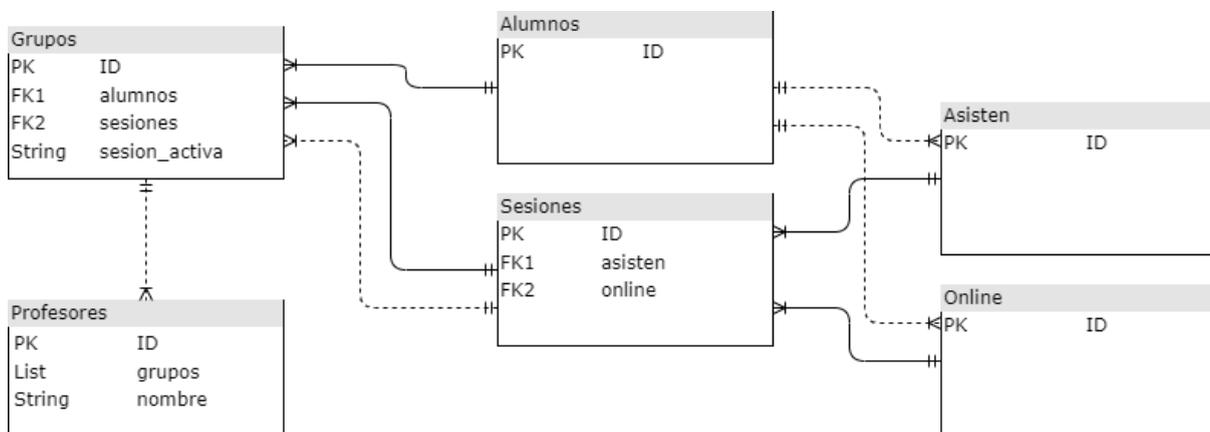


Imagen 7. Esquema de la base de datos

En este esquema, las colecciones más a la izquierda contienen las colecciones a la derecha, indicado por las flechas continuas. Las referencias indirectas (mismo valor, pero sin una conexión directa en la base de datos) están marcadas con líneas discontinuas.

4. Implementación

4.1. Herramientas de trabajo

El desarrollo exitoso del Sistema de Control de Asistencia para Clases ha sido posible gracias al empleo de un conjunto de herramientas efectivas que han facilitado la colaboración, el desarrollo, y la implementación. Estas herramientas no solo han mejorado la eficiencia del desarrollo, sino que también han asegurado la calidad y la escalabilidad del proyecto. Las siguientes herramientas han desempeñado un papel fundamental en cada fase del ciclo de vida del proyecto:

- **Git:** Git ha sido la columna vertebral del control de versiones en el proyecto, permitiendo un seguimiento preciso de los cambios en el código fuente. Esta herramienta ha facilitado la colaboración con el tutor, permitiendo trabajar en distintas ramas y fusionar cambios de manera eficiente.
- **Visual Studio Code:** Visual Studio Code ha sido el entorno de desarrollo integrado (IDE) principal utilizado para la codificación y edición de archivos del proyecto. Su extensibilidad y soporte para múltiples lenguajes de programación lo han convertido en una elección versátil y potente, proporcionando funcionalidades como depuración de código, gestión de versiones Git integrada y una gran variedad de extensiones que mejoran la productividad.
- **Yarn:** Se trata de una herramienta de gestión de paquetes para la administración de dependencias y la instalación de módulos de terceros en el entorno de desarrollo. Su velocidad y consistencia han optimizado el flujo de trabajo del equipo.
- **Firebase Tools:** Firebase Tools ha sido utilizado para implementar y gestionar servicios de Firebase, como la base de datos en tiempo real, la autenticación y el hosting web. Su integración con el entorno de desarrollo ha permitido un despliegue rápido y eficiente.

4.2. Estructura de la aplicación

El código de la aplicación está separado en diferentes carpetas y ficheros que facilitan la correcta organización del código. En este apartado vamos a hacer una descripción de cómo se ha realizado esta organización. Podemos dividir el código en dos secciones, los ficheros de configuración y el código fuente de la aplicación.

4.2.1. Ficheros de Configuración

La mayoría de los ficheros de esta sección se encuentran en la raíz del proyecto y su finalidad es establecer parámetros y directrices esenciales para el funcionamiento y la estructura del mismo. Estos archivos definen desde las dependencias y herramientas hasta las reglas de seguridad y las configuraciones específicas de servicios. Según su finalidad, podemos agruparlos en:

Configuración de Firebase:

- **firebase.json:** Archivo de configuración principal para proyectos Firebase, que define las reglas y configuraciones para los servicios como Hosting, Functions, y Firestore.
- **indexes.json:** Especifica los índices personalizados para la base de datos Firestore, optimizando las consultas y el rendimiento.
- **firestore.rules:** Contiene las reglas de seguridad para el acceso a la base de datos Firestore, definiendo quién puede leer o escribir en las colecciones y documentos.
- **src/fire.ts:** Este archivo contiene la inicialización y configuración del SDK de Firebase para su uso en el proyecto, incluyendo la autenticación y la referencia a la base de datos Firestore.
- **.firebase.rc:** Un archivo de configuración oculto que almacena metadatos y configuraciones locales específicas del proyecto Firebase.

Configuración del proyecto y dependencias:

- **package.json:** Define las dependencias, scripts, y metadatos generales del proyecto JavaScript o Node.js, como el nombre, versión y licencia.
- **yarn.lock:** Archivo generado por Yarn para asegurar que las dependencias del proyecto se instalan exactamente igual en cada entorno, almacenando las versiones específicas de cada paquete.

Configuración de TypeScript:

- **tsconfig.json:** Configura opciones para el compilador de TypeScript, definiendo cómo se compilan los archivos .ts a .js, incluyendo reglas sobre cómo tratar el código, la especificación del directorio de salida, entre otros.

4.2.2. Código fuente

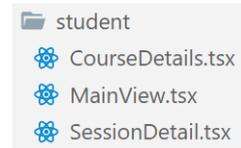
Los ficheros de código fuente, ubicados principalmente en la carpeta src, contienen las diversas funciones y componentes desarrollados para la aplicación. Esta carpeta está organizada en subdirectorios para facilitar la gestión del código según el rol del usuario en la aplicación. Las carpetas más destacadas son 'student', 'teacher', y 'common', cada una albergando elementos específicos para las necesidades y características de esos roles. A continuación, se proporciona un detalle de los ficheros y carpetas más significativos:

Base de la aplicación y router:

- **index.tsx:** Punto de entrada de la aplicación React, encargado de renderizar la aplicación en el DOM y a menudo de configurar proveedores globales como Redux o Context API.
- **App.tsx:** Componente principal de React que define la estructura básica de la aplicación y el enrutamiento entre diferentes vistas o componentes.

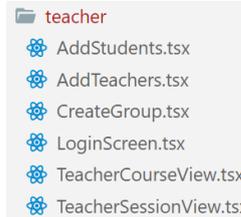
Vistas de la Sección de Estudiante:

- **Carpeta student:** Contiene componentes y vistas específicos para la sección del estudiante, como listados de cursos, perfiles de estudiantes.



Vistas de la Sección de Profesor:

- **Carpeta teacher:** Alberga las vistas y componentes destinados a la sección del profesor, incluyendo la gestión de cursos y listados de estudiantes.



Componentes Comunes:

- **Carpeta common:** Incluye componentes reutilizables en toda la aplicación, como botones, formularios, y elementos de navegación, facilitando la coherencia y mantenimiento del diseño.



Definición de Tipos:

- **Carpeta types:** Contiene archivos de definición de tipos para TypeScript, utilizados para estandarizar y validar la estructura de datos a lo largo del proyecto.

Funciones de Utilidad:

- **Carpeta utils:** Agrupa funciones de utilidad y ayuda que se utilizan en varias partes de la aplicación, como funciones de formato, validación o cálculos comunes.

4.3. Seguridad y autenticación

4.3.1. Autenticación

La autenticación está toda gestionada por Firebase. De esa forma, nos aseguramos de que las credenciales se almacenen de forma segura, y además firebase gestiona todos los procesos de gestión de cuenta, como la creación y la recuperación de contraseña.

Se han habilitado dos métodos de autenticación en Firebase. El primero es mediante cuentas de Google, y es el preferido. De esta forma, el usuario no tiene que gestionar una contraseña extra, sino que utiliza su cuenta de Google ya existente. El segundo método es directamente con email y contraseña. Este método es más incómodo para los usuarios, pero se permite para cualquier usuario que por cualquier razón no quiera utilizar una cuenta de Google.

4.3.2. Autorización

Para la autorización, se utiliza el motor de reglas de Firestore Database. Este motor nos permite definir reglas que se evaluarán cada vez que se quiere realizar cada una de las operaciones disponibles sobre la base de datos, y permite definir bajo qué condiciones se permitirán o rechazarán estas operaciones. Las reglas se almacenan en el mismo repositorio git que el resto del código, dentro del fichero firestore.rules, y se explican a continuación:

Cualquier usuario, autenticado o no, puede leer la lista de grupos que existen:

```
//Un alumno puede ver la lista de grupos
match /grupos/{grupo} {
  allow read;
}
```

Cualquier usuario, autenticado o no, puede leer la lista de sesiones de cualquier grupo que exista:

```
//Un alumno puede ver la lista de sesiones
match /grupos/{grupo}/sesiones/{sesion} {
  allow read;
}
```

Cualquier usuario, autenticado o no, puede leer la lista de alumnos de cualquier grupo que exista:

```
//Un alumno puede ver la lista de alumnos
match /grupos/{grupo}/alumnos/{alumno} {
  allow read;
}
```

Cualquier usuario, autenticado o no, puede añadir elementos a la lista de asistencia (pero no eliminarlos). Solo alumnos que pertenezcan a la lista de alumnos pueden ser creados

```
//Un alumno puede añadir a la lista de asistencia
match /grupos/{grupo}/sesiones/{sesion}/asisten/{nombre} {
  allow create: if
exists(/databases/{database}/documents/grupos/{grupo}/alumnos/{nombre});
}
```

Un usuario autenticado (profesor) puede leer y escribir cualquier documento que desee dentro de un grupo al que pertenezca:

```
//Un profesor puede leer y escribir cualquier documento de sus grupos
match /grupos/{grupo}/{rest=**} {
```

```
    allow read, write: if request.auth != null && grupo in
get(/databases/$(database)/documents/profesores/$(request.auth.uid)).data.gr
upos;
}
```

Un usuario autenticado (profesor) puede crear grupos nuevos si su objeto en la lista de profesores existe:

```
//Un profesor puede crear asignaturas nuevas
match /grupos/{grupo} {
    allow create: if request.auth != null
    &&
exists(/databases/$(database)/documents/profesores/$(request.auth.uid));
}
```

Un usuario autenticado (profesor) puede leer la información de cualquier otro usuario autenticado, y puede modificar su propia entrada de usuario (pero no ajenas) para asignarse grupos nuevos:

```
//Un profesor puede asignarse como profesor de una asignatura
match /profesores/{uid} {
    allow read: if request.auth != null; //NUEVA
    allow write: if request.auth != null && uid == request.auth.uid;
}
```

4.3.3. Seguridad

En cuanto a los datos en tránsito, se ha implementado una capa adicional de seguridad mediante el protocolo HTTPS [14][15] (Hypertext Transfer Protocol Secure). La encriptación de datos en tránsito, respaldada por el estándar de seguridad Transport Layer Security [14][16] (TLS), asegura que toda la comunicación entre el cliente y el servidor esté protegida. Esta medida garantiza que toda la información se transmita de manera segura y no pueda ser interceptada o alterada durante la transmisión.

Para abordar la seguridad de los datos en reposo, la robusta encriptación proporcionada por el servidor de Firebase es AES de 256 bits [14][17]. Los datos almacenados en la base de datos en tiempo real de Firebase se encuentran cifrados utilizando este algoritmo de cifrado avanzado. Esto significa que, incluso en el improbable caso de acceso no autorizado a la base de datos, los datos permanecen ilegibles sin la clave de cifrado correspondiente.

5. Manuales

5.1. Manual de instalación

En esta sección definiremos cómo trabajar con el repositorio para desplegar la aplicación desde cero.

5.1.1. Instalación de requerimientos

El primer paso será instalar las herramientas necesarias. Para instalar estas herramientas, será necesario seguir los pasos indicados en las páginas oficiales de instrucciones, que dependen del sistema operativo. Este es el único paso que depende del sistema operativo donde se vaya a compilar y desplegar la aplicación. Estas son:

- git (<https://git-scm.com/downloads>) para descargar el repositorio con todo el código
- node (<https://nodejs.org/en/download/>) para ejecutar todo el resto de herramientas que serán necesarias para compilar y desplegar la aplicación

A continuación, se instalarán las herramientas de node necesarias. Para instalarlas, es necesario ejecutar los siguientes comandos:

- yarn (`npm install --global yarn`) Esta herramienta es una alternativa a npm, que se ha utilizado en el proyecto ya que mejora la experiencia de desarrollo.
- Firebase tools (`npm install -global firebase-tools`) Esta herramienta será la encargada de gestionar la conexión a Firebase y el proceso de despliegue.

5.1.2. Creación del proyecto de Firebase

El segundo paso consistirá en crear un proyecto de Firebase que realizará la función de todo el backend, incluyendo la base de datos, autenticación y el hosting de la web.

Para ello, será necesario ir a la consola de Firebase (<https://console.firebase.google.com/>) y seleccionar “Crear un proyecto”:

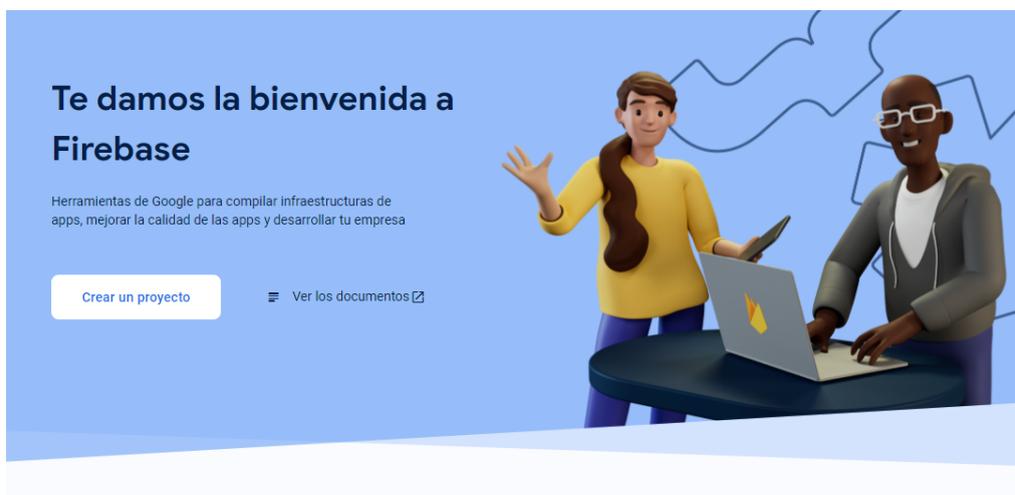


Imagen 8. Crear un proyecto en Firebase

En la siguiente ventana, dale un nombre al proyecto. Este nombre será interno, pero podemos también elegir el ID del proyecto, que se utilizará en la url por defecto de la página web donde se servirá la aplicación. Esta ID es el único dato que no se puede modificar posteriormente



X Crear un proyecto(paso 1 de 3)

Comencemos con el nombre de tu proyecto[?]

Nombre del proyecto

pasarlista

pasarlistaupv

Continuar

Imagen 9. Nombrar un proyecto en Firebase

En el siguiente paso, podemos activar o no Google Analytics. En nuestro caso, lo hemos desactivado para simplificar el proceso de instalación. Hacemos click en Crear proyecto:

Google Analytics para tu proyecto de Firebase

Google Analytics es una solución de analítica ilimitada y gratuita que permite usar la segmentación, los informes y otras funciones en Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing y Cloud Functions.

Google Analytics habilita las siguientes funciones:

- × Pruebas A/B [?]
- × Segmentación de usuarios y orientación a ellos en los productos de Firebase [?]
- × Usuarios que no experimentan fallas [?]
- × Activadores de Cloud Functions basados en eventos [?]
- × Informes ilimitados y gratuitos [?]

Habilitar Google Analytics para este proyecto
Recomendado

[Anterior](#)

[Crear proyecto](#)

Imagen 10. Configurando Google Analytics para el proyecto

Una vez creado, tendremos que activar las funcionalidades que necesitamos. Para este proyecto, será Authentication, Firestore Database y Hosting.

Primero, accedemos a la sección de Authentication en el menú lateral izquierdo, y hacemos click en Comenzar:



Imagen 11. Activando autenticación

Nos pedirá que seleccionemos el primer método de acceso. Seleccionamos “Correo electrónico/contraseña”:

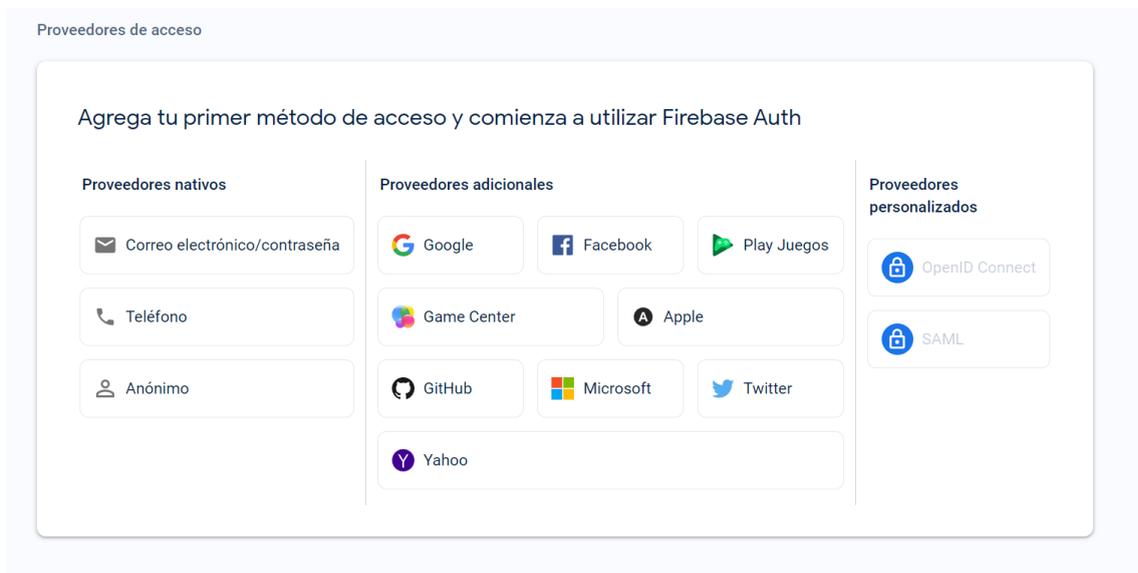


Imagen 12. Lista de métodos de autenticación

Aquí, simplemente lo habilitamos y hacemos click en Guardar:



Imagen 13. Activando autenticación por email y contraseña

Podemos ver que ya está habilitado. A continuación, hacemos click en “Agregar proveedor nuevo”, y seleccionamos Google esta vez:

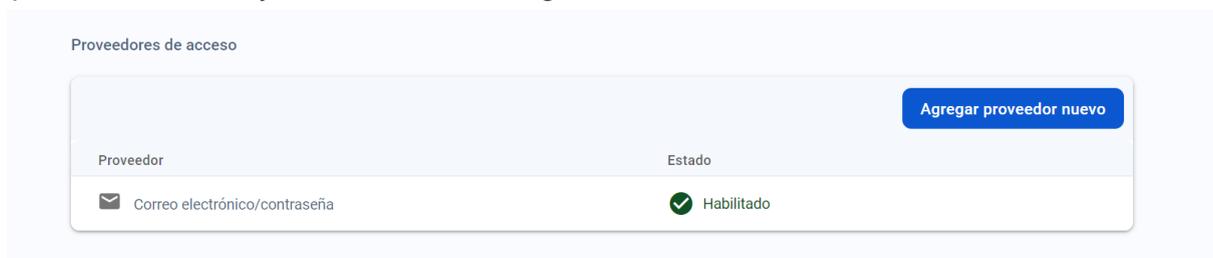


Imagen 14. Agregando un segundo método de autenticación

Aquí tenemos que añadir el nombre público del proyecto, que se visualizará cuando un usuario quiera autenticarse mediante google. También tenemos que añadir un email de asistencia, donde los usuarios nos pueden contactar si tienen problemas.

Proveedores de acceso

Agregar proveedor nuevo

Proveedor Estado

Configura el proveedor (paso 2 de 2)

Google Habilitar

Importante: Para habilitar el Acceso con Google en tus apps para Android, debes proporcionar la [huella digital de lanzamiento SHA-1](#) para cada app (ve a [Configuración del proyecto](#) > la sección *Tus apps*).

Actualiza la siguiente configuración a nivel de proyecto para continuar

Nombre público del proyecto

Correo electrónico de asistencia del proyecto

Agrega a la lista de entidades seguras los ID de cliente de proyectos externos (opcional)

Configuración del SDK web

Cancelar Guardar

Correo electrónico/contraseña Habilitado

Imagen 15. Activando autenticación mediante Google

Finalmente, ya tenemos ambos métodos de autenticación activados

Proveedores de acceso

Agregar proveedor nuevo

Proveedor	Estado
Correo electrónico/contraseña	✓ Habilitado
Google	✓ Habilitado

Imagen 16. Múltiples autenticaciones activadas

A continuación, tendremos que activar Firestore Database. Las seleccionamos en el menú lateral izquierdo, y hacemos click en “Crear base de datos”:



Imagen 17. Activando Firebase Datastore

En la ventana emergente, seleccionamos la ubicación donde se almacenarán los datos. Aquí seleccionaremos una ubicación lo más cercana a la mayoría de nuestros usuarios. Esta selección también afectará a los costes de la base de datos según la lista de precios de Firestore, pero como veremos en el capítulo de resultados, el coste es mínimo en cualquier caso:

A screenshot of the 'Crear base de datos' (Create database) dialog box. The title bar shows 'Crear base de datos' and a close button. Below the title bar, there are two steps: '1 Establece el nombre y la ubicación' (selected) and '2 Reglas de seguridad'. The 'ID de la base de datos' field contains '(default)'. The 'Ubicación' dropdown menu is set to 'eur3 (Europe)'. Below the dropdown, there is an information icon and the text: 'La configuración de ubicación es el lugar donde se almacenarán tus datos de Cloud Firestore'. A warning box with an exclamation mark icon contains the text: 'No podrás cambiar la ubicación después de configurarla. Además, la configuración de ubicación será la de tu bucket predeterminado de Cloud Storage.' and a link for 'Más información'. At the bottom, there is a note: 'Si habilitas Cloud Firestore, no podrás usar Cloud Datastore en este proyecto', and two buttons: 'Cancelar' and 'Siguiente'.

Imagen 18. Eligiendo ubicación de la base de datos

La siguiente ventana nos pregunta la lista de reglas por defecto. Esto es irrelevante para nosotros, porque desplegaremos las nuestras propias que sobrescribirán estos valores. Dejamos el modo de producción seleccionado por defecto y hacemos click en habilitar:

Crear base de datos

1 Establece el nombre y la ubicación — 2 Reglas de seguridad

Después de definir la estructura de tus datos, debes crear reglas para protegerlos.

[Más información](#)

Iniciar en modo de producción

Tus datos son privados de forma predeterminada. El acceso de lectura/escritura de los clientes solo se otorgará como se indica en tus reglas de seguridad.

Comenzar en modo de prueba

Para permitir una configuración rápida, los datos se abren de forma predeterminada. Sin embargo, debes actualizar las reglas de seguridad en un plazo de 30 días a fin de habilitar el acceso de lectura/escritura a largo plazo para los clientes.

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if false;
    }
  }
}
```

i Se denegarán todas las operaciones de lectura y escritura de terceros.

Si habilitas Cloud Firestore, no podrás usar Cloud Datastore en este proyecto

Cancelar

Habilitar

Imagen 19. Eligiendo reglas por defecto

A continuación, tendremos que seleccionar Hosting en el menú lateral izquierdo, y hacer click en el botón de “Comenzar”:



Imagen 20. Activando Hosting

En el primer paso, marcaremos la opción de “Mostrarme también los pasos para agregar el SDK de Firebase” y hacemos click en “Siguiente”:

× Configurar Firebase Hosting

- 1 Instala Firebase CLI**

Necesitas Firebase CLI (una herramienta de línea de comandos) para alojar tu sitio con Firebase Hosting.

Ejecuta el siguiente comando de [npm](#) para instalar la CLI o actualizar a su versión más reciente.

```
$ npm install -g firebase-tools
```

¿No funciona? Consulta la [referencia de Firebase CLI](#) o cambia tus [permisos de npm](#)

Mostrarme también los pasos para agregar el SDK de Firebase JavaScript a mi app web
El SDK incluye Cloud Firestore, Authentication, Performance Monitoring y mucho más.
Puedes agregarlo ahora o más adelante.

Siguiente
- 2 Inicializa el proyecto**
- 3 Registra tu app**
- 4 Agrega el SDK de Firebase**
- 5 Implementa en Firebase Hosting**

Imagen 21. Hosting paso 1

Seleccionamos “Siguiente” también en la sección 2, y le damos un nombre a la app web. Este nombre es solo interno y no lo verán los usuarios. Hacemos click en “Registrar y continuar”:

× Configurar Firebase Hosting

✓ Instala Firebase CLI

✓ Inicializa el proyecto

3 Registra tu app

Si planeas usar productos de Firebase, como Cloud Firestore, Authentication o Performance Monitoring, debes registrar tu app web. Para ello, asígnale un sobrenombre:

Registrar tu app web

Sobrenombre de la app ⓘ

Registrar y continuar

4 Agrega el SDK de Firebase

✓ Implementa en Firebase Hosting

Imagen 22. Hosting paso 3

Hacemos click en **Siguiente** también en la sección 4. No es necesario que guardemos nada de aquí ya que lo recuperaremos más tarde:

× Configurar Firebase Hosting

✓ Instala Firebase CLI

✓ Inicializa el proyecto

✓ Registra tu app

4 Agrega el SDK de Firebase

Usar npm Usar una etiqueta <script>

Si ya usas [npm](#) y un agrupador de módulos como [Webpack](#) o [Rollup](#), puedes ejecutar el siguiente comando para instalar la versión más reciente del SDK ([más información](#)):

```
$ npm install firebase
```

Luego, inicializa Firebase y comienza a usar los SDK de los productos que quieres utilizar.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyA4-B7cv9tKdFdI03EBx1cpimBUFidNuN8",
  authDomain: "pasarlistaupv.firebaseio.com",
  projectId: "pasarlistaupv",
  storageBucket: "pasarlistaupv.appspot.com",
  messagingSenderId: "861482956786",
  appId: "1:861482956786:web:bb89c77042bb30c4039d6b"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Nota: Esta opción utiliza el [SDK de JavaScript modular](#), que proporciona un tamaño reducido del SDK.

Obtén más información sobre Firebase para la Web: [primeros pasos](#), [referencia de la API del SDK web](#) y [muestras](#)

Siguiente

✓ Implementa en Firebase Hosting

Imagen 23. Hosting paso 4

Con esto, ya tenemos todo el proyecto de Firebase creado y configurado.

Sin embargo, necesitamos también los parámetros de configuración del proyecto. Para recuperarlos, hacemos click en el engranaje en la esquina superior izquierda, y en el menú que se abre, seleccionamos “Configuración del proyecto”:



Imagen 24. Abriendo configuración del proyecto

Aquí, navegamos hacia abajo, y en la sección de apps, dentro de la configuración, seleccionamos “Config” y copiamos el contenido para más tarde.

Configuración del SDK

npm CDN Config

Obtén el fragmento para el objeto de configuración de Firebase de tu app.

[Obtén más información](#) .

El siguiente objeto de configuración de Firebase contiene identificadores y claves para tu app:

```
const firebaseConfig = {
  apiKey: "AIzaSyA4-B7cv9tKDfdI03EBx1cpimBUFidNuN8",
  authDomain: "pasarlistaupv.firebaseio.com",
  projectId: "pasarlistaupv",
  storageBucket: "pasarlistaupv.appspot.com",
  messagingSenderId: "861482956786",
  appId: "1:861482956786:web:bb89c77042bb30c4039d6b"
};
```



Imagen 25. Configuración de Firebase

5.1.3. Clonado del repositorio, configuración y despliegue

Finalmente, tendremos que clonar el repositorio, añadir nuestra configuración y desplegarlo.

Para ello, en una línea de comandos, primero clonamos el repositorio:

```
git clone git@github.com:Daniferrito/attendance.git
```

A continuación, accedemos a la carpeta que acabamos de clonar:

```
cd attendance
```

Instalamos las dependencias de nuestro proyecto:

```
yarn install
```

Ahora será necesario añadir la configuración de nuestro proyecto. Para ello, modificaremos dos ficheros, añadiendo la información que hemos recuperado en el último paso de la sección anterior.

Modificamos el fichero `.firebaseerc`, y sustituimos la `id` por la `id` de nuestro proyecto (el campo `projectId` de la configuración). De esta forma, el fichero quedará así:

```
{
  "projects": {
    "default": "pasarlistaupv"
  }
}
```

A continuación, modificamos el fichero `src/fire.ts`, sustituyendo el valor de la variable `config` por el valor de la configuración que hemos obtenido en la sección anterior. El fichero quedará así:

```
import firebase from "firebase/app";
import "firebase/firestore";
import "firebase/auth";
var config = {
  apiKey: "AIzaSyA4-B7cv9tKDfdI03EBx1cpimBUFidNuN8",
  authDomain: "pasarlistaupv.firebaseio.com",
  projectId: "pasarlistaupv",
  storageBucket: "pasarlistaupv.appspot.com",
  messagingSenderId: "861482956786",
  appId: "1:861482956786:web:bb89c77042bb30c4039d6b"
};
var fire = firebase.initializeApp(config);
export default fire;
```

Con esto, ya tenemos todo configurado, y podemos proceder a desplegar la aplicación. Para ello simplemente será necesario autenticarse en la misma línea de comandos de antes con `firebase` mediante el siguiente comando:

```
firebase login
```

Y finalmente desplegar:

```
yarn run publish
```

Este comando compilará todo el proyecto, generando el bundle necesario para desplegar la aplicación, y desplegará la aplicación web. También subirá las reglas definidas en firestore.rules a firestore database, sobrescribiendo lo que hayamos definido en el paso de reglas iniciales de la sección anterior.

Con todo esto, ya podemos acceder a la aplicación web y utilizarla mediante la url que podemos ver dentro de la sección de Hosting.

5.2. Manual de uso

5.2.1. Instrucciones de Gestión:

Registro de un Nuevo Profesor:

- Ingresa a <https://pasar-lista-75130.web.app/teacher>
- Crea un nuevo usuario con correo y contraseña de Google.
- Notifica a jtomas@upv.es para activar el nuevo usuario.

Registro de Grupos:

- Para cada asignatura, crea varios grupos, típicamente uno de teoría y dos o tres de prácticas o seminarios.
- Utiliza el formato "1A_física_P1" para nombrar los grupos, indicando curso, cuatrimestre, asignatura, tipo (P-prácticas y T-teoría) y número de grupo si hay más de uno.
- Accede a <https://pasar-lista-75130.web.app/teacher> y haz clic en "CREAR NUEVO GRUPO". Introduce un nombre adecuado según las instrucciones y añade los alumnos del grupo.

Añadir Alumnos de Poliformat a un Grupo:

- En Poliformat, ve a asignatura / Calificaciones / Importar/Exportar / Exportación personalizada. Selecciona solo "Nombre de estudiante" e indica el "grupo".
- Abre la hoja de cálculo generada y selecciona la primera columna desde la fila 2.
- Pega la selección en el cuadro de texto. Si varios profesores dan clase en este grupo, dales acceso seleccionando el grupo, haz clic en "PERMITIR QUE OTROS PROFESORES GESTIONEN EL GRUPO" y sigue las instrucciones.

5.2.2. Instrucciones para el Profesor:

Para Controlar la asistencia en una clase:

- Accede al enlace: <https://pasar-lista-75130.web.app/teacher>
- Selecciona uno de tus grupos.
- Crea una nueva sesión, proporciona un nombre que puedas cambiar más tarde. Haz clic en "AÑADIR SESIÓN" y, luego, selecciona la sesión añadida.
- Activa el pase de lista haciendo clic en "ABRIR SESIÓN".
- Verifica que el número de alumnos que han registrado asistencia coincida con los presentes en clase.
- Detén el pase de lista haciendo clic en "CERRAR SESIÓN".

- Si algún alumno no tiene móvil o portátil, agrégalo usando el botón "+".
- Si algún alumno asiste desde casa, agrégalo utilizando el botón "casita".

Para visualizar el listado de alumnos que han asistido a una clase:

- Accede al enlace: <https://pasar-lista-75130.web.app/teacher>
- Selecciona uno de tus grupos
- Selecciona una de las sesiones del grupo
- En la página, se pueden ver los alumnos separados en tres listas, alumnos que asistieron presencialmente, alumnos que asistieron desde casa, y alumnos que no asistieron

5.2.3. Instrucciones para el Alumno:

- Accede al enlace: <https://pasar-lista-75130.web.app/>.
- Verifica que la sesión de clase sea la correcta y selecciónala.
- Elige tu nombre y haz clic en el botón. Para futuros accesos, el sistema recordará tu elección.

IMPORTANTE: Sólo puedes hacer clic en el botón si estás físicamente en el aula.

6. Resultados y discusión

6.1. Evaluación de los resultados

La aplicación fue puesta a prueba en un entorno real de trabajo en la EPSG desde septiembre de 2021. Se utilizó en asignaturas diversas como Teoría de la Comunicación, CDIO, Matemáticas 1, Matemáticas para el Turismo, Proyecto de Internet de las Cosas y Desarrollo de aplicaciones móviles. Se formaron diferentes grupos en cada asignatura, sumando un total de 12. En uno de los grupos, se realizó control de asistencia en un máximo de 20 sesiones. En total, 362 alumnos utilizaron la herramienta y 12 profesores participaron en su gestión.

Tras este periodo de pruebas con profesores y alumnos reales, podemos concluir que su uso ha sido generalmente un éxito. El control de asistencia se realiza muy rápidamente, y los alumnos generalmente no encuentran problema alguno.

Por otro lado, la sección de profesores ha resultado ser ligeramente más confusa. La arquitectura de cursos y sesiones no era totalmente intuitiva, y el concepto de sesión activa confunde a algunos profesores. Se ha subsanado esto creando manuales de uso para los profesores.

También podemos hacer una evaluación de la aplicación desde un punto de vista más técnico. El peso más importante va a recaer sobre los servicios en la nube que ofrece Firebase. En este sentido podemos evaluar tres aspectos: seguridad, disponibilidad y cosmos:

La **seguridad** ha sido descrita en el capítulo 4. Se concluye que la aplicación consigue unos estándares satisfactorios. Respecto a la **disponibilidad** destacar que nunca se ha detectado denegaciones de servicio o caídas del sistema.

Finalmente pasemos a analizar el **coste**. Firebase ofrece muchos de sus servicios de forma gratuita, como el servicio de autenticación. El servicio de Hosting ofrece un almacenamiento gratuito de hasta 10Gb y una transferencia diaria suficiente para los requisitos de la aplicación. El mayor coste podría estar en el uso de la base de datos Firestore. Se ha realizado un cálculo que indica que haríamos un uso gratuito para aproximadamente los primeros 10.000 asistencias de alumnos al día. En caso de superarse este volumen de uso el coste extra sería de tan solo 0,03€ por cada 10000 asistencias extra. El coste por cada uso de la aplicación por un alumno son aproximadamente 0,000003€. En ningún momento de las pruebas se ha llegado a incurrir en costes.

Como conclusión, el coste de usar la instalación sería de 0€ para menos de 10.000 usos diarios. Esto permite que cualquier centro educativo pueda utilizar la herramienta de forma totalmente gratuita. No tendría más que seguir el manual de instalación descrito en el capítulo 5.

6.2. Comparación con soluciones existentes

Una distinción clave de nuestro sistema respecto a las soluciones existentes radica en que no se limita exclusivamente a depender del profesor para el registro de asistencia,

aunque esta sigue siendo una opción disponible. La flexibilidad de nuestra solución permite que los alumnos se autoregistro, posibilitando que el profesor se dedique a preparar el material educativo mientras los estudiantes confirman su asistencia. Además, al permitir que todos los alumnos marquen su presencia simultáneamente, el proceso se lleva a cabo de manera completamente paralela, reduciendo significativamente el tiempo perdido en comparación con las soluciones convencionales.

Sin embargo, es importante señalar que, a diferencia de algunos otros sistemas de control de asistencia, nuestra solución no incluye funciones avanzadas de reportes. Mientras que otros sistemas pueden ofrecer análisis más detallados y herramientas de visualización de datos, nuestra prioridad ha sido optimizar la eficiencia del proceso de registro de asistencia. Reconocemos que existen soluciones con capacidades analíticas más robustas, y estamos abiertos a considerar la incorporación de funciones avanzadas en futuras actualizaciones para ofrecer una experiencia aún más completa.

7. Conclusiones y trabajos futuros

7.1. Conclusiones

El desarrollo del Sistema de Control de Asistencia para Clases representa un paso significativo hacia la mejora de la gestión y seguimiento de la asistencia en entornos educativos. Tras un exhaustivo proceso de planificación, diseño e implementación, se pueden extraer las siguientes conclusiones clave:

- **Satisfacción de Requisitos Funcionales:**

El sistema ha logrado satisfacer de manera efectiva los requisitos funcionales establecidos, proporcionando una plataforma integral para el registro preciso y eficiente de la asistencia en clases.

- **Interfaz de Usuario Intuitiva:**

La atención dedicada a la usabilidad se refleja en una interfaz de usuario intuitiva y fácil de usar. La retroalimentación positiva de los usuarios durante las pruebas sugiere que el sistema cumple con su objetivo de ser accesible para usuarios con diversos niveles de familiaridad con la tecnología.

- **Fiabilidad y Disponibilidad:**

La arquitectura del sistema ha demostrado ser altamente fiable, no habiendo dejado de estar disponible en ningún momento, gracias a utilizar los servidores de Firebase con su gran sistema de alta disponibilidad ha evitado que el sistema deje de responder.

En resumen, el Sistema de Control de Asistencia para Clases ha superado las expectativas al proporcionar una solución integral, eficiente y segura para la gestión de la asistencia en entornos educativos. Las lecciones aprendidas durante el desarrollo serán fundamentales para futuras mejoras y expansiones del sistema, asegurando su relevancia y utilidad a largo plazo. La colaboración continua con usuarios y la atención a las tendencias tecnológicas serán esenciales para mantener el sistema en la vanguardia de las soluciones de control de asistencia.

Sin embargo, también cabe destacar algún aspecto negativo como son la necesidad de mejorar la experiencia de usuario en la sección de profesor, como se ha explorado en el capítulo anterior.

7.2. Trabajos futuros

El Sistema de Control de Asistencia para Clases proporciona una base sólida para el seguimiento eficiente de la asistencia en entornos educativos. Sin embargo, para asegurar su continua relevancia y mejora, se identifican áreas específicas para futuros desarrollos y expansiones:

- **Funcionalidades Avanzadas de Reportes:** Explorar la implementación de funciones avanzadas de generación de informes, como análisis de tendencias de asistencia, correlación con el rendimiento académico y visualización gráfica de

datos. Esto proporciona a los profesores y administradores una visión más detallada y analítica de la asistencia.

- **Integración con Plataformas Educativas Existentes:** Mejorar la integración con plataformas educativas existentes, específicamente Poliformat, pero también plataformas abiertas como Moodle o Canvas, para garantizar una experiencia más cohesionada y simplificar la gestión de datos académicos y de asistencia.
- **Mejora de la interfaz de usuario en la sección de profesor:** Algunas de las interacciones en la sección de profesor no son totalmente claras, y la interfaz de usuario (y especialmente la experiencia de usuario) podría mejorarse con otra pasada de mejoras.
- **Optimización de la base de datos:** Optimizar la base de datos reduciría los costes en aproximadamente entre un orden de magnitud y uno y medio. Por como los costes se calculan en Firebase, la estructura actual podría ser mejorada condensando algunas colecciones como valores dentro de la colección que las contiene en vez de colecciones independientes

Referencias bibliográficas

[1] Instituto Nacional de Evaluación Educativa (2014) Hacer novillos y Pisa 2012: El caso de las CC. AA. Españolas

<https://www.educacionyfp.gob.es/dctm/inee/boletines/especialccaa/educaineautonomias1.pdf?documentId=0901e72b8190a44e>

[2] Hoofman, Jacob; Secord, Elizabeth (2021) The Effect of COVID-19 on Education. Pediatric Clinics of North America, vol. 68, no. 5, octubre, pp. 1071-1079. DOI: 10.1016/j.pcl.2021.05.009.

[3] Carbonell, Vicente; Bataller, Jordi; Tomás, Jesús; Lloret, Jaime (2019) Firebase: trabajar en la nube. MARCOMBO S.A. ISBN: 9788426726605

[4] (2023) Página oficial de Firebase.

<https://firebase.google.com/>

[5] (2023) Pagina oficial de Git.

<https://git-scm.com/>

[6] Cardellino, Fernando (2021). El Manual Esencial de Git.

<https://www.freecodecamp.org/espanol/news/el-manual-esencial-de-git/>

[7] (2023) Página oficial de Glthub

<https://github.com/features>

[8] (2023) Página oficial de JavaScript

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

[9] (2023) Página oficial de Node

<https://nodejs.org/en/about>

[10] Altadil Izura, Pello Xabier (2019) Desarrollo Web con React. ANAYA MULTIMEDIA. ISBN: 9788441541634

[11] (2023) Página oficial de React

<https://react.dev/>

[12] (2023) Página oficial de Typescript

<https://www.typescriptlang.org/>

[13] (2023) Página oficial de Yarn

<https://yarnpkg.com/>

[14] Firebase (2023) Server-side encryption

<https://cloud.google.com/firestore/docs/server-side-encryption>

- [15] E. Rescorla, RTFM, Inc. (2000) HTTP Over TLS
<https://datatracker.ietf.org/doc/html/rfc2818>
- [16] E. Rescorla, Mozilla (2018) The Transport Layer Security (TLS) Protocol Version 1.3
<https://datatracker.ietf.org/doc/html/rfc8446>
- [17] ISO/IEC 18033-3:2010 (2010) Encryption systems (ciphers) for the purpose of data confidentiality
<https://www.iso.org/standard/54531.html>
- [18] (2023) Página principal de Connecteam
<https://connecteam.com/>
- [19] (2023) Página principal de Hubstaff
<https://hubstaff.com/>
- [20] (2023) Página principal de Alexia
<https://www.alexiaeducaria.com/>
- [21] (2023) Página principal de Dinantia
<https://dinantia.com/es/>
- [22] (2023) Página principal de Esemtia
<https://esemtia.com/>
- [23] (2023) Página principal de iEduca
<https://www.ieduca.com/es/>
- [24] (2023) Página principal de Additio App
<https://additioapp.com/>

Adjunto A: Repositorio git con el código fuente

En el siguiente repositorio publicado en GitHub, se dispone del software de la aplicación:

<https://github.com/Daniferrito/attendance>