



Aplicación Android de gestión de guardias en centros docentes – On Call App (OCA)

María Isabel Marín Altaba

Máster Desarrollo de aplicaciones para dispositivos móviles – Curso 2021/2022

Tutor: Pau Dominkovics Coll

Fecha de entrega: 30/05/2022



Esta obra está sujeta a una licencia de Reconocimiento-
NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Aplicación Android de gestión de guardias en centros docentes – On Call App (OCA)
Nombre del autor:	María Isabel Marín Altaba
Nombre del consultor:	Pau Dominkovics Coll
Fecha de entrega (dd/mm/aaaa):	30/05/2022/
Titulación:	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>

Resumen del Trabajo (máximo 250 palabras):

Este proyecto describe el desarrollo de una aplicación móvil nativa en Android cuyo objetivo se centra en la gestión de guardias en centros de educación.

La aplicación está orientada para que sus usuarios sean los docentes y el equipo directivo del centro, sobre el cual recae la mayor parte del trabajo relacionado con este punto.

Los objetivos principales de la aplicación son: ahorrar tiempo al equipo docente; automatizando la gestión de las guardias; mantener a los usuarios actualizados e informados en todo momento, independientemente de donde se encuentren; y mantener una equidad y parcialidad a la hora de asignar profesores de guardia.

Para obtener estos objetivos se han implementado diferentes funcionalidades y requisitos que proporcionan dichos servicios al usuario. La aplicación permite tener un visionado actual y claro de las guardias de cada día en el centro escolar, facilita la tarea de asignar a los profesores de guardia a las diferentes tareas y permite al usuario incluir nuevas ausencias en el sistema.

En este documento desarrolla como se ha llevado a cabo el diseño de la aplicación, la implementación y los resultados obtenidos.

La implementación del proyecto se ha realizado usando Android Studio. Se ha hecho uso de una base de datos en Firebase, la cual se ha unido a la lógica de la aplicación mediante arquitectura MVVM (Model View ViewModel). De esta forma el usuario se mantiene actualizado en todo momento, cumpliendo así uno de sus objetivos.

Abstract (in English, 250 words or less):

This project describe the developing of a native mobile application on Android, whose target is focused in the on call management in the high school.

The application is created to be used by teachers and directive team of high school, this one last is who works mainly in this point.

The main targets of the applications are: to save time to the directive team; to automatize the on call management; to keep update the users all the time; and to keep the equalite and objectivity in the task related to assign teachers to on call.

Different functionalities and requirements have been implemented to obtain these targets. The application lets the user watch the list of on-call teachers in the high school, it makes that the task to assign a teacher to on-call be easier and it lets the users join new absences in the system.

This document shows how the design and implementation work have been developed and the results obtained.

The implementation has been made using Android Studio and Firebase to keep data. The MVVM (Model View ViewModel) architecture has been implemented to relate the app logic to the data. This way helps users to keep updated all the time, and then the targets are achieved.

Palabras clave (entre 4 y 8):

Android, Firebase, Firestore, MVVM, guardias, profesores.

Índice

1	Introducción	1
1.1	Contexto y justificación del Trabajo	1
1.1.1	Introducción sobre la temática – Gestión de las guardias en un centro docente 1	
1.1.2	Puntos de mejora o necesidades observadas	3
1.1.3	Benchmarking	4
1.2	Objetivos del Trabajo	7
1.3	Enfoque y método seguido	8
1.4	Planificación del Trabajo	9
1.5	Breve resumen de productos obtenidos	12
1.6	Breve descripción de los otros capítulos de la memoria	12
2	Diseño	13
2.1	Usuarios	13
2.2	Contextos de uso	15
2.2.1	Escenario 1	15
2.2.2	Escenario 2	16
2.2.3	Escenario 3	17
2.3	Diseño conceptual	18
2.3.1	Clasificación de la información	18
2.3.2	Árbol de navegación	19
2.4	Prototipado	21
2.4.1	Pestaña principal – Pestaña de Guardias	23
2.4.2	Pestaña Nueva Ausencia	23
2.4.3	Pestaña Usuario	23
2.4.4	Administración	23
2.5	Casos de uso	24
2.5.1	Relacionados con la identificación del usuario	24
2.5.2	Relacionados con las funcionalidades de la app para el usuario	26
2.5.3	Relacionados con la gestión de la app	28
2.6	Diseño de la arquitectura	32
2.6.1	Absence (Ausencias)	32
2.6.2	GroupOnCall (Grupos de guardias)	33
2.6.3	User (Usuario)	33
3	Implementación	35
3.1	Herramientas utilizadas	35
3.1.1	Android Studio	35
3.1.2	Firebase	35
3.1.3	Git	35

3.2	Modelo de trabajo	35
3.2.1	UI (view).....	35
3.2.2	Models (data).....	36
3.2.3	ViewModel	36
3.2.4	Domain\Data\NetWork.....	37
3.3	Base de datos	37
3.4	Pantallas	37
3.4.1	Pantalla - Inicio de sesión	37
3.4.2	Pantalla Principal - Guardias.....	38
3.4.3	Pantalla Secundaria - Grupo de guardia.....	40
3.4.4	Pantalla Secundaria - Nueva Ausencia	41
3.4.5	Pantalla Usuarios.....	42
3.5	Test de validación.....	43
3.5.1	Pantalla Principal – Guardias	43
3.5.2	Pantalla Secundaria – Grupo de guardias.....	44
3.5.3	Pantalla Secundaria – Nueva ausencia.....	45
3.5.4	Pantalla Secundaria – Usuario	46
3.5.5	Pantalla Inicio de sesión	46
4	Conclusiones	48
4.1	Planificación.....	49
4.2	Funcionalidades y requisitos	50
4.2.1	Funcionalidades y requisitos cumplidos.....	50
4.2.2	Funcionalidades y requisitos no cubiertos	50
4.3	Líneas futuras de desarrollo	51
4.3.1	Mejoras de la aplicación.....	51
4.3.2	Futuros desarrollos	51
5	Glosario	53
6	Bibliografía.....	54
7	Anexos	56
7.1	Resultados de la encuesta.	56

Lista de figuras

Ilustración 1-1 Planificación del Trabajo – Hoja de cálculo basada en un diagrama de Gannt. Versión 04/04/22.....	11
Ilustración 2-2 Árbol de navegación	20
Ilustración 2-3 Prototipo mano alzada: Pantalla principal - Guardias.....	21
Ilustración 2-4 Prototipo mano alzada: De izquierda a derecha (Pantalla Nueva ausencia – Pantalla Usuarios – Pantalla configuración para Administrador).....	22
Ilustración 2-5 Prototipo mano alzada: Pantalla de inicio.....	22
Ilustración 2-6 Diagrama UML de las clases de la aplicación	34
Ilustración 2-7 Esquema MVVM.....	34
Ilustración 3-1 Pantalla de inicio de sesión	38
Ilustración 3-2 Pantalla principal – Guardias (Izquierda: El profesor de guardia no ha sido seleccionado; Derecha: El profesor de guardia ha sido seleccionado).....	39
Ilustración 3-3 Pantalla Grupos (secundaria) – Muestra el listado de grupos de guardias para cada hora y día de la semana	40
Ilustración 3-3 Pantalla Nueva Ausencia (secundaria) – Muestra el formulario a rellenar por el usuario.	41
Ilustración 3-5 Pantalla Grupos (secundaria) – Muestra el listado de grupos de guardias para	42

Lista de Tablas

Tabla 1-1 Horario semanal de un profesor de secundaria de física y química.	2
Tabla 1-2 Aplicaciones web para la gestión de la actividad docente.	5
Tabla 1-3 Aplicaciones web para la gestión de un centro educativo.	5
Tabla 1-4 Aplicaciones web para la gestión de un centro educativo y del aula.....	5
Tabla 1-5 Aplicaciones web para la gestión de un centro educativo – Comparativa de funcionalidades.....	6
Tabla 1-6 Planificación del Trabajo – Horas asignadas a las principales tareas.	9
Tabla 2-1 Casos de uso relacionados con la identificación del usuario – CU 001.	25
Tabla 2-2 Casos de uso relacionados con la identificación del usuario – CU 002.	25
Tabla 2-3 Casos de uso relacionados con la identificación del usuario – CU 003.	26
Tabla 2-4 Casos de uso relacionados con la identificación del usuario – CU 004.	26
Tabla 2-5 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 005.....	26
Tabla 2-6 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 006.....	27
Tabla 2-7 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 007.....	28
Tabla 2-8 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 008.....	29
Tabla 2-9 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 009.....	30
Tabla 2-10 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 010.....	31

Tabla 2-11 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 011.....	32
Tabla 3-1 Prueba de validación – Pantalla principal.....	44
Tabla 3-2 Prueba de validación – Pantalla Grupo de guardias.....	45
Tabla 3-3 Prueba de validación – Pantalla Grupo de guardias.....	45
Tabla 3-4 Prueba de validación – Pantalla Usuario.....	46
Tabla 3-5 Prueba de validación – Pantalla Inicio de sesión (AuthActivity.js).....	47

1 Introducción

1.1 Contexto y justificación del Trabajo

La aplicación propuesta en este proyecto final de máster se ha centrado en ciertas necesidades observadas en el sector educativo, concretamente en la gestión de las guardias de profesores en los centros educativos cuando un docente se ausenta de su puesto de trabajo.

El motivo principal de esta elección es la cercanía y la familiaridad de la autora del proyecto con este sector. Trabaja como profesora en un instituto de educación secundaria, y ha podido comprobar que los jefes de estudios deben invertir mucho tiempo en el campo de la gestión de las guardias, obteniendo unos resultados que no siempre facilitan el control de las ausencias por parte de los profesores de guardia

Se ha observado un campo sobre el que indagar para ver en qué puntos se podría aportar valor y mejorar aquellas situaciones que se consideren no óptimas.

1.1.1 Introducción sobre la temática – Gestión de las guardias en un centro docente

Este proyecto se va a centrar en los centros educativos públicos de la comunidad autónoma de Aragón, concretamente en los centros de educación secundaria; ya que la autora del proyecto trabaja en dicha comunidad y en este tipo de centros. Sin embargo, todo el proyecto es extrapolable a cualquier otro centro y a cualquier comunidad autónoma.

Los profesores de la comunidad autónoma de Aragón que trabajan a jornada completa, tienen una jornada lectiva de 20 horas semanales, a las que se les debe sumar un porcentaje de horas complementarias; concretamente un 15% de las lectivas que son 7 horas a la semana [1].

- Las horas lectivas son aquellas en las que el profesor imparte docencia directa con el alumnado;
- Las horas complementarias se destinan a las siguientes tareas:
 - AP: atención a padres para resolver dudas de la asignatura. Obligatorias para todo el profesorado, como máximo 1 hora a la semana;
 - TP: tutoría con padres. Obligatorias para todos los tutores, como máximo 1 hora a la semana;
 - AC: colaboración en actividades complementarias. No obligatoria, como máximo 2 horas a la semana;
 - PP: preparación de prácticas. No obligatoria, como máximo 2 horas a la semana;
 - Reunión semanal del departamento al que pertenezca el profesor;
 - Horas de guardia, como mínimo 3 horas a la semana.

El número de horas que se destina a horas de guardia suele ser aproximadamente de 3 horas a la semana. Estas 3 horas se incluyen dentro del horario semanal del profesor completándolo con las horas lectivas. Es decir, que durante todo el curso escolar se mantienen las horas de guardia de cada profesor: por ejemplo, un profesor de matemáticas podría tener guardia todos los lunes a 2º hora, martes a 6º hora, jueves a 5º hora y viernes a 6º hora. De este modo, los profesore que tiene guardia a una hora concreta, coincidirán durante todo el curso en esa hora.

Un profesor que este contratado a jornada parcial, tendrá la parte proporcional de horas complementarias que le corresponda en función del número de horas lectivas que tenga en contrato; las horas complementarias serán un 15% de las lectivas. Estos profesores también podrán realizar horas de guardia.

Si un profesor está de baja laboral, será sustituido por un profesor interino; el cual tendrá el mismo horario que el profesor al que sustituye y por lo tanto también las mismas horas de guardia.

A continuación, se muestra un ejemplo de horario semanal de un profesor, en el que están coloreadas de azul las horas complementarias, y resaltadas en negrita las tres horas de guardia.

Horas\Días	Lunes	Martes	Miércoles	Jueves	Viernes
8:30 / 9:20	FQ 4ªA	FQ 4º A	FQ 3ºB	FQ 3ªA	FQ 4ªA
9:25 / 10: 15	CC 1º Bt A	FPB 1º E	FPB 1º E	Reunión Dpto.	GUARDIA
10:20 / 11:10	FQ 4ªB	CC 4ªB	TP	FPB 1ªE	PP
11:10 / 11:40	RECREO	RECREO	RECREO	RECREO	RECREO
11:40 / 12:30	FQ 3ªA	AF		GUARDIA	Reunión tutores
12:35 / 13:25	FPB 1º E	FQ 4ªB	FQ 4ªB	CC 1º Bt. A	GUARDIA
13:30 / 14:20		FQ 3ªB	CC 4ªB		

Tabla 1-1 Horario semanal de un profesor de secundaria de física y química.

Durante las horas de guardia, los profesores tienen que desempeñar los siguientes trabajos:

- Comprobar que en todas las clases se encuentra un profesor con sus alumnos; Si no fuese así, debido a que el profesor correspondiente se ha retrasado, debería permanecer en dicha aula hasta que llegase, para evitar que los alumnos estuviesen solos;
- Permanecer en la sala de profesores por si surge alguna incidencia o se les necesita en algún momento por cualquier motivo.
- Sustituir en el aula, a un profesor que se ha ausentado del centro por el motivo que sea. Las tareas que debe realizar el profesor de guardia, en estos casos son las siguientes:
 - Pasar lista para comprobar que no se ha ausentado ningún alumno y registra las faltas de asistencia si las hubiese;
 - Entregar la tarea que haya dejado el profesor ausente para que la realicen los alumnos durante la hora lectiva;
 - Mantener el orden y el control en la clase;

Normalmente, desde jefatura se notifica que profesores estarán ausentes durante la jornada lectiva, para que así, los profesores de guardia sepan a quienes tienen que sustituir y en que aula. Se suele usar una Excel que se muestra en una pantalla de ordenador en la sala de profesores, donde se indican las aulas y los grupos que no tendrán profesor ese día. También es jefatura la que proporciona la tarea que el profesor ausente ha dejado para que los alumnos la realicen.

El número de profesores de guardia en cada una de las horas del horario escolar, varía en función de la disponibilidad de los mismos y del número de profesores del centro. Se podría decir que el número suele encontrarse entre 3 y 6 profesores. Si el número de incidencias y ausencias no puede ser cubierto por el número de profesores que se encuentra de guardia, el jefe de estudios tendría que resolverlo según el protocolo que tenga establecido el centro.

Debido a que las tareas que tiene que desempeñar el profesorado durante las horas de guardia es muy variado, se intenta que todos los profesores vayan realizando un porcentaje parecido de cada una de las tareas. Es decir, si en una hora de guardia hay 4 profesores, se debería evitar que uno de ellos se encargue siempre de sustituir a los

profesores ausentes mientras los otros realizan el resto de tareas. Lo ideal es que hubiese un reparto equitativo, ya que los profesores que permanecen en la sala de profesores pueden trabajar en su asignatura adelantando tareas; sin embargo, aquel que está en el aula con los alumnos sustituyendo a un compañero, difícilmente podrá trabajar en su asignatura.

Este reparto equitativo no siempre se da, ya que depende de la voluntad y del consenso que haya entre los profesores que compartan hora de guardia. Este tema puede dar pie a discusiones o malentendidos que hacen que el ambiente de trabajo sea más estresante.

En algunos centros, los jefes de estudio se encargan de hacer este reparto el día anterior, conociendo las ausencias que van a tener el próximo día. Esta tarea requiere de tiempo y puede estar sujeta a cambios que habría que resolver en el momento, haciendo perder mucho tiempo más a los jefes de estudio. Por ejemplo: el jefe de estudios asigna a un profesor, en su hora de guardia, sustituir a un compañero que se va a ausentar. Sin embargo, el profesor de guardia se ausenta ese día por motivos de salud sin poder haber avisado con tiempo. Otro profesor de guardia tendrá que cubrir la ausencia que se le había asignado y además las horas lectivas de dicho profesor tendrán que ser cubiertas por profesores de guardia.

1.1.2 Puntos de mejora o necesidades observadas

Se ha discutido con profesores de diferentes centros sobre la gestión de las horas de guardia, para conocer diferentes opiniones e inquietudes sobre el tema. Y así, conocer que puntos son los que se considera que deberían ser mejorados y que necesidades existen que no son cubiertas con el método actual.

Se han recopilado los siguientes puntos:

- El hecho de que sea jefatura quien tenga que gestionar las ausencias de los profesores, hace que se cree un cuello de botella, haciendo perder tiempo al personal docente y a la propia jefatura de estudios;
- El reparto de las diferentes tareas de las guardias debe ser equitativo. Tendría que registrarse el número de sustituciones en aula que hace cada profesor a lo largo del curso, y así, de esta forma se evitaría que esta tarea siempre recaiga sobre los mismos compañeros;
- Los profesores y jefatura quieren que se les libere de la responsabilidad de decidir que profesor es el que debe cubrir la ausencia de un compañero, a pesar de tener un registro del número de guardias de clase que ha hecho cada profesor. Es decir, que este proceso sea automático y a la vez equitativo como se ha mencionado en el punto anterior;
- Desde primera hora de la jornada, cada profesor debería saber si tiene que hacer una sustitución en aula durante una guardia, para así organizarse mejor. Cualquier incidencia o cambio relacionado con las guardias, debería notificarse inmediatamente a los profesores que tienen guardia durante esa jornada;
- Si los profesores pudiesen acceder a la información relacionada con la gestión de las guardias, desde cualquier punto del centro, y no sólo desde la sala de profesores; ayudaría mucho a evitar situaciones de descontrol;
- Sería de gran ayuda saber en cada momento que aulas se encuentran vacías, ya que es posible que un profesor de guardia necesite usar un aula con: un grupo de alumnos que ha sido expulsado de clase; un alumno que debe repetir un examen fuera de su aula habitual; y otras incidencias que se den durante la hora de guardia;

- Cuando un profesor es contratado para sustituir a otro profesor que se encuentra de baja laboral, desconoce el método que se usa en el centro para la gestión de guardias, además de que no conoce a los profesores y ellos tampoco lo conocen a él. Este punto causa mucho descontrol y situaciones en las que los profesores que tienen mejor conocimiento del centro, acaban realizando la mayoría de las tareas de guardia. Si esta gestión fuese más metódica y estuviese digitalizada, ayudaría a que los nuevos profesores se adaptasen mejor y más rápido a la dinámica del centro.

En resumen, se han identificado los siguientes puntos de mejora:

- Ahorrar tiempo a jefatura y al personal docente, a la hora de gestionar las ausencias y de repartir las tareas de la hora de guardia, además de garantizar equitatividad y objetividad en el reparto;
- Responsabilidad nula sobre cualquier usuario (jefatura y profesores) sobre el reparto de las tareas de guardia; esto se lograría si el método fuese automatizado;
- Que todos los usuarios estuviesen informados en todo momento de la gestión de las horas de guardia, así como de cualquier incidencia o cambio, e independientemente del lugar en el que se encuentren.

Observando los puntos de mejora, se ha considerado diseñar una aplicación web para mejorar el método de gestión de las horas de guardia en los centros de educación y solucionar los problemas encontrados.

1.1.3 Benchmarking

Se ha desarrollado un análisis de la competencia buscando productos o servicios similares al que se quiere diseñar, o que den soluciones a los problemas indicados anteriormente.

No se ha encontrado ninguna herramienta cuya principal funcionalidad fuese similar a la presentada en este documento, gestionar las guardias del personal docente.

Sin embargo, si se han encontrado muchas y diferentes herramientas enfocadas en la gestión del centro y en la gestión del aula [2].

Aplicaciones para la gestión de la actividad docente en el aula.

Estas aplicaciones ayudan al docente en las tareas diarias de gestión del aula, como: registrar faltas de asistencia; planificación y programación del curso; seguimiento de las tareas; mantener comunicación con las familias; entre otras. Funcionan como un cuaderno del profesor digital.

Aplicaciones para la gestión de un centro educativo.

Estas aplicaciones están centradas en las tareas desarrolladas por el personal administrativo del centro y por jefatura. Ayuda en las tareas de: matriculación; facturación; creación y gestión de horarios; gestiones administrativas; entre otras.

Aplicaciones para la gestión del centro y para la gestión del aula.

Aquí encontramos aplicaciones que muestran funcionalidades para los dos puntos de arriba, pueden ser usadas por el personal administrativo del centro y también por el personal docente en el aula.

En las siguientes tablas se indican cada una de las aplicaciones encontradas para cada uno de los grupos identificados.






Aplicaciones para la gestión de la actividad docente		
Additio App	https://www.additioapp.com/es	
ClassLink	https://www.classlink.com/	
Clickedu	https://clickartedu.com/inicio-plataforma-colegios.html	
SM Educamos	https://www.grupo-sm.com/es/sm-educamos	
Classroom	https://classroom.google.com/u/0/h?hl=es	

Tabla 1-2 Aplicaciones web para la gestión de la actividad docente.





Aplicaciones para la gestión de un centro educativo		
aGora	https://www.agora-erp.com/	
Alexia		
ApliAula	http://www.apliaula.com/	
Aula1	https://www.aula1.com/	
Classlife		
DocDF	http://www.grupocfdeveloper.com/plataforma-academias-colegios-jardines-infantiles.php	

Tabla 1-3 Aplicaciones web para la gestión de un centro educativo.

Aplicaciones para la gestión de un centro y del aula		
CifraEducación	https://www.cifraeducacion.com/	

Tabla 1-4 Aplicaciones web para la gestión de un centro educativo y del aula.

De las aplicaciones mostradas arriba, se va a estudiar aquellas cuyo objetivo es la gestión de un centro educativo. Estas son las que podrían mostrar similitudes con las funcionalidades y puntos interesantes que quizás serían válidos para la aplicación que se muestra en este proyecto. Los resultados obtenidos se muestran en la siguiente tabla.

FUNCIONALIDADES	aGora	Alexia	ApliAula	Aula1	ClassLife	DocDF	CifraEducación
Generación de horarios.	N/S	SI	SI	N/S	N/S	N/S	N/S
Comunicación en tiempo real de entornos seguros.	SI	SI	SI	SI	SI	SI	SI
Configurar reservas del comedor.			SI				
Gestión del horario del personal.	SI			SI	SI	SI	
Gestión de la información de los docentes.	SI	SI	SI	SI	SI	SI	SI
Reserva de aulas.					SI		
Gestión de la asistencia de los docentes.					SI		

Tabla 1-5 Aplicaciones web para la gestión de un centro educativo – Comparativa de funcionalidades.

Las conclusiones que se obtienen de la tabla anterior es que la mayoría de las aplicaciones centradas en la gestión de centros educativos se centra en:

- Mantener una comunicación activa entre los diferentes actores: familias, alumnos y docentes;
- Facilitar herramientas para el trabajo diario de los equipos directivos relacionado con los alumnos: elaboración de horarios, por ejemplo;
- Gestionar la información de los docentes: horarios, grupos, alumnado, entre otros.

Pocas aplicaciones se han enfocado también en la reserva de diferentes actividades, como el comedor y las aulas.

Sólo una de las aplicaciones tiene en cuenta la gestión de la asistencia de los docentes. Sin embargo, no es muy extensa la información que da sobre esta funcionalidad, no pudiendo así ser útil para este proyecto.

Se puede concluir que muy pocas aplicaciones tienen en cuenta la gestión de las ausencias de los profesores.

Estudiando la bibliografía relacionada con el tema concreto de este proyecto, se han encontrado dos únicas referencias a aplicaciones cuyo objetivo fuese la gestión de las guardias en centros docentes. Las dos hacen referencia a proyectos de fin de carrera [3] en 2017 y [4] en 2012; y en los dos se ha desarrollado una aplicación web. En el desarrollo de este proyecto se tendrán en cuenta las consideraciones y conclusiones obtenidas en estos dos proyectos mencionados, a pesar de que las herramientas utilizadas han evolucionado en este tiempo.

1.2 Objetivos del Trabajo

La función principal de esta aplicación será la de mantener informados al equipo docente sobre las ausencias del profesorado, es decir de las guardias que deben realizarse; repartir las diferentes tareas asociadas a las horas de guardia entre los profesores, de forma equitativa y automática; así como permitir a los docentes registrar las ausencias que tengan previstas.

Los objetivos principales serían:

- Ahorrar tiempo a jefatura y al personal docente en la tarea de gestionar las ausencias de los profesores;
- Lograr equidad y objetividad en el reparto de las tareas de las horas de guardia entre los profesores;
- Mantener informados y actualizados a todos los profesores sobre la gestión de las guardias, sin importar el lugar donde se encuentren ni si llevan poco tiempo trabajando en el centro;

Esta aplicación facilitaría el trabajo del personal docente, disminuyendo posibles situaciones difíciles de resolver y liberando de trabajo al personal de jefatura.

Se ha decidido llamar a la aplicación OCA, siendo las siglas de (*On call – App*), que traducido al castellano significa (*en servicio de guardia – Aplicación*). Se ha considerado que es un nombre corto, fácil de recordar y de pronunciar.

A continuación, se enumeran los diferentes requerimientos funcionales y no funcionales de la aplicación que se acaba de introducir:

Requisitos funcionales

- Visualización de una tabla con las tareas asociadas a las guardias¹ y los profesores a los que se les ha asignado cada una de ellas;
- Notificación de cualquier incidencia o cambio asociado a las guardias;
- Posibilidad de acceder a la app para registrar una ausencia prevista;
- Posibilidad de acceder a un listado de las aulas vacías para poder hacer uso de ellas;

Requisitos no funcionales

- Facilitar la navegación mediante una interfaz fácil e intuitiva;
- Garantizar el proceso de inicio de sesión seguro;
- Almacenar y consumir los datos mediante un servicio de base de datos;
- Actualizar la información que se muestra por la interfaz, en el momento que haya una modificación de la misma;
- Implementar una aplicación web multidispositivo;

1.3 Enfoque y método seguido

El objetivo de este proyecto es facilitar la gestión de las guardias en los centros docentes. Existen diversas aplicaciones que se centran en la gestión de un centro educativo y también en la gestión del trabajo del profesor. Pero la mayoría de ellas no se enfocan en la gestión de guardias, priorizan otras funcionalidades como: la gestión de las horas lectivas, la programación didáctica, la gestión y creación de horarios, la gestión administrativa del centro, entre otras.

De modo que esta aplicación se va a desarrollar como un producto nuevo, teniendo en cuenta las consideraciones y conclusiones obtenidas en el estudio de aplicaciones similares.

Método seguido

Primeramente, se ha realizado un visionado del contexto en el que vamos a trabajar, consultando a futuros posibles usuarios, profesores de diferentes centros, sobre posibles mejoras o problemas que encontraban en la gestión de las guardias docentes en sus centros. También se ha incluido las conclusiones y opiniones de la autora del proyecto ya que también es profesora en un centro educativo. Una vez recopilada toda esta información se han definido los objetivos principales y las funcionalidades de la aplicación.

Estudiando el entorno donde se va a utilizar, se cree que la mejor opción sería desarrollar una aplicación web para multidispositivo. De esta forma, podrá ser usada:

- en ordenador de sobremesa por parte de jefatura; pudiendo acceder a la aplicación en modo administrador y así gestionar todas las posibles funcionalidades de la misma;
- en móvil (Android o IOS) por parte de los docentes, desde cualquier aula o punto del centro; de esta forma no se discriminará la tecnología que cada profesor tenga;
- en pantalla táctil desde la sala de profesores. Muchos centros tienen una pantalla táctil en la sala de profesores, en esta se podría visualizar la app para que pudiese ser usada desde esa estancia por los profesores de guardia;

¹ Cuando se habla de guardias, se refiere a la sustitución de un profesor en su aula porque este se ha ausentado del centro.

Otro tipo de aplicación no daría tanta versatilidad en cuanto a dispositivos de uso.

Sin embargo, enfocándonos en el alcance del proyecto, en el tiempo disponible y en la formación del alumno, se ha optado por desarrollar una aplicación móvil Android. Dejando el desarrollo de una aplicación web como desarrollo futuro compatible con la aplicación que aquí se muestra.

Tras conocer los objetivos y definir las funcionalidades, se ha procedido a llevar a cabo el diseño de la aplicación. Tras este, se siguió con la implementación, tanto del front-end (la cara visible de la web) como del back-end (donde se almacena la información necesaria para su ejecución).

Por último, el producto obtenido ha sido sometido a una evaluación y testeo, para poder indicar posibles mejoras, líneas de desarrollo futuro y conclusiones del proyecto.

1.4 Planificación del Trabajo

El trabajo total del proyecto se ha dividido en cuatro tareas, coincidiendo cada una de ellas con las entregas de las PEC. Cada una de estas tareas está dividida en diferentes secciones, y a su vez, en el caso del desarrollo front-end y back-end, también están subdivididas en diferentes subsecciones.

A cada una de estas tareas, secciones y subsecciones se les ha asignado un número de horas, así como una fecha de inicio y otra de fin (una horquilla de tiempo para su ejecución).

A la hora de planificar el trabajo, se ha considerado que cada semana se invertirán 21 horas en el proyecto. Estas se repartirán de igual forma entre los días laborables y los festivos, asignándose a cada día 3 horas de trabajo.

La alumna se encuentra desempleada en estos momentos, de modo que tiene disponibilidad tanto de mañana como de tarde. Además del TFM, está cursando otras asignaturas. A las otras asignaturas del máster les dedica las horas de las mañanas del lunes a sábado, de modo que quedan las tardes del lunes al domingo para invertir en el TFM.

La representación de la planificación del trabajo y la división de las tareas se ha realizado usando la herramienta Excel, elaborando una hoja de cálculo que se actualizaría automáticamente si se hace cualquier modificación. Esta hoja de cálculo está basada en un diagrama de Gantt y se muestra en la Ilustración 1-1.

La planificación anterior también muestra el grado de progreso del proyecto, de forma que se podrá hacer un seguimiento del mismo, y modificar si es necesario el reparto de horas de las diferentes tareas.

En la siguiente tabla se puede observar de forma esquemática el número de horas que se ha asignado a cada tarea y sección.

TAREA	INICIO	FIN	HORAS
PEC1 – Plan de trabajo	16-02-2022	09-03-2022	51
PEC2 – Diseño	10-03-2022	04-04-2022	48
PEC3 - Implementación	05-04-2022	16-05-2022	111
PEC4 – Memoria	07-03-2022	30-05-2022	81
TOTAL			300

Tabla 1-6 Planificación del Trabajo – Horas asignadas a las principales tareas.

Hay que tener en cuenta que en la tarea 4 Memoria, se ha incluido el tiempo invertido en cada capítulo de la memoria que se va a incluir en cada una de las entregas

anteriores: PEC1, PEC2 y PEC3; de forma que la duración de la tarea Memoria se solapa en el tiempo con el resto de tareas.

La tarea que más tiempo tiene invertido es la de implementación, ya que abarca el grueso del proyecto, el desarrollo de la aplicación web: front-end y back-end.

Observando la tabla anterior, podríamos pensar que se ha invertido más tiempo en la PEC1 Plan de trabajo, que en la PEC2 Diseño; pero no es así, si a cada una de ellas le sumamos el tiempo que hemos invertido en completar parte de la memoria, observaremos que la PEC2 Diseño tiene asignado más tiempo.

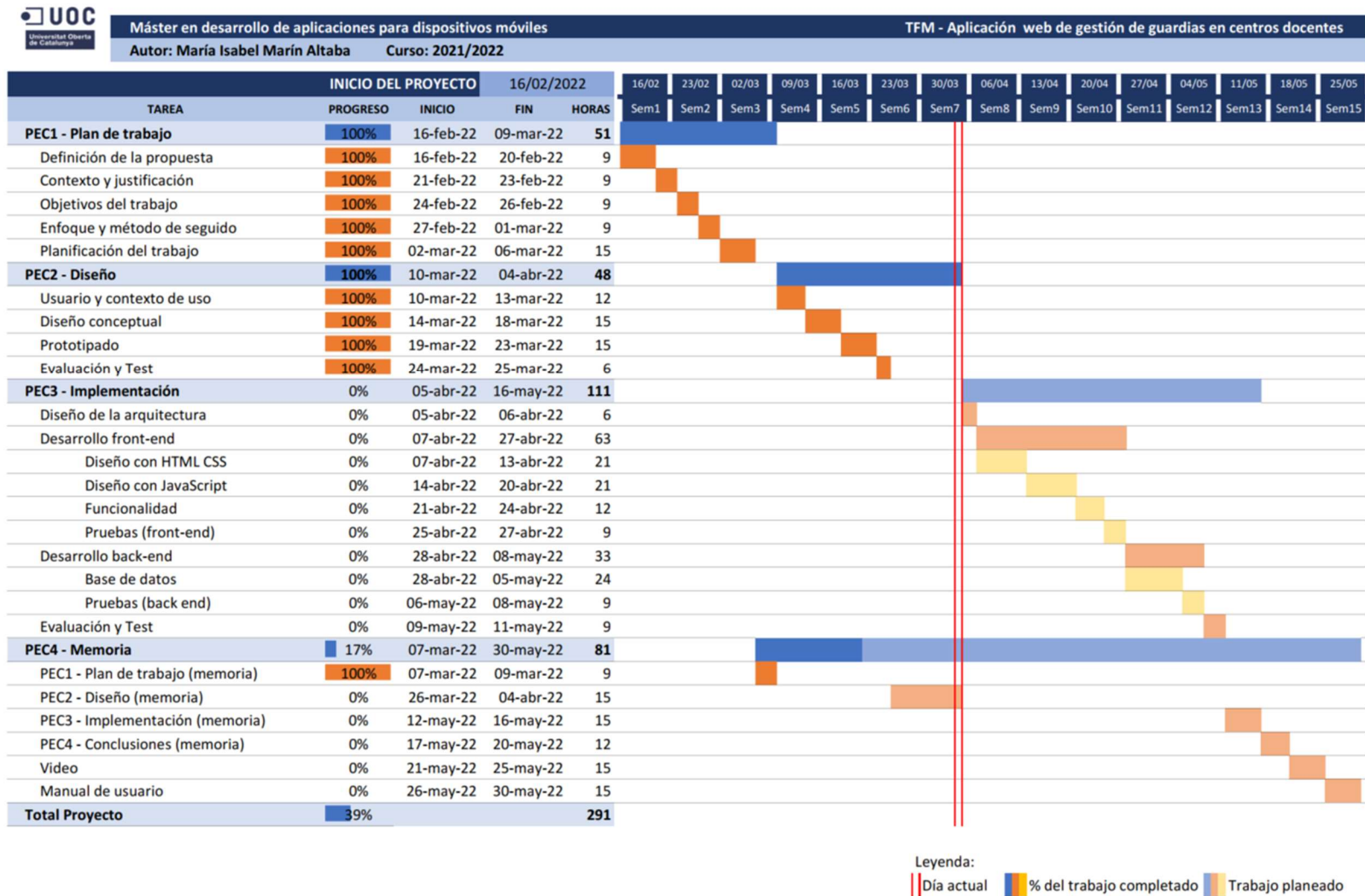


Ilustración 1-1 Planificación del Trabajo – Hoja de cálculo basada en un diagrama de Gantt. Versión 04/04/22

1.5 Breve resumen de productos obtenidos

Los productos resultantes del proyecto son:

- Una aplicación Android desarrollada para dispositivo móvil de la misma tecnología, que permitirá gestionar las guardias en los centros docentes.
- Un manual de usuario que será de guía en el uso de la aplicación;

1.6 Breve descripción de los otros capítulos de la memoria

El documento está formado por cuatro secciones:

1. Introducción:

Donde se introduce el contexto y justificación del trabajo; se indican los puntos de mejora o las necesidades que los posibles usuarios exigen; se realiza un estudio de Benchmarking para conocer que herramientas hay en el mercado que solucionen esta problemática; y finalmente se describe la aplicación que se va a desarrollar, indicando su funcionalidad y objetivos.

En esta sección, también se planifica el trabajo en una línea temporal; tarea importante para poder llevar a cabo un proyecto y hacerle un seguimiento durante su curso. Esta planificación puede ayudar a evitar o resolver cualquier incidencia que se presente durante su ejecución;

2. Diseño:

En esta sección se van a describir y explicar las diferentes etapas que componen el desarrollo del diseño de una aplicación en un multidispositivo: empezando por el estudio del perfil de usuario y contexto de uso; siguiendo por el diseño conceptual del producto; el prototipado; y finalmente la evaluación de la experiencia de usuario;

3. Implementación:

Esta sección abarcará el desarrollo del front-end de la aplicación, la parte visible de la app que interactuará con el usuario: el diseño de la página, su formato, su funcionalidad y el dinamismo que ayudará a la experiencia de usuario. También se mostrará el desarrollo del back-end de la aplicación, la parte oculta de la app que almacena la información que esta necesita para su ejecución;

4. Conclusiones:

En este punto, se mostrarán los pasos que se han seguido, así como los puntos más destacables del proyecto. También se indicarán que dificultades se han encontrado durante el desarrollo del mismo, y de qué forma se han resuelto. Se lanzarán posibles líneas futuras que mejorarán o ampliarán los objetivos de la app.

Como ya se ha mencionado, la aplicación que se presenta en este trabajo sería indicada para ser desarrollada como una app-web, de forma que pudiese ser utilizada en múltiples dispositivos: ordenador, Tablet, pantallas digitales y smartphones.

Sin embargo, el alcance del trabajo sólo va a abarcar el desarrollo de la app en un dispositivo móvil. El resto de posibles desarrollos se indicarán como futuras líneas de trabajo o aplicaciones.

Hay que tener en cuenta que esta memoria se ha desarrollado a medida que el proyecto iba avanzando. El proyecto ha sido un elemento vivo que ha sufrido modificaciones, variando su alcance y cambiando parcialmente alguno de sus objetivos. Todo esto queda reflejado en la memoria, la cual puede presentar unos objetivos e indicaciones iniciales, que pueden ser modificadas o incluso descartadas a lo largo del documento.

Para facilitar su lectura, se han incluido puntos en los que se describen las modificaciones en el diseño y en la implementación a medida que el trabajo avanzaba. Estos puntos se muestran en la mayoría de las secciones y ayudan a entender como ha ido evolucionando el proyecto.

2 Diseño

En este capítulo se va a describir los aspectos más relevantes del diseño y desarrollo del proyecto.

- Se van a definir los usuarios potenciales y los diferentes contextos o escenarios de uso de la app;
- A partir de esta información se desarrollará un árbol de navegación y se definirán las funciones y servicios que dará la app;
- Por último, se definirán las pantallas o ventanas de la app, un prototipo de baja definición;

2.1 Usuarios

Objetivo: conocer las características de los usuarios, sus necesidades y objetivos, así como el contexto de uso.

Este primer ejercicio servirá para poder detectar las funcionalidades que deberá tener la aplicación para satisfacer a sus usuarios.

Tras desarrollar una serie de métodos de indagación, se ha podido extraer información sobre los potenciales usuarios de esta aplicación, conocer sus objetivos y necesidades, los contextos de uso del producto y el servicio que desempeñaría.

El usuario potencial de esta aplicación es todo el cuerpo docente que trabaja en centros de educación y que por lo tanto realizan guardias durante su jornada laboral. Pero desconocemos si la aplicación tendrá el mismo interés en los diferentes grupos que conforman este cuerpo; si todos ellos tendrán las mismas necesidades, si darán el mismo valor a las diferentes soluciones y si estos usarán la aplicación en contextos muy diferentes.

Para ello, tras llevar a cabo una dinámica de grupo entre un grupo de docentes con diferentes responsabilidades y experiencia en el campo: personal del equipo directivo, con plaza fijo, itinerante; y con la experiencia profesional de la autora del proyecto, que también es docente, se concluyó que las principales necesidades entre los usuarios potenciales de la aplicación eran:

- Ahorrar tiempo a la hora de gestionar las ausencias de los profesores y las guardias;
- Lograr imparcialidad a la hora de repartir las tareas de guardia entre los docentes;
- Mantener actualizado al usuario con toda la información referente a las guardias y a posibles incidencias que tengan lugar a lo largo del día;
- Utilizar un método automatizado para llevar a cabo esta gestión;

Sin embargo, es necesario llevar a cabo otro tipo de estudio para corroborar que la mayoría de los posibles usuarios valorarían el uso de una aplicación para gestionar las guardias en un centro educativo y si todos ellos consideran con el mismo grado que los puntos anteriores deberían ser las funciones principales de una aplicación para así solucionar los problemas existentes o mejorar la situación actual.

Para ello se ha llevado a cabo una encuesta (5), para poder clasificar a los posibles usuarios en varios grupos si se observa alguna tendencia entre ellos sobre la identificación de los problemas y necesidades.

Los resultados del formulario se pueden ver en el Anexo, apartado 7.1:

- La encuesta ha sido contestada por 102 personas, con edades comprendidas entre 25 y 65 años. Todos los cargos (profesores de secundaria; profesores de formación profesional; maestros de primaria; docentes con cargos directivos como jefatura de estudios, secretaría, dirección) se han cubierto.

Los diferentes tipos de centros (institutos de educación secundaria; colegios de educación primaria e infantil; CRA colegio rural agrupado; y otros), así como el tamaño de los mismos se ha tenido en cuenta, obteniendo una muestra representativa.

- La mayoría de los encuestados, un 93%, indica que la gestión de las guardias es una tarea llevada a cabo por el equipo directivo. Mientras que la tarea de repartir las guardias entre los profesores se divide principalmente en dos: en un 58,4% de casos en los que se encarga el equipo directivo y un 42,6% donde son los profesores de guardia quienes hacen el reparto.
- El método usado por la mayoría de los centros para comunicar las guardias y gestionarlas es físico (pizarra, libreta...etc), un 54,5%; seguido del método digital (pantalla o PC) un 21,2%.
- En cuanto a la satisfacción con el método usado actualmente para gestionar las guardias: un 41,6% de los encuestados consideran eficiente el método, pero consideran que se podría mejorar; un 22,8% considera que es muy eficiente y no necesita mejora; y un 19,8% considera que necesita mejorar, pero hace su función.

Cabría esperar que aquellos que consideran que no se necesita una mejora en la gestión de este servicio, consideraran que no fuese necesario elaborar una aplicación para este fin. Sin embargo, todos los encuestados han contestado que, si creen que es necesaria una aplicación, a excepción de un 18,4% que opina que no.

De aquí podemos concluir que es necesaria una aplicación para elaborar y gestionar un calendario de guardias, usando una metodología similar a la que se usa actualmente, pero automatizada.

- Hay una gran diversidad a la hora de valorar que factor se valora más, siendo los que mayor puntuación han recibido: ahorrar tiempo y estar informado en todo momento; seguidos muy de cerca de imparcialidad y automatización del proceso.

En resumen, el perfil de usuario para esta aplicación sería:

- Todos los docentes, independientemente de la edad, de la antigüedad en el campo de la enseñanza y del manejo de las nuevas tecnologías.

Dentro de este grupo podríamos diferenciar a los que forman parte del equipo directivo y a los que sólo desarrollan trabajo docente; ya que como hemos visto en los resultados de la encuesta, la mayoría del trabajo de gestión de guardias lo lleva a cabo el equipo directivo.

- Centros de educación secundaria. Nos centramos en estos ya que las guardias se gestionan de forma diferente en los colegios de infantil y primaria, como nos ha reflejado la encuesta. Sin embargo, esta app podría extrapolarse también por estos centros, modificando algunos puntos.

Destacaremos que esta app y/o web no tiene sentido si no se usa en conjunto por todos los profesores en un centro, no tiene ninguna funcionalidad si se utiliza de forma individual. De ahí la importancia en considerar a los centros como usuarios potenciales, y no sólo a los docentes.


- Los factores que más se valorarían serían: el ahorro del tiempo y la automatización de la gestión; seguidos muy de cerca de la imparcialidad en el reparto de las guardias entre los docentes.
- El contexto principal de uso sería durante la jornada laboral, es decir durante el horario escolar. Sin embargo, podríamos ampliar esta horquilla, introduciendo el tiempo de itinerancia al centro, donde el profesor podría utilizar la app para comprobar sus guardias; y el tiempo de descanso, donde el docente puede encontrarse fuera del centro y podría acceder a esta información desde la app.

2.2 Contextos de uso


A continuación, se van a definir posibles escenarios de uso, a partir de la información recopilada en el punto anterior. Para así, establecer las funcionalidades de la aplicación.

Se muestran tres escenarios, los tres tienen una ficha personal muy diferente, siendo por lo tanto diferentes los posibles contextos en los que se usaría la aplicación.

2.2.1 Escenario 1

Datos personales	
<p>Nombre: Rosa</p> <p>Edad: 40 años</p> <p>Nivel de estudios: Licenciada en química</p> <p>Profesión: profesora de física y química</p>	
<p>Ficha Personal:</p> <p>Rosa trabaja en un IES (Instituto de educación secundaria) como profesora. En sus clases, introduce herramientas digitales muy a menudo, teniendo cierta habilidad en temas informáticos. En su día a día, hace uso del smartphone y de diferentes apps que tiene instaladas para facilitarle las tareas rutinarias.</p>	
<p>Escenario:</p> <p>Es lunes y Rosa tiene hora de guardia a 4ª y 6ª hora. De camino al centro, ha consultado el calendario de guardias de la app OCA y ha comprobado que:</p> <ul style="list-style-type: none"> • a 4º hora tendrá que sustituir al profesor de matemáticas con el grupo 2º ESO D. La app le muestra el aula donde se dará la clase y también la tarea que el profesor ausente ha dejado para que realicen los alumnos; en este caso el profesor indica que hagan los ejercicios de la página 135 del libro; • y a 6ª hora no tiene guardias asignadas, de modo que tendrá que vigilar los pasillos durante los primeros 10 minutos de clase y ya después quedarse en la sala de profesores trabajando. <p>Al saber que va a poder adelantar trabajo a 6ª hora, llama a una amiga que se encuentra cerca del centro para tomar un café a la hora del recreo. Ya no va a necesitar la media hora del recreo para realizar unas tareas, las hará a 6ª hora en la sala de profesores.</p> <p>Rosa podría haber consultado en la pantalla de televisión de la sala de profesores si tenía guardias asignadas esa mañana, ya que allí se proyecta también el calendario de la app OCA; sin embargo, ella prefiere usar su móvil.</p> <p>Durante el recreo, mientras está tomando un café con su amiga fuera del centro, Rosa revisa la app OCA por si hubiese algún cambio en las guardias. Comprueba que sí, que la guardia de 4ª hora ya no aparece porque el profesor de matemáticas ausente ya ha llegado al centro. Sin embargo, se le ha asignado otro grupo y aula porque el profesor de geografía e historia ha tenido un imprevisto personal y se ha tenido que ausentar del centro el resto de la mañana. La app OCA le indica a Rosa que a 4ª hora tendrá que ir al aula 201 con el grupo 4º ESO C, la tarea que ha preparado el profesor es que estudien para el examen que tienen el próximo día.</p> <p>Rosa termina de tomar el café con su amiga y se dirige al centro para hacer la guardia asignada a 4ª hora.</p>	

2.2.2 Escenario 2

Datos personales	
<p>Nombre: Rubén</p> <p>Edad: 29 años</p> <p>Nivel de estudios: Licenciado en historia del arte</p> <p>Profesión: profesor de geografía e historia</p>	
<p>Ficha Personal:</p> <p>Rubén trabaja en un IES (Instituto de educación secundaria) como profesor de geografía e historia. Usa herramientas digitales en sus clases siempre que puede, y utiliza dispositivos también digitales para la gestión de su trabajo, entre ellos una Tablet.</p>	
<p>Escenario:</p> <p>Es lunes y Rubén no tiene ninguna hora de guardia ese día. Los lunes sólo tiene tres horas de clase (2ª hora, 4ª hora y 5ª hora), además de la reunión de departamento que la tiene a 3ª hora.</p> <p>Durante la reunión de departamento tiene una llamada de teléfono, es su hermana diciéndole que su madre ha tenido un accidente doméstico y ha sido trasladada a las urgencias del hospital cercano a su domicilio. Ella no puede acudir porque se encuentra en un viaje de trabajo y le pide que se acerque él para poder ayudarla y acompañarla.</p> <p>Rubén abandona la reunión de departamento rápidamente y toma un taxi para acudir al hospital. Tiene demasiada prisa para pasarse por jefatura de estudios para comunicarles que va a ausentarse y que le tendrá que sustituir un profesor de guardia.</p> <p>En el taxi, abre la app OCA en su móvil y abre el menú en el que se comunican las ausencias. Allí indica que va a estar ausente la 4ª y 5ª hora de la mañana por motivos personales, y anota las tareas que deberán hacer sus alumnos en esas clases. Recibe una notificación del equipo directivo aprobando su ausencia.</p> <p>Inmediatamente puede observar en la pestaña de calendario de la app OCA, que ya aparece su ausencia a 4ª hora y el profesor al que se le ha asignado la guardia.</p> <p>Una vez en el hospital, mientras está esperando que le hagan unas pruebas médicas a su madre, consulta su Tablet para repasar unas clases y una programación. De repente recuerda que para el viernes necesita utilizar la sala de informática para realizar una tarea con sus alumnos de 3º ESO D. Abre la app OCA en su móvil y va a la pestaña de gestión de material, allí reserva la sala de informática para el viernes a 2ª hora; está libre según la app OCA, de modo que puede seguir con su plan de trabajo.²</p>	

² La funcionalidad de permitir reservar aulas o material del centro se ha descartado del alcance del proyecto. Sin embargo, se propone como una línea futura de desarrollo. Por este motivo se ha mantenido en el contexto de uso.

2.2.3 Escenario 3

Datos personales

Nombre: Paloma

Edad: 50 años

Nivel de estudios: Ingeniera industrial

Profesión: profesora de matemáticas

**Ficha Personal:**

Paloma trabaja en un IES (Instituto de educación secundaria) como profesora de matemáticas. Además de docente, ese curso forma parte del equipo directivo, es jefa de estudios junto a dos compañeros más. Tienen un despacho donde trabajan los tres y disponen de un portátil para cada uno de ellos, para gestionar las diferentes tareas que se les encomiendan.

Escenario:

Es lunes y Paloma acude al instituto a primera hora para empezar su jornada. Esa hora la tiene reservada para realizar trabajo relacionado con la jefatura de estudios. Ella es la persona encargada de la gestión de las guardias.

Abre la app-web³ OCA en su ordenador como administrador y comprueba el calendario de guardias, enciende la pantalla de televisión de la sala de profesores para que se visualice el calendario de la app con las guardias de la mañana.

Tiene varias peticiones de profesores que se van a ausentar a lo largo de la semana, las revisa y aprueba e inmediatamente se muestran en el calendario semanal de la app. También es la app la que se encarga de repartir las guardias entre los profesores de guardia, para Paloma es un gran alivio porque ya no tiene que trabajar con tablas donde contar el número de guardias de cada profesor para poder hacer un reparto equitativo.

Un profesor se presenta en el despacho para comunicarle que mañana no asistirá al centro durante toda la mañana porque tiene que acompañar a su padre a una cita médica. El profesor le deja la tarea que quiere que hagan sus alumnos. Paloma abre la app-web en su portátil y va al menú donde se comunican las ausencias, al ser administrador puede incluir las ausencias de cualquier profesor. Indica las horas que va a faltar el profesor y la tarea que tienen que hacer los alumnos, inmediatamente al confirmarlo, estas ausencias ya quedan reflejadas en el calendario semanal y ya están asignadas a profesores de guardia.

Observa que esta semana hay un gran número de ausencias, esto es debido a la pandemia de COVID que está en uno de sus puntos más altos de incidencia; y se alegra de trabajar con la app OCA para gestionar las guardias, ya que le facilita mucho más el trabajo por automatizarlo.

Ese jueves tienen una reunión con los jefes de departamento donde se va a hablar de las guardias. Prepara un informe para comentarlo en la reunión, donde se visualizará el número de guardias que ha realizado cada uno de los profesores durante el último mes. Esta información la obtiene de la app OCA, en la pestaña de usuarios; así como el número de ausencias de cada uno de los profesores.

³ Se indica app-web, ya que cómo se ha mencionado al principio del documento, lo ideal sería que la aplicación fuese en desarrollo web, para poder ser usada en ordenadores. Sin embargo, no es el alcance de este proyecto. De todas formas, se tiene en cuenta en los posibles escenarios.

Gracias al trabajo de conceptualización llevado a cabo, se ha obtenido un listado de las funciones que el usuario esperaba encontrar en la aplicación OCA. Estas son:

- Elaboración automática de un horario de guardias diario, según la información que han proporcionado los profesores ausentes.
- Asignación automática e imparcial de las guardias entre los profesores de guardia.
- Posibilidad de subir la tarea de la guardia, en caso de ser el profesor ausente.
- Posibilidad de gestionar las ausencias usando la aplicación.
- Poder visualizar el histórico de guardias de cada profesor.

Las siguientes funciones tuvieron menos acogida por parte de los encuestados (potenciales usuarios). No se introducirán en la aplicación actual, pero se tendrán en cuenta para futuros desarrollos y mejoras de la misma:

- Posibilidad de reservar material y aulas específicas del centro.
- Creación del calendario del centro, añadiendo el horario de todos los profesores, para gestionar las guardias.
- Posibilidad de gestionar los cambios de guardias asignadas entre profesores.
- Poder visualizar los compañeros con los que compartes hora de guardia;
- Poder visualizar las aulas libres del centro en cada momento;

2.3 Diseño conceptual

2.3.1 Clasificación de la información

Se ha procedido a identificar todos los puntos y piezas que van a formar parte de la aplicación, agrupándolas en secciones para así poder conformar el árbol de navegación de la misma.

Todas ellas han quedado agrupadas en cinco secciones:

1. Ausencias – Relacionada con las ausencias que deben cubrir los profesores de guardia:
 - Visualización de las guardias del día actual, y de los profesores asignadas a las mismas;
 - Visualización de las guardias de los próximos días;
 - Actualización del listado de ausencias cuando haya un cambio;
2. Gestionar ausencias – Relacionada con la solicitud de permiso al equipo directivo para una ausencia:
 - Poder notificar ausencias y subir el material para la clase;
3. Usuarios - Relacionadas con la visualización de los usuarios y con el contador de guardias
 - Poder visualizar el histórico de guardias de cada usuario;
 - Poder visualizar los perfiles de todos los usuarios;
4. Material - Relacionadas con la gestión del material y de las aulas del centro:
 - Poder reservar material;

- Poder reservar aulas;
 - Poder visualizar las aulas vacías;
5. Configuración - Relacionadas con la gestión del administrador (esta opción sólo será visible para el administrador, es decir el equipo directivo):
- Subir la información relativa a cada uno de los profesores que va a trabajar en el centro durante el curso actual;
 - Añadir información sobre los grupos de guardias (días, horas y profesores) en la base de datos;
 - Aprobar las ausencias notificadas por los docentes;

Con toda esta información se ha decidido que la mejor estructura de navegación para esta aplicación es la **estructura en pestañas**. La aplicación mostrara una barra menú con las tres pestañas clave que clasifican toda la información que esta contendrá. Ninguna de las tres está por encima de las otras en cuanto a jerarquía, todas está en el mismo nivel.

Guardias – Material – Usuarios

2.3.2 Árbol de navegación

El árbol de navegación indica claramente el recorrido que hará el usuario pantalla a pantalla en la aplicación.

Se ha usado la herramienta digital *cacoo* para crear un diagrama de flujos donde cada cuadrícula representa una pantalla por donde navegará el usuario. Todas ellas están enlazadas, indicando así, como llegar a cada una de ellas.

De color amarillo encontramos las pestañas primarias, de color verde las secundarias, teniendo todas en común el listado de guardias relacionado entre ellas. De color azul tenemos las opciones que no se encuentran en la pestaña en sí, que son el registro de ausencias y los cambios de guardias.

Todas las pestañas están enlazadas con flechas de color negro. Las flechas de color rojo nos indican que comparten información.

La aplicación se inicia en la pestaña de Guardias, ya que es la que presenta la funcionalidad principal. Las otras dos pestañas, Usuarios y Nueva ausencia son secundarias.

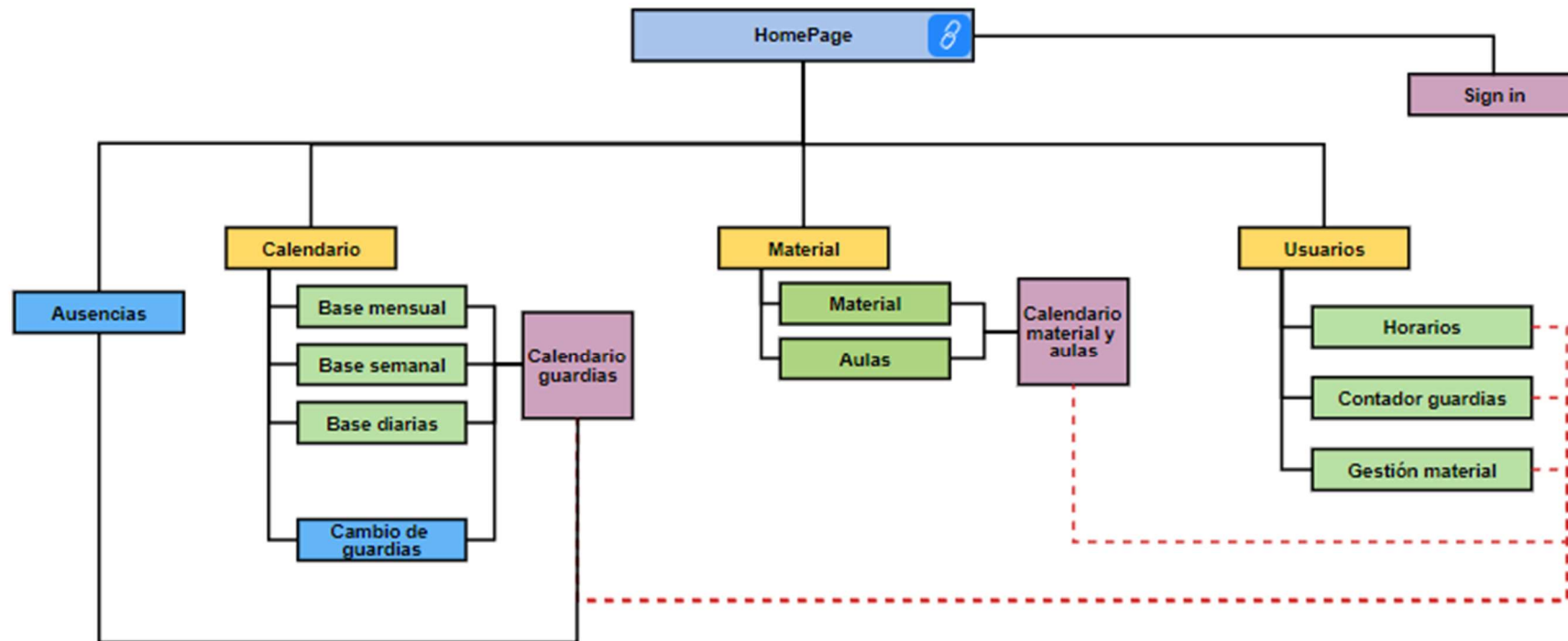


Ilustración 2-1 Árbol de navegación⁴

⁴ La opción Material será excluida del alcance del proyecto como se menciona en las siguientes secciones, dando prioridad al resto de funcionalidades.

Variaciones tras el prototipado

Al empezar el prototipo de la aplicación se realizó una segunda revisión de las funcionalidades que debería incluir la misma, dando prioridad a aquellas que los usuarios habían dado mayor valoración, para evitar que por cuestiones de tiempo alguna de ellas no pudiese ser implementada. En este punto se observó que la opción de reserva de material y de aulas no eran las más prioritarias para los usuarios, por lo que se decidió dejarla fuera del alcance del proyecto, y presentarla como una futura mejora de la aplicación.

De modo que, a partir de este punto, se deja de incluir la sección de reserva de material y aulas en el alcance del proyecto. Desapareciendo así del prototipo, y por lo tanto desapareciendo la opción de Material del menú principal.

El nuevo diseño de la aplicación mostrara en su pantalla principal el listado de guardias junto a los profesores asignados. En esta pantalla principal encontraremos un menú en la parte inferior para acceder a las dos pantallas secundarias: Usuarios y Gestión de ausencias. Jerárquicamente, la pantalla de Guardias estará por encima de las otras dos. Ya que la función principal de la pestaña Gestión de ausencias es notificar de una nueva ausencia, se ha considerado llamarla "Nueva Ausencia", para facilitar su comprensión.

GUARDIAS

Usuarios – Nueva Ausencia

Existirá un menú sándwich desplegable en la parte superior, donde aparecerá la opción de configuración, a la que podrá acceder sólo el administrador (equipo directivo) y le llevará a la sección de configuración, indicada en el párrafo anterior.

2.4 Prototipado

Una vez elaborado el árbol de navegación. Se ha comenzado a desarrollar un prototipo de la aplicación en lápiz y papel. Se ha intentado plasmar aquellas pantallas más importantes. El objetivo es visualizar lo que se quiere conseguir y como se quiere hacer.

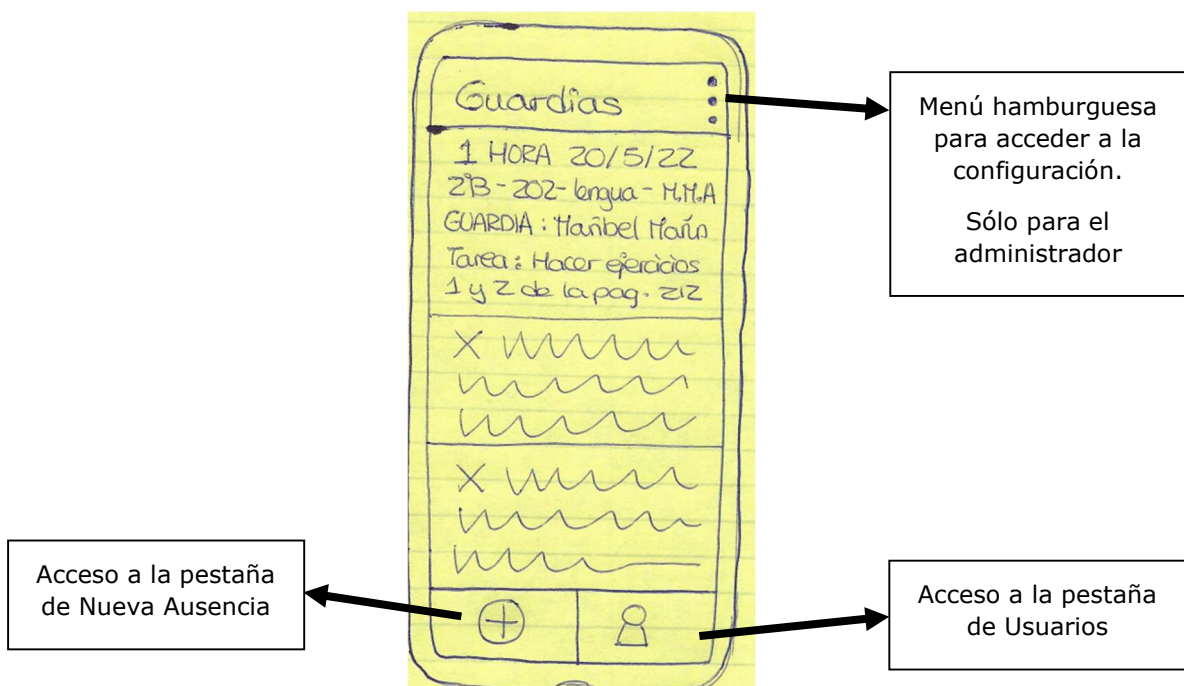


Ilustración 2-2 Prototipo mano alzada: Pantalla principal - Guardias

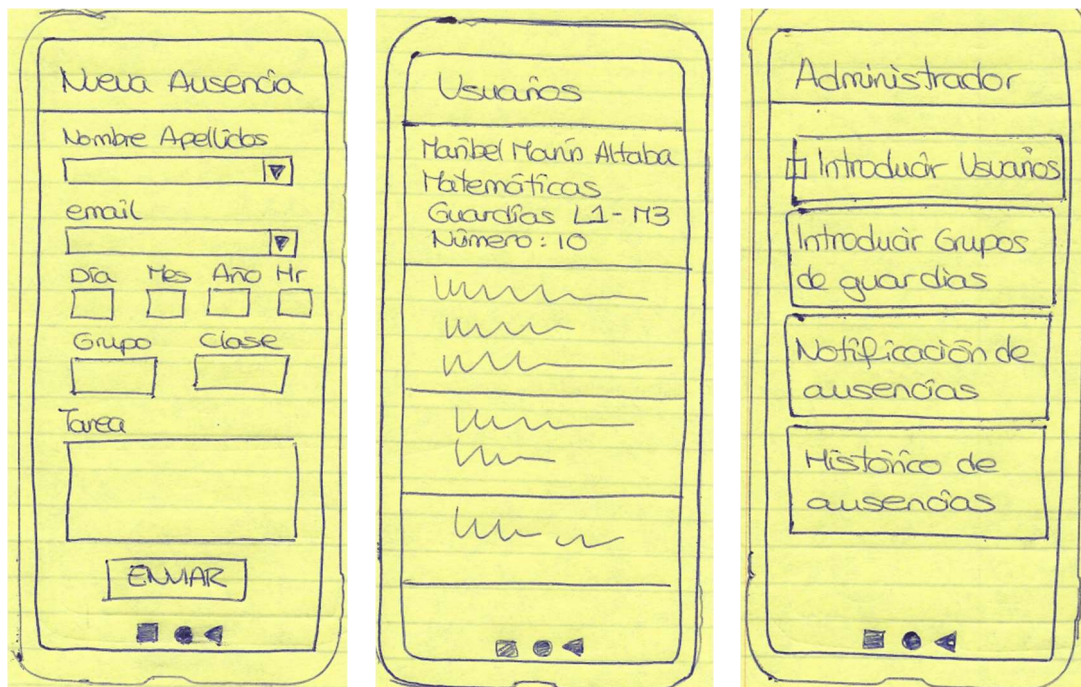


Ilustración 2-3 Prototipo mano alzada: De izquierda a derecha (Pantalla Nueva ausencia – Pantalla Usuarios – Pantalla configuración para Administrador)



Ilustración 2-4 Prototipo mano alzada: Pantalla de inicio

A continuación, se va a describir el prototipo, lo que el usuario va a encontrar en cada una de las pestañas y pantallas que contiene y como sus necesidades van a ir siendo cubiertas. Para ello vamos a imaginar dos tipos de usuarios. Un administrador, en este caso el jefe de estudios; y un usuario común, un docente.

La pantalla principal es la pantalla donde se muestran las ausencias de los docentes, acompañadas del profesore que va a realizar la guardia.

2.4.1 Pestaña principal – Pestaña de Guardias

Todos los usuarios, cuando inicien su aplicación y se hayan identificado, entrarán en la pestaña de guardias. Allí se mostrará el listado de ausencias de docentes junto a los profesores de guardia asignados a ellas, para el día en el que nos encontremos y los próximos días. El usuario podrá hacer scroll vertical para poder visualizar todas las ausencias. El administrador podrá limitar el número de días para los que se muestra las guardias, cinco días próximos podría ser un buen rango, ya que permite ver las guardias de toda la semana.

Si el usuario selecciona una hora cualquiera, se abrirá una view que mostrará los nombres de las personas que tienen guardia esa hora y como se han repartido las guardias.

En esta pestaña encontramos el menú de la app en la parte inferior, el cual muestra dos botones para acceder a las dos pestañas secundarias de la aplicación: Nueva ausencia y Usuarios.

2.4.2 Pestaña Nueva Ausencia

La segunda pestaña de la aplicación se llama Nueva Ausencia. Accedemos a ella con el menú de la parte inferior de la pestaña principal.

Al entrar en ella se abrirá un formulario para completar por el docente que solicite o comunique esta nueva ausencia. Tendrá que indicar las horas o días ausentes e incluyendo la tarea para los alumnos. Esta acción deberá ser confirmada por jefatura. Una vez se confirme se visualizará en la pantalla principal, en el listado de guardias.

Para cada hora que vaya a faltar, tendrá que completar una petición, ya que cada hora corresponde a un profesor de guardia diferente.

2.4.3 Pestaña Usuario

La tercera pestaña de la aplicación se llama Usuario. Accedemos a ella con el menú de la parte inferior de la pestaña principal.

Al entrar en ella tendremos un listado de los profesores ordenados alfabéticamente, mostrándonos la asignatura que imparten, los días y horas que tienen guardias y un contador con las guardias realizadas. De forma opcional se podrá añadir una foto de perfil del docente.

2.4.4 Administración

En la parte superior de la pantalla principal, encontraremos un menú hamburguesa, en el que se desplegarán varias opciones, entre ellas la de administración. En este campo sólo podrá acceder el administrador, es decir el equipo directivo. Este podrá realizar las siguientes acciones desde este campo⁵:

- Introducir los horarios de los docentes, así como la información de los docentes, la cuál será registrada en la base de datos que usará la app-web.
- Introducir los grupos de guardias de cada hora lectiva de la semana.
- Aquí se mostrarán todas las notificaciones de cambios de guardias y de comunicación de ausencias que los docentes harán, y que deberán ser aceptadas por el administrador.

⁵ Para el uso de la aplicación como administrador, será más útil usar la aplicación web en un ordenador que la aplicación que se presenta en este proyecto para dispositivo móvil. Es decir, que las funcionalidades indicadas en este punto serían competencia de la app-web.

2.5 Casos de uso

Se han diferenciado tres grupos de casos de uso en función de la prioridad y del actor:

- **Relacionados con la identificación del usuario;**

Estos casos de uso están relacionados con el registro e inicio y finalización de la sesión del usuario en la aplicación. Su prioridad es media y los actores pueden ser tanto los docentes (usuarios comunes) como el equipo directivo (administrador).

- **Relacionados con las funcionalidades principales de la app;**

En estos casos tendremos las prioridades más altas, ya que cumplen las principales funcionalidades de la app: poder visualizar las guardias de clase y poder añadir una nueva ausencia. Hay algún caso de uso más, pero cuya prioridad es inferior, tratándose de funcionalidades añadidas en la aplicación tras realizar la encuesta. Los actores serían los docentes (usuario común).

- **Relacionadas con la gestión de la app;**

Estos usos sólo podrán ser llevados a cabo por el administrador, es decir el equipo directivo. Están relacionados con la inicialización de la aplicación, es decir con completar la base de datos con la información necesaria; y con la confirmación de las peticiones de ausencia por parte del profesorado. Su prioridad es media.

2.5.1 Relacionados con la identificación del usuario

CU-001 – Registro del usuario	
Descripción	El usuario hace un registro para así poder iniciar sesión en la aplicación y poder hacer uso de sus funcionalidades.
Actor	Docente y/o Administrador (equipo directivo)
Prioridad	Media
Precondiciones	Aplicación instalada, acceso a la pantalla de inicio de sesión y usuario no registrado.
Flujo	<ol style="list-style-type: none"> 1. En la pantalla de inicio, selecciona crear una cuenta. 2. Usuario introduce su email corporativo y una contraseña. <ol style="list-style-type: none"> a) Si el email no pertenece a ningún docente, se le mostrará un error y no se permitirá crear la cuenta. b) Si el email ya ha sido registrado, se le mostrará un error y no se permitirá crear la cuenta. c) Si la contraseña contiene menos de 6 caracteres, se le mostrará un error y no se permitirá crear la cuenta. 3. Una vez creada la cuenta, se inicia la sesión automáticamente accediendo a la pantalla principal de la app.
Postcondiciones	El usuario puede acceder a todas las funcionalidades de usuario de la app.
Observaciones	Existe un <i>checkbox</i> ; si está marcado, permite que el dispositivo recuerde la contraseña y email del usuario, y así no necesite identificarse cada vez que quiere acceder a la aplicación.

Tabla 2-1 Casos de uso relacionados con la identificación del usuario – CU 001.

CU-002 – Inicio del sistema	
Descripción	El usuario quiere acceder a la app para usar sus funcionalidades.
Actor	Docente y/o Administrador (equipo directivo)
Prioridad	Media
Precondiciones	Usuario no identificado, pero si registrado. App instalada y acceso a la pantalla de inicio de sesión.
Flujo	<ol style="list-style-type: none"> 1. En la pantalla de inicio, completa las celdas con su correo corporativo y su contraseña. <ol style="list-style-type: none"> a) Si el correo y/o la contraseña son incorrectos, se muestra un error y no se permite acceder a la app. b) Si no se completan todos los campos, se muestra un error y no se permite acceder a la app. 2. El usuario accede a la pantalla principal de la aplicación.
Postcondiciones	El usuario puede acceder a todas las funcionalidades de usuario de la app.
Observaciones	Existe un <i>checkbox</i> ; si está marcado permite que el dispositivo recuerde la contraseña y email del usuario, y así no necesite identificarse cada vez que quiere acceder a la aplicación.

Tabla 2-2 Casos de uso relacionados con la identificación del usuario – CU 002.

CU-003 – Restablecer contraseña	
Descripción	El usuario necesita restablecer su contraseña para poder acceder a la aplicación.
Actor	Docente y/o Administrador (equipo directivo)
Prioridad	Media
Precondiciones	El usuario ha recibido un mensaje de error tras intentar iniciar sesión (CU-002), porque la contraseña o el usuario son incorrectos.
Flujo	<ol style="list-style-type: none"> 1. El usuario hace click en el link de restablecer contraseña, en la pantalla de inicio de sesión. 2. El sistema envía un correo electrónico al email con el que el usuario se registró. En él se adjunta un enlace para restablecer la contraseña. 3. El usuario deberá acceder a su correo electrónico para recuperar la contraseña.

Postcondiciones	Una vez restablecida la contraseña, el usuario ya puede acceder a iniciar sesión (CU-001).
Observaciones	

Tabla 2-3 Casos de uso relacionados con la identificación del usuario – CU 003.

CU-004 – Cerrar Sesión	
Descripción	El usuario cierra sesión de la aplicación.
Actor	Docente y/o Administrador (equipo directivo)
Prioridad	Media
Precondiciones	El usuario debe tener la aplicación instalada y una sesión inicializada.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede al menú hamburguesa del banner superior, un desplegable aparece, entre las opciones está "Cerrar sesión". 2. El usuario hace click en la opción "Cerrar Sesión". 3. La sesión finaliza y la aplicación se redirige a la pantalla de inicio de sesión.
Postcondiciones	El usuario ha cerrado la sesión.
Observaciones	

Tabla 2-4 Casos de uso relacionados con la identificación del usuario – CU 004.

2.5.2 Relacionados con las funcionalidades de la app para el usuario

CU-005 – Consultar guardias	
Descripción	El usuario quiere consultar las guardias del día de hoy y/o de los próximos días.
Actor	Docente
Prioridad	Alta
Precondiciones	Aplicación instalada y sesión inicializada.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla principal de la aplicación, donde se muestran las guardias del día de hoy. 2. Haciendo scroll vertical con su dedo puede ver todas las guardias del día de hoy y de los próximos 5 días.
Postcondiciones	El usuario queda informado sobre la gestión de las guardias de inmediato.
Observaciones	

Tabla 2-5 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 005.

CU-006 – Añadir una ausencia	
Descripción	El usuario quiere añadir una ausencia.
Actor	Docente
Prioridad	Alta
Precondiciones	Aplicación instalada y sesión inicializada.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla principal de la aplicación, donde se muestran las guardias del día de hoy. 2. Hace click en el menú inferior, en el botón donde se puede ver la imagen del signo "+". 3. Una ventana se abre, mostrando una serie de celdas que el usuario debe completar para informar de su ausencia, un formulario. Los campos serán: nombre y apellidos, email corporativo, día y hora de la ausencia, asignatura, grupo de alumnos, aula y tarea que deberán realizar los alumnos. El profesor tendrá que completar un formulario como este para cada una de las horas que vaya a ausentarse. <ol style="list-style-type: none"> a) Si el formulario no queda completo, un mensaje de error será mostrado, no permitiendo enviar el formulario. b) Si alguna celda contiene algún error, un mensaje de error será mostrado, no permitiendo enviar el formulario. 4. Una vez completado, el usuario hará click al botón de enviar y la solicitud será enviada al equipo directivo. Automáticamente, la ventana se cerrará y se redirigirá a la pantalla principal de la aplicación. 5. Cuando el equipo directivo acepte la solicitud de ausencia, la aplicación enviará una notificación al usuario donde se indicará que ha sido confirmada.
Postcondiciones	Una vez aceptada la solicitud, la ausencia de este usuario será mostrada dentro del listado de ausencias en la página principal junto al profesor que realizará la guardia; si esta ausencia está dentro de los próximos cinco días.
Observaciones	

Tabla 2-6 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 006.

CU-007 – Consultar número guardias profesores	
Descripción	El usuario desea consultar el número de guardias que ha hecho uno de los docentes.
Actor	Docente
Prioridad	Baja

CU-007 – Consultar número guardias profesores	
Precondiciones	Aplicación instalada y sesión inicializada.
Flujo	<ol style="list-style-type: none"> 1. Hace click en el menú inferior, en el botón donde se puede ver la imagen de un contacto. 2. Una nueva pantalla se abre, donde se puede ver el listado de todos los profesores junto a su información: nombre y apellidos, email corporativo, asignatura que imparte, días y horas que realiza guardias y número de guardias realizadas durante el curso. 3. El usuario puede encontrar al profesor que busca haciendo scroll vertical en la pantalla. Estos están ordenados alfabéticamente.
Postcondiciones	El usuario puede conocer el número de guardias que ha realizado uno de los docentes.
Observaciones	

Tabla 2-7 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 007.

2.5.3 Relacionados con la gestión de la app

CU-008 – Validar una ausencia	
Descripción	El equipo directivo tiene que validar una ausencia solicitada por un docente.
Actor	Equipo directivo
Prioridad	Alta
Precondiciones	Aplicación instalada, sesión como administrador inicializado.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación como administrador, usando su email corporativo. La aplicación ya tiene registrados los correos que forman parte del equipo directivo. 2. El usuario accede al menú hamburguesa, donde se despliegan varias opciones, entre ellas la de administrador. Hace clic en esta opción y se le abre una ventana. 3. Una pantalla nueva se abre, donde se puede ver un listado de diferentes operaciones que puede realizar el administrador de la aplicación. Hace clic en la frase que dice “gestionar solicitudes ausencia”. 4. Una ventana se abre con el listado de las ausencias solicitadas por los docentes, en la parte inferior hay dos botones uno pone confirmar y otro denegar. 5. El equipo directivo hace clic en los checkbox que hay al lado de cada una de las notificaciones y después pulsa el botón confirmar. 6. El listado de solicitudes se queda de color gris, mostrando así que han sido aceptadas.

	7. El usuario sale de esta pantalla haciendo clic a cualquiera de los otros dos botones que se muestran en el menú inferior. Accediendo así a la pantalla principal o a la de usuarios.
Postcondiciones	Una notificación será enviada automáticamente al usuario que ha solicitado el permiso para ausentarse, informándole que ha sido aceptada. Esta ausencia será creada en la base de datos y se mostrará en la aplicación en la pantalla principal cuando se encuentre entre los 5 días próximos al actual junto al profesor de guardia asignado.
Observaciones	

Tabla 2-8 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 008.

CU-009 – Denegar una ausencia	
Descripción	El equipo directivo tiene que gestionar si validar una ausencia solicitada por un docente.
Actor	Equipo directivo
Prioridad	Alta
Precondiciones	Aplicación instalada, sesión como administrador inicializado.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación como administrador, usando su email corporativo. La aplicación ya tiene registrados los correos que forman parte del equipo directivo. 2. El usuario accede al menú hamburguesa, donde se despliegan varias opciones, entre ellas la de administrador. Hace clic en esta opción y se le abre una ventana. 3. Una pantalla nueva se abre, donde se puede ver un listado de diferentes operaciones que puede realizar el administrador de la aplicación. Hace clic en la frase que dice "gestionar solicitudes ausencia". 4. Una ventana se abre con el listado de las ausencias solicitadas por los docentes, en la parte inferior hay dos botones uno pone confirmar y otro denegar. 5. El equipo directivo hace clic en los checkbox que hay al lado de cada una de las notificaciones y después pulsa el botón denegar. 6. El listado de solicitudes se queda de color rojo, mostrando así que han sido denegadas. 7. El usuario sale de esta pantalla haciendo clic a cualquiera de los otros dos botones que se muestran en el menú inferior. Accediendo así a la pantalla principal o a la de usuarios.
Postcondiciones	Una notificación será enviada automáticamente al usuario que ha solicitado el permiso para ausentarse, informándole que ha sido denegada.

Observaciones

Tabla 2-9 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 009.

CU-010 – Introducir Grupos de guardias	
Descripción	El equipo directivo tiene que inicializar la aplicación introduciendo los grupos de guardias de los profesores para este curso.
Actor	Equipo directivo
Prioridad	Media
Precondiciones	<p>Ningún docente ha sido registrado todavía, la base de datos se encuentra vacía.</p> <p>La aplicación-web debe ser inicializada y el usuario registrado como administrador, accediendo a la pantalla principal.</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación como administrador, usando su email corporativo. La aplicación ya tiene registrados los correos que forman parte del equipo directivo. 2. El usuario accede al menú hamburguesa, donde se despliegan varias opciones, entre ellas la de administrador. Hace clic en esta opción y se le abre una ventana. 3. Una pantalla nueva se abre, donde se puede ver un listado de diferentes operaciones que puede realizar el administrador de la aplicación. Hace clic en la frase que dice "gestionar solicitudes ausencia". 4. Una pantalla nueva aparece mostrando una serie de acciones que puede realizar el administrador. 5. Hace clic en la opción introducir grupos de guardias. 6. Una ventana se abre con una celda vacía en la que pone "arrastrar documento con la información necesaria". 7. El administrador arrastra el documento con la información de los grupos de guardias hasta la celda. 8. Hace clic en el botón que se encuentra debajo de la celda, que pone enviar. 9. Automáticamente esta ventana se cierra y aparece la pantalla de configuración. 10. El usuario puede ir a la pantalla principal si lo desea pulsando en el banner inferior. 11. Un correo electrónico es enviado automáticamente al equipo directivo con un enlace a la base de datos Firebase, para poder ver como se ha cargado la información.
Postcondiciones	La base de datos contiene los grupos de guardias de profesores junto a sus días y horas.

CU-010 – Introducir Grupos de guardias

Observaciones Para este uso, se necesitará la opción app-web⁶ de la aplicación.

Tabla 2-10 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 010.

Para el uso de la aplicación como administrador, será más útil usar la aplicación web en un ordenador que la aplicación que se presenta en este proyecto para dispositivo móvil. Es decir, que las funcionalidades indicadas en este punto serían competencia de la app-web.

CU-011 – Introducir Datos profesores

Descripción El equipo directivo tiene que inicializar la aplicación introduciendo los profesores que van a trabajar este curso en el centro.

Actor Equipo directivo

Prioridad Media

Precondiciones Ningún docente ha sido registrado todavía, la base de datos se encuentra vacía.

La aplicación-web debe ser inicializada y el usuario registrado como administrador, accediendo a la pantalla principal.

Flujo

1. El usuario accede a la aplicación como administrador, usando su email corporativo. La aplicación ya tiene registrados los correos que forman parte del equipo directivo.
2. El usuario accede al menú hamburguesa, donde se despliegan varias opciones, entre ellas la de administrador. Hace clic en esta opción y se le abre una ventana.
3. Una pantalla nueva se abre, donde se puede ver un listado de diferentes operaciones que puede realizar el administrador de la aplicación. Hace clic en la frase que dice "gestionar solicitudes ausencia".
4. Una pantalla nueva aparece mostrando una serie de acciones que puede realizar el administrador.
5. Hace clic en la opción introducir profesores.
6. Una ventana se abre con una celda vacía en la que pone "arrastrar documento con la información necesaria".
7. El administrador arrastra el documento con la información de los profesores hasta la celda.
8. Hace clic en el botón que se encuentra debajo de la celda, que pone enviar.
9. Automáticamente esta ventana se cierra y aparece la pantalla de configuración.
10. El usuario puede ir a la pantalla principal si lo desea pulsando en el banner inferior.

⁶ Para el uso de la aplicación como administrador, será más útil usar la aplicación web en un ordenador que la aplicación que se presenta en este proyecto para dispositivo móvil. Es decir, que las funcionalidades indicadas en este punto serían competencia de la app-web.

	Un correo electrónico es enviado automáticamente al equipo directivo con un enlace a la base de datos Firebase, para poder ver como se ha cargado la información.
Postcondiciones	La base de datos contiene todos los profesores que se van a registrar en la aplicación para usarla, durante el curso.
Observaciones	Para este uso, se necesitará la opción app-web ⁷ de la aplicación.

Tabla 2-11 Casos de uso relacionados con las funcionalidades de la app para el usuario – CU 011.

Variaciones tras el diseño

Una vez finalizado el diseño, se centró el trabajo en los servicios relacionados con el docente y no con el administrado o equipo directivo. De modo que, a partir de este punto, sólo se hará referencia a las funcionalidades que el usuario necesita, dejando las del administrador como futuros desarrollo de la aplicación.

2.6 Diseño de la arquitectura

A continuación, se muestra un diagrama UML (6) mostrando las diferentes clases que participarán en la lógica de la aplicación y como se relacionan entre ellas. Esta será la información que albergará la base de datos y que será consultada y modificada desde la aplicación.

Este diagrama ha sido elaborado antes de llevar a cabo la implementación total de la aplicación, de modo que existen algunas variaciones con el resultado final y lo que se muestra en él. Estas diferencias se indican a continuación, pero serán descritas en mayor detalle en las siguientes secciones.

El diagrama muestra tres clases principales:

2.6.1 Absence (Ausencias)

Esta clase registrará cada una de las ausencias solicitadas por los profesores, con toda la información necesaria almacenada en sus atributos. Estas se almacenarán en la base de datos una vez que sean aprobadas por el administrador, y se mostrarán en la aplicación a través de la pantalla principal.

Presenta un método llamado *createAbsence()*, que se llamará cuando el usuario quiera solicitar el permiso para una ausencia.

Variaciones tras la implementación

Sin embargo, en la implementación final no se define el método *createAbsence()* en la clase *Absence*. La funcionalidad que presenta este método se encuentra dentro del código de la app.

Se han añadido cuatro atributos más que proporcionan información importante para la lógica de la app. Estas son: *DayWeek* (día de la semana de la guardia); *Year*; *UserName* (email del docente ausente); *UserOnCall* (nombre del docente asignado para realizar la guardia).

⁷ Para el uso de la aplicación como administrador, será más útil usar la aplicación web en un ordenador que la aplicación que se presenta en este proyecto para dispositivo móvil. Es decir, que las funcionalidades indicadas en este punto serían competencia de la app-web.

2.6.2 GroupOnCall (Grupos de guardias)

Aquí se mostrarán todos los grupos de profesores de guardia, uno para cada hora y día del horario semanal. La información correspondiente a esta clase será subida por el administrador a la base de datos.

El grupo de profesores será formado por un array de tipo *Users*. El método *orderUsers()* se utilizará para saber cuál es el profesor de cada grupo que tiene menos número de guardias hechas; se utilizará para asignar las guardias.

Variaciones tras la implementación

Tras finalizar la implementación, el número de profesores de cada grupo de guardias no se registró formando un array de tipo *Users*. Cada uno de los usuarios que formaba parte del grupo de guardias fue registrado como un atributo más de la clase *GroupOnCall*, mostrando tan sólo el nombre del usuario. Estos atributos se llaman *User1*, *User2*, *User3* y *User4*, son cuatro porque normalmente este es el número de profesores de guardia cada hora. Se añadieron cuatro atributos más, haciendo así referencia al contador de cada uno de los docentes que forma parte de cada grupo, denominados *Counter1*, *Counter2*, *Counter3* y *Counter4*.

El método *orderUsers()* no se define en la clase *GroupOnCall*, porque finalmente no se pudo implementar la funcionalidad que asignaba un profesor de guardia a cada una de las guardias.

2.6.3 User (Usuario)

Esta clase muestra la información de cada uno de los profesores, incluyendo los atributos *Counter* que mostrará el número de guardias que ha realizado ese profesor, y *Status* para saber si un profesor es el ausente. Se debe tener en cuenta que no se le podrá asignar ninguna guardia a un profesor ausente. Contiene el método *addCounter()* que se utilizará para añadir una guardia al atributo *Counter* cuando el profesor sea asignado a una guardia.

Variaciones tras la implementación

El método *addCounter()* no se define en la clase *User*, porque finalmente no se pudo implementar la funcionalidad que asigna un profesor de guardia a cada una de las guardias, de modo que tampoco se completó el hecho de añadir una guardia al contador del profesor asignado.

Dos atributos más han sido añadidos en la clase *User*, uno hace referencia al password que introducirá el usuario cuando se registre, *Password*; y el otro a la url de una imagen de perfil, para cada uno de los usuarios, *Picture*.

La clase *Absences* y *GroupOnCall* están relacionadas entre sí, de tal forma que para cada objeto *Absences* debe existir un objeto *GroupOnCall*, y para cada *GroupOnCall* se pueden encontrar ninguno o infinitos objetos *Absences*.

La clase *Users* forma parte de las otras dos clases, *Absences* y *GroupOnCall*. En la clase *Absences* es un atributo y en la clase *GroupOnCall* es un atributo en forma de array.

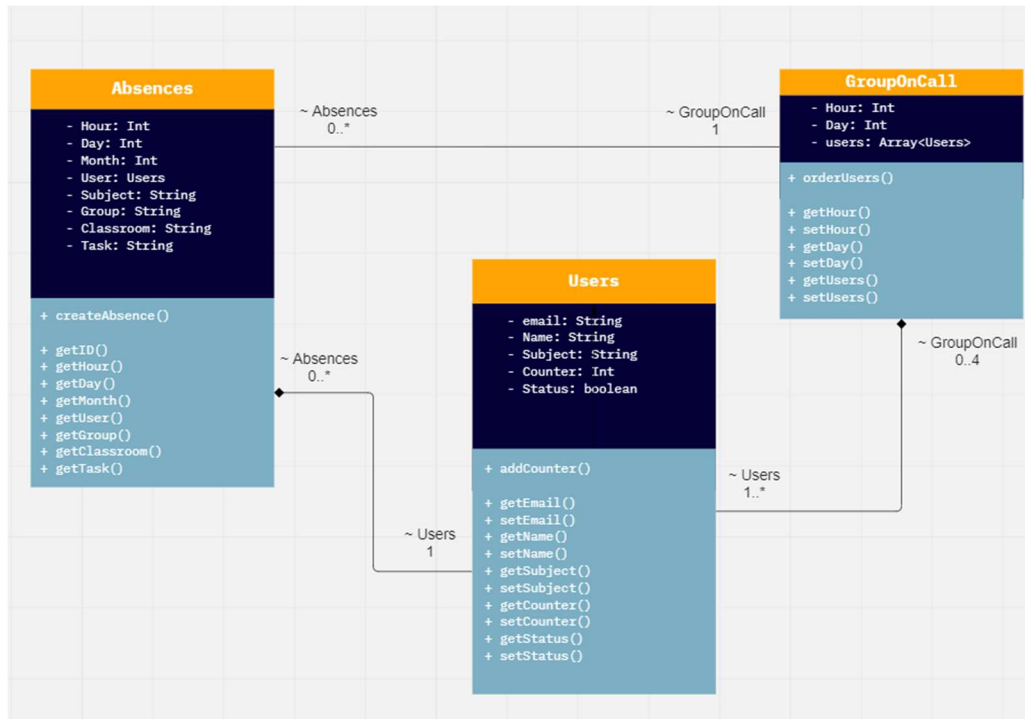


Ilustración 2-5 Diagrama UML de las clases de la aplicación

Seguindo las indicaciones de la documentación oficial de Android (8), se ha decidido escoger el Modelo View ViewModel (MVVM). Esta estructura diferencia tres campos:

- **Model:** donde encontramos los datos y la estructura de los mismos; En nuestro caso, model se alimentará de la base de datos (Absence, User y GroupOnCall).
- **View:** que muestra la interfaz (Activity, fragments, archivos xml), lo que se muestra al usuario;
- **ViewModel:** que actúa como intermediario entre las dos. Si Model cambia al añadir datos o al modificarlos, ViewModel se encarga de que esto quede reflejado en la View. Así del mismo modo, si el usuario realiza alguna interacción en la view, ViewModel notifica a Model con estas indicaciones.

El esquema siguiente (7) muestra cómo actúan las diferentes partes.

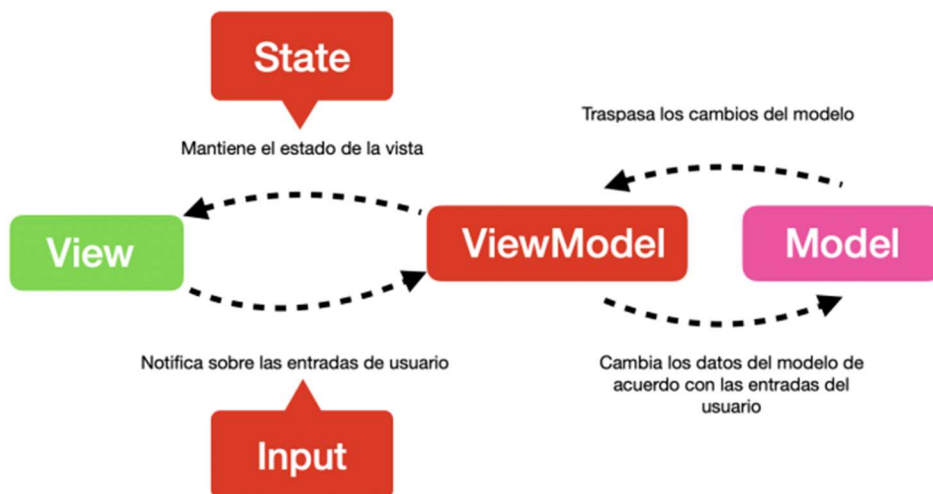


Ilustración 2-6 Esquema MVVM

3 Implementación

En esta sección se va a describir los puntos más importantes llevados a cabo en la implementación y desarrollo de la aplicación móvil. Se comentará las herramientas utilizadas, tanto para el desarrollo del *front-end*, la parte visible de la app que interactuará con el usuario, como del *back-end* de la aplicación, la parte oculta de la app que almacena la información que esta necesita para su ejecución.

3.1 Herramientas utilizadas

3.1.1 Android Studio

Para el desarrollo de la aplicación (*front-end*) se ha utilizado el software Android Studio, concretamente la versión 6, API 23. Con esta versión, la aplicación puede ser utilizada por un 95.6% de los dispositivos móviles, lo cual es una buena elección.

Android Studio tiene una muy buena documentación oficial para el desarrollo de aplicaciones, lo que ha facilitado el trabajo en este punto (9).

3.1.2 Firebase

En la parte de *back-end*, se ha optado por utilizar una base de datos Firebase (10), de Google. Es una plataforma para el desarrollo de aplicaciones web y aplicaciones para móvil. Al ser compatible para las dos tecnologías, facilita la posibilidad de desarrollar la aplicación web para ser utilizada en ordenador de sobremesa por parte de los administradores; utilizando la misma base de datos.

Entre sus muchas posibilidades, se han utilizado en el desarrollo de esta aplicación las siguientes:

- Firebase Autenticación de usuarios: para el inicio y cierre de la sesión.
- Firestore Database (Cloud firestore): base de datos orientada al desarrollo de apps para dispositivos móviles. Es una base de datos no relacional que permite hacer consultas rápidas y en tiempo real.

Al utilizar Firebase, es posible acceder desde múltiples dispositivos, Tablet, pc y smartphone, mediante API que ya está creada. Esto facilita que se pueda usar una app ANDROID para los smartphones y que no sea incompatible con una app-web para PC.

3.1.3 Git

Es un sistema de gestión de control de versiones (11). Todos los cambios y modificaciones que se realizan en el proyecto son almacenados en este software que ayuda en la recuperación de trabajos, en entender el proceso llevado a cabo.

Junto a Git se ha utilizado el software SourceTree (12) y GitLab (13), ambos utilizan Git y facilitan su gestión.

3.2 Modelo de trabajo

El modelo de trabajo como ya se ha mencionado ha sido el de MVVM, el cual separa por un lado todo lo relacionado con las vistas e interfaz (ui o view), por otra parte, están las clases relacionadas con la base de datos (models) y, por último, el viewmodel que hace de enlace para las dos partes.

Las carpetas del proyecto están estructuradas según la siguiente tecnología:

3.2.1 UI (view)

Aquí se muestran las Activity y fragmentos que contienen la lógica de la aplicación para que está presente las funcionalidades requeridas:

- MainActivity.js (pantalla Guardias)

- MainAdapter.js (para la recyclerView de las guardias)
- NewAbsActivity.js (nueva ausencia)
- UserActivity.js (usuarios)
- UserAdpater.js (para la recyclerView de User)
- AuthActivity.js (para la autenticación de los usuarios, inicio de la app)

Variaciones tras la implementación

A los archivos anteriores se han añadido los siguientes:

- GroupActivity.js (pantalla que muestra los grupos de guardias por día y hora)
- GroupAdapter.js (para la recyclerView de los grupos de guardias)

La pantalla que muestra los grupos de guardias de cada día y hora (GroupActivity.js) no estaba prevista en el momento del diseño de la app. Se ha implementado finalmente para solventar el hecho de que no se ha podido completar la funcionalidad de asignar automáticamente un profesor a cada una de las guardias.

Cómo esta función debe hacerse de forma manual, el tener una pantalla adicional en la que se muestran que profesores están de guardia cada hora, facilita la asignación siendo esta más rápida.

- NoAuthActivity.js (pantalla de inicio de la app, sin autenticación, utilizada al fallar la AuthActivity.js)

Esta última pantalla se ha implementado al tener problemas con la activity AuthActivity.js. El alumno no logró que se ejecutará, a pesar de conectarse a *Firestore Authentication* y de seguir todas las indicaciones de la documentación oficial (XXX). Finalmente se optó por mantener la pantalla AuthActivity.js, para que pudiese ser evaluada por el tribunal; y se creó una nueva Activity *NoAuthActivity.js* cuya función simplemente es hacer de paso entre el inicio y la pantalla principal, sin registrar a los usuarios.

3.2.2 Models (data)

En este directorio se almacenan las clases de datos, que nos indican la estructura que deben tener los datos que se recogerán de la base de datos. Estas clases coinciden con las almacenadas en la base de datos Firebase.

- Absence
- User
- GroupOnCall

Variaciones tras la implementación

Las diferencias entre lo que se planteó al definir la base de datos y lo que finalmente se ha desarrollado, simplemente son atributos adicionales que se han incluido en cada una de las class data, por mejorar así la arquitectura de la base de datos y la lógica de la app. Estos ya se han mencionado en la sección 2.6, cuando se describía la arquitectura de la base de datos.

3.2.3 ViewModel

En esta carpeta se encuentran los archivos que hacen de unión entre los datos recibidos de la base de datos, y la lógica de las Activity que mostrarán esta información al usuario.

- MainViewModel (para la pestaña principal)
- UserViewModel (para la pestaña de usuarios)

Estas dos clases mantienen una observación continua a la base de datos, para detectar si hay alguna modificación. Si la hubiese, la aplicación recibiría los datos actualizados de forma inmediata. Esta es una de las razones por las que se ha escogido la arquitectura MVVM.

3.2.4 Domain\Data\NetWork

En este directorio se presenta una clase llamada *Repo (repositorio)*, la cual hace las llamadas a la base de datos FireStore (19), para recibir la información necesaria de las diferentes clases de datos: *Absence*, *User* y *GroupOnCall*.

3.3 Base de datos

Se ha utilizado Firestore Database (10) para almacenar toda la información necesaria para la app. Los datos se han subido haciendo uso de archivos nodejs (14), los cuales se han adjuntado en la entrega, en el archivo zip donde se encuentra la aplicación. La información almacenada se divide en tres colecciones:

- *Absence* (ausencias)
- *GroupOnCall* (grupos de profesores de guardia por hora lectiva)
- *User* (docentes)

En el esquema Ilustración 2-5, se muestra con más detalle los atributos y métodos de cada uno de las clases. Así como las variaciones que ha habido tras completar la implementación.

El alumno ha utilizado información ficticia para completar estas tres clases y así tener una base de datos funcional.

Para el caso de los usuarios, se ha listado un equipo docente donde el nombre de cada uno de los profesores hace referencia a la asignatura que imparten. Es decir, si se han listado cuatro profesores de matemáticas, estos se llamarán: Profesor m1, Profesor m2, Profesor m3 y Profesor m4. Y así con el resto de usuarios.

3.4 Pantallas

Las pantallas implementadas son las descritas en el apartado 2 Diseño:

- Pantalla de inicio de sesión;
- Pantalla de Guardias (pantalla principal);
- Pantalla Crear nueva ausencia (secundaria);
- Pantalla Usuarios (secundaria);

Variaciones tras la implementación

A estas cuatro pantallas se les ha añadido una más: Pantalla Grupo de guardias (secundaria);

El alumno no logró completar la funcionalidad en la que se asigna de forma automática un profesor a cada una de las guardias. De modo que para facilitar esta función que deberá hacerse de forma manual, se ha incluido una pantalla donde se muestran los profesores de cada grupo de guardia por día y hora.

Para la realización de cada una de las pantallas se han seguido las directrices que se mencionan a continuación:

3.4.1 Pantalla - Inicio de sesión

El cometido de esta pantalla es que el usuario que utiliza la app por primera vez tenga que registrarse con un email y password y estos queden registrados en la base de datos. Para el

resto de inicios de sesión, se deberá introducir de nuevo el email y el password y pulsar el botón de acceso.

De esta forma se cumplía uno de los objetivos no funcionales de la aplicación, descritos en la sección 1.3, el de asegurar una navegación segura.

Para ello, se ha utilizado la opción de Firebase Authentication, que facilita el registro de usuarios en una aplicación. Sin embargo, a pesar de haber seguido los pasos indicados (20), no se ha conseguido que la pantalla registre a los usuarios ni de paso a la pantalla principal de la app.

De modo que la solución que se ha encontrado, es hacer una réplica de la Activity que contenía esta lógica, pero eliminando la funcionalidad de registro de email y password por parte de Firebase. La nueva Activity tan sólo permite pasar a la pantalla principal si se completan los campos email y password y se pulsa después uno de los botones, "Registrarse" o "Acceder".

Esta nueva pantalla, no comprueba si el campo introducido es un email correcto; no se ha incluido lógica para esta comprobación ya que tan sólo se ha creado para tener una pantalla de transición entre el inicio y la pantalla principal de la app. Ya que la pantalla original de autenticación no ha funcionado.

La pantalla original de autenticación se llama: AuthActivity.js y su archivo xml es activity_auth.xml.

La pantalla que sustituye a la original, pero pierde la funcionalidad de registro se llama: NoAuthAcvity.js y su archivo xml es activity_noauth.xml.

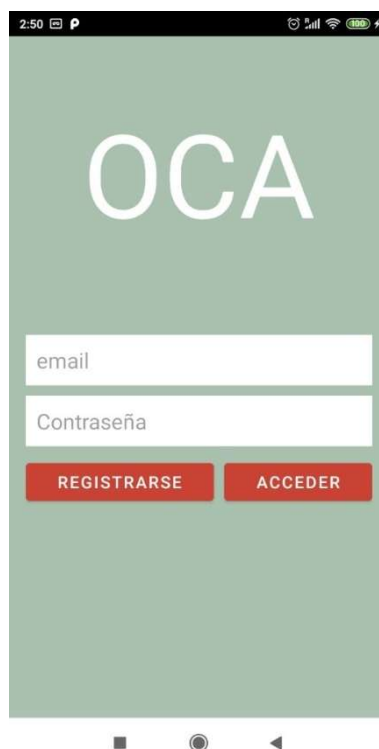


Ilustración 3-1 Pantalla de inicio de sesión

3.4.2 Pantalla Principal - Guardias

Esta es la pantalla principal de la aplicación, en ella se muestra un listado de todas las guardias que hay en el día de hoy y en los próximos días, se puede acceder a ellas haciendo un scroll vertical. En la parte de abajo muestra un menú para acceder a las pantallas secundarias: Usuarios (Usuarios), Grupo de guardias (Grupos) y Nueva ausencia (Añadir). Se escogieron estos nombres para el menú porque se consideró que con una sola palabra se hacía entender cuál era su cometido.

Como se puede observar en la imagen de la izquierda, para cada una de las guardias se muestra toda la información necesaria (hora, fecha, asignatura, profesor ausente, grupo, aula y tarea que deben realizar los alumnos). Se puede ver que en el campo donde debería indicarse el profesor de guardia asignado, se encuentra vacío.

El alumno no ha podido completar esta funcionalidad, al no saber cómo combinar los datos de diferentes clases provenientes de la base de datos. Los principales problemas fueron la falta de experiencia con bases de datos y la falta de tiempo.

Esta función deja de ser automática y pasa a ser manual, siendo el equipo directivo quien debería realizarla; aunque se deja la opción de que los propios docentes se asignen las guardias entre ellos. Para facilitar esta tarea, el alumno ha incluido en el menú una nueva opción que no se indicaba en el diseño de la app, la opción de "Grupos". Esta opción nos lleva a una pantalla secundaria donde se muestra un listado de los diferentes grupos de guardias para cada día y hora. En ella también se indica el contador de guardias de cada uno de los docentes, de forma que facilita la tarea de asignación de guardias.

Una vez que el equipo directivo ha consultado la pestaña secundaria "Grupos", ya puede asignar la guardia a un docente. Se utilizará la opción desplegable que se encuentra debajo de cada guardia, donde se muestran todo el equipo docente. Se escogerá una de las opciones y la pantalla de guardias quedará como se indica en la ilustración de la derecha.

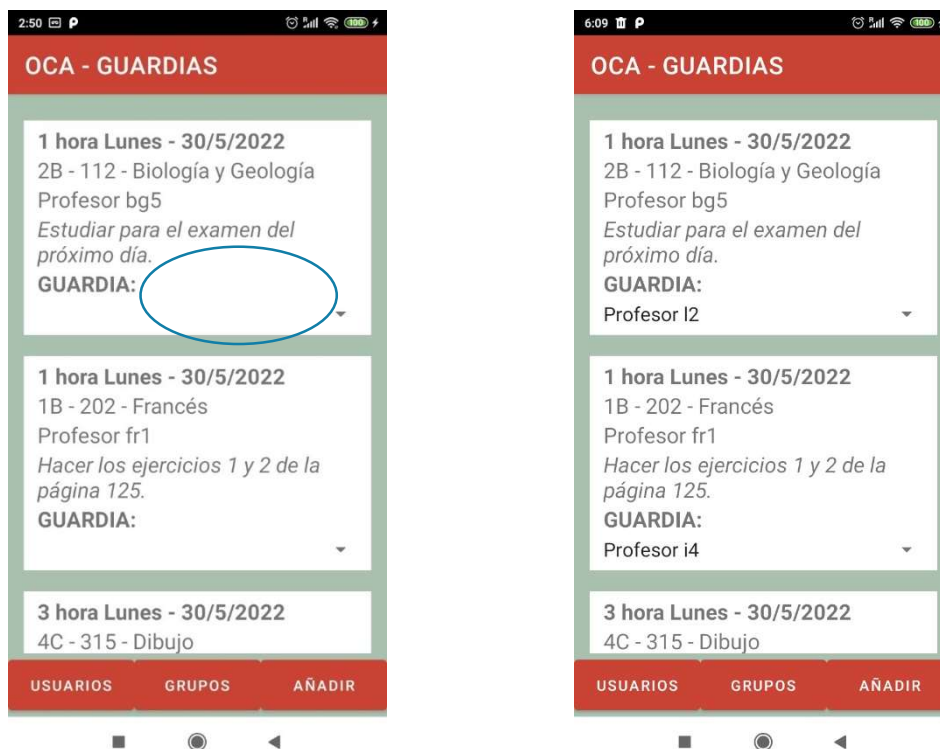


Ilustración 3-2 Pantalla principal – Guardias (Izquierda: El profesor de guardia no ha sido seleccionado; Derecha: El profesor de guardia ha sido seleccionado)

Esta pantalla está formada por el archivo MainActivity.js. La información es tomada de la base de datos Firestore, concretamente de la clase Absence.

El contenido se muestra en una recyclerView (19), asociada a un archivo adaptador MainAdapter.js; los archivos xml son activity_main.xml para el contenedor de la recyclerView y main_item_row.xml para mostrar cada uno de los elementos. Para el desplegable de las opciones de profesores de guardia se ha utilizado un spinner (17), que se alimenta de un archivo xml llamado array.

Se ha decidido utilizar una recyclerview, porque el contenido a mostrar puede variar y puede llegar a ser muy extenso, dependiendo del número de días para el que se permita mostrar las

guardias. De esta forma el usuario puede acceder a toda la información haciendo un scroll vertical.

Se ha completado el código para que cada vez que el usuario seleccione un profesor de guardia en el desplegable, este sea almacenado en la base de datos en el atributo "UserOnCall" de la colección "Absence" y así se muestre en esta misma pantalla de forma permanente. Sin embargo, no se ha conseguido, la aplicación no realiza esta función. En el archivo MainAdapter.js, se puede observar el código desarrollado para tal fin.

El motivo de que el código no realice esta función podría ser que estamos intentando almacenar un dato de una recyclerView, siendo su principal función el destruir los ítems una vez que dejan de mostrarse.

De modo que, aunque se seleccione un profesor de guardia para una guardia concreta, este no queda registrado en la base de datos. Cuando el usuario sale de la pantalla principal, la selección se deshace y no se recupera. Este sería un punto que habría que trabajar para mejorar la aplicación.

3.4.3 Pantalla Secundaria - Grupo de guardia

A esta pantalla se accede a través del menú de la pantalla principal, pulsando el botón "Grupo". Esta pantalla ha sido añadida tras no poder haber implementado la funcionalidad de automatizar la asignación del profesor de guardia a la misma.

Su objetivo es facilitar la asignación manual. Accediendo a esta pantalla el usuario puede observar los profesores que tienen guardia en cada una de las horas y días, además del contador de cada uno de ellos.



Ilustración 3-3 Pantalla Grupos (secundaria) – Muestra el listado de grupos de guardias para cada hora y día de la semana

Esta pantalla está formada por el archivo GroupActivity.js. La información es tomada de la base de datos Firestore, concretamente de la clase GroupOnCall.

El contenido se muestra en una recyclerView (19), asociada a un archivo adaptador GroupAdapter.js; los archivos xml son activity_group.xml para el contenedor de la recyclerView y group_item_row.xml para mostrar cada uno de los elementos.

Se ha decidido utilizar una recyclerview, porque el contenido a mostrar es extenso. De esta forma el usuario puede acceder a toda la información haciendo un scroll vertical. Se ha añadido un botón en la parte inferior, "Buscar"; la función del mismo sería facilitar la búsqueda de un grupo de guardia, para evitar tener que hacer scroll en toda la lista. Sin embargo, no se ha completado el código del mismo, tan sólo se ha añadido un Toast con un mensaje que aparece si lo pulsa el usuario, para indicarle que todavía no tiene funcionalidad.

Con el botón de la esquina inferior izquierda se vuelve de nuevo a la página principal de guardias.

3.4.4 Pantalla Secundaria - Nueva Ausencia

La función de esta pantalla es crear una nueva ausencia por parte del docente, de forma automática. Una vez creada, el equipo directivo debería aprobarla o denegarla, dependiendo de su elección, la ausencia sería incluida o no en la base de datos, en la colección "Absence".

La pantalla muestra un pequeño formulario para completar por el usuario. Cada uno de los campos tiene un desplegable para que el usuario escoja la opción que requiere, de esta forma se evitan posibles errores y facilita la tarea al docente.

Si alguno de los campos no es completado, la aplicación enviará un mensaje de aviso (Toast) indicando al usuario que debe completar el formulario para poder enviarlo. Tan sólo el campo de "Tarea" es opcional.

Para enviarlo se debe presionar el botón que se encuentra en la parte inferior de la pantalla. Este botón hace que la información rellena se adjunte a la base de datos, en la colección "Absences", ya que los campos a completar son los mismos que los atributos que esta clase presenta.

Las funcionalidades relacionadas con la administración no se han implementado en este proyecto, no estaban en el alcance. De modo, que la confirmación por parte de la administración de la ausencia no ha sido implementada. Por defecto, se ha permitido que toda nueva ausencia sea incorporada en la base de datos y tratada como si hubiese sido aceptada. De esta forma se ha podido validar esta funcionalidad de la aplicación.

Ilustración 3-4 Pantalla Nueva Ausencia (secundaria) – Muestra el formulario a rellenar por el usuario.

Esta pantalla está formada por el archivo `NewAbsActivity.js` y `activity_newabs.xml`. La información es almacenada en la base de datos `Firestore`, concretamente de la colección `Absence`.

Con el botón de la esquina inferior izquierda se vuelve de nuevo a la página principal de guardias.

3.4.5 Pantalla Usuarios

En esta pantalla aparece un listado con todos los profesores que trabajan en el centro, se hayan o no registrado en la app. El usuario puede ver información de los mismos como:

- Nombre y apellidos;
- Asignatura que imparte;
- Número de guardias (sustituciones) que ha realizado;
- Foto de perfil.

Esta pantalla se alimenta de la base de datos `Firestore`, concretamente de la colección `"Users"`. El listado está ordenado por el contador de guardias de forma descendente.

El contenido se muestra en una `recyclerView`, asociada a un archivo adaptador `UserAdapter.js`; los archivos `xml` son `activity_user.xml` para el contenedor de la `recyclerView` y `user_item_row.xml` para mostrar cada uno de los elementos.

Se ha decidido utilizar una `recyclerview` (19), porque el contenido a mostrar es extenso. De esta forma el usuario puede acceder a toda la información haciendo un `scroll` vertical. Se ha añadido un botón en la parte inferior, "Buscar"; la función del mismo sería facilitar la búsqueda de un usuario, para evitar tener que hacer `scroll` en toda la lista. Sin embargo, no se ha completado el código del mismo, tan sólo se ha añadido un `Toast` con un mensaje que aparece si lo pulsa el usuario, para indicarle que todavía no tiene funcionalidad.

Con el botón de la esquina inferior izquierda se vuelve de nuevo a la página principal de guardias.

Para la incorporación de las imágenes se ha utilizado la librería `Glide` (18).



Ilustración 3-5 Pantalla Grupos (secundaria) – Muestra el listado de grupos de guardias para

3.5 Test de validación

En este apartado se muestra los resultados obtenidos tras el desarrollo de una serie de test de validación manual, para comprobar si la aplicación cumple los requisitos que se establecieron al principio del desarrollo, tanto funcionales como no funcionales.

Estos test consisten en una serie de pruebas específicas para comprobar y validar si el software hace lo que se pide y espera. Para llevarlas a cabo se va a ir pasando por cada una de las pantallas, comprobando que cada una de las funcionalidades y/o acciones funciona correctamente.

Los dispositivos utilizados para este test fueron un dispositivo físico Xiaomi y un emulador Android Pixel 5.

En resumen, todas las pantallas superan esta validación, quizás algún requisito se supera de forma parcial, pero el resultado global es satisfactorio.

A continuación, se describen, de forma más explícita, las pruebas realizadas para cada una de las pantallas.

3.5.1 Pantalla Principal – Guardias

De esta pantalla se espera que muestre el listado de guardias actualizado, mostrando toda la información de la misma. Así mismo se espera que la información se actualice en el momento que haya una modificación en los datos

Al no incluir la funcionalidad automática de asignar un profesor de guardia, se ha modificado su planteamiento para que se realice de forma manual. Es decir, que lo que se espera de la pantalla es que esta acción se pueda realizar manualmente, o al menos que la facilite.

Las pruebas realizadas están en línea con estos requisitos.

Pantalla Principal - Guardias	
Prueba realizada	Resultado obtenido
Se muestra el listado de guardias completo.	Si
Se muestra toda la información almacenada en la base de datos para la colección Absence.	Si
La apariencia y formato de los ítems de la recyclerView es correcta.	Si
Se muestra el listado de guardias en orden temporal.	Si , pero con ayuda del administrador.
El usuario puede acceder y visualizar todo el listado haciendo scroll.	Si
Si se añade una ausencia, esta se actualiza y visualiza en la pantalla principal.	Si
El usuario puede acceder a todos los desplegados de profesor de guardia.	Si
Si el usuario accede a uno de los desplegados, puede seleccionar un profesor de guardia.	Si

Pantalla Principal - Guardias	
La selección del desplegable es visible una vez que este se cierra.	Si
Si se sale de la pantalla de guardias, cuando se regresa se mantiene la selección del profesor de guardia.	No , como ya se ha comentado, esta selección no queda registrada en la base de datos.
Los botones del menú inferior cumplen su función cuando son pulsados.	Si

Tabla 3-1 Prueba de validación – Pantalla principal.

Observamos que esta pantalla cumple perfectamente los requisitos de mantener informado al usuario mostrando todo el listado de guardias actualizado y de una forma clara.

La segunda funcionalidad queda cubierta de forma parcial, la pantalla sí que permite seleccionar de forma manual al profesor de guardia, sin embargo, este no queda registrado en la base de datos.

Como ya se ha comentado en otras secciones, el alumno ha desarrollado el código para tal fin, y este se puede visualizar en el archivo MainAdapter.js de la aplicación. Sin embargo, la aplicación no cumple esta funcionalidad.

Se cree que el motivo pueda ser el hecho de trabajar en una recyclerView. Es decir, que la función de registro del profesor seleccionado se realiza en el item que se va a mostrar en la recyclerView. Este item es temporal, a medida que el usuario hacer scroll, los ítems van siendo creado y eliminados. De modo, que la función no puede realizar la acción encomendada.

Una posible solución, sería hacer que cada uno de los ítems fuesen seleccionables, mostrando un fragment cada vez que fuesen seleccionados. En este fragment (ventana) se mostraría la información necesaria para seleccionar el profesor de guardia y un botón para guardar dichos datos en la base de datos. De esta forma, se podría solucionar el problema anterior.

Por falta de tiempo, no se han podido incluir esta medida para solventar esta carencia de la aplicación.

3.5.2 Pantalla Secundaria – Grupo de guardias

Esta pantalla fue creada para facilitar la asignación manual del profesor de guardia. Su cometido es mostrar el listado de grupos de guardias para cada hora y día de la semana. Las pruebas realizadas se centran en comprobar dicha funcionalidad.

Pantalla Secundaria - Grupo de Guardias	
Prueba realizada	Resultado obtenido
Se muestra el listado de grupo de guardias completo.	Si
Se muestra toda la información almacenada en la base de datos para la colección GroupOnCall.	Si
La apariencia de los ítems de la recyclerView es correcta.	Si
El usuario puede acceder y visualizar todo el listado haciendo scroll.	Si

Pantalla Secundaria - Grupo de Guardias	
Los botones del menú inferior cumplen su función cuando son pulsados.	Si. El botón "Buscar" no está habilitado, pero se muestra un mensaje informativo.

Tabla 3-2 Prueba de validación – Pantalla Grupo de guardias.

El test nos indica que esta pantalla cubre las funcionalidades esperadas, ayudar al usuario a asignar a un profesor de guardia.

3.5.3 Pantalla Secundaria – Nueva ausencia

La pantalla "Nueva Ausencia" está destinada para incluir una nueva ausencia en la base de datos y que sea mostrada en la aplicación, en la pantalla principal.

El formulario que se muestra intenta ser amigable con el usuario y facilitar su entrega, así como evitar que posibles se registren datos erróneos en la base de datos.

Las pruebas llevadas a cabo comprueban que se cumplen los puntos anteriores.

Pantalla Secundaria - Nueva ausencia	
Prueba realizada	Resultado obtenido
Se muestra el formulario completo para incluir una nueva ausencia.	Si, haciendo scroll vertical.
La apariencia de los campos es correcta.	Si
Se permite seleccionar las opciones en cada desplegable de cada campo.	Si
Se mantienen seleccionadas las opciones una vez que el desplegable se retira.	Si
En el campo "Tarea", se puede incluir texto, independientemente de la extensión del mismo.	Si
Los botones del menú inferior cumplen su función cuando son pulsados.	Si
Si no se completa algún campo del formulario, la ausencia no es registrada.	Si, un mensaje informativo es mostrado.
La ausencia es registrada en la base de datos, en la colección Absence, cuando el formulario está completo y se pulsa el botón "Enviar".	Si

Tabla 3-3 Prueba de validación – Pantalla Grupo de guardias.

La pantalla Nueva ausencia supera las pruebas de validación, confirmando así que cubre los requisitos esperados, incluir una nueva ausencia en la base de datos y que esta sea mostrada en la aplicación al resto de usuarios.

3.5.4 Pantalla Secundaria – Usuario

Esta pantalla debe mostrar un listado de todos los docentes que trabajan en el centro, junto a información significativa para la aplicación, como la asignatura, correo electrónico y número de guardias realizadas.

Las pruebas llevadas a cabo comprueban el cumplimiento de dicha funcionalidad.

Pantalla Secundaria - Usuario	
Prueba realizada	Resultado obtenido
Se muestra el listado de todos los usuarios completo.	Si , haciendo scroll vertical.
Se muestra toda la información almacenada en la base de datos para la colección User.	Si
Se puede observar el número de guardias de cada uno de los docentes de una forma rápida.	Si , vienen ordenados según el número de guardias en orden descendente, lo que ayuda a visualizar el contenido global.
Se muestra información adicional que puede ser útil para el usuario.	Si , se muestra una foto de perfil de cada docente.
La apariencia de los ítems de la recyclerView es correcta.	Si
El usuario puede acceder y visualizar todo el listado haciendo scroll.	Si
Los botones del menú inferior cumplen su función cuando son pulsados.	Si . El botón "Buscar" no está habilitado, se muestra un mensaje informativo.

Tabla 3-4 Prueba de validación – Pantalla Usuario.

Esta pantalla supera el test de validación, ya que permite visualizar toda la información de los usuarios almacenada en la base de datos.

De hecho, de forma adicional muestra una foto de perfil de los usuarios.

3.5.5 Pantalla Inicio de sesión

En este punto se va a testear la pantalla de inicio de sesión que realmente se está usando en la aplicación. Es decir, la que corresponde a la actividad NoAuthActivity.js y activity_noauth.xml.

La pantalla original (AuthActivity.js y activity_auth.xml) tenía como funcionalidad registrar el email y password de cada usuario que se registrase. Sin embargo, como ya se ha mencionado en la sección anterior, no se logró completar de forma satisfactoria, no pudiendo ser usada en la aplicación.

Esta pantalla NoAuthActivity, supe a la original, perdiendo parte de la funcionalidad de la anterior. Esta pantalla que se va a testear, tiene como cometido permitir el acceso a la pantalla principal, cuando el usuario completa los campos de email y password. No es su cometido, registrarlos en la base de datos.

De modo que las pruebas de testeo siguen esa línea.

Pantalla Inicio de sesión	
Prueba realizada	Resultado obtenido
La apariencia de la pantalla es la correcta y esperada.	Si
Se permite editar los campos de email y password.	Si
Los botones "Registrarse" y "Acceso" cumplen su funcionalidad, acceder a la pantalla principal.	Si
Si el usuario no completa los campos, los botones "Registrarse" y "Acceso" no permiten acceder a la aplicación.	Si, un mensaje informativo se muestra.

Tabla 3-5 Prueba de validación – Pantalla Inicio de sesión (AuthActivity.js).

Este proceso de pruebas nos ha indicado que la funcionalidad y aspecto de la aplicación es el esperado. Salvando aquellos puntos donde ya se conocía que la funcionalidad no era la esperada, entre ellos:

4 Conclusiones

Este proyecto describe el desarrollo de una aplicación en Android cuyo objetivo se centra en la gestión de guardias en centros de educación.

Los objetivos principales de la aplicación son:

- Ahorrar tiempo al equipo docente, automatizando la gestión de las guardias;
- Mantener a los usuarios actualizados e informados en todo momento, independientemente de donde se encuentren;
- Mantener una equidad y parcialidad a la hora de asignar profesores de guardia;

Para su desarrollo se ha llevado a cabo un estudio de benchmarking, analizando las aplicaciones disponibles en el mercado que cubriesen este campo de la docencia. Encontrando muy pocas o ninguna que se centre en la gestión de guardias.

Los requisitos funcionales y no funcionales que se le han exigido a la aplicación desde su inicio, y que se han mantenido hasta el final de su desarrollo han sido:

- Visualización de las guardias y de los profesores asignados a las mismas;
- Posibilidad de registrar una nueva ausencia;
- Actualización inmediata de la información mostrada en la aplicación, en el momento que haya una modificación de la misma;
- Facilitar la navegación mediante una interfaz fácil e intuitiva;
- Garantizar el proceso de inicio de sesión seguro;
- Almacenar y consumir los datos mediante un servicio de base de datos;

Alguno de los requisitos que se propuso al inicio del desarrollo, fueron desestimados a medida que el trabajo avanzaba, porque se decidió priorizar aquellos puntos más valorados por los usuarios potenciales. Como es el caso de:

- Implementar una aplicación web multidispositivo;
- Permitir reservar material docente y visualizar aulas vacías a través de la app;

Estos puntos serán indicados como futuros desarrollos de la aplicación, ya que aportarían valor a la misma.

El trabajo de desarrollo se ha dividido en dos partes principales, la de diseño y la de implementación.

Diseño

En la parte de diseño, se ha realizado un estudio del ecosistema potencial de la app; analizando diferentes contextos de uso y recopilando información de usuarios potenciales mediante herramientas de análisis como encuestas. Esta información se ha utilizado para desarrollar un árbol de navegación y definir las funciones y servicios que debería proporcionar la aplicación. Este punto ha servido para priorizar unas funcionalidades frente a otras, modificando así el planteamiento inicial de la aplicación.

En este punto, fue cuando se desestimó sacar del alcance el requisito de permitir reservar material docente y visualizar aulas vacías; priorizando el resto de funcionalidades y servicios. Los requisitos que se mantuvieron, mencionados arriba, se ampliaron para albergar algún servicio secundario más, como: permitir incluir la tarea de los alumnos cuando se registra una nueva ausencia; y mostrar un histórico de guardias realizadas por cada profesor.

Se elaboró un prototipo de las pantallas más importantes de la aplicación, para poder visualizar lo que se quería conseguir y cómo hacerlo. El esquema planteado fue de mantener una pantalla principal que asumiese el servicio más prioritario, mostrar

información actual de las guardias diarias; y presentar un menú en la parte inferior de la misma que permitiese acceder a las pantallas secundarias. Las pantallas secundarias nos mostrarían el resto de servicios: histórico de usuarios y registrar nuevas ausencias.

Diferentes casos de uso se elaboraron para así completar el flujo de uso de la aplicación.

El diseño de la arquitectura escogido fue MVVM (Model View ViewModel), el cual facilita la comunicación entre la base de datos y la lógica de la aplicación. Permite que cualquier modificación de los datos sea notificada y actualizada en la aplicación. Este fue el principal motivo de su elección, ya que, para esta aplicación, es vital que los datos sean actualizados al momento.

Implementación

En esta parte del diseño se seleccionaron las diferentes herramientas a utilizar en el desarrollo de la aplicación, así como la estructura de la base de datos. Destacamos las siguientes:

- Para el software de la app se ha utilizado Android Studio;
- Se escogió una base de datos Firebase, de Google, permitiendo así poder usar su opción de autenticación y de almacenamiento de datos.

Se desarrollaron las pantallas descritas en el prototipo de diseño de la aplicación. A estas se le añadió una más, Pantalla Grupo de guardias (secundaria):

- Pantalla de inicio de sesión;
- Pantalla de Guardias (pantalla principal);
- Pantalla Crear nueva ausencia (secundaria);
- Pantalla Usuarios (secundaria);
- Pantalla Grupo de guardias(secundaria);

Durante el desarrollo de las mismas, el alumno no pudo completar alguna de las funcionalidades o requisitos establecidos durante todo el proyecto: la asignación automática del profesor de guardia a cada una de las ausencias. De modo que se modificó dicho requisito pasando a ser una asignación manual, realizada por el usuario, pero en la propia aplicación para que así no perdiese la funcionalidad principal de la app: mantener informado en todo momento a los usuarios.

Para ello, se añadió una pantalla más, Pantalla Grupo de guardias (secundaria), la cual facilitaría la tarea de asignar un profesor de guardia por parte del usuario o equipo directivo. También se modificó la estructura de la pantalla principal para poder cumplir así este cometido.

Las pantallas arriba indicadas, permiten al usuario disfrutar de todas las funcionalidades indicadas en esta sección.

Una serie de pruebas de validación fueron realizadas, para así comprobar el funcionamiento de las mismas.

4.1 Planificación

La planificación inicial del proyecto se vio alterada sustancialmente por la subestimación de las horas de cada apartado, principalmente en la parte de implementación.

A este punto se le debe añadir que, a lo largo del desarrollo, el tiempo disponible de dedicación se vio disminuido porque la situación laboral del empleado cambió, comenzando a trabajar como docente. La falta de tiempo disponible hizo que se priorizase sobre algunos aspectos de la aplicación, dejando a otros fuera del alcance de la misma.

Otro punto a tener en cuenta ha sido el uso de modelos y herramientas como Firestore y MVVM, para las que el alumno no tiene mucha experiencia, suponiendo así un hándicap

añadido. Sin embargo, han sido resueltos con satisfacción por parte del alumno, consiguiendo buenos resultados.

La suma de todos estos puntos ha sido la causa de que no se haya seguido la planificación inicial, no consiguiendo todos sus hitos en el tiempo marcado, pero si solucionándolos.

4.2 Funcionalidades y requisitos

Como ya se ha mencionado, las funcionalidades que se han mantenido durante todo el desarrollo del proyecto son las siguientes:

- Visualización de las guardias y de los profesores asignados a las mismas;
- Posibilidad de registrar una nueva ausencia;
- Actualización inmediata de la información mostrada en la aplicación, en el momento que haya una modificación de la misma;
- Facilitar la navegación mediante una interfaz fácil e intuitiva;
- Garantizar el proceso de inicio de sesión seguro;
- Almacenar y consumir los datos mediante un servicio de base de datos;

Las cuales tenían como finalidad cubrir los siguientes puntos:

- Ahorrar tiempo al equipo docente, automatizando la gestión de las guardias;
- Mantener a los usuarios actualizados e informados en todo momento, independientemente de donde se encuentren;
- Mantener una equidad y parcialidad a la hora de asignar profesores de guardia;

4.2.1 Funcionalidades y requisitos cumplidos

Prácticamente todas las funcionalidades han sido cubiertas. La aplicación desarrollada permite:

- Visualización de las guardias;
- Posibilidad de registrar una nueva ausencia;
- Actualización inmediata de la información mostrada en la aplicación, en el momento que haya una modificación de la misma;
- Facilitar la navegación mediante una interfaz fácil e intuitiva;
- Almacenar y consumir los datos mediante un servicio de base de datos;

Estos servicios permiten que el equipo directivo y docente ahorre tiempo en la gestión de las guardias y los usuarios se mantienen informados en todo momento.

4.2.2 Funcionalidades y requisitos no cubiertos

Algunos de los requisitos incluidos en el proyecto no han podido ser completados de forma total, lo cual debe ser comentado para poder analizar el motivo de ese alcance y como podría solucionarse.

1. La aplicación no permite asignar de forma automática a los profesores de guardia.

Este era uno de los requisitos iniciales, ya que ahorraría tiempo al equipo directivo. Sin embargo, no ha sido posible desarrollar la lógica que se encargaría de realizar esta acción.

La causa principal es la falta de tiempo disponible para desarrollar el proyecto, ya que al ser el alumno un desarrollador con poca experiencia requiere de un tiempo extra para formarse y profundizar en determinados aspectos.

Para solventar esta carencia, el alumno ha modificado esta funcionalidad pasando a ser:

- La aplicación permite asignar de forma manual a los profesores de guardia;

Para ello, se ha modificado el diseño de la aplicación, añadiendo una pestaña más para facilitar esta tarea, así como incluir un desplegable en la pantalla principal para que el usuario pudiese seleccionar a los profesores asignados a las guardias.

Estas modificaciones funcionan dando este servicio al usuario, pero de forma parcial, ya que no se ha conseguido que el profesore seleccionado para la guardia quede registrado en la base de datos.

De modo que se podría considerar que esta funcionalidad o requisito, se ha completado de forma parcial.

En las secciones anteriores se ha indicado posibles soluciones, como el uso de fragments u otros elementos que proporciona Android. Este punto se mostrará como posibles mejoras o desarrollos futuros.

2. La aplicación no garantiza el proceso de inicio seguro

Este requisito era uno de los que se ha mantenido durante todo el desarrollo.

El alumno ha desarrollado la lógica para tal fin, creando la pantalla, su interfaz y su código y haciendo uso de la opción de la base de datos Firebase Authentication. Sin embargo, no ha podido conseguir que este trabajo funcionase.

Para solucionar parcialmente este punto, se ha creado una pantalla similar a la indicada pero sólo con la funcionalidad de acceder a la aplicación, sin registro de usuario.

Ambas pantallas se encuentran dentro de la aplicación, pero sólo una de ellas es la que está activa.

De nuevo el motivo principal de esta carencia en la aplicación es el tiempo disponible por parte de la alumna para su desarrollo, sumado a la corta experiencia en desarrollo de app, lo que hace que necesite invertir más tiempo en formación cuando se enfrenta a uno de estos casos.

4.3 Líneas futuras de desarrollo

4.3.1 Mejoras de la aplicación

Incluiremos primero aquellos puntos que deberían ser mejorados:

1. Permitir que la asignación manual del profesor de guardia sea registrada en la base de datos para su visualización por parte del resto de usuarios;
Para este punto, el uso de fragment sería adecuado.
2. Conseguir que el inicio de sesión sea de forma segura, manteniendo un registro de los usuarios. Incluir también un apartado de cierre de sesión;
3. Introducir la API Calendar Picker (15) y (16) de Android, para seleccionar las fechas de nuevas ausencias, y así registrar las ausencias en la base de datos con una variable tipo Timestamp. Esto ayudaría en su visualización, para ordenar los ítems o mostrar sólo los que se encuentran en un rango determinado;
4. Utilizar los elementos Fragment en lugar de Activities para optimizar la fluidez de la aplicación;

4.3.2 Futuros desarrollos

Dentro de este punto incluiremos requisitos que se indicaron en un inicio, pero después fueron sacados del alcance del proyecto. Estos puntos mejorarían la aplicación aumentando su funcionalidad:

1. Desarrollo de la aplicación en versión web, para que pueda ser utilizada en ordenadores y pantallas. Esta podría utilizar la misma base de datos que usa la app aquí presentada;
2. Desarrollar la parte de administrador, implementando entre otros, los servicios de:

- a. Alimentación de la base de datos;
- b. Confirmación de las ausencias;

Esta parte estaría enlazada con la primera, ya que la opción de administrador tiene sentido si se realiza desde una aplicación web, porque facilita su uso.

- 3. Añadir un apartado relacionado con la reserva de material y aulas, así como para poder visualizar que aulas se encuentran libres en cada momento.

5 Glosario

1. App: *Application* en inglés, aplicación en español;
2. Firebase: Plataforma de Google, que tiene diferentes funcionalidades entre ellas ser utilizada como base de datos, para el desarrollo de aplicaciones en dispositivos móviles.
3. Firestore: Cloud Firestore, opción que proporciona Firebase como almacenamiento de datos;
4. MVVM: *Model View ViewModel*, arquitectura de diseño de una aplicación. Se basa en distribuir la lógica del desarrollo en tres campos: model, view y viewModel.
5. RecyclerView: elemento para la interfaz de una aplicación, que proporciona Android Studio para desarrollar aplicaciones, el cual es un contenedor que muestra un listado de ítems;
6. Fragment: elemento para la interfaz de una aplicación, que proporciona Android Studio para desarrollar aplicaciones, el cual muestra una vista invirtiendo menos recursos de la aplicación;
7. Activity: elemento de la lógica de una aplicación, que proporciona Android Studio para desarrollar aplicaciones. Está relacionada con cada una de las pantallas que interactúa el usuario;
8. UML: Unified Modeling Language: Es el lenguaje de modelado de sistemas software más estandarizado;

6 Bibliografía

1. Orden de 22 de agosto de 2002 del Departamento de Educación y Ciencia del Gobierno de Aragón, que modifica el Real Decreto 83/1996 de 26 de enero, por el que se aprueba el Reglamento orgánico de los institutos de Educación secundaria. *Boletín Oficial de Estado*, Madrid, 21 de febrero de 1996, número 45.
<https://www.boe.es/buscar/pdf/1996/BOE-A-1996-3834-consolidado.pdf>
2. Educación 3.0 Líder informativo en innovación educativa. [Visitado el 28-02-2022]
<https://www.educaciontrespuntocero.com/recursos/plataformas-gestion-escolar/>
3. Rodríguez García, A. M.: *App de Guardias – Grado en Ingeniería Informática*. Universidad de Alicante. Alicante, septiembre 2017. [en línea]
https://rua.ua.es/dspace/bitstream/10045/69688/1/Aplicacion_multiplataforma_de_gestion_de_cen_RODRIGUEZ_GARCIA_ANTONIO_MIGUEL.pdf
4. Erustes Lobato, J. A.: *iNotHere – Proyecto Final de Carrera .NET*. Universitat Oberta de Catalunya. Barcelona, 2011-2022. [en línea]
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/14978/8/jerustesTFC062012memoria.pdf>
5. Marín Altaba M.A. *Gestión de las guardias – Centros de educación*. Encuesta realizada por el alumno en Google formularios [on line]. Disponible en el enlace;
<https://forms.gle/fzREDyz6fWwvpDTG6>
6. Web para realizar mapa UML <https://miro.com/es/>
7. Esquema MVVM [en línea] <https://www.adictosaltrabajo.com/2020/06/05/patron-mvvm-en-swiftui/>
8. Arquitectura de app recomendada. [en línea]
<https://developer.android.com/jetpack/guide?hl=es-419#recommended-app-arch>
9. Android Studio developers [en línea] <https://developer.android.com/>
10. Firebase [en línea]
https://firebase.google.com/?gclid=Cj0KCQjwvqeUBhCBARIsA0dt45ZdYvZDDqLoaO8pRvnwzKWq1JZ3uL7fKERRwjkQ7PAI_aPylKfggaArnuEALw_wcB&gclidsrc=aw.ds
11. Git – control de versiones [en línea] <https://git-scm.com/>
12. Sourcetree [en línea] <https://www.sourcetreeapp.com/>
13. GitLab [en línea] <https://about.gitlab.com/>
14. Tutorial – uso de json en Firestore [en línea]
<https://www.youtube.com/watch?v=BWNiMQkW3u4>
15. Tutorial – uso de la API DatePicker en Kotlin [en línea]
<https://www.youtube.com/watch?v=GOYIat0QC6A>
16. DatePicker - Información oficial – Android [en línea]
<https://developer.android.com/reference/android/widget/DatePicker>
17. Cómo agregar un menú desplegable en Android Studio [en línea]
<https://code.tutsplus.com/es/tutorials/how-to-add-a-dropdown-menu-in-android-studio--cms-37860>

18. Glide – Create apps & websites [en línea] <https://www.glideapps.com/>
19. Tutorial - Uso de Android Studio MVVM + Firebase + RecyclerView [en línea] https://www.youtube.com/watch?v=T-rZJ_NZM8Q
20. Autentica con Firebase en Android mediante un sistema de autenticación personalizado. *Firebase Documentation* [en línea] <https://firebase.google.com/docs/auth/android/custom-auth?hl=es>

7 Anexos

7.1 Resultados de la encuesta.

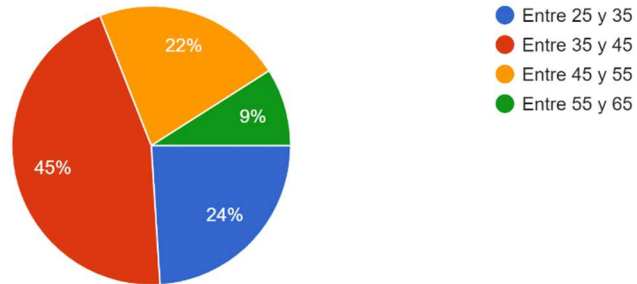
Gestión de las guardias – Centros de educación

Encuesta realizada por el alumno María Isabel Marín

Resultados

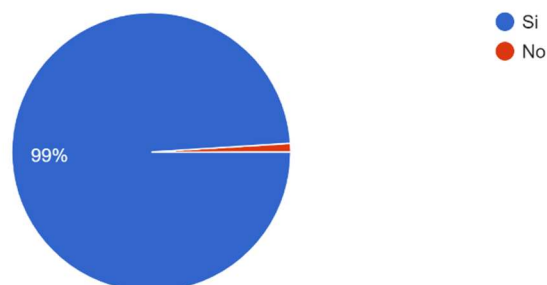
Edad

100 responses



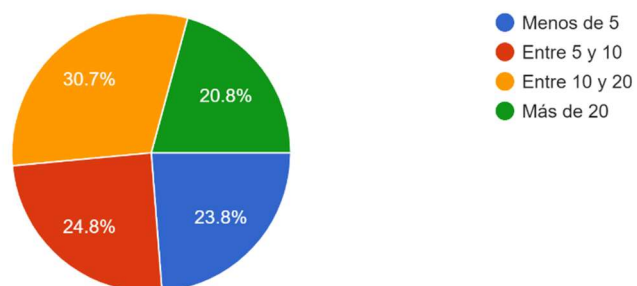
¿Trabajas como docente?

100 responses



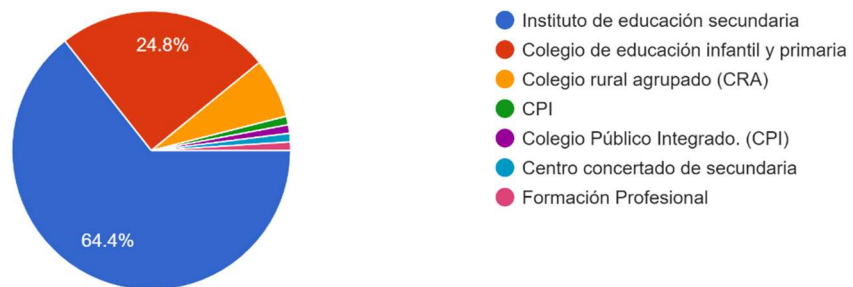
Años de experiencia en la docencia

101 responses



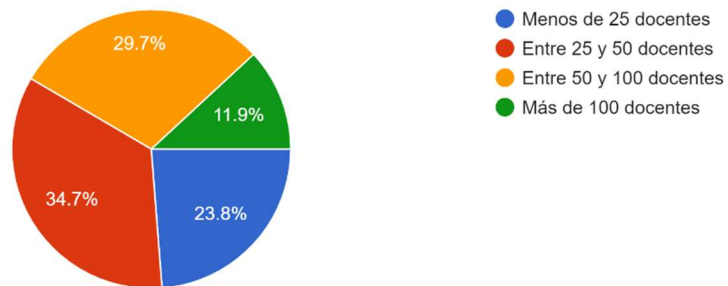
¿En qué tipo de centro has trabajado por última vez o actualmente trabajas?

101 responses



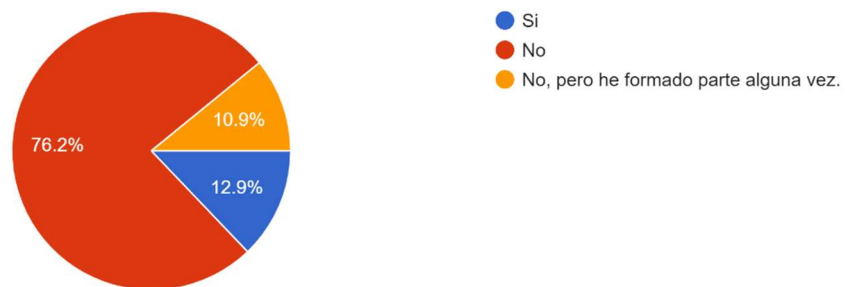
¿En que rango estaría el centro en el que trabajas o el último en el que has trabajado?

101 responses



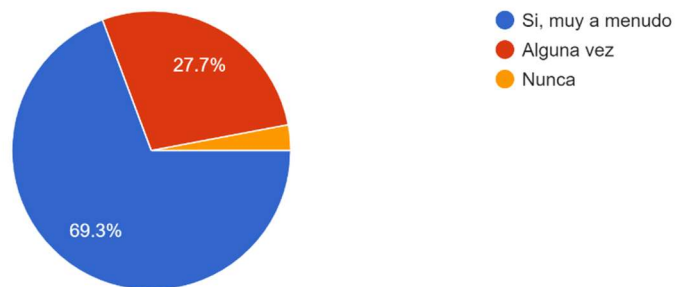
¿Formas parte del equipo directivo? (dirección, jefatura de estudios o secretaría)

101 responses



¿Utilizas aplicaciones de móvil, webs o herramientas similares en tu trabajo?

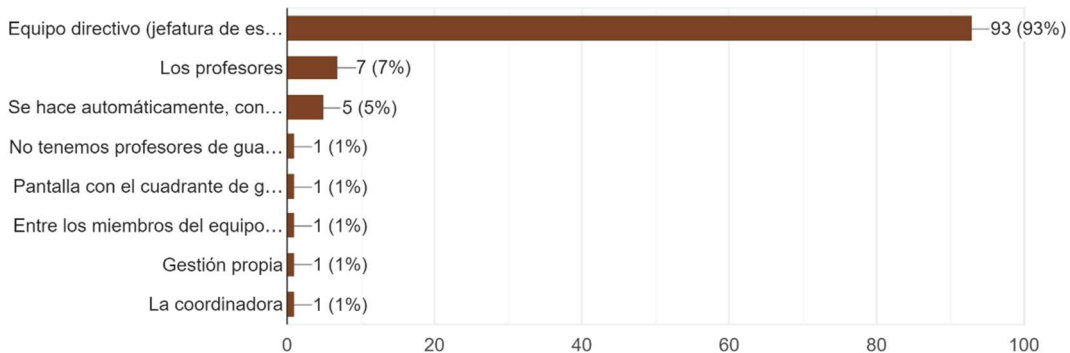
101 respuestas



¿Quién se encarga de registrar los profesores ausentes y comunicarlo a los profesores de guardia?

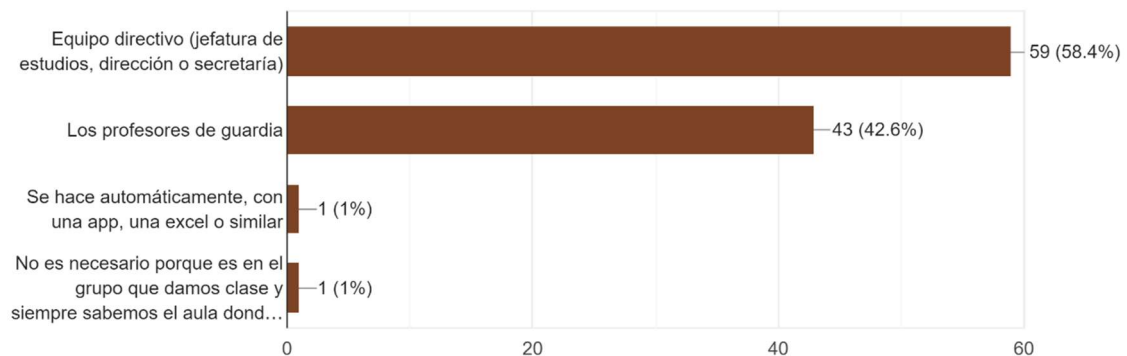
Selecciona todas las opciones que consideres.

100 respuestas



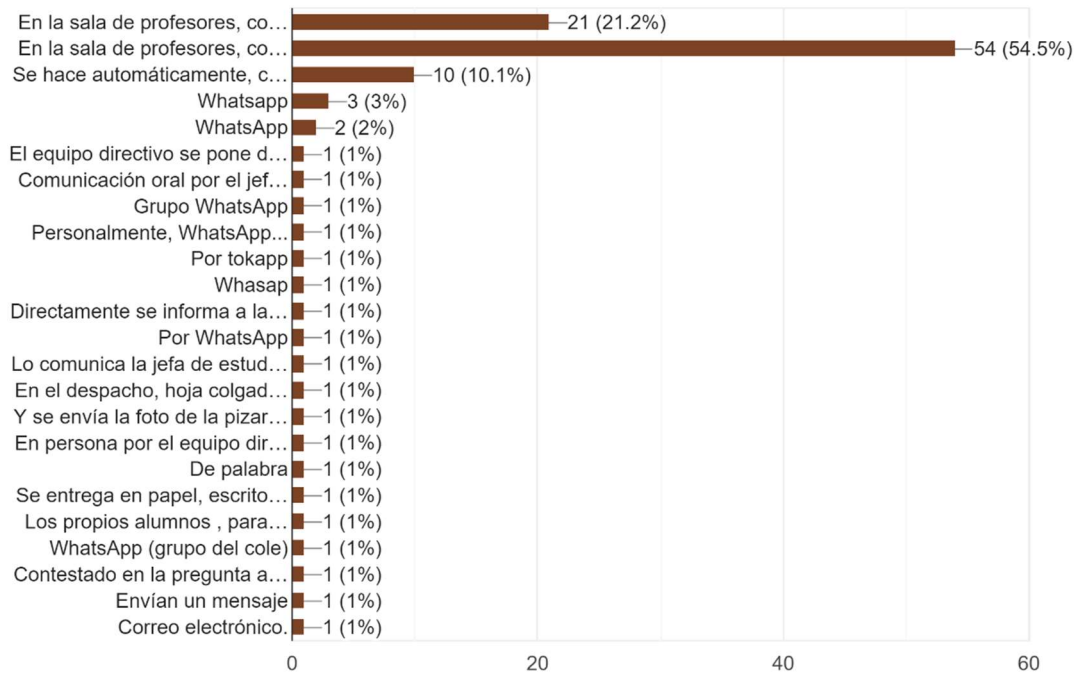
¿Quién se encarga de repartir entre los profesores de guardia, las aulas que hay que cubrir por ausencia de un docente?

101 respuestas



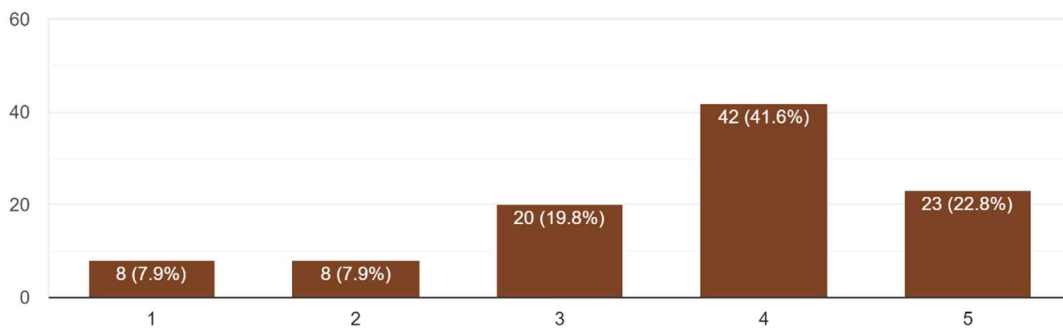
¿Cómo se comunica a los profesores de guardia cuáles son las aulas que no tienen profesor?

99 respuestas



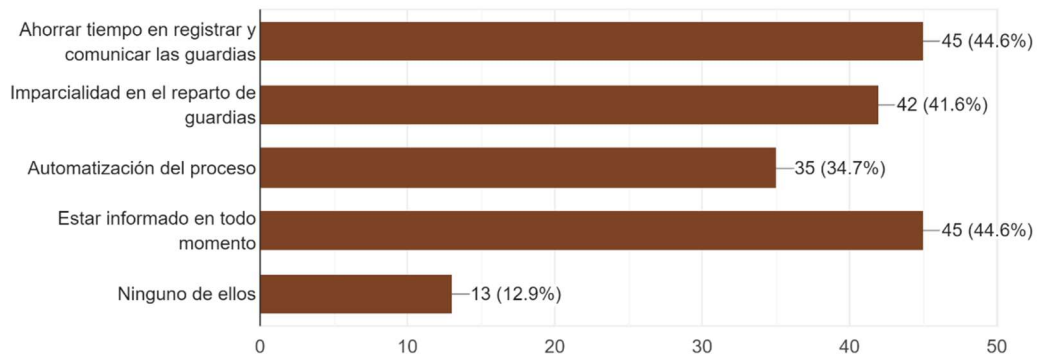
Valora del 1 al 5 el método usado para gestionar las guardias, siendo 1 muy mejorable y 5 muy eficiente.

101 respuestas



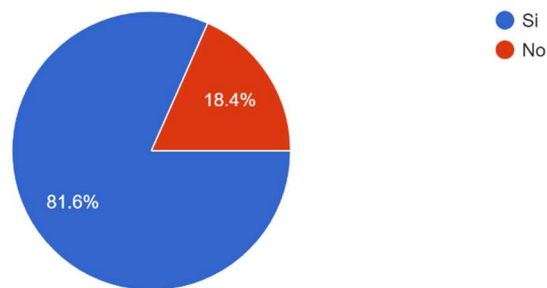
Si una app de móvil de gestión de guardias te ofreciese estos servicios, selecciona los dos que más valorarías.

101 respuestas



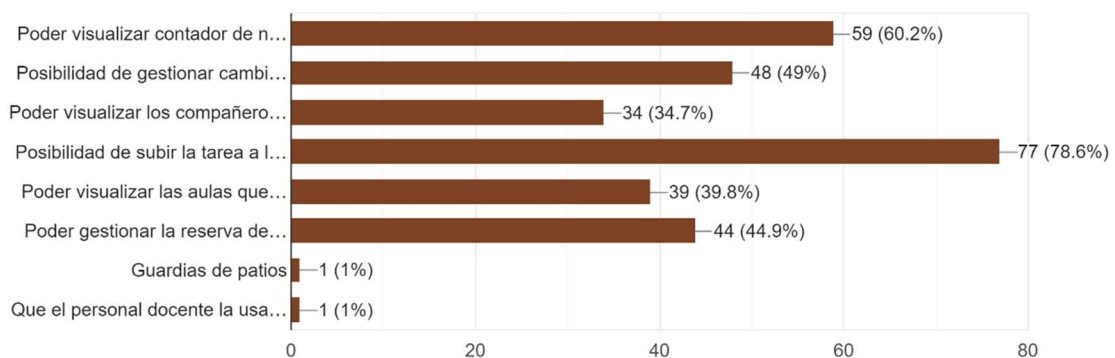
¿Crees que sería útil disponer de una app móvil y/o web con todos los puntos anteriores para gestionar las guardias?

98 respuestas



¿Qué características adicionales añadirías a la aplicación?

98 respuestas



Si existiera una app o web para la gestión de guardias, ¿la usarías?
99 responses

