

REMINDER

PROYECTO DE FIN DE CICLO

ISRAEL SERRANO

JORGE BUENDÍA

PABLO BALDAZO

Tutora:

Raquel Cerdá Losa

DESARROLLO DE APLICACIONES
MULTIPLATAFORMA

EDIX



Información del documento

Nombre del documento	PROYECTO_REMINDER
Autores del documento	Israel Serrano, Pablo Baldazo, Jorge Buendía
Versión del proyecto	v1.0
Fecha de entrega	02.06.2023
Repositorios GitHub	Aplicación: https://github.com/PIJ-Group/Proyecto_Fin_De_Ciclo.git Web: https://github.com/PIJ-Group/ReminderWeb.git



1 Introducción

Es un momento emocionante para presentar nuestro proyecto final de fin de grado, Reminder. Durante los últimos tres meses, hemos invertido tiempo y esfuerzo en la creación de esta aplicación innovadora, que tiene como objetivo facilitar la vida diaria de las personas al mejorar la forma en que recordamos eventos y tareas pendientes.

El desarrollo de Reminder surge como resultado de nuestro deseo de aplicar los conocimientos y habilidades adquiridos a lo largo de nuestra formación en el campo de las tecnologías móviles. Durante este proceso, hemos puesto en práctica una amplia gama de conceptos y técnicas aprendidas en asignaturas como desarrollo de aplicaciones móviles, bases de datos y diseño de interfaces de usuario, entre otras.

La idea central detrás de Reminder es crear una aplicación que simplifique y optimice la gestión de tareas y eventos en la vida cotidiana. Observamos que los calendarios existentes carecían de ciertos aspectos personalizables y de una experiencia de usuario intuitiva. Por lo tanto, nos propusimos superar estas limitaciones y desarrollar una solución que se adapte a las necesidades individuales de cada usuario.

Una de las características destacadas de Reminder es la capacidad de crear perfiles de usuario, lo que permite a las personas personalizar su propia agenda de recordatorios y adaptarla a su estilo de vida y preferencias. Asimismo, hemos integrado la funcionalidad de inicio de sesión mediante usuario y contraseña, a través de Firebase y otros proveedores, para ofrecer una experiencia segura y conveniente.

En términos de diseño, nos hemos enfocado en desarrollar una interfaz atractiva y fácil de usar. Hemos prestado atención a cada detalle, desde la elección de colores y tipografía hasta la disposición de los elementos en pantalla, con el objetivo de proporcionar una experiencia fluida y agradable para nuestros usuarios.

En cuanto a la funcionalidad, Reminder permite a los usuarios agregar, modificar y eliminar tareas pendientes de manera rápida y sencilla. La aplicación también les



brinda la posibilidad de establecer recordatorios personalizados, lo que garantiza que nunca se olviden eventos importantes y se cumplan todas las responsabilidades.

Nos emociona presentar Reminder como un proyecto que reúne nuestro conocimiento y pasión por el desarrollo de aplicaciones móviles. Esperamos que esta aplicación sea una herramienta útil y valiosa para las personas que buscan una forma eficiente y personalizada de gestionar sus tareas y eventos diarios. Estamos orgullosos de los logros obtenidos y confiamos en que Reminder tendrá un impacto positivo en la vida de quienes la utilicen.



Contenido

1	Introducción	3
2	Agradecimientos.....	7
3	Palabras clave.....	8
4	Sumario	11
5	Asignaturas aplicadas.....	12
6	Tecnologías y herramientas.....	14
7	Componentes del equipo y aportación realizada por cada estudiante	19
8	Fases del proyecto	20
8.1	Diseño.....	20
8.2	Desarrollo	26
8.2.1	Temas	26
8.2.2	Login y registro	28
8.2.3	Actividad Principal.....	32
8.2.4	Añadir, Actualizar y listar eventos	35
8.2.5	Página Web	39
8.3	Pruebas	40
9	Mejoras.....	41
10	Conclusiones	42
11	Bibliografía.....	44
	Anexos.....	45



Indice de imágenes

Ilustración 1 Splash Mockup	22
Ilustración 2 Login Mockup	23
Ilustración 3 Registro Mockup.....	23
Ilustración 4 Main Mockup	24
Ilustración 5 Añadir/Actualizar Mockup	24
Ilustración 6 Detalle Mockup.....	25
Ilustración 8 Login (Claro)	31
Ilustración 7 Registro (Claro)	31
Ilustración 9 Registro (Oscuro).....	31
Ilustración 10 Login (Oscuro).....	31
Ilustración 11 Main (Oscuro).....	34
Ilustración 12 Main (Claro)	34
Ilustración 13 Add Event (Oscuro).....	37
Ilustración 14 Add Event (Claro)	37
Ilustración 15 Update Event (Claro).....	38
Ilustración 16 Update Event (Oscuro)	38
Ilustración 17 List Event (Oscuro)	38
Ilustración 18 List Event (Claro).....	38
Ilustración 19 QR Web.....	39
Ilustración 20 Anexo WorkManager 1	45
Ilustración 21 Anexo WorkManager 2	46
Ilustración 22 Anexo WorkManager 3	46
Ilustración 23 Anexo MainActivity.....	47
Ilustración 24 Anexo AddEvent 1	47
Ilustración 25 Anexo AddEvent 2	47



2 Agradecimientos

Agradecemos a las personas que sin ánimo de lucro se han prestado a hacernos de beta testers:

- María Cano
- Daniel Arroyo
- Jorge Zuloaga
- Patricia Herreros

También queremos expresar nuestro más sincero agradecimiento a los siguientes profesores por su dedicación apoyo y enseñanzas:

- Tomás Escudero Delgado
- Félix de Pablo Lobo
- Raquel Cerdá Losa
- Alfredo Cordero Marcos
- Raúl Salgado Vilas
- Cristina Gómez Monsalve

Por último, pero no menos importante, agradecer a Chon Blázquez por su liderazgo como directora de la UNIR y a José Daniel Díaz por su trabajo como jefe de estudios.



3 Palabras clave

- **UX** (Experiencia de Usuario): Se refiere a la experiencia general de un usuario al interactuar con un producto, como una aplicación o un sitio web. Incluye aspectos como la usabilidad, accesibilidad, diseño visual, flujo de interacción y satisfacción del usuario.
- **UI** (Interfaz de Usuario): Es la parte visual y tangible con la que los usuarios interactúan en un producto, como botones, iconos, menús y campos de entrada. La UI se centra en la presentación y la interacción visual de los elementos de la interfaz.
- **Layout**: En el contexto del diseño de interfaces, el layout se refiere a la disposición y organización de los elementos en una pantalla. Define cómo se distribuyen visualmente los componentes, como botones, texto, imágenes y espacios en blanco, para lograr una interfaz equilibrada y fácil de usar.
- **View**: En el desarrollo de aplicaciones móviles o web, una view (vista) es un componente visual que representa una parte específica de la interfaz de usuario. Puede contener elementos interactivos, mostrar información o ser utilizada para interactuar con el usuario.
- **Activity**: En el desarrollo de aplicaciones móviles Android, una activity es un componente fundamental que representa una pantalla o una ventana con la que el usuario puede interactuar. Las activities son responsables de gestionar la interfaz de usuario y las interacciones dentro de una aplicación.



- **Git:** Es un sistema de control de versiones ampliamente utilizado para el seguimiento de cambios en el código fuente de un proyecto. Permite a los desarrolladores colaborar, mantener un historial de revisiones, deshacer cambios y gestionar ramas de desarrollo de manera eficiente. Git es conocido por su capacidad de manejar proyectos de cualquier tamaño y su integración con plataformas como GitHub y GitLab.
- **ListView:** Es un componente de la interfaz de usuario que muestra una lista de elementos desplazables en una sola columna. Se utiliza para representar conjuntos de datos en forma de lista, donde cada elemento de la lista puede ser seleccionado por el usuario.
- **TextView:** es un control básico en la programación de interfaces gráficas de usuario (GUI) en Android. Se utiliza para mostrar texto al usuario.
- **Action Bar:** Es una barra de herramientas ubicada en la parte superior de una aplicación Android. Proporciona acciones y opciones de navegación relevantes para la pantalla actual. El Action Bar puede contener botones, menús desplegables y elementos de título.
- **DatePickerDialog:** Es un cuadro de diálogo predefinido en Android que permite al usuario seleccionar una fecha de un calendario. Se utiliza para recopilar datos de fecha de manera fácil e intuitiva, mostrando un calendario para elegir el día, mes y año deseados.
- **TimePickerDialog:** Similar al DatePickerDialog, es un cuadro de diálogo predefinido en Android que permite al usuario seleccionar una hora. Se utiliza para recopilar datos de tiempo, mostrando una interfaz para seleccionar horas y minutos.



- **APK:** Es el formato de archivo utilizado para distribuir e instalar aplicaciones en dispositivos Android. APK significa "Android Package" y contiene todos los archivos necesarios para ejecutar una aplicación en un dispositivo Android, incluido el código compilado, recursos, manifiestos y certificados de firma.
- **Spinner:** Un Spinner es un componente de la interfaz de usuario en Android que permite seleccionar un único elemento de los mostrados en una lista desplegable. Equivale al típico select de HTML, pero con una diferencia interesante: de serie, Spinner es capaz de mostrar la lista en un cuadro de diálogo.
- **Intent:** Los Intents permiten la comunicación entre componentes de la aplicación, como actividades, servicios y receptores de difusión.
- **Toolbar:** es un componente de interfaz de usuario utilizado para crear una barra de navegación o una barra de herramientas en una aplicación web o móvil.
- **Sidenav:** es un componente importante que se utiliza para crear paneles laterales deslizantes en una aplicación web o móvil. Proporciona una forma conveniente de mostrar contenido secundario o de navegación en un panel que se puede abrir y cerrar.



4 Sumario

Reminder es una aplicación móvil innovadora diseñada para mejorar la gestión de eventos y tareas pendientes. Desarrollada como proyecto final de nuestro grado, esta aplicación tiene como objetivo brindar a los usuarios una solución personalizada y eficiente para organizar su vida diaria.

Mediante la aplicación de los conocimientos y habilidades adquiridos a lo largo de nuestros estudios en desarrollo de aplicaciones móviles, hemos creado una aplicación fácil de usar y personalizable. Reminder se diferencia de los calendarios tradicionales al ofrecer perfiles de usuario, lo que permite a las personas gestionar sus tareas y eventos de acuerdo con sus necesidades específicas. Además, la aplicación ofrece una experiencia de usuario única y personalizada a través de estilos y diseños personalizables.

A través de una interfaz intuitiva, los usuarios pueden agregar, modificar y eliminar tareas, asegurándose de que los eventos importantes nunca sean pasados por alto. Reminder está dirigida a un amplio público, que incluye a estudiantes, profesionales y padres que buscan optimizar su tiempo y aumentar su productividad.

En resumen, Reminder es una aplicación innovadora de recordatorios que ofrece perfiles de usuario y opciones de personalización. Su interfaz intuitiva y características convenientes la convierten en una herramienta valiosa para la gestión efectiva del tiempo y las tareas. Estamos orgullosos de presentar Reminder como la culminación de nuestros estudios y creemos que tendrá un impacto positivo en la vida diaria de los usuarios.



5 **Asignaturas aplicadas**

Nos hemos apoyado en diferentes asignaturas para la realización de este proyecto, a continuación, resumiremos los aspectos más importantes que nos han llevado a apoyarnos en ellas:

- **Programación:**

Asignatura principal del proyecto, ya que recoge los aspectos fundamentales de la programación, independientemente del lenguaje usado (lógica de desarrollo, buenas prácticas y conocimientos generales del uso de los lenguajes de programación).

- **Programación multimedia y dispositivos móviles:**

El proyecto está realizado en Android Studio, para ser una aplicación orientada a dispositivos móviles. Por tanto, se utilizan todos los conocimientos adquiridos en esta asignatura, incluido el lenguaje de programación usado.

- **Base de datos:**

La persistencia de los datos es fundamental para nuestra aplicación, ya que debemos registrar los eventos que desee el usuario y mostrárselos una vez lo requiera. Para ello nos ayudaremos de herramientas que nos ayuden con esa persistencia, las cuáles explicaremos en su correspondiente apartado.

- **Diseño de interfaces:**

Otra de las asignaturas más importantes en la que nos hemos apoyado ha sido diseño de interfaces, puesto que como bien sabemos, es fundamental que todos los usuarios que van a ser partícipes de la aplicación tengan una experiencia correcta a la hora de interactuar con la misma. No solo es importante el tema de colores y contrastes, para que la aplicación se visualice correctamente si no también que el usuario final sepa cómo tiene que utilizar la aplicación, por tanto, es importante la realización de una serie de buenas prácticas que proporcionarán en gran medida una buena experiencia de usuario al utilizar nuestro producto.



- **Entornos de desarrollo:**

Es fundamental para el correcto desarrollo de la aplicación usar un control de versiones para poder reflejar los cambios realizados, y poder deshacer ciertas actualizaciones que no se comportan de la manera esperada. A su vez, podemos llevar un control de los tiempos de desarrollo con aplicaciones de organización de proyectos.

- **Lenguaje de marcas:**

Puesto que hemos desarrollado nuestra aplicación en Android Studio, implica que necesitemos conocimientos de esta asignatura, para la realización de los **layouts**.

Además, para la distribución de la aplicación hemos creado una página web, donde destacamos sus principales características además de ubicar el descargable de esta.

6 Tecnologías y herramientas

Durante la realización del proyecto final, hemos utilizado una serie de herramientas, lenguajes y entornos de desarrollo, que han sido fundamentales para la realización del mismo. A continuación explicaremos por qué hemos decidido utilizarlas y qué enfoque han aportado a nuestro proyecto.

En primer lugar, tenemos que destacar la utilización de herramientas, servicios y plataformas que hemos utilizado durante el desarrollo del proyecto:

- **Trello:**

Debido a la metodología de trabajo propuesta en el anteproyecto, SCRUM con sprints de 1 semana, se ha utilizado esta herramienta para poder llevar un control de las tareas, pudiendo tener diferentes listas con las tareas, ordenadas por su estado de realización y pudiendo añadir comentarios e interactividad con el resto del grupo. Para poder acceder a dicho tablón, dejamos el enlace correspondiente aquí.

Lo hemos dividido en los siguientes apartados:

- *Lista de tareas*: es una lista en la que los miembros proponen las tareas a realizar (ya sean tareas principales como la realización de funcionalidad, como pequeños arreglos y cambios). Se asigna un integrante del equipo para su realización y se incluye una explicación de la tarea.
- *To-Do Sprint*: una vez la tarea está creada y asignada, y se pretende que se empiece con su realización en el siguiente sprint, se mueve de “Lista de tareas” a este apartado, para que el equipo tenga conocimiento de las siguientes tareas a realizar.
- *En proceso*: como su nombre indica, son las tareas que están siendo desarrolladas en ese momento, pudiendo incluir comentarios para que los otros miembros del equipo revisen el estado y den su aprobación.
- *Finalizado*: la última fase de una tarea, habiendo pasado por el resto de los estados y teniendo la aprobación del equipo para su puesta en la aplicación después de pasar por el estado de “en proceso”. En ella, se introduce un comentario a modo de resumen, contando la funcionalidad aplicada o que se ha realizado para su solución.



- **GitHub:**

Debido a la magnitud del proyecto, es necesario contar con una herramienta para controlar los cambios realizados en dicha aplicación, y poder verificar su correcto funcionamiento, o deshacer una actualización si esta llega a conflicto con el resto de la aplicación. Para ello, utilizamos GitHub, herramienta basada en comandos Git, para almacenar el repositorio remoto y poder tener el trabajo de todo el equipo centralizado.

La decisión de la elección de esta herramienta no ha sido otra que haber trabajado anteriormente, durante toda la realización del ciclo, con ella, pudiendo controlar fácilmente todos los aspectos debido a la familiarización que tenemos después de todos los trabajos realizados.

Dejamos un enlace correspondiente al repositorio pinchando aquí.

- **Firebase:**

Nuestra aplicación se basa en la persistencia de ciertos eventos que el usuario actual de nuestra aplicación necesita recordar. Para ello, utilizaremos una herramienta para almacenar dicha información, que nos permita un fácil manejo de la conexión con la base de datos y su posterior mantenimiento.

Nuestra elección ha sido Firebase, por esa simplicidad a la hora del manejo de la base de datos y de la transferencia de los datos para su persistencia.

Podemos comprobar la organización de la base de datos pinchando aquí.

Para poder mantener un correcto control de los datos almacenados, hemos separado la base de datos en dos colecciones:

- *Users*: en esta colección almacenamos los datos de los usuarios registrados de cualquiera de las maneras disponibles (introducción de un email y contraseña o el registro mediante un proveedor), siendo los datos del email del usuario (que usaremos para la funcionalidad de la aplicación) y su nombre (utilizado para mensajes personalizados a ese usuario) los datos que incluiremos en cada documento de dicha colección.



- *Events*: mantenemos separados los datos de los usuarios de sus eventos creados, para poder acceder mejor a cada uno de ellos desde la interfaz de la herramienta. En el caso de los eventos registrados por los usuarios, añadimos una serie de información acerca del mismo (hora de registro, hora del evento, fecha de registro, fecha del evento, título, descripción del evento, identificación del evento) además de una referencia al usuario que ha creado el evento (identificación del mismo y su email) para poder llevar un control de los eventos de cada uno y poder personalizar la aplicación de acuerdo con el usuario activo.

En segundo lugar, tenemos los entornos de desarrollo que todo el equipo decidió utilizar para poder desarrollar el proyecto:

- **Android Studio:**

Debido a la elección de realizar una aplicación móvil Android, el IDE elegido no puede ser otro que Android Studio.

Es un IDE muy completo que brinda mucha funcionalidad, como poder instalar emuladores de los terminales más importantes del mercado y poder probar en tiempo real como está quedando la aplicación, sin necesidad de empaquetarla e instalarla en un teléfono real.

Además, añade funcionalidad de Git, que nos permite llevar ese control de versiones de manera mucho más sencilla.

- **Visual Studio Code:**

Es una herramienta muy popular y ampliamente utilizada por desarrolladores debido a su versatilidad, funcionalidad y facilidad de uso, por esa razón, es el elegido para desarrollar nuestra web con el framework Angular.



A continuación, explicaremos un poco acerca del framework escogido para la realización de la página web de nuestra aplicación, así como del módulo utilizado durante el desarrollo de esta:

- **Angular:**

Angular es un framework de desarrollo de aplicaciones web de código abierto desarrollado por Google. Proporciona una estructura sólida para crear aplicaciones web dinámicas y escalables, razón por la que lo hemos escogido para la realización de la web de nuestro proyecto.

- **Angular Material:**

Proporciona una amplia gama de componentes predefinidos y estilos de diseño para crear interfaces de usuario modernas y atractivas, además de responsivas, que nos ayuda a dar un estilo profesional y minimalista a nuestra web.

Por último, pero no menos importante, detallaremos todos los lenguajes que hemos ido utilizando durante la realización del proyecto final:

- **Java:**

Para la realización de una aplicación para Android, podíamos elegir entre dos lenguajes diferentes: Java y Kotlin.

Debido a su utilización a lo largo del curso, y, por tanto, su familiarización, decidimos utilizar el lenguaje Java, que, si bien no es lo demandado actualmente para realizar este tipo de aplicaciones, tenemos unos conocimientos adquiridos a lo largo del curso, suficientes para poder haber realizado con éxito la aplicación y todos los problemas que han existido a lo largo de su desarrollo.



- **Xml:**

Lenguaje de marcas utilizado para la parte visual de la aplicación, los layouts, y así poder formar la interfaz de usuario a nuestro criterio, siguiendo como pilar fundamental la correcta experiencia de usuario.

- **HTML:**

HTML (HyperText Markup Language) es el lenguaje de marcado estándar utilizado para crear y estructurar el contenido de las páginas web. En nuestro caso es el lenguaje utilizado para definir la estructura y los elementos de la web que hemos implementado para dar a conocer nuestra app, así como para poder descargarla.

- **CSS:**

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para describir la presentación y el diseño visual de nuestra página web, en consonancia a los elementos usados de Angular Material.

- **TypeScript:**

Es un lenguaje derivado de JavaScript que añade características adicionales y se ha convertido en el lenguaje de programación principal utilizado en aplicaciones web con Angular. En nuestro proyecto, hemos optado por utilizar TypeScript para desarrollar la página web de nuestra aplicación.



7 Componentes del equipo y aportación realizada por cada estudiante.

El equipo de trabajo de este proyecto está compuesto por Pablo Baldazo, Jorge Buendía e Israel Serrano.

En la fase de Diseño intervinimos todos por igual proponiendo ideas, cambios y/o mejoras en lo que sería el resultado final de nuestro proyecto.

Una vez decidido como hacerlo nos repartimos el trabajo en tres grandes bloques:

- **Pablo Baldazo** se encargó de toda la parte de las actividades del Login y registro, así como implementar el registro con los distintos proveedores de servicios.
- **Jorge Buendía** ha sido el encargado de hacer todo lo referente a la Main Activity.
- **Israel Serrano**, desarrolló las actividades de añadir eventos, de modificar eventos y la de detalle de eventos, así como de realizar la página web.

Aun con estos tres bloques bien definidos, la necesidad de la realización de nuevas tareas en cada Sprint desencadenó que, según la disponibilidad de los miembros del equipo, se pudiera realizar tareas de un bloque diferente al asignado inicialmente. De esta manera, cumplimos con los plazos marcados en los Sprints semanales.

8 Fases del proyecto

8.1 Diseño

Una vez realizada la fase del anteproyecto, donde dimos unas ideas generales de que tipo de aplicación queríamos realizar y cuáles serían sus componentes, llegamos al momento de diseñar la interfaz de usuario.

Para ello, nos basamos en un desarrollo centrado en la experiencia de usuario, donde, además de dotar a la aplicación de un estilo minimalista y moderno, intentamos que el usuario se sienta cómodo con el diseño, pudiendo entender fácilmente para que sirve cada elemento, siguiendo convenciones tales como, por ejemplo, la dimensión de los elementos seleccionables como pueden ser los botones, de un mínimo de 48 “dp”.

Para lograr esa armonía y funcionalidad, nos centramos en dos pilares fundamentales: la elección del color y la disposición de los elementos.

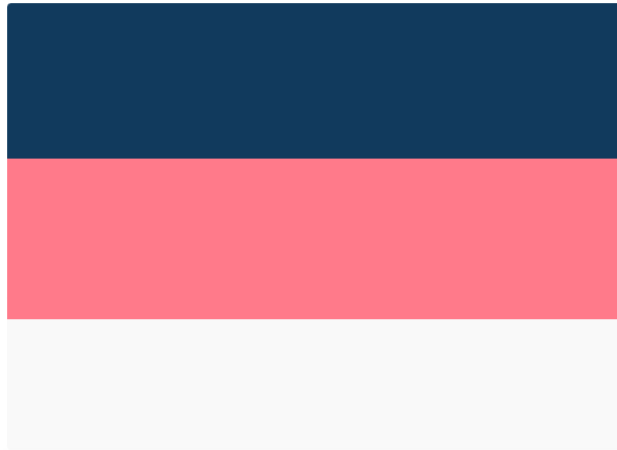
- *Elección de color:*

Además del tema estético, la elección de color es un factor fundamental, ya que, de no ser adecuado en cuanto a combinación y tonalidades, puede llevar a confusión de los elementos para personas con problemas de visión. Dividiremos la elección de color en dos apartados:

- *Estética:*

En la aplicación predominará el fondo blanco para el modo claro, y fondo gris oscuro para el modo oscuro. Con ese predominio del color, podremos utilizar el resto de los colores para elementos como la barra de menú y botones, y crearemos una interfaz elegante y minimalista.

La gama cromática elegida para el resto de los elementos será la siguiente:



El primer color será un tono azul, correspondiente a el color **#113A5D**, y que lo utilizaremos en las barras de menú que estarán disponibles en ciertos layout de la aplicación.

El segundo color será un tono rosa, correspondiente a el color **#FF7A8A**, y lo utilizaremos principalmente en los elementos seleccionables de la aplicación cuando estos se sitúen sobre el color de fondo.

El tercer color será un tono gris claro, correspondiente a el color **#F9F9F9**, y lo utilizaremos para resaltar ciertas zonas de introducción de texto.

- *Contraste:*

Para que la aplicación pueda ser funcional para todo tipo usuarios, se han seguido dos pasos fundamentales para ofrecer un buen contraste entre elementos y que no haya problemas en su visualización:

En primer lugar, por cada color de la paleta de colores elegida, se ha añadido una versión más clara y otra más oscura, con esto conseguimos abarcar cualquier gama de colores y poder combinarlos cuando se necesite dotar de mayor contraste a ese elemento.

En segundo lugar, hemos hecho uso de páginas para garantizar el ratio mínimo de contraste, asegurándonos un mínimo de puntuación de 8.4 para el caso más desfavorable:

Color del texto

Color del fondo

Ratio mínimo

☒ 11.75

Color en hexadecimal, RGB o palabras clave de color CSS.
Por ejemplo: #FFFFFF, rgb(255,255,255) o white

Color del texto

Color del fondo

Ratio mínimo

☒ 8.4

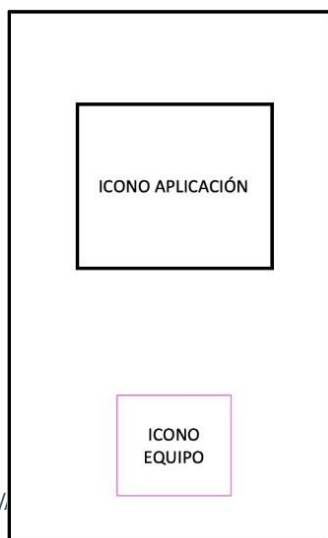
Color en hexadecimal, RGB o palabras clave de color CSS.
Por ejemplo: #FFFFFF, rgb(255,255,255) o white

- Disposición de los elementos:

Para poder seguir un orden preestablecido del diseño de la aplicación, se crearon una serie de mockups que, si bien no tenían que ser el diseño final, sí que seguiríamos para la inclusión de la mayoría de los elementos. Así, dotábamos a la aplicación de una armonía, la cual hace que cada nueva pantalla recuerde al conjunto del proyecto.

A continuación, mostramos los mockups de cada sección:

- Splash:



La distribución de esta sección consta de un elemento predominante, el icono de la aplicación, junto al icono del equipo de desarrollo que incluimos en todos los trabajos grupales.

- Login:

Ilustración 2 Login Mockup

Consta de un icono de un calendario, dos elementos para poder introducir las credenciales del usuario, un botón para poder acceder con las credenciales anteriormente introducidas, un botón o frase de enlace al layout de crear una cuenta, y tres botones inferiores que serán los iconos de los proveedores de servicio que nos permitirán registrarnos e identificarnos con sus credenciales.

- Registro:

Ilustración 3 Registro Mockup

A partir de este layout, todos contarán con la barra de menú, en este caso solo a nombre de recordatorio de nombre de la aplicación.

Incorporaremos un icono que haga alusión a la creación de cuentas, unos elementos donde poder escribir todas las credenciales requeridas y el botón para confirmar la introducción de los elementos.

- Página principal:

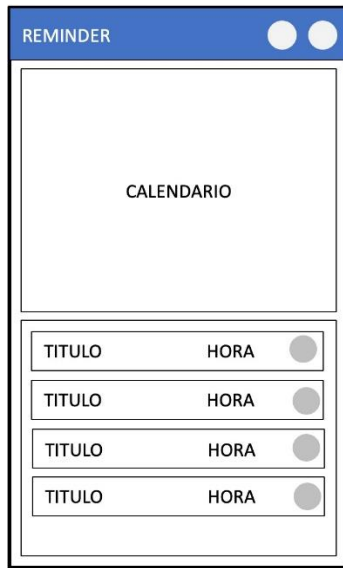


Ilustración 4 Main Mockup

En esta página, incluiremos dos botones en la barra de menú, los cuáles servirían para añadir un nuevo evento y poder cerrar sesión y volver a la pantalla de “Login”.

Esta pantalla quedará dividida con dos elementos, el calendario de la parte superior y la lista de eventos en la parte inferior, con cada uno de ellos mostrando el título, hora y un botón desplegable de opciones.

- Añadir y actualizar eventos:

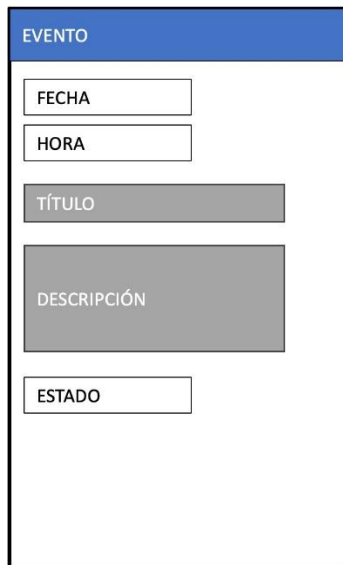


Ilustración 5 Añadir/Actualizar Mockup

Incluiremos el mismo diseño tanto para añadir como para actualizar los eventos.

La barra de menú contará con un botón para poder guardar el evento y la pantalla contará con los datos del usuario en activo, dos elementos para escribir el título y descripción, dos botones para poder seleccionar una fecha distinta y la hora de realización del evento y el estado del mismo.

- Ver detalles del evento:



EVENTO

FECHA

HORA

TÍTULO

DESCRIPCIÓN

ESTADO

Por último, tendremos un layout donde podremos ver en detalle el evento seleccionado, mostrando la fecha, hora, título, descripción y su estado.

.

Ilustración 6 Detalle Mockup

Una vez decidido y diseñado estos aspectos, pasaremos al desarrollo de la aplicación, donde distribuiremos las tareas, como ya dijimos anteriormente en este documento, utilizando metodología SCRUM con sprints de una semana y trazando las tareas a través de un tablero Kanvan, en este caso nos decidimos por Trello. Seguiremos los datos aportados en la fase anterior para su desarrollo.

Por último, decidimos hacer una página web como complemento a la aplicación móvil, para dar a conocer la aplicación, a nosotros, y poder descargar el **apk**. Esta debería de ser responsiva, para poder visualizarla tanto en navegadores de escritorio, como en navegadores móviles. Y debido al reciente conocimiento de uno de los integrantes del equipo sobre Angular, decidimos realizarla con este framework debido a su naturaleza responsiva, además de tener como apoyo Angular material para dar estilos simples y de aspecto amigable, a la par que profesionales.

8.2 Desarrollo

En la fase de Desarrollo del proyecto Reminder, se han dividido las tareas en tres partes principales: el login con proveedores, el registro de usuarios, y la funcionalidad principal del calendario. Además, se han identificado tres subpartes adicionales: la actividad principal (main activity) que muestra los eventos de los usuarios y permite añadir nuevos eventos, y una sección dedicada a actualizar o listar en detalle los eventos existentes.

Además, desarrollaremos un punto importante de la aplicación, como son los temas, en los cuales hemos tenido especial cuidado, puesto que hemos escogido una paleta de colores que haga resaltar ciertos elementos de la interfaz de usuario.

8.2.1 Temas

Durante el diseño de la aplicación, se ha tenido especial cuidado en la implementación de los colores, lo cual se ha detallado en la fase de diseño, pero también queríamos hacer una mención a los modos de visualización (claro y oscuro).

Por lo tanto, hemos implementado una gama de colores para cada escenario, centrándonos en obtener una correcta visualización que garantice una experiencia correcta para todos los usuarios, cuando su teléfono cambie automáticamente el tema de visualización, en función de su propia configuración y que nuestra aplicación se adapte perfectamente a esas condiciones.

Esto, lo hemos conseguido, aplicando una paleta de colores oscuros, mezclando azules marinos y rosas fuertes, así como programando el uso de colores blancos y negros de los textos, según se esté usando el tema claro o el tema oscuro en la aplicación.

A continuación, proporcionaremos una breve descripción de ambos temas, así como las correspondientes imágenes para apreciar la visualización de estos:



- **Modo claro:**

Hemos utilizado colores más claros en la interfaz, como un fondo blanco o de tonos claros y texto oscuro. Este tema es popular por su aspecto limpio y moderno, además de su legibilidad en entornos bien iluminados. El tema claro es especialmente adecuado para su uso durante el día, cuando la luz ambiental es suficiente y el contraste entre la pantalla y el entorno no es un problema.

- **Modo oscuro:**

Nuestro tema oscuro se caracteriza por utilizar colores más oscuros en la interfaz, lo que resulta en un fondo negro o gris oscuro y texto de color claro. Este tema tiene varias ventajas importantes. En primer lugar, reduce la fatiga visual al disminuir el contraste entre la pantalla y el entorno circundante, lo que resulta en una menor exposición a la luz intensa. Esto es especialmente beneficioso en entornos con poca iluminación o durante la noche, ya que la pantalla no resulta tan intensa y los ojos no se esfuerzan tanto.

En resumen, tanto el tema oscuro como el tema claro desempeñan un papel importante en la experiencia del usuario al utilizar la aplicación Reminder. El tema oscuro reduce la fatiga visual y puede contribuir a un menor consumo de energía en dispositivos OLED, mientras que el tema claro ofrece un aspecto moderno y una buena legibilidad en entornos bien iluminados. Brindar opciones de temas personalizables nos permite satisfacer las preferencias individuales de los usuarios y mejorar su experiencia global.



8.2.2 Login y registro

- **Inicio de sesión con usuario y contraseña:**

En esta primera pantalla, se ha implementado un sistema de inicio de sesión tradicional de usuario y contraseña, que está directamente vinculado a Firebase, donde los usuarios pueden introducir sus credenciales de acceso para entrar a la aplicación.

- **Verificación y autenticación de credenciales:**

Una vez que los usuarios ingresan sus credenciales, la aplicación verifica la validez de la combinación de usuario y contraseña. Esto implica una verificación en tiempo real con Firebase para garantizar que los datos ingresados sean correctos y coincidan con los registros almacenados en la base de datos. Si las credenciales son correctas, se autentica al usuario y se le permite acceder a su cuenta personalizada.

- **Integración con Firebase Authentication:**

El sistema de inicio de sesión con usuario y contraseña está directamente vinculado a Firebase Authentication. Firebase proporciona una plataforma segura y confiable para la gestión de autenticación de usuarios. Al conectarse con Firebase Authentication, se asegura la confidencialidad y protección de las credenciales de los usuarios.

- **Seguridad y privacidad:**

El inicio de sesión con usuario y contraseña brinda seguridad y privacidad a los usuarios de Reminder. Las contraseñas se almacenan de forma segura y se utilizan métodos de cifrado para proteger la información confidencial. Además, la integración con Firebase Authentication ofrece funciones adicionales de seguridad, como la gestión de sesiones y la detección de intentos de inicio de sesión no autorizados.

Además, durante esta fase inicial del desarrollo de la aplicación, hemos decidido implementar un sistema extra de inicio de sesión que ofrece a los usuarios la opción de utilizar diferentes proveedores como Google, Twitter o GitHub. A continuación, proporcionaremos más detalles acerca de cada uno de estos proveedores adicionales y como se han integrado en nuestra aplicación.

- **Proveedor de inicio de sesión con Google:**

Se ha desarrollado una funcionalidad que permite a los usuarios iniciar sesión utilizando sus cuentas de Google. Al seleccionar la opción de inicio de sesión con Google, los usuarios son redirigidos a una página de autorización de Google donde se les solicita conceder permiso a Reminder para acceder a cierta información de su cuenta de Google. Una vez que los usuarios otorgan el permiso, son redirigidos de vuelta a la aplicación y se les permite acceder a su cuenta de Reminder. Esta integración con Google es muy usual hoy en día, puesto que facilita el inicio de sesión para aquellos usuarios que ya tienen una cuenta de Google.

- **Proveedores de inicio de sesión con Twitter y GitHub:**

Para los proveedores de inicio de sesión menos comunes, como Twitter y GitHub, se han implementado dos actividades adicionales. Al seleccionar la opción de inicio de sesión con Twitter o GitHub, los usuarios son redirigidos a un navegador web dentro de la aplicación. En este navegador, se les muestra la página correspondiente de autorización de Twitter o GitHub, donde pueden iniciar sesión y otorgar permiso a Reminder para acceder a su cuenta. Después de que los usuarios aceptan el inicio de sesión, son redirigidos de vuelta a la aplicación y se les permite acceder a su cuenta de Reminder.



En cuanto a la pestaña de registro, se han implementado varias validaciones en el formulario para garantizar que los datos proporcionados por los usuarios sean correctos y cumplan con los requisitos necesarios. Esto incluye la verificación de que el nombre de usuario no puede quedarse en blanco, por poner un ejemplo. Aquí tenemos que aclarar que el nombre de usuario no tiene por qué ser único, ya que no es una aplicación donde haya interacción por parte de los usuarios, como puede ser Twitter u otras redes sociales donde ver nombres repetidos puede llevar a confusión, por tanto, el identificador único de cada usuario será el correo electrónico.

Como comentábamos anteriormente, también tenemos la validación de la dirección de correo electrónico y la implementación de medidas de seguridad para asegurar contraseñas fuertes. Una vez que los usuarios completan el formulario de registro y hacen clic en el botón de registro, se establece una conexión con Firebase, una plataforma de desarrollo de aplicaciones móviles y web, para crear el usuario en la base de datos y almacenar su información de manera segura.

En resumen, se han implementado diferentes proveedores de inicio de sesión, incluyendo Google, Twitter y GitHub, en el proyecto Reminder. Además, se han realizado validaciones en el formulario de registro y se ha utilizado Firebase para crear y almacenar los usuarios registrados. Estas medidas garantizan la seguridad y privacidad de los usuarios, así como una experiencia de inicio de sesión y registro precisa y correcta.

A continuación, mostraremos varias imágenes, las primeras muestran el tema claro de la aplicación, tanto en el login como el registro, mientras que las siguientes podremos visualizar el enfoque contrario, el tema oscuro.

Reminder

30

Email

Password

LOGIN

Don't have an account yet? [Sign up now!](#)

or sign up with





  

Ilustración 7 Login (Claro)

Register

Create a new account



Complete name

Email

Password

Confirm password

USER REGISTRATION

I already have an account

Ilustración 8 Registro (Claro)

Reminder

30

Email

Password

LOGIN

Don't have an account yet? [Sign up now!](#)

or sign up with





  

Ilustración 9 Login (Oscuro)

Register

Create a new account



Complete name

Email

Password

Confirm password

USER REGISTRATION

I already have an account

Ilustración 10 Registro (Oscuro)



Por último, hemos añadido una interesante función que ayudará a los clientes que utilicen solo una cuenta en sus dispositivos, haciendo que, una vez el usuario se registre o inicie sesión, dicha sesión se mantenga aun cerrando la aplicación.

Siempre que vuelva a la aplicación, comenzará a visualizarla desde la actividad principal, obviando los pasos anteriores, debiendo de cerrar sesión desde el botón situado en la actividad principal para tal efecto y así poder ingresar con nuevas credenciales.

8.2.3 Actividad Principal

En esta parte se ha implementado la funcionalidad principal del calendario. Al iniciar sesión, los usuarios son recibidos con una vista de calendario que ocupa la mitad de la pantalla. Dicha vista se compone de las siguientes partes:

- **Barra de menú:**

Además de incluir como referencia el nombre de la aplicación, nos encontramos con dos botones, los cuales tienen la siguiente funcionalidad:

- *Botón de añadir tarea* (primer icono): si presionamos en él, nos llevará a una nueva ventana, en la que se pedirá una serie de datos para poder ingresar un nuevo evento, los cuales quedarán registrados en la base de datos y servirán para poder ofrecer esa experiencia personalizada al usuario.
- *Botón de cerrar sesión* (segundo icono): esta experiencia personalizada no sería posible si el usuario no tiene la posibilidad de tener varias cuentas en el sistema (su correo personal, correo laboral, correo organizativo, etc), por lo que brindamos la oportunidad de poder cerrar sesión y volver a la pantalla de inicio para poder ingresar desde otra cuenta.

- **Calendario:**

La primera mitad de la pantalla la compone un calendario.

Con este calendario (que muestra la fecha actual de forma predeterminada) puedes seleccionar cualquier fecha que esté incluido en él, con el fin de poder registrar los eventos de la aplicación.

Gracias a una serie de métodos incluidos en el calendario, podemos recoger esa fecha que selecciona el usuario y almacenarla/distribuirla, para poder personalizar el resto de los elementos, como son los eventos y la visualización de estos.

- **Zona de visualización de los eventos:**

La segunda mitad de la pantalla la compone un “listView”, que no es más que un elemento vacío que sirve como contenedor de los artículos que saldrán en su interior.

Esos artículos están desarrollados en un archivo diferente, en el cual especificamos los siguientes elementos:

- *Título del evento*: cuando el usuario registra un nuevo evento, además de los datos requeridos que debe ingresar, también se pide que ingrese un título, el cual mostramos en el artículo a modo de resumen del evento completo.
- *Hora de realización del evento*: otro de los campos a introducir a la hora de registrar el evento es la hora de su realización, la cual también recogemos y mostramos en el artículo.
- Icono de “*otras acciones*”:
- Debido a la importancia de poder acceder a los eventos en mayor profundidad, editarlos si ha habido algún error e incluso borrarlos debido a su finalización o por un error a la hora de añadirlo, tenemos un icono que despliega un diálogo con tres acciones:
- *Editar evento*: accederemos a una pantalla nueva pero muy parecida a la pantalla de añadir evento, ya que se muestra toda la información referente a ese evento y puede ser editado en su totalidad.

- *Ver detalles del evento*: accedemos a una nueva pantalla en la que se muestra toda la información del artículo seleccionado, sin posibilidad de modificación de cada punto.
- *Eliminar evento*: podremos eliminar el evento seleccionado.

Toda la funcionalidad de esta sección se realiza mediante métodos de llamada a la base de datos, ya que accediendo al usuario que ha registrado el evento, y el día en que lo ha hecho, podemos mostrar todos los eventos referentes a ese usuario y día en este apartado mediante los artículos creados, que se irán mostrando a modo de lista y estarán ordenados por la hora de realización para mayor claridad.

A continuación, mostramos dos imágenes, referentes al modo claro y al modo oscuro:

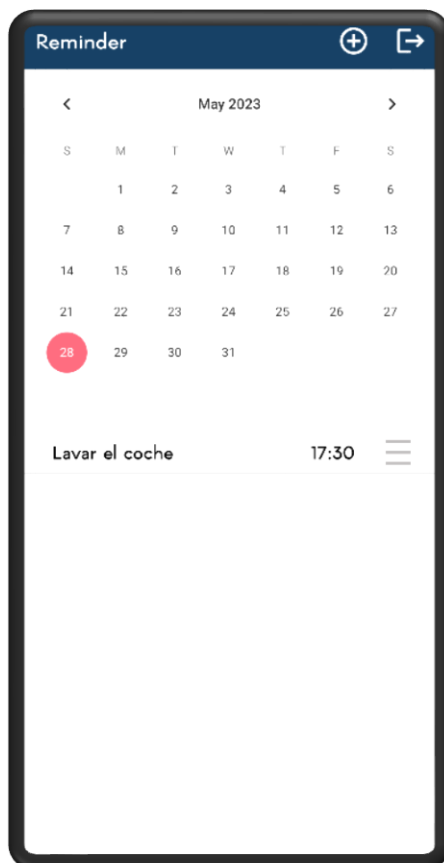


Ilustración 12 Main (Claro)

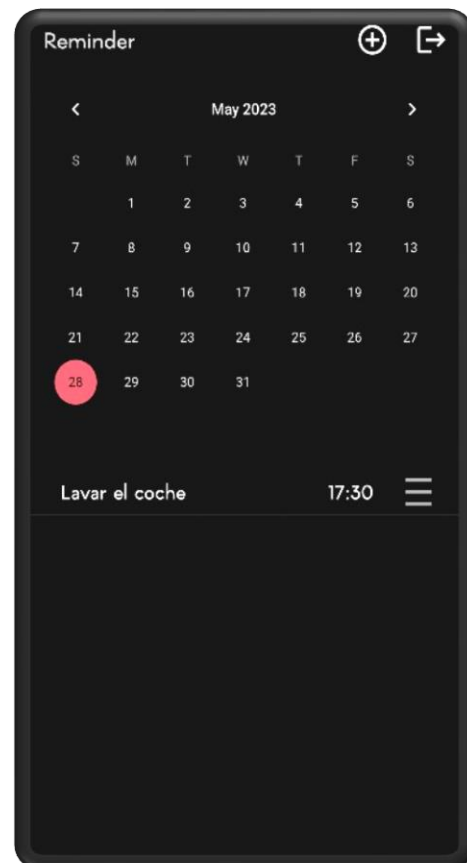


Ilustración 11 Main (Oscuro)



8.2.4 Añadir, Actualizar y listar eventos

- **Action Bar:**

Lo primero que nos encontramos de forma común en estas tres activities, es esta Action Bar, el cual incluye el nombre de la activity en el que nos encontramos, y una flecha para volver a la anterior.

En esta barra, además, en las actividades de añadir evento y actualizar evento, nos encontramos unos botones para añadir o actualizar/modificar un evento, según en la que nos encontremos.

- **Añadir evento:**

- *Nombre de usuario:* procedemos a recoger el nombre del usuario que está logueado, de su correspondiente documento en la base de datos Firestore, para mostrarlo en la parte superior del interfaz. Da una sensación de personalización que siempre nos atrae más a los usuarios.
- *Fecha:* Se recoge la fecha y hora actual del sistema para indicar al usuario la fecha y hora en la que está en este justo momento.
- *Title y Description:* habilitamos dos TextView en el que poner un título al evento, que es lo que mostrará en la vista de calendario y una descripción más detallada del evento.
- *Botones de Calendar y Hour:* estos botones son un datePickerDialog, y un timePickerDialog, los cuales al pulsarlos muestran respectivamente un dialog con un calendario, en el que podemos elegir el día, mes y año que queramos, y la hora y minutos en el segundo.
- *Fecha y Hora:* a la derecha de los botones, tenemos sendos textviews, para mostrar la hora y la fecha seleccionadas, formateándolas para que aparezcan de una manera amigable.

En el caso de la fecha, cabe destacar, que, al entrar en esta activity, ya viene por defecto seleccionada la fecha que teníamos marcada en la vista de calendario de la activity principal, la cual se envía como parámetro a través del intent de cambio de actividad, y se recoge para mostrarla en este campo.



- *Estado*: por último, nos aparece el estado del evento, que, por defecto, en esta interfaz aparecerá como no finalizado, al ser la pantalla en la que creamos los eventos.

En esta activity, al guardar el evento, en lugar de guardar uno a uno los distintos elementos del evento, se crea un objeto de evento, que es añadido a la correspondiente colección de eventos, creando un nuevo documento.

- **Actualizar eventos:**

Esta activity en configuración y usabilidad es muy parecida a la anterior, con la salvedad, de que cuando entramos, ya tenemos cargados los datos del evento a modificar/actualizar es sus correspondientes views, recogidos de los parámetros enviados a través del intent de la actividad principal.

- *Estado*: en este caso el estado del evento aparece arriba del todo, con unos imageviews que inician ocultos, y que muestran con un cuadro de color, una ayuda visual a saber el estado en el que se encuentra el evento, además del correspondiente texto de finalizado o no finalizado.
- *Tittle y Description*: como en la actividad anterior son dos campos de texto para modificar el título y/o la descripción del evento.
- *Calendar, Hour, fecha k hora*: estos botones con sus respectivos textviews, funcionan idénticos a la actividad anterior.
- *New Status*: es este caso nos encontramos con un spinner, que nos hace de selector para cambiar el estado del evento y un textview al lado para mostrar, lo que está elegido en ese momento.

- **Listar evento:**

En esta activity, tenemos los mismos componentes o views que en las anteriores actividades, pero en este caso, los campos están bloqueados para sólo permitir su lectura. De nuevo los datos para cada componente es enviado a través del intent de cambio de actividad y recogidos para mostrarlos en su correspondiente view.

En esta actividad también se usa la ayuda visual en forma de colores para ver rápidamente el estado del evento.

A continuación, mostramos imágenes, referentes al modo claro y al modo oscuro de las actividades comentadas:

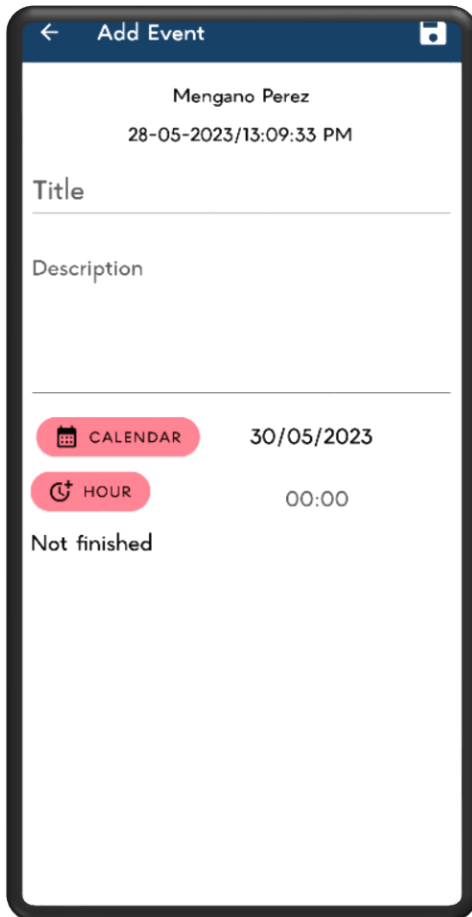


Ilustración 14 Add Event (Claro)

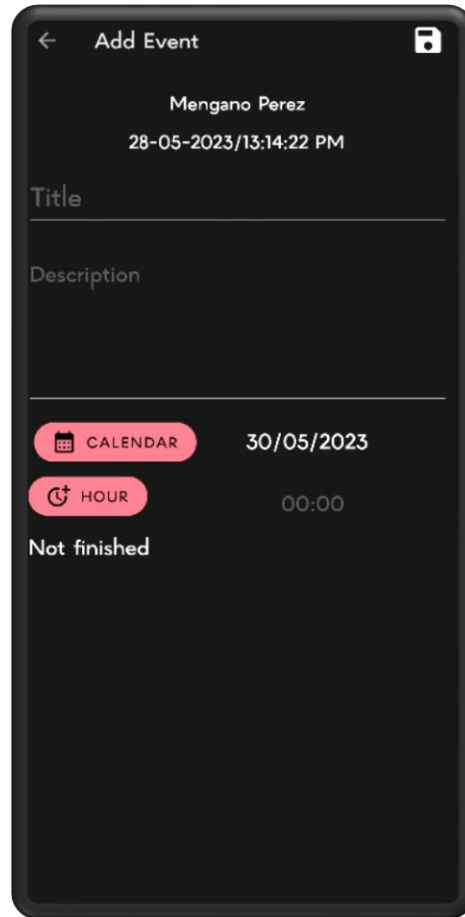


Ilustración 13 Add Event (Oscuro)

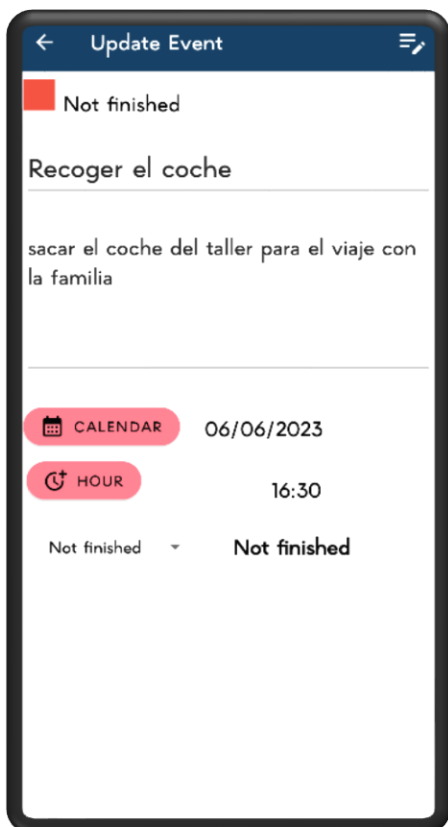


Ilustración 15 Update Event (Claro)

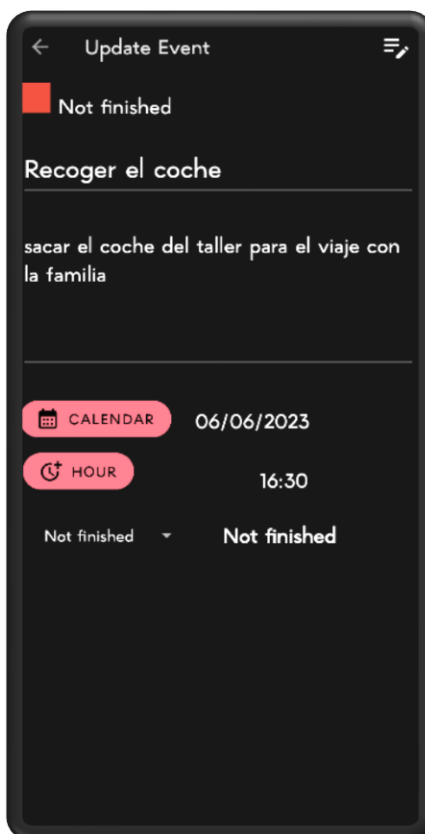


Ilustración 16 Update Event (Oscuro)

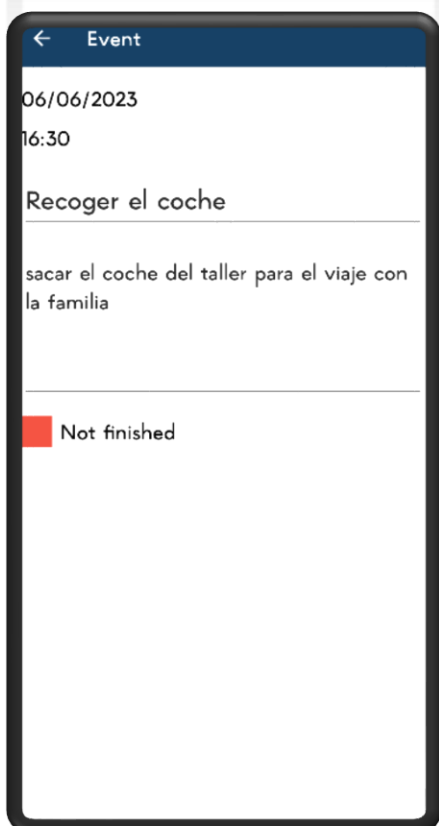


Ilustración 18 List Event (Claro)

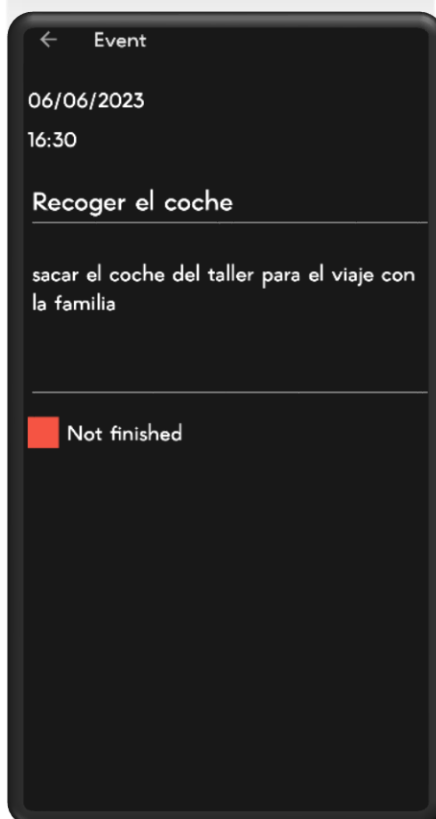


Ilustración 17 List Event (Oscuro)

8.2.5 Página Web

La web al estar realizada con Angular es una aplicación de página única, es decir, una aplicación web o un sitio web que cabe en una sola página con el propósito de dar una experiencia más fluida a los usuarios, como si fuera una aplicación de escritorio.

La web visualmente consta de un **toolbar** en la parte superior y un **sidenav** en el lateral izquierdo con el menú para recorrer los tres componentes que hemos generado, estos son:

- Home: donde damos a conocer la aplicación y su funcionalidad.
- Quienes somos: componente en el que hablamos un poco sobre nosotros
- Descarga: componente en el cual se puede descargar el apk de Reminder.



Ilustración 19 QR Web

8.3 Pruebas

En este apartado, se abordarán las pruebas realizadas en el proyecto, incluyendo pruebas manuales y la distribución de la aplicación entre varios testers para obtener retroalimentación y detectar posibles problemas. Aunque no se hayan realizado pruebas unitarias ni utilizado las herramientas de prueba proporcionadas por Android Studio, se ha buscado obtener información valiosa sobre el funcionamiento de la aplicación.

- **Pruebas manuales:** Dado el tiempo limitado disponible, se priorizó la realización de pruebas manuales para evaluar el rendimiento y la funcionalidad del sistema. Durante esta fase, el equipo se dedicó a explorar y probar manualmente las diversas características y escenarios de la aplicación. Se llevaron a cabo pruebas exhaustivas para identificar posibles errores, fallos en la interfaz de usuario y cualquier comportamiento inesperado.
- **Distribución entre testers:** Además de las pruebas manuales, se distribuyó la aplicación a un grupo de testers seleccionados. Estos testers representaban diferentes perfiles de usuarios y utilizaron la aplicación en diversos dispositivos y sistemas operativos. La intención era obtener una variedad de perspectivas y retroalimentación sobre el rendimiento, la usabilidad y posibles problemas técnicos. Se recopilaron los comentarios de los testers para identificar áreas de mejora y corregir posibles errores.

A pesar de no haber utilizado las pruebas unitarias ni las herramientas de prueba automatizadas proporcionadas por Android Studio en esta etapa, es fundamental reconocer su valor para asegurar la estabilidad y calidad del software.

En conclusión, aunque el tiempo limitado no permitió realizar pruebas unitarias ni utilizar herramientas de prueba automatizadas, se llevaron a cabo pruebas manuales exhaustivas y se obtuvo retroalimentación de testers externos. Estas acciones han proporcionado información valiosa para mejorar la funcionalidad y calidad general del proyecto. En etapas futuras de desarrollo, se recomienda encarecidamente, priorizar la implementación de pruebas unitarias y automatizadas con el objetivo de mejorar la robustez del sistema y facilitar la detección temprana de problemas para garantizar una mayor estabilidad y eficacia del software.



9 Mejoras

Reminder es una aplicación en continua evolución, mejorando la experiencia de usuario en cada versión publicada.

Para ello, una vez terminado el desarrollo de la aplicación, los miembros del equipo proponen ideas sobre esas mejoras a implementar, las cuáles son:

- Inicio de sesión mediante biometría, ya sea mediante huella digital o reconocimiento facial.
- Mejora del registro del usuario enviando una confirmación al email aportado mediante métodos de Firebase, asegurándonos de que es un email real y que el usuario no se ha equivocado en su escritura.
- Control de los eventos para mayor comodidad del usuario, ofreciendo filtros para que el cliente pueda organizarlos a su gusto (por ejemplo, si el evento es del trabajo del usuario, si son cumpleaños, etc)
- Implementación de notificaciones a programar por el usuario, las cuales mostrarán el título y la hora del evento con una antelación a elegir.

Es importante destacar que la implementación de las notificaciones no alcanzó los resultados deseados en esta etapa específica. Sin embargo, nuestro enfoque se centra en el aprendizaje y la superación de estos desafíos para lograr una mejor implementación en futuros intentos.

Estas mejoras buscan proporcionar una visión clara de los desafíos técnicos enfrentados durante la implementación de las notificaciones en el proyecto. Aunque estos desafíos no se superaron en esta instancia, confiamos en que la información recopilada y el análisis de los obstáculos nos permitirán tomar medidas correctivas y mejorar en futuras iteraciones.



10 Conclusiones

La culminación del proyecto Reminder nos ha dejado con una serie de conclusiones profundas que trascienden más allá de los aspectos técnicos y abarcan el crecimiento personal y profesional del equipo. A medida que reflexionamos sobre nuestra experiencia, se destacan varios aspectos que merecen ser resaltados.

En primer lugar, hemos aprendido la importancia de la perseverancia y la determinación. A lo largo del proyecto, nos enfrentamos a desafíos técnicos y obstáculos que parecían insuperables en un principio. Sin embargo, con una actitud resiliente y un enfoque centrado en la solución, logramos superar cada obstáculo y alcanzar resultados exitosos. Esta experiencia nos ha enseñado la valiosa lección de que la determinación y la voluntad de no rendirse son fundamentales para alcanzar nuestros objetivos.

Además de las habilidades técnicas adquiridas, el proyecto Reminder nos ha permitido desarrollar nuestras habilidades de trabajo en equipo y comunicación efectiva. A medida que colaborábamos estrechamente, aprendimos a escuchar y respetar las ideas de cada miembro del equipo, aprovechando la diversidad de perspectivas para lograr un enfoque más completo y creativo. La sinergia generada dentro del equipo demostró ser esencial para el éxito del proyecto y nos ha dejado una valiosa lección sobre la importancia del trabajo en equipo y la colaboración.

A medida que enfrentábamos los desafíos técnicos y trabajábamos juntos para superarlos, también nos dimos cuenta de la importancia de la planificación y la organización. El proyecto nos ha enseñado que una buena planificación, junto con una gestión efectiva del tiempo y los recursos, puede marcar una gran diferencia en el resultado final. La capacidad de establecer metas claras, dividir el trabajo en tareas manejables y llevar un seguimiento constante de nuestro progreso ha sido fundamental para mantenernos encaminados hacia el éxito.



En términos de valores, el proyecto Reminder nos ha permitido ejercitar la responsabilidad, la ética y el compromiso. Hemos asumido la responsabilidad de cumplir con los plazos establecidos, de mantenernos actualizados con las mejores prácticas de desarrollo y de garantizar la seguridad y la confidencialidad de los datos de los usuarios. Estos valores fundamentales nos han guiado en cada paso del proyecto y nos han proporcionado un marco ético sólido en el cual basar nuestras decisiones y acciones.

Además de las habilidades técnicas y los valores adquiridos, el proyecto Reminder nos ha dejado con una sensación de logro y satisfacción personal. Hemos comprobado que, con dedicación y esfuerzo, podemos materializar nuestras ideas y alcanzar resultados exitosos. Esta experiencia nos ha dado confianza en nuestras capacidades y nos ha motivado a seguir buscando nuevos desafíos y oportunidades de crecimiento.

En conclusión, el proyecto Reminder ha sido una experiencia enriquecedora que va más allá del ámbito académico. Hemos adquirido habilidades técnicas, fortalecido nuestros valores y experimentado el poder del trabajo en equipo y la determinación. Estamos orgullosos del resultado final y confiamos en que las lecciones aprendidas nos acompañarán en nuestro futuro personal y profesional.



11 Bibliografía

Contrastes de colores

[Contrast Finder, encuentra los contrastes adecuados de color para crear webs accesibles \(WCAG\) \(contrast-finder.org\)](#)

Consultas

[Stack Overflow -](#)

Documentación Android

[Desarrolladores de Android | Android Developers](#)

Documentación de Angular

[Angular - Introduction to the Angular docs](#)

Documentación de Angular Material

[Sidenav | Angular Material](#)

Documentación Firebase Authentication

[Firebase Authentication | Firebase Authentication \(google.com\)](#)

Documentación Firebase Firestore

[Agrega datos a Cloud Firestore | Firebase \(google.com\)](#)

Paleta de colores

[Color Palettes | Beautiful Color Palettes](#)

Anexos

Anexo 1: Desafíos en la implementación de las notificaciones del proyecto

En este anexo, se presentan imágenes que documentan los desafíos técnicos encontrados durante la implementación de las notificaciones en el proyecto.

Las imágenes muestran de manera resumida la implementación de la clase `WorkManager` y la utilización de sus métodos en las clases `AddEvent` y `MainActivity`.

La clase `WorkManager` es una parte fundamental de la arquitectura de Android que permite la ejecución programada y controlada de tareas en segundo plano. Es responsable de gestionar las tareas en función de diferentes condiciones, como disponibilidad de recursos o estado del dispositivo.

En la implementación específica de la clase `AddEvent`, se utiliza el `WorkManager` para programar y ejecutar tareas relacionadas con la adición de eventos.

```
package com.example.reminder;

import ...

public class WorkManagerNotification extends Worker {
    public WorkManagerNotification(@NonNull Context context, @NonNull WorkerParameters workerParams) {
        super(context, workerParams);
    }

    public static void saveNotification(Long duration, Data data, String tag){
        OneTimeWorkRequest notification = new OneTimeWorkRequest.Builder(WorkManagerNotification.class)
            .setInitialDelay(duration, TimeUnit.MILLISECONDS).addTag(tag)
            .setInputData(data).build();

        WorkManager instance = WorkManager.getInstance();
        instance.enqueue(notification);
    }
}
```

Ilustración 20 Anexo WorkManager 1



```
@NonNull
@Override
public Result doWork() {
    String title = getInputData().getString( key: "title");
    String description = getInputData().getString( key: "description");
    int id = (int) getInputData().getLong( key: "idNotification", defaultValue: 0);

    notificationFirebase(title, description);
    return Result.success();
}
```

Ilustración 21 Anexo WorkManager 2

```
public void notificationFirebase(String title, String description){
    String id = "message";
    NotificationManager nm = (NotificationManager) getApplicationContext().getSystemService(Context.NOTIFICATION_SERVICE);
    NotificationCompat.Builder builder = new NotificationCompat.Builder(getApplicationContext(), id);

    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
        NotificationChannel nc = new NotificationChannel(id, name: "new", NotificationManager.IMPORTANCE_HIGH);
        nc.setDescription("Notification FirebaseCloudManager");
        nc.setShowBadge(true);
        assert nm!= null;
        nm.createNotificationChannel(nc);
    }

    builder.setAutoCancel(true)
        .setWhen(System.currentTimeMillis())
        .setContentTitle(title)
        .setTicker("New Notification")
        .setSmallIcon(R.mipmap.ic_launcher)
        .setContentText(description)
        .setContentInfo("New");

    Random random = new Random();
    int idNotify = random.nextInt( bound: 100000);

    assert nm!= null;
    nm.notify(idNotify, builder.build());
}
```

Ilustración 22 Anexo WorkManager 3

```
//Method to delete the notification.  
public void deleteNotification(String tag){  
    WorkManager.getInstance(this).cancelAllWorkByTag(tag);  
}
```

Ilustración 23 Anexo MainActivity

```
String tag = generateKey();  
Calendar currentDateTime = Calendar.getInstance();  
long alertTime = calendar.getTimeInMillis() - (2 * 60 * 1000) - System.currentTimeMillis();  
  
//Long alertTime = calendar.getTimeInMillis() - System.currentTimeMillis();  
int random = (int) (Math.random() * 50 +1);  
  
Data data = saveData(title, hour, random);  
WorkManagerNotification.saveNotification(alertTime, data, tag);
```

Ilustración 24 Anexo AddEvent 1

```
// Method to generate a random key  
public String generateKey() { return UUID.randomUUID().toString(); }  
  
// Method to save data to send it to the WorkManagerNotification class.  
public Data saveData(String title, String description, int idNotification){  
    return new Data.Builder()  
        .putString("title", title)  
        .putString("description", description)  
        .putInt("idNotification", idNotification).build();  
}
```

Ilustración 25 Anexo AddEvent 2



}
}