

IES Santiago Hernández

Desarrollo de un sistema para el control de la asistencia

Desarrollo De Aplicaciones Multiplataforma Dual

Curso 2022 - 2023

30 de marzo de 2023

Lucas Tejel Alejo

Tutores: María Pilar Lozano Bernal y Edgar Bernáldez Mangas

2. ABSTRACT.....	4
3. DATOS DESCRIPTIVOS.....	5
DESCRIPCIÓN GENERAL.....	5
MARCO TEÓRICO.....	5
PRINCIPIOS TECNOLÓGICOS.....	8
NORMAS TÉCNICAS APLICABLES.....	8
ANÁLISIS DE LA REALIDAD Y ENTORNO DE LA EMPRESA COLABORADORA.....	9
JUSTIFICACIÓN FUNCIONAL.....	9
JUSTIFICACIÓN TECNOLÓGICA.....	9
MARCO LEGAL.....	10
DESTINATARIOS.....	11
4. DESARROLLO DEL PROYECTO.....	12
4.1. ACUERDO DE PROYECTO.....	12
REQUISITOS FUNCIONALES.....	12
REQUISITOS NO FUNCIONALES.....	13
ESPECIFICACIONES TÉCNICAS.....	14
4.2. ACTIVIDADES Y CRONOLOGÍA.....	15
4.2.1. DIAGRAMA DE GANTT.....	15
4.2.2. DOCUMENTO DE ANÁLISIS Y DISEÑO.....	17
MODELO ENTIDAD-RELACIÓN.....	17
MODELO RELACIONAL.....	18
CASOS DE USO.....	19
DESCRIPCIONES DE CASOS DE USO.....	20
DIAGRAMA DE CLASES DEL SERVIDOR.....	40
DIAGRAMA DE CLASES EN LA APLICACIÓN ANDROID.....	41
DIAGRAMA DE SECUENCIA EDICIÓN GRUPO.....	43
4.2.3. PRUEBAS DEL SISTEMA.....	44
4.2.4. DOCUMENTACIÓN PARA EL CLIENTE.....	47
MANUAL DE INSTALACIÓN Y CONFIGURACIÓN.....	47
MANUAL DE USUARIO.....	48
4.3. METODOLOGÍA SEGUIDA.....	52
4.4. RECURSOS.....	52
MATERIALES.....	52
INFRAESTRUCTURA Y EQUIPAMIENTO.....	53
HUMANOS.....	54
5. PRESUPUESTO.....	56
5.1. GASTOS.....	56
5.2. INGRESOS.....	56
5.3. BENEFICIO.....	57

6. RESULTADOS Y CONCLUSIONES.....	58
7. BIBLIOGRAFÍA.....	59
8. ANEXOS.....	61
DISEÑO DE INTERFACES GRÁFICAS.....	61
RELACIÓN DE HORAS RESPECTO A LA ESTIMACIÓN.....	74

2. ABSTRACT

The purpose of the system is to facilitate and streamline attendance management in various contexts, whether it's in a company, an educational environment, or any other group that needs to keep track of attendance.

The key feature of this system is the ability for attendees to use the NFC technology on their phones to register their entry into a group. This functionality allows users to mark their presence quickly and easily, eliminating the need for traditional registration methods such as paper attendance lists and saving administrators from manually inputting attendance.

Furthermore, the system provides administrators with a comprehensive set of tools to manage groups and their members. They can easily create new groups, add or remove members, and receive notifications when attendees arrive at designated locations, enabling them to stay updated on the attendance situation in real-time. The system gives administrators full control over the attendance process, as they can view a calendar for each user that displays the days they have attended, the days they have been absent, the days they have arrived late, days they have taken vacations, and holidays.

In summary, this attendance management system aims to simplify and enhance the attendance tracking process in various contexts. Attendees benefit from the convenience of using their phone's NFC technology to register their presence, while administrators have access to powerful tools for group and member management. With this system, the time-consuming task of attendance management is streamlined, allowing for efficient tracking and reducing the reliance on manual processes.

3. DATOS DESCRIPTIVOS

DESCRIPCIÓN GENERAL

El propósito del sistema es facilitar y agilizar la gestión de asistencia en diversos contextos, ya sea en una empresa, en un entorno educativo o en cualquier otro grupo que necesite llevar un registro de la asistencia.

La característica clave de este sistema es la posibilidad de que los asistentes utilicen la tecnología NFC de sus teléfonos para registrar su entrada en un grupo. Esta funcionalidad permite a los usuarios marcar su presencia de manera rápida y sencilla, eliminando la necesidad de métodos tradicionales de registro como listas de asistencia en papel, y evitando que los administradores gasten tiempo en colocar manualmente las asistencias.

Además, el sistema ofrece a los administradores un conjunto completo de herramientas para gestionar los grupos y sus miembros. Pueden crear nuevos grupos y añadir miembros o eliminarlos de forma sencilla. Además, reciben notificaciones cuando los asistentes acuden a los lugares designados, lo que les permite estar al tanto de la situación de asistencia en tiempo real. Les brinda un control total sobre el proceso de asistencia ya que para cada usuario pueden ver un calendario con aquellos días que los usuarios han asistido, los días que no han asistido, los días que han llegado con retraso, los días que tienen vacaciones y los días festivos.

MARCO TEÓRICO

- **Sistema operativo Android:** Android, creado por Android Inc. y adquirido por Google en 2005, se basa en Linux y Unix, y su objetivo inicial fue promover estándares abiertos en dispositivos móviles. Actualmente, Android es utilizado en una amplia gama de dispositivos y cuenta con millones de aplicaciones disponibles en su catálogo Google Play, incluyendo tanto aplicaciones gratuitas como de pago. Android se destaca por su seguridad y se han detectado pocas vulnerabilidades en su estructura.

Otras características notables de Android incluyen envío de mensajes MMS y SMS, navegador web, streaming, capacidad para trabajar con archivos multimedia en varios formatos, conectividad como WiFi, Bluetooth, NFC y datos móviles, lo que lo hace ideal para dispositivos móviles.

- **Android Studio:** es el entorno de desarrollo integrado (IDE) oficial para la creación de aplicaciones móviles para Android, y se basa en IntelliJ IDEA.

Además de ofrecer un potente editor de códigos y herramientas para desarrolladores, Android Studio también incluye una variedad de funciones adicionales que aumentan la productividad durante el desarrollo de aplicaciones móviles, tales como:

- Un sistema de compilación flexible basado en Gradle.
 - Un emulador rápido y completo.
 - Un entorno unificado para el desarrollo de aplicaciones móviles en todos los dispositivos Android.
 - La capacidad de aplicar cambios en tiempo real al código y recursos de la aplicación sin necesidad de reiniciarla.
 - Integración con GitHub y plantillas de código para ayudar en la creación de funciones comunes de aplicaciones móviles e importar código de ejemplo.
 - Una variedad de frameworks y herramientas de prueba.
 - Herramientas de Lint para identificar problemas de rendimiento, usabilidad, compatibilidad y otros.
- **Java:** Java es uno de los lenguajes de programación más populares y ampliamente utilizados en el mundo del desarrollo de software. Es conocido por su portabilidad, ya que el código Java se puede ejecutar en cualquier plataforma que tenga instalado el entorno de ejecución Java (JRE). En el caso de Android, Java es el lenguaje principal utilizado para desarrollar aplicaciones para este sistema operativo móvil.

En el contexto de una aplicación Android que utiliza la tecnología NFC (Near Field Communication), Java es el lenguaje utilizado para escribir el código que interactúa con el hardware NFC y procesa la información intercambiada a través de él.

Java cuenta con un conjunto completo de APIs (Application Programming Interfaces) para el desarrollo de aplicaciones. Estas APIs permiten a los desarrolladores acceder a diferentes funciones del hardware NFC, como la lectura y escritura de etiquetas NFC, la comunicación con otros dispositivos NFC y la creación de servicios en segundo plano que responden a eventos NFC.

Además, el framework de desarrollo de Android proporciona a los desarrolladores una amplia variedad de herramientas y bibliotecas para crear aplicaciones para Android. El framework de Android está construido en Java y se basa en una arquitectura de componentes reutilizables. El framework de Android proporciona una serie de ventajas que hacen que el desarrollo de aplicaciones sea más fácil y eficiente. Algunas de estas ventajas incluyen: la posibilidad de desarrollar aplicaciones en múltiples lenguajes de programación, la compatibilidad con múltiples dispositivos y resoluciones de pantalla, la disponibilidad de una amplia variedad de bibliotecas y

herramientas de desarrollo, la integración con tecnologías de la nube, la capacidad de desarrollar aplicaciones con capacidades avanzadas como NFC y realidad aumentada, y una comunidad de desarrolladores activa y grande que proporciona soporte y recursos adicionales. Todo esto hace que el desarrollo de aplicaciones para Android sea más accesible, flexible y escalable.

- **Eclipse IDE:** Eclipse IDE es un entorno de desarrollo integrado (IDE) de código abierto, que se utiliza principalmente para la creación de aplicaciones Java. Eclipse ofrece una amplia gama de características, herramientas y plugins para el desarrollo de software en diferentes lenguajes de programación.

El IDE está compuesto por una plataforma base, que proporciona los componentes fundamentales para el desarrollo de software, y una serie de plugins que se pueden instalar para ampliar la funcionalidad del IDE. La plataforma base está construida sobre el framework de aplicaciones OSGi (Open Services Gateway initiative), que permite la gestión dinámica de módulos y la interoperabilidad entre ellos.

Entre las características clave de Eclipse IDE se encuentran:

- Editor de código avanzado: cuenta con un editor de código completo que incluye resaltado de sintaxis, autocompletado de código, refactorización y navegación de código.
 - Integración con herramientas de construcción: se integra con diferentes herramientas de construcción, como Ant y Maven, lo que permite la automatización de la construcción, el empaquetado y la distribución del software.
 - Depuración y pruebas de software: proporciona herramientas para depurar y probar software, incluyendo un depurador integrado, una vista de variables y una consola de salida.
 - Integración con sistemas de control de versiones: Eclipse IDE se integra con sistemas de control de versiones, como Git y SVN, lo que permite la gestión de código fuente y el control de versiones del software.
- **MySQL:** es un sistema de gestión de bases de datos relacional de código abierto. Fue creado en 1995 y es uno de los sistemas de gestión de bases de datos más populares en la actualidad, utilizado por muchas organizaciones en todo el mundo. Se basa en el lenguaje de consulta estructurado (SQL) y se puede utilizar para almacenar, organizar y acceder a grandes cantidades de información. Ofrece una amplia gama de funciones y características, como la capacidad de manejar múltiples conexiones de cliente, la compatibilidad con varios sistemas operativos y lenguajes de programación, la seguridad y

el control de acceso, y la replicación y la escalabilidad para manejar grandes cantidades de datos y usuarios simultáneos.

MySQL también es conocido por su estabilidad, rendimiento y facilidad de uso, lo que lo hace ideal para aplicaciones de gran escala, desde pequeñas aplicaciones web hasta grandes sistemas empresariales.

PRINCIPIOS TECNOLÓGICOS

La aplicación utiliza la tecnología NFC (Near Field Communication) de los teléfonos móviles para ello. El administrador creará un nuevo grupo y añadirá a todas las personas que deban asistir. Además, configurará una etiqueta NFC y la colocará en el lugar de asistencia. Cuando la persona acuda, acercará su teléfono a la etiqueta colocada con la aplicación abierta y la tecnología NFC activada. Cuando la aplicación reconozca esta etiqueta, enviará la confirmación de asistencia a la base de datos.

NORMAS TÉCNICAS APLICABLES

- Normas ISO: la Organización Internacional de Normalización (ISO) tiene varias normas técnicas relevantes para el desarrollo de aplicaciones móviles, como la norma ISO 9241 para la usabilidad y la experiencia del usuario o la norma ISO 14443, que establece las especificaciones para las tarjetas de proximidad, incluyendo las tarjetas NFC.
- Normas de seguridad de la información: existen varias normas que se centran en la seguridad de la información, como la norma ISO/IEC 27001, que establece los requisitos para un sistema de gestión de seguridad de la información o la norma ISO/IEC 18092, que establece los requisitos de seguridad para la comunicación NFC.
- Normas de privacidad de datos: para garantizar la protección de la privacidad de los usuarios, pueden aplicarse normas como el Reglamento General de Protección de Datos de la Unión Europea (GDPR) o la Ley de Protección de Datos Personales en los Estados Unidos.
- Normas de accesibilidad: las normas de accesibilidad, como las directrices de accesibilidad del contenido web (WCAG), también son importantes para garantizar que la aplicación sea accesible para todas las personas, incluidas aquellas con discapacidades.
- El NFC Forum ha desarrollado varias normas técnicas, como la norma NFC Data Exchange Format (NDEF), que establece un formato estándar para los datos intercambiados a través de NFC.

ANÁLISIS DE LA REALIDAD Y ENTORNO DE LA EMPRESA COLABORADORA

Identificar la necesidad de control de asistencia de los empleados o alumnos de la empresa colaboradora.

Evaluar la factibilidad de implementar la tecnología NFC en el lugar, es decir, si los dispositivos móviles utilizados por los empleados o alumnos tienen esta funcionalidad o, en caso contrario, habilitar una funcionalidad para que el administrador introduzca manualmente las asistencias.

Estudiar la infraestructura y la capacidad de la empresa para soportar la implementación de la aplicación y la tecnología NFC, incluyendo el acceso a Internet, los servidores, los sistemas de seguridad, entre otros aspectos relevantes.

JUSTIFICACIÓN FUNCIONAL

Durante los últimos meses y años me he estado percatando de lo lentas, inefficientes y vulnerables que eran las aplicaciones del control de la asistencia. En algunas el administrador tiene que perder en cada jornada o clase minutos, observando a cada persona y colocando manualmente la asistencia o no asistencia de cada uno de ellos. En otras aplicaciones, cada persona marca su asistencia, pero realmente nada que verifique que esa asistencia en fecha y hora marcadas sea correcta. Por ello, he visto una gran necesidad en utilizar la tecnología NFC, ya que corrige el problema de la lentitud de que el administrador deba colocar las asistencias una a una y también resuelve el problema del posible fraude, ya que la etiqueta a escanear se encontrará en el lugar a asistir.

Pensé también en utilizar la tecnología de códigos QR, sin embargo, sigue pudiendo haber fraude, ya que las personas pueden hacer fotografías al código y escanear este fuera del lugar.

JUSTIFICACIÓN TECNOLÓGICA

Para mi desarrollo de un sistema para el control de la asistencia voy a utilizar las siguientes tecnologías:

- **Java:** Se utiliza como lenguaje de programación para la aplicación completa. Es un lenguaje orientado a objetos muy popular para el desarrollo de aplicaciones empresariales y de software de escritorio. Es conocido por su

seguridad y confiabilidad, lo que lo hace una buena opción para aplicaciones críticas de negocio, como el control de la asistencia. Además, cuenta con una gran cantidad de bibliotecas y frameworks que facilitan el desarrollo de aplicaciones complejas. En el ciclo, es el lenguaje con el que hemos aprendido en el módulo de Programación en el primer curso, y en el segundo curso ha sido utilizado en los módulos de Acceso a datos, Programación de servicios y procesos y Programación multimedia y dispositivos móviles.

- **Android Studio:** al crear una aplicación para dispositivos android, el entorno de desarrollo Android Studio es una excelente elección para desarrollar un sistema de control de asistencia. Este entorno incorpora acceso a una amplia variedad de bibliotecas y frameworks que te permiten crear una aplicación móvil robusta y escalable. Además, Android Studio cuenta con herramientas de depuración y emulación para probar las aplicaciones en diferentes dispositivos y asegurarse de que estén funcionando correctamente. En el ciclo lo hemos estado utilizando en el módulo de Programación multimedia y dispositivos móviles para el desarrollo de aplicaciones Android.
- **Eclipse IDE:** para crear la parte del servidor, Eclipse IDE es una herramienta de desarrollo de software de código abierto y gratuita que ofrece una amplia gama de características y herramientas para escribir, depurar y desplegar aplicaciones en Java de manera eficiente. Además, su flexibilidad y capacidad de extensión mediante plugins, lo hacen una herramienta muy versátil y accesible para cualquier desarrollador. En particular, su conjunto de características diseñadas para el desarrollo en paralelo o concurrente, lo hacen una opción atractiva para proyectos de este tipo. En el ciclo lo hemos utilizado en los módulos de Programación, Acceso a datos y Programación de servicios y procesos.
- **MySQL:** para la parte de base de datos, se utiliza el sistema gestor de bases de datos MySQL es una base de datos de código abierto y uno de los SGBD más populares en el mundo. Es conocido por ser altamente escalable, confiable y seguro. Lo he escogido por todo lo anterior y porque voy a utilizar una base de datos relacional cliente-servidor. En el ciclo lo hemos utilizado en el módulo de Acceso a datos.

MARCO LEGAL

La aplicación debe cumplir con las normas de privacidad de datos y seguridad de la información, como la Ley de Protección de Datos Personales (LOPD) y la normativa GDPR de la UE. Además, se han implementado medidas de seguridad físicas y tecnológicas para proteger los datos de los empleados y evitar posibles brechas de seguridad.

DESTINATARIOS

Los destinatarios son todas aquellas empresas, instituciones educativas y grupos que necesiten controlar la asistencia de las personas. Estos son los que pagarán por la aplicación en sus departamentos. Dentro de ellos encontramos como destinatarios a los administradores de cada grupo, quienes podrán gestionarlos y consultar las asistencias, días de vacaciones, festivos, etc; y por otro lado, los asistentes, quienes escanearán las etiquetas NFC para marcar su asistencia.

4. DESARROLLO DEL PROYECTO

4.1. ACUERDO DE PROYECTO

REQUISITOS FUNCIONALES

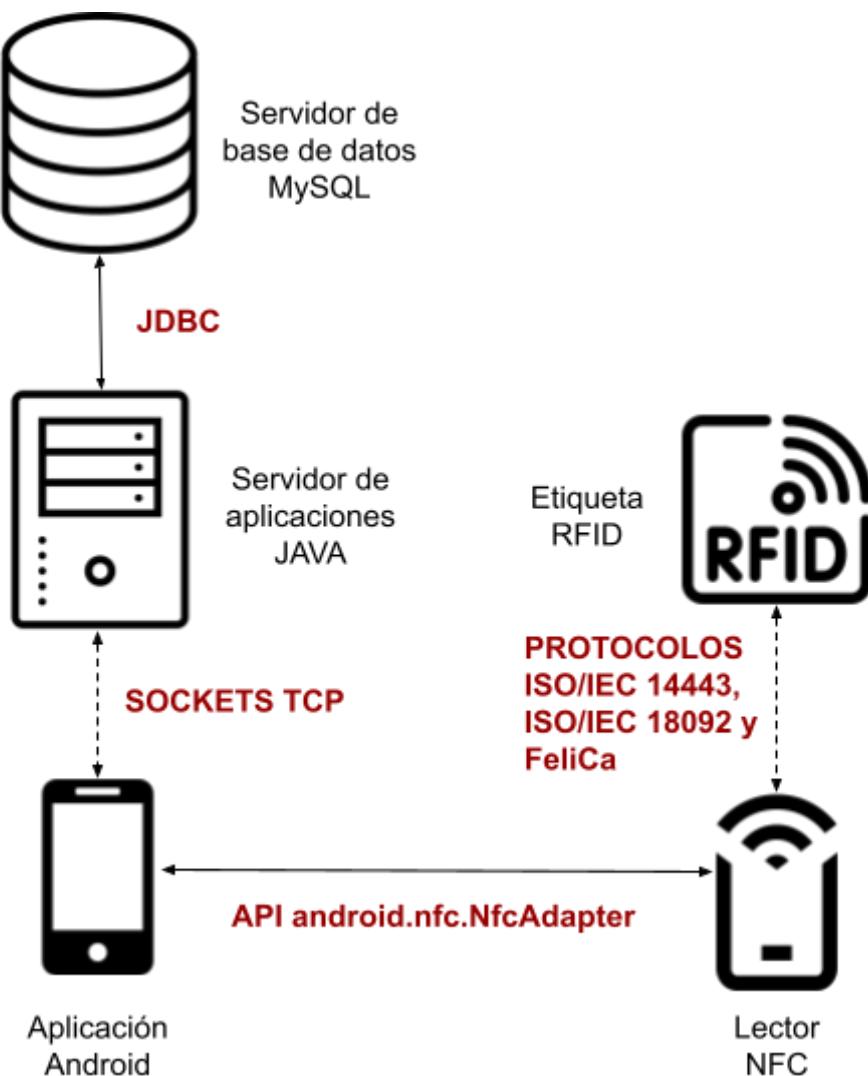
Requisito Funcional	Descripción
Registro e inicio de sesión de usuarios	Debe permitir el registro de usuarios y su posterior inicio de sesión. Los usuarios deben proporcionar su información personal como nombre, correo electrónico, número de teléfono y contraseña.
Inicio de sesión automático	Cuando se inicie sesión se deben guardar los datos de inicio de sesión para que cuando la próxima vez que se abra la aplicación, se inicie la sesión automáticamente.
Tipos de usuarios	Debe tener un sistema de autenticación en el que solo los usuarios autorizados puedan acceder a la información y funciones correspondientes. Los administradores tendrán acceso a más funcionalidades en la aplicación que los usuarios asistentes.
Registro de asistencia	Debe permitir a los asistentes registrar su entrada al lugar mediante la tecnología NFC de teléfono móvil. Los administradores también pueden registrar el ingreso manualmente en caso de que los usuarios no puedan utilizar la tecnología NFC.
Visualización de registro de asistencia	Debe permitir a los usuarios administradores ver un registro histórico de la asistencia de cada usuario.
Ver de notificaciones de asistencia	Debe permitir a los administradores consultar en forma de notificaciones cuando un usuario registre su asistencia.
Gestión de grupos	Debe permitir a los administradores crear y eliminar grupos, además de editar sus nombres y descripciones.
Gestión de etiquetas RFID	Los administradores deben poder vincular una etiqueta a un lugar correspondiente a un grupo, para que los asistentes la escaneen. También desvincularla después.

Gestión de usuarios	Debe permitir a los administradores gestionar los usuarios de los grupos, como crear, editar o eliminar cuentas de usuario.
Gestión de calendario	Debe permitir a los administradores gestionar los calendarios de los grupos, como añadir festivos. También los administradores tendrán podrán añadir los períodos de vacaciones de los empleados.

REQUISITOS NO FUNCIONALES

Requisito no funcional	Descripción
Seguridad	Debe contar con medidas de seguridad para evitar el acceso no autorizado a la información de los usuarios y para proteger la privacidad de los datos personales de los usuarios. Para ello se implementa un inicio de sesión con correo electrónico y contraseña.
Escalabilidad	Debe ser escalable para poder soportar una gran cantidad de usuarios y registros de asistencia, sin comprometer el rendimiento y la calidad de la aplicación.
Usabilidad	Debe ser fácil de usar para los asistentes y administradores, con una interfaz intuitiva y clara que les permita realizar sus tareas de manera eficiente y sin confusiones.
Disponibilidad	Debe estar disponible en todo momento para que los asistentes puedan registrar su asistencia y los administradores puedan acceder al registro. Esto implica que el servidor debe estar siempre en funcionamiento.
Rendimiento	Debe tener un rendimiento rápido y eficiente para que los asistentes puedan registrar su asistencia sin retrasos y los administradores puedan acceder al registro de manera rápida.
Comunicación cliente-servidor mediante sockets	La aplicación móvil y el servidor deben estar conectados e intercambiarán información gracias al uso de sockets. Estos utilizarán el protocolo TCP para tener una conexión fiable.

ESPECIFICACIONES TÉCNICAS



4.2. ACTIVIDADES Y CRONOLOGÍA

4.2.1. DIAGRAMA DE GANTT

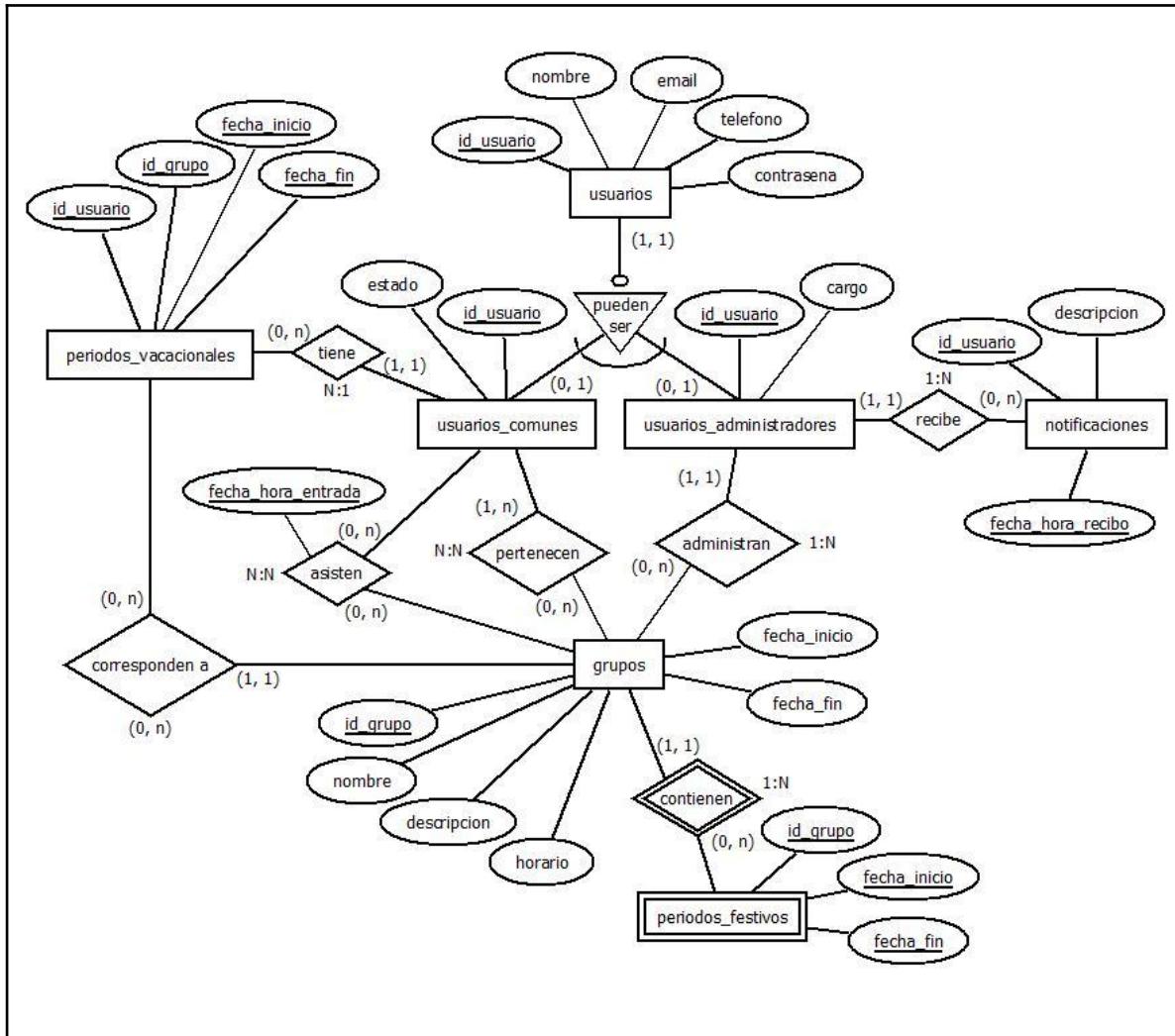
Para el siguiente diagrama de gantt, se han estimado unas 120 horas de trabajo repartidas en 40 días. Por lo tanto, debe tenerse en cuenta un trabajo de 3 horas diarias.

FASE	DETALLES	ABRIL															
		SEMANA DE PROYECTO		SEMANA 1			SEMANA 2			SEMANA 3			SEMANA 4				
		DÍA DEL MES	3	4	5	6	10	11	12	13	17	18	19	20	24	25	26
1	Investigación	- Planear cual es la mejor arquitectura posible para el desarrollo del sistema - Investigación sobre funcionamiento e implementación de la tecnología NFC - Investigación sobre estilos nuevos de vistas y layouts - Investigación sobre el funcionamiento de la vista de calendario	3h				12h										
2	Memoria del proyecto	- Portada - Datos descriptivos - Acuerdo del proyecto - Diagrama de Gantt - Diagramas de base de datos - Diagramas de casos de uso - Descripciones de casos de uso - Diagramas de clases - Diagramas de secuencia - Pruebas del sistema - Manual de instalación - Manual de usuario - Metodología seguida - Recursos - Presupuesto - Resultados y conclusiones - Bibliografía y anexos - Abstract					0.5h	4.5h		1.5h	3h	4h	1.5h	3h	7.5h	1.5h	
3	Desarrollo del entorno cliente	- Clases pojo - Diseño de interfaces gráficas - Programación de actividades - Funciones de sockets - Exportación												1.5h		11h	
4	Desarrollo del entorno servidor	- DAOs - Controladores - Funciones de sockets - Exportación															
5	Fin del proyecto	- Presentación - Cambios y corrección de errores															

En el [anexo](#), se ha incluido un apartado en el que se comenta la relación de horas invertidas respecto a la estimación realizada en el diagrama de Gantt.

4.2.2. DOCUMENTO DE ANÁLISIS Y DISEÑO

MODELO ENTIDAD-RELACIÓN



MODELO RELACIONAL

USUARIOS	#id_usuario: entero entero	nombre: cadena(50) cadena(50)	email: cadena(100) cadena(100)	telefono: cadena(15) cadena(15)	contrasena: cadena(100) cadena(100)
		Valor no nulo	Valor no nulo		Valor no nulo

USUARIOS_COMUNES	#id_usuario: entero USUARIOS	estado: booleano Valor no nulo
-------------------------	--	--

USUARIOS_ADMINISTRADORES	#id_usuario: entero USUARIOS	cargo: cadena(100)
---------------------------------	--	---------------------------

PERIODOS_VACACIONALES	#id_usuario: entero USUARIOS_COMUNES	#id_grupo: entero GRUPOS
	#fecha_inicio: fecha Valor no nulo	#fecha_fin: fecha

NOTIFICACIONES	#id_usuario: entero USUARIOS_ADMINISTRADORES	fecha_hora_recibo: timestamp	descripcion: cadena(500)
-----------------------	--	-------------------------------------	---------------------------------

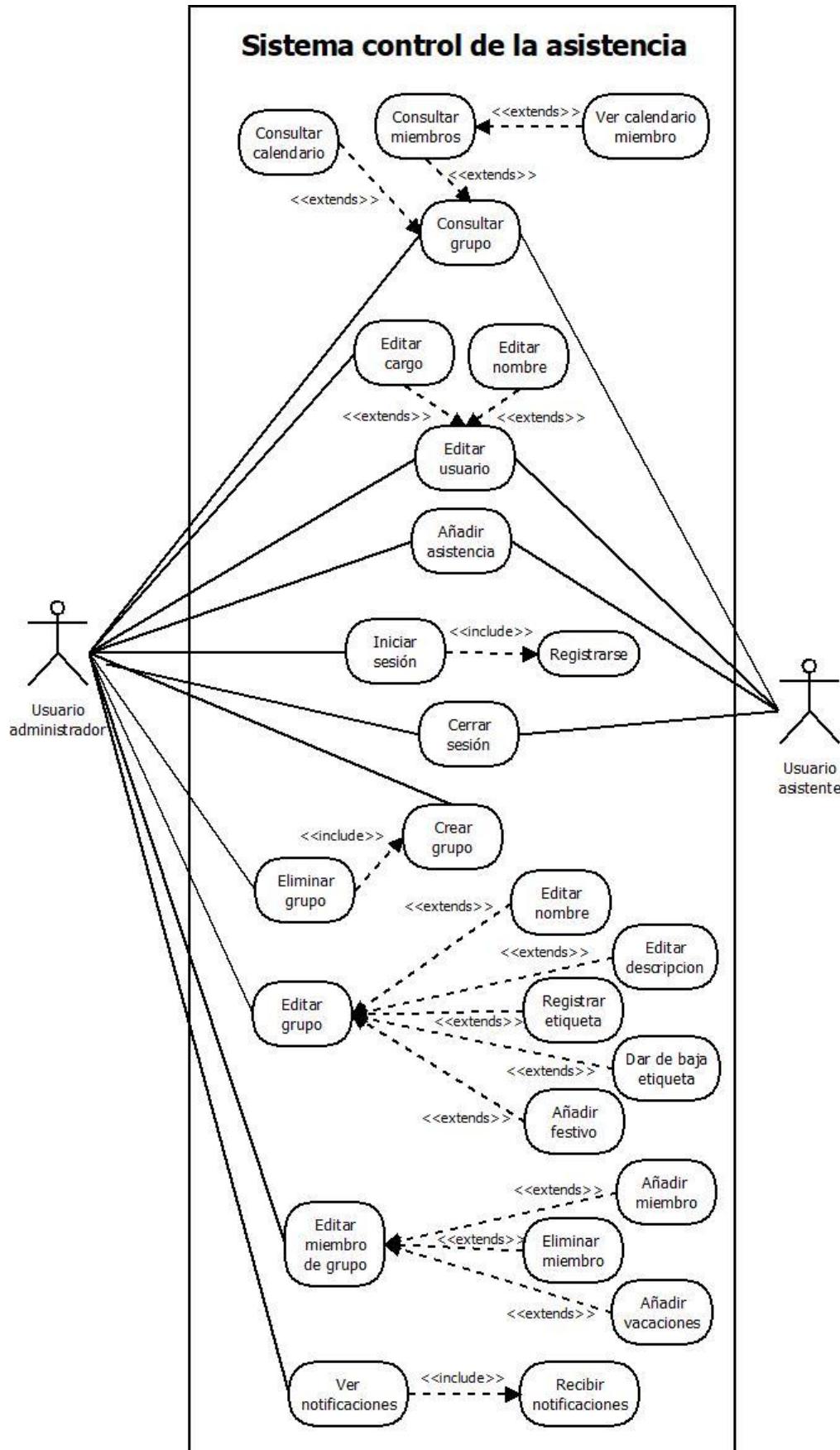
GRUPOS	#id_grupo: entero	nombre: cadena(100) cadena(100)	descripcion: cadena(15)
	horario: cadena(100)	fecha_inicio: fecha Valor no nulo	fecha_fin: fecha
	id_usuario_administrador: entero USUARIOS_ADMINISTRADORES		

USUARIOS_COMUNES_GRUPOS	#id_grupo: entero GRUPOS	#id_usuario: entero USUARIOS_COMUNES
--------------------------------	------------------------------------	--

ASISTENCIAS	#id_usuario: entero USUARIOS_COMUNES	#id_grupo: entero GRUPOS
	#fecha_hora_entrada: timestamp	

PERIODOS_FESTIVOS	#id_grupo: entero GRUPOS	fecha_inicio: fecha	fecha_fin: fecha
--------------------------	------------------------------------	----------------------------	-------------------------

CASOS DE USO



DESCRIPCIONES DE CASOS DE USO

Caso de uso: Registrarse
Actores: Usuario asistente, usuario administrador
Descripción: Un usuario nuevo puede registrarse en la aplicación para obtener acceso a todas las funcionalidades.
Precondiciones: El usuario no tiene una cuenta en la aplicación.
Curso normal del caso de uso: <ol style="list-style-type: none">1. El usuario abre la aplicación y selecciona la opción "Registrarse".2. El sistema muestra un formulario de registro que solicita al usuario su información personal, como usuario, dirección de correo electrónico, teléfono y contraseña.3. El usuario ingresa su información personal en el formulario y selecciona "Registrarse".4. El sistema valida la información ingresada por el usuario y crea una nueva cuenta de usuario en la aplicación.5. El sistema redirige al usuario a la página principal de inicio de sesión, donde colocará sus datos y podrá comenzar a utilizar todas las funcionalidades.
Alternativa: <ol style="list-style-type: none">4. Si el usuario ya ha registrado una cuenta con la dirección de correo electrónico o teléfono proporcionados, el sistema muestra un mensaje de error y solicita al usuario que ingrese unos datos diferentes.
Postcondiciones: El usuario tiene una cuenta registrada en la aplicación y puede iniciar sesión con su nueva cuenta.

Caso de uso: Iniciar sesión

Actores: Usuario asistente, usuario administrador

Descripción:

Un usuario registrado puede iniciar sesión en la aplicación para acceder a sus funcionalidades.

Precondiciones:

El usuario debe tener una cuenta registrada en la aplicación.

Curso normal del caso de uso:

1. El usuario abre la aplicación y selecciona la opción "Iniciar sesión".
2. El sistema muestra un formulario de inicio de sesión que solicita al usuario el correo electrónico o teléfono y la contraseña
3. El usuario ingresa los datos en el formulario y selecciona "Iniciar sesión".
4. El sistema valida la información ingresada por el usuario y verifica si la cuenta existe en la base de datos.
5. Si la cuenta existe y la información es correcta, el sistema redirige al usuario a la página principal de la aplicación, donde puede acceder a todas las funcionalidades.

Alternativa:

5. Si la cuenta no existe o la información ingresada es incorrecta, el sistema muestra un mensaje de error y solicita al usuario que verifique su información y la vuelva a ingresar.

Postcondiciones:

El usuario ha iniciado sesión correctamente y tiene acceso a todas las funcionalidades de la aplicación de control de asistencia.

Caso de uso: Cerrar sesión
Actores: Usuario asistente, usuario administrador
Descripción: Un usuario registrado puede cerrar sesión en la aplicación para salir de la sesión.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación.
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario abre la aplicación y selecciona la opción "Cerrar sesión". 2. El sistema elimina la información de sesión del usuario y lo redirige a la pantalla de bienvenida.
Alternativa: <ol style="list-style-type: none"> 2. Si la sesión no existe por algún error, no se eliminará y directamente redirige al usuario a la pantalla de bienvenida.
Postcondiciones: El usuario ha cerrado sesión correctamente y la información de sesión ha sido eliminada de la aplicación.

Caso de uso: Consultar grupo
Actores: Usuario asistente, usuario administrador
Descripción: Un usuario registrado puede consultar los grupos a los que pertenece.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y pertenecer a uno o varios grupos.
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario abre la aplicación y selecciona el grupo que deseé ver. 2. El sistema muestra los detalles del grupo tales como nombre, descripción, horario, usuarios y calendario.
Alternativa: <ol style="list-style-type: none"> 2. Si el usuario ha dejado de pertenecer al grupo, no lo abrirá y no lo mostrará.
Postcondiciones: El usuario ha iniciado sesión correctamente y tiene acceso a todas las funcionalidades de la aplicación de control de asistencia.

Caso de uso: Consultar calendario

Actores: Usuario asistente, usuario administrador

Descripción:

Un usuario registrado puede consultar el calendario de un grupo. El calendario debe mostrar los días de asistencia, los días festivos y días de descanso programados para el grupo.

Precondiciones:

El usuario debe haber iniciado sesión en la aplicación y pertenecer a un grupo.

Curso normal del caso de uso:

1. El usuario abre la aplicación y selecciona el grupo que desee ver.
2. El sistema muestra los detalles del grupo, entre los cuales están el calendario.
3. El usuario selecciona ver el calendario del grupo.
4. Se muestra el calendario con los días de asistencia, los días festivos, días de descanso y días de vacaciones.

Alternativa:

4. Si el calendario del grupo no tiene información, el sistema muestra el calendario vacío, sin información.

Postcondiciones:

El usuario ha consultado el calendario del grupo y puede ver los días festivos y días de descanso programados para el grupo.

Caso de uso: Consultar miembros
Actores: Usuario asistente, usuario administrador
Descripción: Un usuario registrado puede consultar todos los miembros de un grupo. El listado muestra, en primer lugar, el administrador del grupo y debajo el resto de usuarios del grupo.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y pertenecer a un grupo.
Curso normal del caso de uso: <ol style="list-style-type: none">1. El usuario abre la aplicación y selecciona el grupo que desee ver.2. El sistema muestra los detalles del grupo, entre los cuales están los miembros.3. El usuario selecciona ver la pestaña de miembros del grupo.4. Se muestra el listado de miembros del grupo, al que es posible pulsar en uno y mostrar su información.
Alternativa: <ol style="list-style-type: none">4. Si el listado de miembros del grupo está vacío, el sistema muestra un listado vacío, sin información.
Postcondiciones: El usuario ha consultado el listado de miembros y puede ver toda la información sobre ellos.

Caso de uso: Ver calendario miembro
Actores: Usuario asistente, usuario administrador
Descripción: Un usuario registrado puede consultar un calendario donde se muestran las asistencias, las asistencias con retraso, las no asistencias, las vacaciones del usuario y los festivos del grupo.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación, pertenecer a un grupo y tener permisos para consultar las asistencias del miembro en cuestión.
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario abre la aplicación y selecciona el grupo que deseé ver. 2. El sistema muestra los detalles del grupo, entre los cuales están los miembros del grupo. 3. El usuario mantiene presionado el miembro a consultar y selecciona ver calendario 4. Se muestra el calendario con los datos mencionados.
Alternativa: <ol style="list-style-type: none"> 4. Si el miembro no tiene asistencias ni vacaciones, tan sólo se mostrarán los festivos del grupo.
Postcondiciones: El usuario ha consultado las asistencias de un miembro en un grupo y puede ver información detallada sobre sus registros de asistencia.

Caso de uso: Editar usuario
Actores: Usuario asistente, usuario administrador
Descripción: Un usuario registrado puede modificar sus datos
Precondiciones: El usuario debe haber iniciado sesión en la aplicación.
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario abre la aplicación. 2. El usuario modifica el dato que deseé
Alternativa: <ol style="list-style-type: none"> 2. Si el dato modificado no cumple con los requisitos de formato, la aplicación muestra un mensaje de error y solicita que se introduzca otro válido.
Postcondiciones: Los datos del usuario son actualizados y se reflejan en todas las funcionalidades y grupos de la aplicación.

Caso de uso: Editar nombre usuario
Actores: Usuario asistente, usuario administrador
Descripción: Un usuario registrado puede modificar su nombre.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación.
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario abre la aplicación. 2. El usuario modifica su nombre y confirma los cambios. 3. La aplicación actualiza el nombre del usuario en la aplicación.
Alternativa: <ol style="list-style-type: none"> 3. Si el nombre introducido por el usuario no cumple con los requisitos de formato o ya existe, la aplicación muestra un mensaje de error y solicita que se introduzca un nombre válido.
Postcondiciones: El nombre del usuario es actualizado y se refleja en todas las funcionalidades y grupos de la aplicación.

Caso de uso: Editar cargo
Actores: Usuario administrador
Descripción: Un usuario registrado puede modificar su cargo.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario abre la aplicación. 2. El usuario modifica su cargo como administrador y confirma los cambios. 3. La aplicación actualiza el cargo en la aplicación.
Alternativa: <ol style="list-style-type: none"> 3. Si el cargo introducido por el usuario no cumple con los requisitos de formato, la aplicación muestra un mensaje de error y solicita que se introduzca un cargo válido.
Postcondiciones: El cargo es actualizado y se refleja en todas las funcionalidades y grupos de la aplicación.

Caso de uso: Añadir asistencia

Actores: Usuario asistente, usuario administrador

Descripción:

Un usuario registrado puede registrar su asistencia en un día específico en la aplicación de control de asistencia.

Precondiciones:

El usuario debe haber iniciado sesión en la aplicación y pertenecer a un grupo.

Curso normal del caso de uso:

1. El usuario abre la aplicación y selecciona el grupo que desee registrar su asistencia.
2. El sistema muestra la información general del grupo, como la descripción y los miembros.
3. El usuario selecciona la opción "Registrar asistencia".
4. Se mostrará un texto diciendo que, con la tecnología NFC activa, se acerque el dispositivo a la etiqueta NFC. En caso de que la asistencia la registre manualmente el administrador, aparecerá un formulario donde seleccionará el usuario y hora de asistencia.
5. El sistema registra la asistencia del usuario correctamente

Alternativa:

5. Si hay un fallo en la lectura de la etiqueta NFC, se pedirá al usuario que vuelva a intentarlo.

Postcondiciones:

El usuario ha registrado su asistencia en un día específico en la aplicación de control de asistencia, y la información se ha guardado.

Caso de uso: Crear grupo
Actores: Usuario administrador
Descripción: Un usuario registrado puede crear un nuevo grupo en la aplicación.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario abre la aplicación y selecciona nuevo grupo. 2. El sistema muestra un formulario de creación de grupo que incluye campos para el nombre y descripción . 3. El sistema añade el nuevo grupo.
Alternativa: 3. Si algún dato como el nombre o la descripción no son válidos, aparecerá un mensaje de error.
Postcondiciones: El usuario ha creado un nuevo grupo en la aplicación. El grupo aparece en la lista de grupos disponibles en la página principal de la aplicación.

Caso de uso: Eliminar grupo
Actores: Usuario administrador
Descripción: Un usuario registrado puede eliminar un grupo en la aplicación.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario abre la aplicación y mantiene pulsado el grupo a eliminar. 2. El sistema muestra una ventana de confirmación preguntando si el usuario está seguro de que desea eliminar el grupo. 3. El usuario selecciona la opción "Aceptar" para confirmar que desea eliminar el grupo. 4. El sistema muestra un mensaje de confirmación de que el grupo se ha eliminado correctamente.
Alternativa: 2. Si el usuario cancela con "X" en la ventana de confirmación, el sistema cancela la acción y regresa a la página principal.
Postcondiciones: El usuario ha eliminado un grupo existente. El grupo ya no aparece en la lista de grupos disponibles en la página principal de la aplicación.

Caso de uso: Editar grupo

Actores: Usuario administrador

Descripción:

Un usuario registrado puede editar la información de un grupo existente en la aplicación.

Precondiciones:

El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.

Curso normal del caso de uso:

1. El usuario abre la aplicación y selecciona el grupo a editar.
2. El sistema muestra la información general del grupo.
3. El usuario puede modificar los datos, miembros, festivos y etiquetas vinculadas al grupo.
4. El usuario modifica aquello que desea editar.
5. El sistema valida los cambios, actualiza la información del grupo y muestra un mensaje de confirmación.

Alternativa:

5. Si el usuario inserta algún dato erróneamente, el sistema envía una notificación de error y no se edita el grupo.

Postcondiciones:

El usuario ha editado la información de un grupo en la aplicación. La información actualizada se muestra.

Caso de uso: Editar nombre
Actores: Usuario administrador
Descripción: Un usuario registrado puede editar el nombre de un grupo existente en la aplicación.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: <ol style="list-style-type: none">6. El usuario abre la aplicación y selecciona el grupo a editar.7. El sistema muestra la información general del grupo.8. El usuario selecciona el ícono rojo de edición de nombre al lado del nombre del grupo.9. El usuario inserta el nuevo nombre del grupo.10. El sistema valida los cambios, actualiza el nombre del grupo y muestra un mensaje de confirmación.
Alternativa: 5. Si el usuario inserta el nombre erróneamente, el sistema envía una notificación de error y no se edita el grupo.
Postcondiciones: El usuario ha editado el nombre del grupo en la aplicación. La información actualizada se muestra.

Caso de uso: Editar descripción
Actores: Usuario administrador
Descripción: Un usuario registrado puede editar la descripción de un grupo existente en la aplicación.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: 11. El usuario abre la aplicación y selecciona el grupo a editar. 12. El sistema muestra la información general del grupo. 13. El usuario selecciona el ícono rojo de edición de nombre al lado de la descripción del grupo. 14. El usuario inserta el nuevo nombre del grupo. 15. El sistema valida los cambios, actualiza la descripción del grupo y muestra un mensaje de confirmación.
Alternativa: 5. Si el usuario inserta la descripción erróneamente, el sistema envía una notificación de error y no se edita el grupo.
Postcondiciones: El usuario ha editado la descripción del grupo en la aplicación. La información actualizada se muestra.

Caso de uso: Registrar nueva etiqueta
Actores: Usuario administrador
Descripción: Un usuario registra una nueva etiqueta NFC para ser asociada a un grupo en la aplicación. La etiqueta será utilizada posteriormente para identificar al usuario y registrar su asistencia en el grupo correspondiente.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador. Además, la etiqueta NFC debe estar disponible y lista para ser registrada en la aplicación.
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario abre la aplicación y selecciona el grupo a registrar. 2. El sistema muestra la información general del grupo. 3. El usuario selecciona "Vincular/desvincular etiqueta". 4. El usuario selecciona la pestaña "desvincular etiqueta". 5. El sistema pide que se active la tecnología NFC y que se acerque la etiqueta al lector. 6. El usuario acerca la etiqueta. 7. El sistema escribe los datos en la etiqueta y muestra una animación de confirmación.
Alternativa: <ol style="list-style-type: none"> 7. Si la etiqueta NFC está dañada, mostrará una animación de error.
Postcondiciones: La nueva etiqueta NFC queda asociada al grupo correspondiente, permitiendo el registro de la asistencia de los usuarios mediante el escaneo de la etiqueta.

Caso de uso: Dar de baja etiqueta
Actores: Usuario administrador
Descripción: Un usuario vacía la información de una etiqueta NFC asociada a un grupo en la aplicación.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: <ol style="list-style-type: none">1. El usuario abre la aplicación y selecciona un grupo cualquiera.2. El sistema muestra la información general del grupo.3. El usuario selecciona "Vincular/desvincular etiqueta".4. El usuario selecciona la pestaña "desvincular etiqueta".5. El sistema pide que se active la tecnología NFC y que se acerque la etiqueta al lector.6. El usuario acerca la etiqueta.7. El sistema vacía los datos de la etiqueta y muestra una animación de confirmación.
Alternativa: 7. Si la etiqueta NFC está dañada, mostrará una animación de error.
Postcondiciones: La etiqueta NFC ha sido vaciada de información y está lista para ser asociada a otro grupo.

Caso de uso: Añadir festivos
Actores: Usuario administrador
Descripción: Un usuario registrado puede añadir períodos festivos de un grupo en la aplicación.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: <ol style="list-style-type: none">1. El usuario abre la aplicación y selecciona el grupo a editar.2. El sistema muestra la información general del grupo.3. El usuario selecciona la pestaña de "Ver calendario".4. El usuario presiona el botón "Nuevo festivo".5. El sistema muestra una ventana para seleccionar el inicio y el fin del festivo.6. El usuario coloca las fechas y pulsa el botón "Aceptar".7. El sistema valida los cambios, actualiza los festivos del grupo y muestra un mensaje de confirmación.
Alternativa: 7. Si el usuario añade algún festivo con fechas que ya existieran, mostrará un mensaje de error.
Postcondiciones: El usuario ha editado la lista de vacaciones de los usuarios usuario de un grupo en la aplicación. La información actualizada se muestra.

Caso de uso: Editar miembro del grupo

Actores: Usuario administrador

Descripción:

Un usuario registrado puede editar la lista de miembros de un grupo en la aplicación.

Precondiciones:

El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.

Curso normal del caso de uso:

1. El usuario abre la aplicación y selecciona el grupo a editar.
2. El sistema muestra la información general del grupo.
3. El usuario selecciona la pestaña de "Ver miembros".
4. El usuario puede añadir o eliminar miembros, seleccionando las opciones llamadas de la misma manera.
5. El sistema valida los cambios, actualiza la información del grupo y muestra un mensaje de confirmación.

Alternativa:

5. Si el usuario modifica algún miembro que no existe, el sistema envía una notificación de error y no se añade.

Postcondiciones:

El usuario ha editado la lista de miembros de un grupo en la aplicación. La información actualizada se muestra.

Caso de uso: Añadir miembro
Actores: Usuario administrador
Descripción: Un usuario registrado puede añadir miembros a un grupo en la aplicación.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: <ol style="list-style-type: none">1. El usuario abre la aplicación y selecciona el grupo a editar.2. El sistema muestra la información general del grupo.3. El usuario selecciona la pestaña de "Ver miembros".4. El usuario presiona el botón "Añadir miembro".5. El usuario coloca el email del nuevo usuario a añadir.6. El sistema valida los cambios, actualiza la información del grupo y muestra un mensaje de confirmación.
Alternativa: 6. Si el usuario añade algún miembro que no existe, el sistema envía una notificación de error y no se añade.
Postcondiciones: El usuario ha añadido un miembro de un grupo en la aplicación. La información actualizada se muestra.

Caso de uso: Eliminar miembro
Actores: Usuario administrador
Descripción: Un usuario registrado puede eliminar miembros a un grupo en la aplicación.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: <ol style="list-style-type: none">1. El usuario abre la aplicación y selecciona el grupo a editar.2. El sistema muestra la información general del grupo.3. El usuario selecciona la pestaña de "Ver miembros".4. El usuario mantiene presionado al miembro a eliminar.5. El usuario presiona el botón "Eliminar miembro".6. El sistema valida los cambios, actualiza la información del grupo y muestra un mensaje de confirmación.
Alternativa: 6. Si el usuario selecciona al miembro administrador, el sistema envía una notificación de error y no se elimina.
Postcondiciones: El usuario ha eliminado un miembro de un grupo en la aplicación. La información actualizada se muestra.

Caso de uso: Añadir vacaciones
Actores: Usuario administrador
Descripción: Un usuario registrado puede añadir periodos de vacaciones de un usuario de un grupo en la aplicación.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y debe ser administrador.
Curso normal del caso de uso: <ol style="list-style-type: none">1. El usuario abre la aplicación y selecciona el grupo a editar.2. El sistema muestra la información general del grupo.3. El usuario selecciona la pestaña de "Ver miembros".4. El usuario mantiene presionado el miembro a añadir vacaciones.5. El sistema muestra una ventana y el usuario selecciona "Añadir vacaciones".6. El usuario coloca las fechas de inicio y fin de las vacaciones y pulsa el botón "Aceptar".7. El sistema valida los cambios, actualiza los festivos del grupo y muestra un mensaje de confirmación.
Alternativa: <ol style="list-style-type: none">7. Si el usuario añade algún periodo de vacaciones con fechas que ya existieran, mostrará un mensaje de error.
Postcondiciones: El usuario ha editado la lista de vacaciones de un grupo en la aplicación. La información actualizada se muestra.

Caso de uso: Ver notificaciones
Actores: Usuario administrador
Descripción: El usuario debe haber recibido un listado de las notificaciones que ha recibido.
Precondiciones: El usuario debe haber iniciado sesión en la aplicación y pertenecer a un grupo.
Curso normal del caso de uso: <ol style="list-style-type: none">1. El usuario abre la aplicación y si quiere puede seleccionar un grupo en caso de que quiera ver solo las notificaciones de ese grupo concreto.2. El usuario presiona el ícono de la campana.3. Se muestra el listado con las notificaciones recibidas.
Alternativa: 3. Si no existen notificaciones no se mostrarán.
Postcondiciones: El usuario recibe notificaciones de las actividades del grupo.

DIAGRAMA DE CLASES DEL SERVIDOR

Para facilitar la comprensión de las uniones entre clases, en los diagramas de clases todas las clases mapeadas (pojo) y las clases de las actividades de la aplicación Android, se han introducido en un paquete. Esto significa que cualquier unión con estos paquetes significa una unión del mismo tipo con todas las clases que componen el paquete. De esta manera se eliminan la mayoría de líneas y la comprensión es mucho más clara.

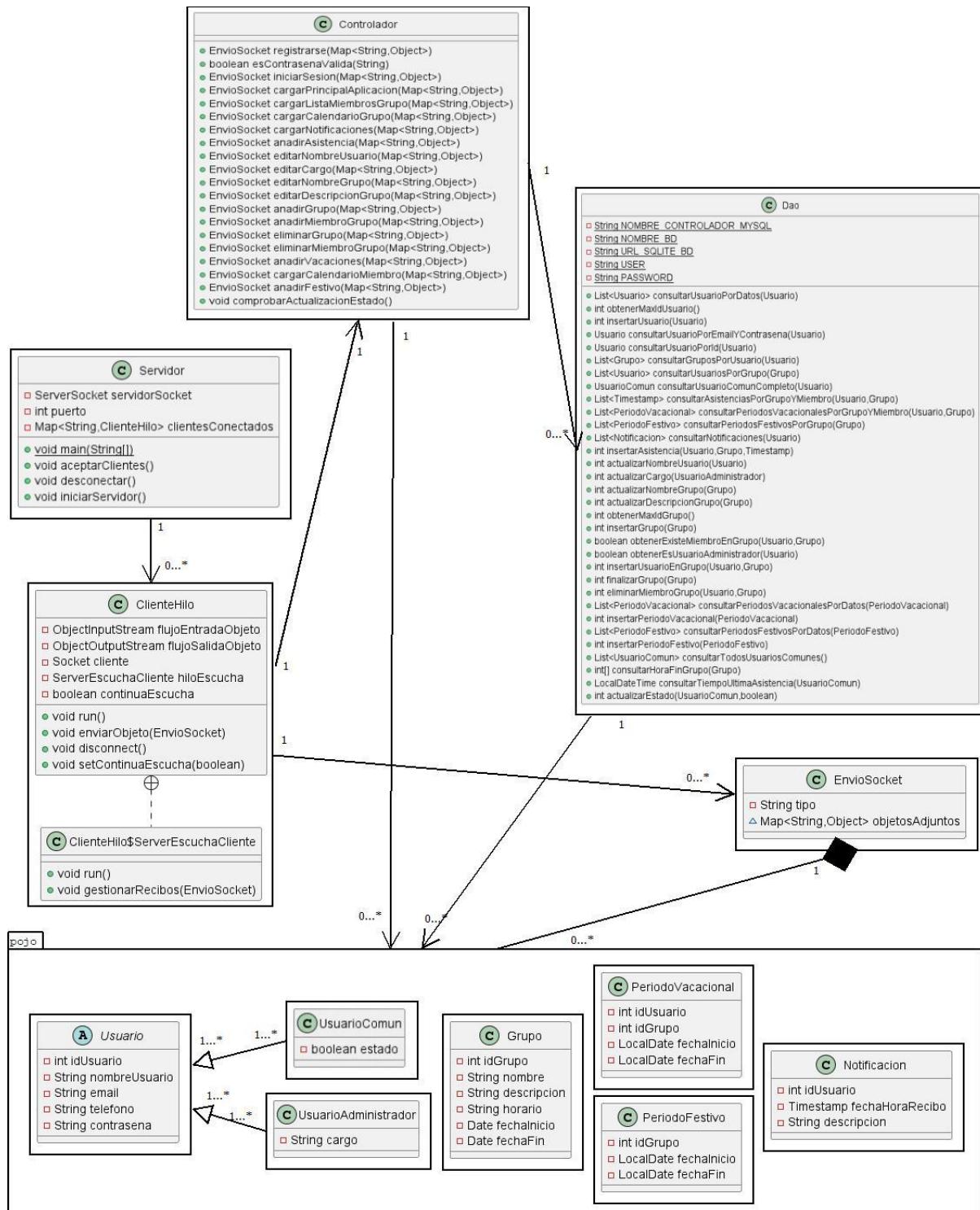
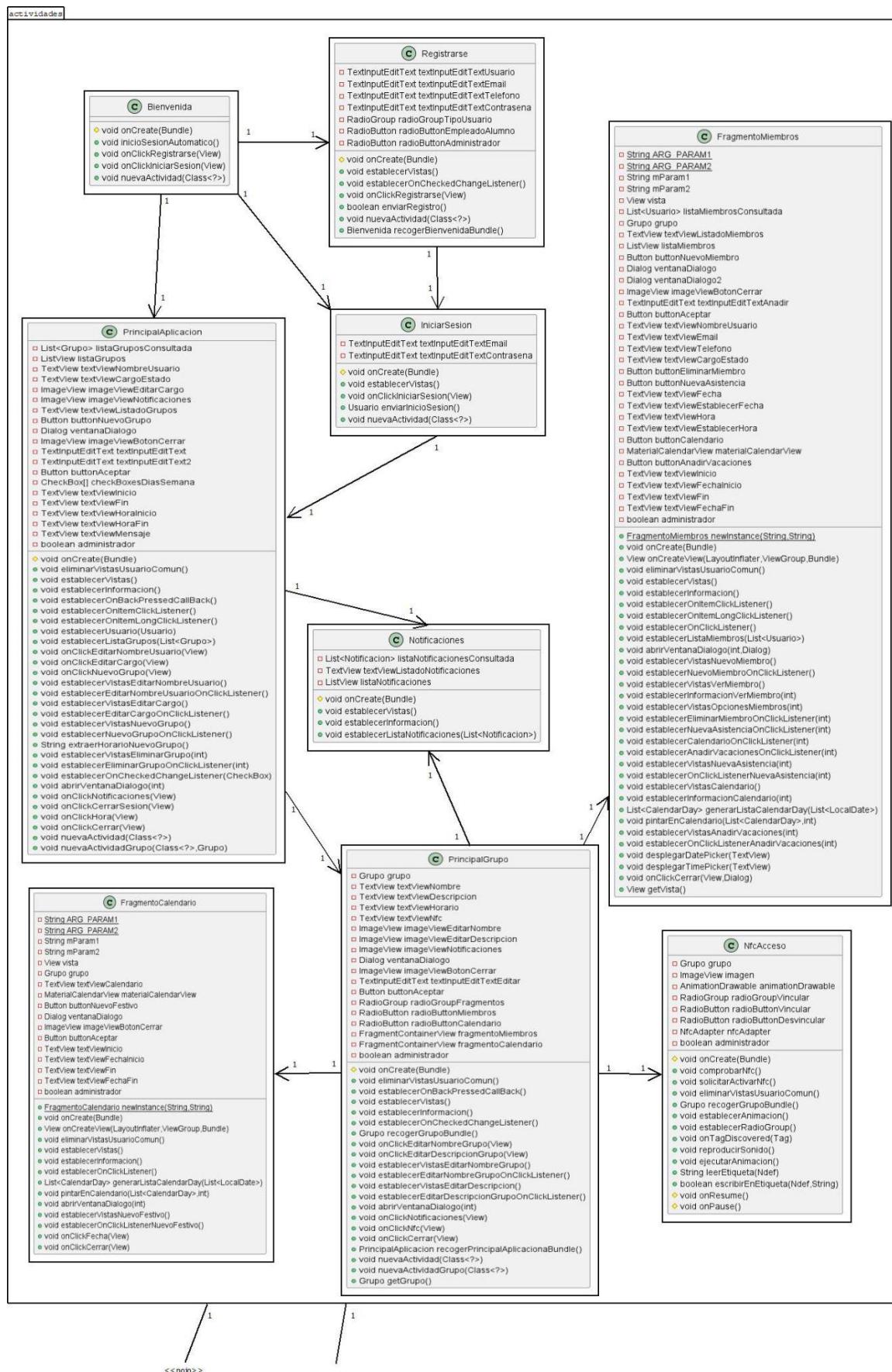


DIAGRAMA DE CLASES EN LA APLICACIÓN ANDROID



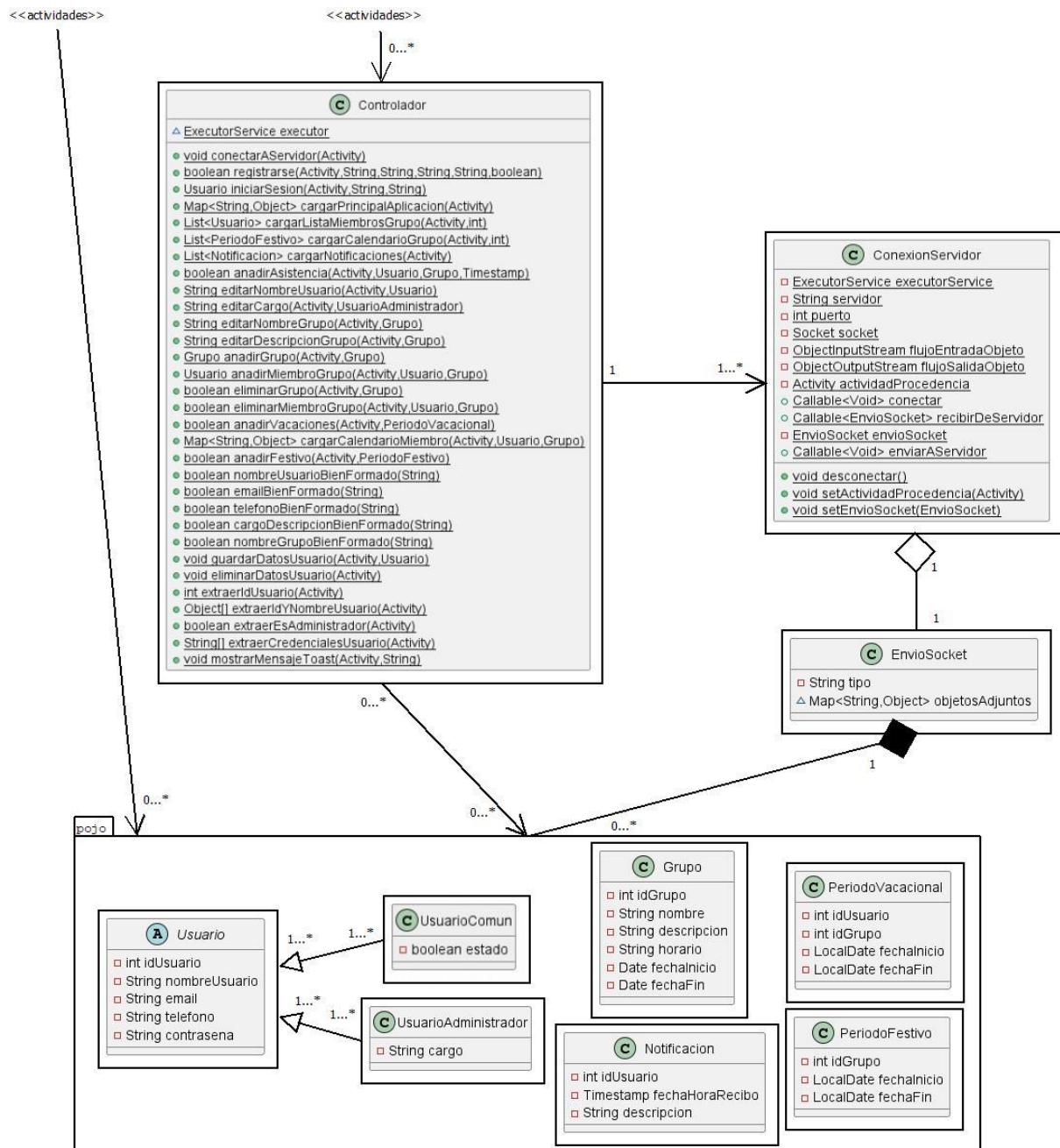
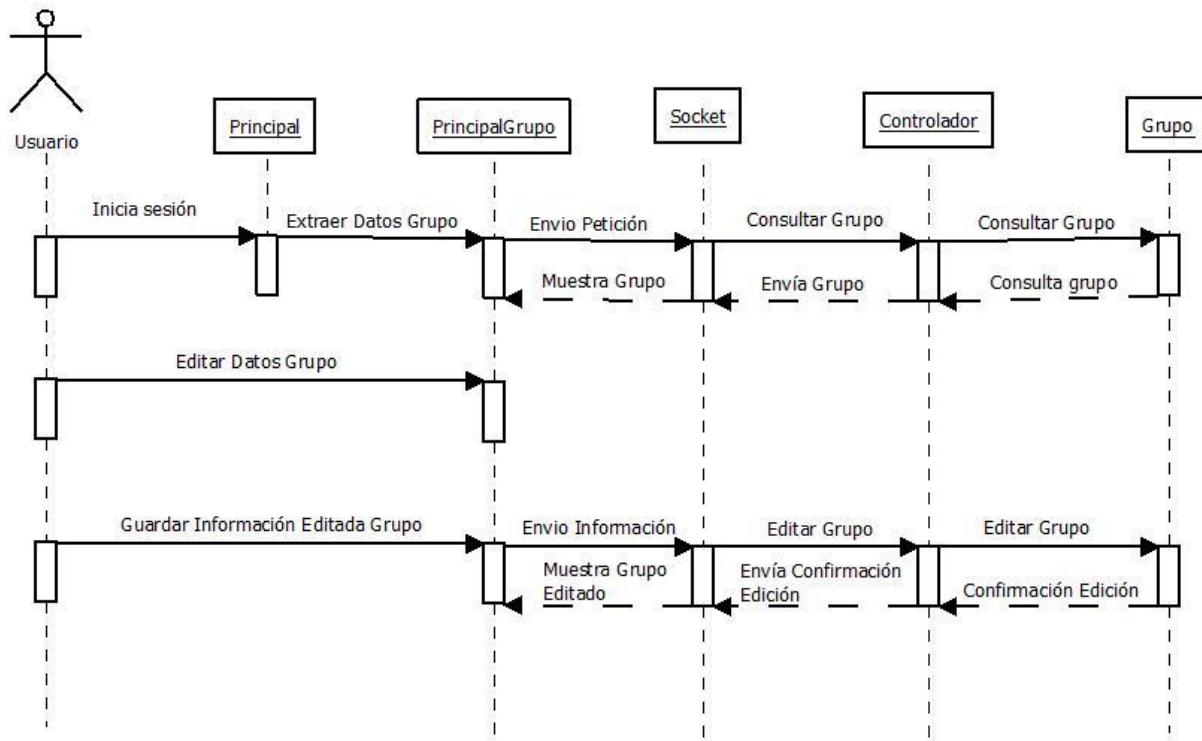


DIAGRAMA DE SECUENCIA EDICIÓN GRUPO



4.2.3. PRUEBAS DEL SISTEMA

Durante el desarrollo del sistema, existen diversas pruebas para garantizar su correcto funcionamiento. A continuación, se detallan algunas de las pruebas existentes a realizar:

1. **Prueba de funcionalidad del sistema:** Se trata de pruebas exhaustivas para verificar que todas las funciones del sistema funcionarán según lo previsto. Esto incluye la verificación de la comunicación entre el cliente y el servidor, la lectura precisa de las tarjetas NFC, el registro de la asistencia y la generación de informes.
2. **Prueba de rendimiento:** Se trata de pruebas para evaluar el rendimiento del sistema en diferentes situaciones de carga. Esto incluye simular un alto volumen de usuarios registrando su asistencia simultáneamente para garantizar que el sistema pudiera manejar eficientemente esta carga de trabajo sin afectar la respuesta y el tiempo de procesamiento.
3. **Prueba de integración:** Se trata de pruebas para comprobar la correcta integración del sistema con otros componentes o sistemas existentes en la organización. Esto incluye la verificación de la interoperabilidad del sistema de control de asistencia con sistemas de gestión de recursos humanos o bases de datos externas.
4. **Prueba de seguridad:** Se trata de pruebas exhaustivas para evaluar la seguridad del sistema. Esto incluye pruebas de penetración para identificar posibles vulnerabilidades, pruebas de autenticación y autorización para garantizar el acceso adecuado al sistema, y pruebas de encriptación de datos para proteger la confidencialidad de la información sensible.
5. **Prueba de usabilidad:** Se trata de pruebas con usuarios reales para evaluar la facilidad de uso y la experiencia del usuario en la aplicación web. Se recopilan comentarios y se realizan ajustes en la interfaz y la navegación para mejorar la usabilidad del sistema.

A continuación se muestra un ejemplo de prueba unitaria de algunos de los métodos de la aplicación. Se han creado métodos de prueba separados para cada uno de los tres métodos que deseas probar.

Para crear una prueba en Android Studio, podemos crear un archivo de prueba de unidad que ejecute la tarea test utilizando el complemento Gradle. Para cada método, se han definido varios casos de prueba que verifican diferentes escenarios. A cada uno se le debe colocar la anotación `@Test` para indicar que es una prueba. Usamos el método `assertTrue` para verificar que el resultado sea `true` cuando se cumpla la condición esperada, y usamos el método `assertFalse` para verificar que el resultado sea `false` cuando no se cumpla la condición esperada.

```
import static org.junit.Assert.*;

import com.shifttracker.shifttracker.conexion_servidor
    .GestionConexiones;

import org.junit.Test;

public class Prueba {
    @Test
    public void testNombreUsuarioBienFormado() {
        //Caso de prueba: nombre de usuario válido
        assertTrue(GestionConexiones
            .nombreUsuarioBienFormado("usuario123"));

        //Caso de prueba: nombre de usuario demasiado corto
        assertFalse(GestionConexiones
            .nombreUsuarioBienFormado("ab"));

        //Caso de prueba: nombre de usuario demasiado largo
        assertFalse(GestionConexiones
            .nombreUsuarioBienFormado
            ("esteNombreDeUsuarioEsMuyLargo"));
    }

    @Test
    public void testEmailBienFormado() {
        //Caso de prueba: email válido
        assertTrue(GestionConexiones
            .emailBienFormado("correo@example.com"));

        //Caso de prueba: email sin símbolo @
        assertFalse(GestionConexiones
            .emailBienFormado("correoejemplo.com"));

        //Caso de prueba: email sin dominio
        assertFalse(GestionConexiones
            .emailBienFormado("correo@"));

        //Caso de prueba: email sin parte local
        assertFalse(GestionConexiones
            .emailBienFormado("@example.com"));
    }

    @Test
    public void testTelefonoBienFormado() {
```

```
//Caso de prueba: número de teléfono válido  
assertTrue(GestionConexiones  
    .telefonoBienFormado("+34612345678"));  
  
//Caso de prueba: teléfono sin prefijo  
assertTrue(GestionConexiones  
    .telefonoBienFormado("678901234"));  
  
//Caso de prueba: teléfono con formato incorrecto  
assertFalse(GestionConexiones  
    .telefonoBienFormado("1234"));  
  
//Caso de prueba: teléfono con caracteres no numéricos  
assertFalse(GestionConexiones  
    .telefonoBienFormado("+34abcdefghijkl"));  
}  
}
```

Conclusiones:

Las pruebas realizadas en el sistema de control de asistencia demostraron su correcto funcionamiento y cumplimiento de los objetivos establecidos. Se identificaron y corrigieron posibles problemas, garantizando la calidad, seguridad y usabilidad del sistema..

4.2.4. DOCUMENTACIÓN PARA EL CLIENTE

MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

El sistema de seguimiento de asistencia ShiftTracker, consta de una base de datos, un programa de escritorio que actúa como servidor y la aplicación Android que actúa como clientes.

Antes de comenzar, hay que asegurarse de tener un sistema operativo compatible con Java y MySQL.

Primero, necesitamos instalar la base de datos. MySQL es el sistema gestor de bases de datos a utilizar. Después de instalar la base de datos, se debe crear una base de datos llamada "ShiftTracker" y se ejecutarán todas las sentencias incluidas en el archivo "ShiftTracker.sql", para crear todo lo necesario en la base de datos.

A continuación, se debe instalar el programa de escritorio en el servidor. Este programa hace de mediador entre la aplicación Android y la base de datos. Hay que asegurarse de tener el entorno de ejecución de Java instalado en el servidor. Después, se debe descargar el programa "ShiftTracker.jar" y ejecutarlo.

Finalmente, se debe instalar la aplicación Android en los dispositivos de los usuarios. La aplicación Android es la interfaz que los usuarios usarán para registrar su asistencia y los administradores para gestionar sus grupos. Para instalar la aplicación Android se debe descargar e instalar el archivo "ShiftTracker.apk".

Debido a que se trata de un sistema cliente-servidor, tanto la base de datos como la aplicación de escritorio que hace de servidor necesitan estar constantemente siendo ejecutados para que puedan comunicarse entre ellos y con los usuarios en la aplicación Android.

Con esto, se estará listo para poner en marcha el sistema de seguimiento de asistencia ShiftTracker.

MANUAL DE USUARIO

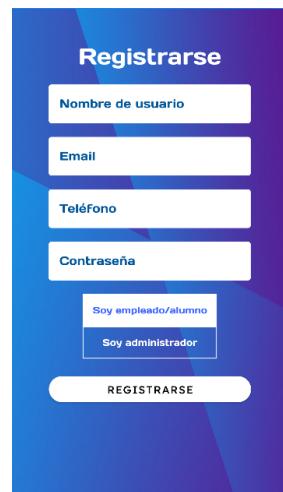
- **Inicio**

Tras instalar la aplicación, veremos la pantalla de bienvenida, en la que podremos seleccionar registrarnos o iniciar sesión si ya tenemos una cuenta.



- **Registro**

Para registrarse habrá que insertar el nombre de usuario email, teléfono y una contraseña, aparte de seleccionar si la cuenta va a ser de un usuario empleado/alumno o administrador.



- **Inicio de sesión**

En el momento en que ya estemos registrados insertaremos nuestro email y la contraseña para iniciar la sesión.

Cabe mencionar que cuando se inicia sesión, se guardan los datos y las siguientes veces que se abre la aplicación el inicio de sesión es automático.



- **Pantalla principal**

En la pantalla principal podremos ver nuestro nombre de usuario y nuestro cargo (administradores) o nuestro estado actual (no administradores) y poder editarlos, si procede. Podremos ver los grupos en los que nos encontramos y al pulsarlos entrar en ellos.

Además, podremos cerrar sesión y ver las notificaciones si somos administradores.



- **Creación y eliminación de grupos**

Siendo administradores, podemos crear un nuevo grupo dándole al botón "Nuevo grupo" en la pantalla principal y podemos eliminarlos manteniendo pulsado el grupo que deseemos eliminar.



- **Pantalla principal de grupo**

En la pantalla principal de cada grupo podremos ver el nombre, descripción y horario del mismo y editarlos (administrador). Podremos seleccionar ver la pestaña de miembros del grupo o la del calendario. En la primera se muestran todos los miembros que están dentro del grupo y en la segunda un calendario con los días festivos.



Aquí también podremos ver las notificaciones si somos administradores.

- **Añadir y eliminar miembros**

Siendo administradores podemos añadir un nuevo miembro al grupo dándole al botón nuevo miembro en la pantalla principal del grupo y podemos eliminarlo manteniendo pulsado el miembro que deseemos eliminar.



- **Añadir festivos**

Siendo administradores podemos añadir un nuevo festivo de grupo dándole al botón nuevo festivo en la pantalla principal del grupo.



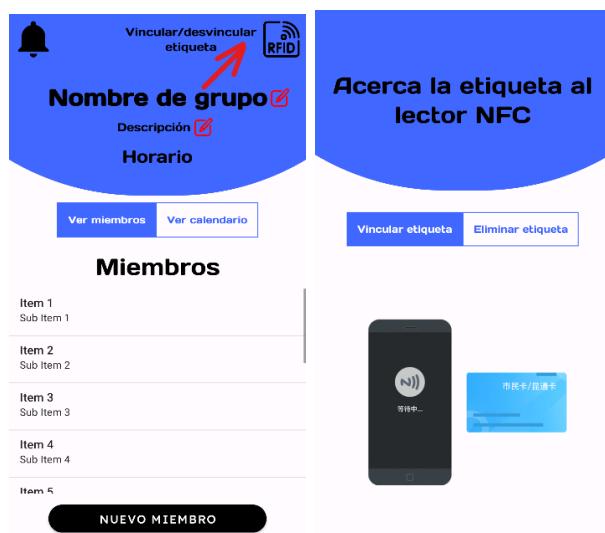
- **Registrar nueva asistencia**

Siendo asistentes, podemos registrar una asistencia en la pantalla principal del grupo, dándole al botón de registrar asistencia y acercando el lector NFC a la etiqueta del lugar. En caso de no poder de esa forma, el administrador al mantener pulsado un miembro puede registrar su asistencia también.



- Registrar y dar de baja etiqueta**

Siendo administradores, podemos registrar a un grupo o eliminar una etiqueta dándole al botón de vincular/desvincular etiqueta en la pantalla principal del grupo. Aquí podemos seleccionar cuál de las dos tareas queremos hacer y, con la tecnología NFC activada, acercamos la etiqueta al lector. Aparecerá un mensaje de éxito, o de error en caso de que no se haya podido realizar.



- Opciones de miembros**

Siendo administradores, podemos realizar algunas acciones si mantenemos pulsado al miembro a gestionar. Ahí podremos eliminar el miembro del grupo, registrar una nueva asistencia, ver las vacaciones y añadir un periodo vacacional nuevo.



- Ver notificaciones**

Siendo administradores, las notificaciones que nos lleguen pueden ser consultadas pulsando en el ícono de la campana en la pantalla principal o en la pantalla principal de un grupo. Cómo se ha mencionado, podremos verlas pulsando el ícono de la campana en la pantalla principal o en cualquier pantalla de grupo.



4.3. METODOLOGÍA SEGUIDA

En el proceso de desarrollo de cualquier software, la metodología que se siga es un aspecto crucial para garantizar el éxito del proyecto. En el caso que estamos desarrollando, hemos optado por seguir una metodología tradicional de cascada debido a la naturaleza clara y definida de los requisitos del sistema.

La metodología de cascada es un enfoque secuencial y lineal, donde cada fase del proyecto se completa antes de pasar a la siguiente. Las fases incluyen la definición de requisitos, el diseño, la implementación, las pruebas y el mantenimiento. En este caso, se comenzó con un borrador de los requisitos del sistema y, a partir de ahí, se empezó la fase de definición de requisitos.

En esta fase, se definen las necesidades y se establecen claramente los requisitos del sistema. A continuación, se pasa a la fase de diseño, donde se crean diagramas de flujo y diagramas de clases para visualizar la estructura y el funcionamiento del sistema. Después de la fase de diseño, se pasa a la implementación, donde se desarrolla y se codifica el software.

Una vez que se completa la implementación, se llevan a cabo pruebas para garantizar que el software funcione de acuerdo con los requisitos establecidos en la fase de definición de requisitos. Si se encuentran errores, se corrigen antes de pasar a la fase de mantenimiento, donde se realiza el mantenimiento y se actualiza el software según sea necesario.

En conclusión, se ha utilizado una metodología de cascada para el desarrollo debido a la naturaleza clara y definida de los requisitos del sistema. La metodología de cascada permite trabajar de manera secuencial y lineal, lo que facilita el seguimiento del progreso y asegura que se satisfagan los requisitos establecidos.

4.4. RECURSOS

MATERIALES

Para el desarrollo y puesta en marcha del sistema, se necesitan varios recursos materiales. Estos recursos son necesarios para asegurar que el sistema funcione correctamente y pueda manejar la carga de tráfico que se espera.

En primer lugar, es importante contar con un hardware adecuado para el servidor de base de datos. Este hardware debe ser lo suficientemente potente como para manejar la carga de la base de datos, lo que incluye un procesador de alta capacidad, suficiente memoria RAM y un disco duro con una capacidad suficiente

para almacenar los datos de registro. Se debe tener en cuenta que el servidor de base de datos debe ser escalable para poder manejar un aumento en la cantidad de usuarios y datos.

En segundo lugar, se requiere hardware adecuado para el servidor de aplicaciones. Esto incluye un procesador potente, suficiente memoria RAM y un disco duro con capacidad para almacenar la aplicación y los datos. Además, se necesita una conexión de red estable para asegurar que los clientes puedan acceder a la aplicación sin problemas.

En tercer lugar, es importante que los clientes cuenten con hardware adecuado. Para los dispositivos móviles, se necesita un smartphone o tablet compatible con Android y, aunque no sea necesario, es importante que sea compatible con la tecnología NFC, ya que si no se pierde la funcionalidad principal de la aplicación. También es esencial que se esté conectado a internet para conectarse con el servidor de aplicaciones.

Por último, para poder registrar las asistencias, son necesarias las etiquetas NFC que serán registradas y se colocarán en los lugares para ser escaneadas.

INFRAESTRUCTURA Y EQUIPAMIENTO

Es importante tener en cuenta los recursos de infraestructura y equipamiento necesarios para desarrollar y poner en marcha el sistema que pueda ser utilizado por los usuarios.

En la parte del servidor, es necesario contar con el servidor de base de datos MySQL que aloja la base de datos. Se utilizará, además, MySQL Workbench como herramienta visual para la administración de la base de datos.

También en el servidor, la aplicación de escritorio que actúa como servidor de aplicación ha sido desarrollada en Java, con el entorno de desarrollo Eclipse, por lo que Java es necesario para ejecutar la aplicación y Eclipse también para el desarrollo de la misma. Para conectar el servidor de la aplicación con el de base de datos, es utilizado el jdbc de MySQL para Java, por lo que es necesario el uso de las bibliotecas.

Al tratarse de un sistema cliente-servidor, es importante tener en cuenta la conectividad a internet de alta velocidad para garantizar la disponibilidad de la aplicación para los usuarios y la fiabilidad de la información almacenada en la base de datos.

Para los clientes que instalen la aplicación es importante contar con dispositivos móviles Android y que cuenten como mínimo con la versión de Android 10 (API 29), ya que es la versión de Android mínima para poder instalar la aplicación, ya

que esta contiene características que se implementaron en esa versión. Además como ya hemos mencionado antes, el dispositivo debería ser compatible con la tecnología NFC porque la funcionalidad principal de la aplicación requiere escanear la etiqueta NFC para registrar la asistencia.

En el desarrollo de la aplicación Android ha sido utilizado el lenguaje de programación Java y el entorno de desarrollo Android Studio. Se ha utilizado el framework de Android para el desarrollo de la mayoría de funcionalidades. Este framework está incluido en Android Studio por lo que no ha sido necesario descargar ningún extra.

Al igual que en el servidor, es necesario contar con una constante conexión a internet al utilizar la aplicación, ya que de otro modo no podrá conectarse con el servidor.

HUMANOS

Para el desarrollo del sistema los recursos humanos necesarios para el proyecto son el desarrollador que crea el sistema desde cero y los usuarios que lo utilizan.

En primer lugar, el desarrollador de software es capaz de diseñar, desarrollar y mantener el sistema completo. Dado que en el caso, el sistema implica un servidor de base de datos MySQL, un servidor de aplicación Java y una aplicación móvil Android, el desarrollador debería tener habilidades y experiencia en estas tecnologías. El desarrollador también debe ser capaz de trabajar de manera autónoma y ser responsable de todo el desarrollo del proyecto, desde el análisis de requisitos hasta el despliegue final.

El desarrollador debe diseñar, crear y mantener la base de datos para asegurarse de que se implemente un diseño de base de datos adecuado y eficiente. Debe ser capaz de optimizar la base de datos para un rendimiento óptimo y asegurarse de que se mantengan los datos confiables y precisos.

Este también debe desarrollar todo diseño de interfaces para crear una interfaz de usuario intuitiva, con modo claro y oscuro y fácil de usar para los usuarios. Debe tener una comprensión profunda de cómo diseñar interfaces de usuario efectivas para asegurarse de que la interfaz sea coherente y se ajuste a las necesidades del usuario.

El desarrollador también debe ejecutar el diseño, desarrollo y mantenimiento del servidor de aplicación Java. Debe ser capaz de optimizar el rendimiento del servidor, diseñar soluciones de escalabilidad, integrar nuevas funcionalidades, y garantizar que el servidor esté configurado correctamente y funcionando de manera óptima, junto a sus conexiones con la aplicación Android y con la base de datos.

Por otro lado, desarrolla y mantiene la aplicación móvil. Es capaz de realizar desde cero una aplicación Android de alta calidad que sea fácil de usar y que ofrezca una excelente experiencia de usuario en varios lenguajes. Además, también debe ser responsable de optimizar el rendimiento de la aplicación, integrar nuevas funcionalidades, y asegurarse de que la aplicación esté configurada y funcionando correctamente.

A parte del desarrollador, contamos con los usuarios clientes, quienes son administradores y no administradores, los principales usuarios del sistema. Los usuarios administradores son aquellos que gestionan los grupos. Estos usuarios pueden acceder a un registro completo de la asistencia de todos los usuarios de sus grupos y ver la hora de entrada y salida. Los administradores también pueden agregar y eliminar usuarios en los grupos que manejen y configurar las opciones como los horarios de trabajo y los días festivos.

Por otro lado, los usuarios comunes son aquellos que utilizan la aplicación móvil para registrar su entrada y salida al lugar de trabajo. Estos usuarios simplemente tienen que abrir la aplicación, registrar su entrada o salida, y guardar el registro en el sistema. Los usuarios comunes también pueden ver su propia información en el sistema, como sus detalles relacionados con su registro de asistencia.

5. PRESUPUESTO

Es importante monitorear los gastos, ingresos y beneficios de manera continua para evaluar la rentabilidad del proyecto. Se pueden realizar ajustes en los precios, estrategias de marketing o enfoques comerciales para mejorar la situación financiera y lograr un beneficio positivo en el futuro.

5.1. GASTOS

Los gastos del proyecto pueden variar según diferentes factores, pero a modo de ejemplo, se podrían considerar los siguientes:

- a) **Desarrollo de software:** Se estima un gasto de 1700€ en salarios para el desarrollador que crea el sistema. Teniendo en cuenta las horas invertidas en el diagrama de Gantt en el proyecto, con un salario es suficiente para crear el sistema. De esta manera los 1700€ corresponden a 120 horas de trabajo, lo que sale a 14.16€ la hora de trabajo, teniendo en cuenta que en este presupuesto se incluye el gasto de material.
- b) **Infraestructura:** Para el alojamiento del servidor y el almacenamiento de datos, se prevé un gasto pequeño al inicio de entre 200€ y 500€ anuales en servicios de alojamiento en la nube. Lo ideal es utilizar una plataforma en la nube escalable como Amazon Web Services por la que se pague en función de los recursos consumidos.
- c) **Material:** Al tratarse de un desarrollador freelance quien crea el proyecto, no se debe gastar en material extra, ya que este coste se incluye en el salario del desarrollador.
- d) **Mantenimiento:** Para la capacitación del personal encargado de operar y mantener el sistema, se destina un presupuesto de 100€ mensuales tras la publicación de la aplicación.
- e) **Marketing y promoción:** Para dar a conocer el sistema, se estima un gasto de 300€ al principio, en la creación de materiales de marketing, participación en eventos y publicidad en línea.

5.2. INGRESOS

Los ingresos del proyecto pueden provenir de diferentes fuentes, aunque pueden considerarse los siguientes:

- a) **Ventas del sistema:** Supongamos que se logran vender licencias del sistema a 100 empresas y entidades educativas (entre otras), a un precio por usuario de 2€ al año. Si el promedio de usuarios por empresa o entidad es de 70 usuarios, esto generaría ingresos de 14000€ anuales.

- b) **Servicios adicionales:** Ofreciendo servicios de soporte técnico y personalización del sistema a un promedio de 1000€ por cliente al año, se podrían generar ingresos adicionales de 10.000€ si lo contratan 10 entidades anualmente.

5.3. BENEFICIO

El beneficio se calcula restando los ingresos generados por el proyecto de sus gastos totales.

Siguiendo los ejemplos mencionados, supongamos que los gastos totales del primer año son de 5000€, incluyendo el desarrollo, infraestructura, material, mantenimiento y marketing. Por otro lado, los ingresos del primer año en base a lo estimado podrían ser de entre 20000€ y 24000€.

El beneficio neto sería de entre 15000€ y 19000€, lo que sería un buen comienzo para ser el primer año. Este beneficio se aumentaría significativamente los siguientes años, ya que habría más facilidad para encontrar clientes y menos gastos ya que el desarrollo estaría realizado.

6. RESULTADOS Y CONCLUSIONES

Este proyecto ha sido un desafío en términos del desarrollo de un sistema de control de asistencia que cumpla con los requisitos establecidos. Sin embargo, después de analizar los resultados obtenidos del sistema, podemos afirmar que se ha logrado obtener un sistema cliente-servidor que funciona correctamente.

El modelo implementado consiste en una aplicación Android que utiliza tecnología NFC y etiquetas RFID para confirmar la asistencia a un lugar cuando esta etiqueta es escaneada. La comunicación entre los dispositivos Android y el servidor de aplicaciones se ha conseguido mediante sockets TCP y este interactúa con el servidor de base de datos MySQL.

Además, el objetivo del proyecto ha sido asegurar que el sistema esté disponible para todos los usuarios que necesiten utilizarlo en sus actividades de seguimiento. Para lograr esto, el servidor ha de estar constantemente operativo y proporcionar los datos que se le soliciten. Esto permite a los usuarios que con tan solo iniciar sesión con su usuario en la aplicación, y sin necesidad de realizar instalaciones adicionales, puedan acceder a su cuenta con sus grupos y datos de asistencias. También se ha trabajado en mejorar de manera constante la aplicación de forma transparente para los usuarios. Se ha puesto especial hincapié en diseñar una interfaz intuitiva y fácil de usar.

El sistema, además, se ha diseñado con el patrón de arquitectura modelo–vista–controlador, separando de esa manera las clases de las actividades (interfaces gráficas), los controladores y el DAO (acceso a datos). Esto permite que se puedan añadir nuevas funcionalidades fácilmente en el sistema, sin necesidad de modificar lo existente apenas.

La experiencia adquirida a lo largo del ciclo en áreas como programación paralela, bases de datos, desarrollo de interfaces gráficas y sistemas, entre otras, ha sido fundamental para llevar a cabo este proyecto, y, junto con la investigación externa realizada, se ha podido llevar a cabo el proyecto con éxito. No solo he podido aplicar los conocimientos adquiridos en estas áreas, sino que también he logrado tener una visión global de su conjunto y ponerlos en práctica de manera efectiva.

7. BIBLIOGRAFÍA

[Android 10 para desarrolladores](#)

[Exploring Android Q: Settings Panels | by Joe Birch | Google Developer Experts | Medium](#)

- ▶ [How to Integrate Google Sign In in Android | 2022](#)
- ▶ [How to make responsive Registration and Login Activity using constraint Layout...](#)
- ▶ [Como implementar los text fields de material design en Android](#)

[TextInputEditText | Android Developers](#)

- ▶ [TextInputEditText - Material Design Edit Text | Android Studio Tutorial](#)
- ▶ [How Filter Credentials \(Email & Password \) & Show Error to EditText in Androi...](#)
- ▶ [Profile design UI Diseño UI de perfil // De Adobe XD a Android Studio parte2](#)
- ▶ [Custom Dialog box - Materiel Design UI Alert Dialog - Beautiful Alert Dialog An...](#)

[Fragments | Desarrolladores de Android](#)

[Fragment | Android Developers](#)

- ▶ [4. Fragments en Activity \(Colocar y Reemplazar fragments\) 📱 || Curso de Andr...](#)
- ▶ [64. Como Agregar Eventos En Fragments - Android](#)

[Descripción general de los avisos | Desarrolladores de Android](#)

- ▶ [Android Studio - Toast Personalizado](#)
- ▶ [CREAR UN TOAST PERSONALIZADO - ANDROID STUDIO | PARTE 7](#)

[Descripción general sobre la Comunicación de campo cercano | Desarrolladores de Android](#)

[Conceptos básicos de NFC | Desarrolladores de Android | Android Developers](#)

[Descripción general de conceptos avanzados de NFC | Desarrolladores de Android](#)

[NfcAdapter.ReaderCallback | Android Developers](#)

[Is it possible to read an NFC tag from a card without using an intent? - Stack Overflow](#)

[GitHub - SundeepK/CompactCalendarView: An android library which provides a compact calendar view much like the one used in google calenders.](#)

[GitHub - prolificinteractive/material-calendarview: A Material design back port of Android's CalendarView](#)

[A simple and customizable calendar widget for Android based on Material Design](#)

[MaterialCalendarView – customizable calendar widget for Android \[updated 2022\]](#)

[com.prolificinteractive.materialcalendarview.MaterialCalendarView java code examples | Tabnine](#)

[Prueba tu app | Android Studio](#)

[Cómo realizar pruebas en Android Studio](#)

[Prueba Unitaria JUnit](#)

8. ANEXOS

DISEÑO DE INTERFACES GRÁFICAS

Se han diseñado cuidadosamente cada una de las pantallas de la interfaz con el objetivo de garantizar una experiencia de usuario óptima, teniendo en cuenta aspectos como la facilidad de aprendizaje, la tasa de errores, la estética y la consistencia en el diseño.

En primer lugar, se ha enfocado en facilitar el aprendizaje para que los usuarios puedan utilizarla de manera intuitiva desde el principio. Se ha optado por una interfaz sencilla y clara, donde la información y las opciones se presentan de forma organizada y fácilmente comprensible. Por ejemplo, en la pantalla principal de un grupo, se muestra de manera destacada el botón de "Nueva asistencia", acompañado de instrucciones claras sobre los pasos a realizar en cada momento.

La reducción de la tasa de errores ha sido una prioridad en el diseño. Se han implementado mecanismos para minimizar las equivocaciones por parte de los usuarios. Por ejemplo, hemos utilizado botones de gran tamaño y con un diseño distintivo, lo que facilita la selección precisa y evita acciones accidentales. Además, se han incluido validaciones en tiempo real para detectar los errores. Cuando en las operaciones se producen estos errores, siempre se especifica en un mensaje de alerta el error específico que ha ocurrido para que el usuario esté informado de todo aquello que haya podido ir mal.

En términos de estética, se ha buscado crear una interfaz atractiva y profesional. Se ha utilizado una combinación de colores y tipografía legibles para transmitir seriedad y confianza en el contexto educativo, laboral y profesional. Asimismo, se ha tenido cuidado la disposición de los elementos visuales para mantener un diseño ordenado y equilibrado, evitando la sobrecarga de información en cada pantalla.

La consistencia en el diseño ha sido clave para facilitar la familiaridad y la eficiencia en el uso de la aplicación. Han sido empleados patrones de diseño y acciones similares a lo largo de todas las pantallas. Por ejemplo, se ha mantenido una cabecera de información y opciones constante en la parte superior para acceder fácilmente a nombres de usuario y grupo, descripciones, notificaciones y etiquetas RFID. Además, han sido utilizados iconos reconocibles para representar acciones comunes, como los de etiquetas RFID, notificaciones y edición de nombres y descripciones.

A continuación se van a mostrar capturas de pantalla de todas las interfaces gráficas de la aplicación Android. Al ser compatible con el modo claro y oscuro del dispositivo, aquellas interfaces en las que haya diferencia entre ambos modos, se

mostrarán capturas de los dos. Además, y aunque no se muestre en las capturas, la aplicación es compatible con los idiomas inglés y español, que se mostrará dependiendo de la configuración de idioma del dispositivo. También cabe mencionar que todas las interfaces a continuación son mostradas desde un usuario administrador ya que es este quien tiene desbloqueadas todas las funcionalidades de la aplicación.

El servidor no tiene ninguna interfaz gráfica, por lo que no se mostrará nada relacionado con ello.

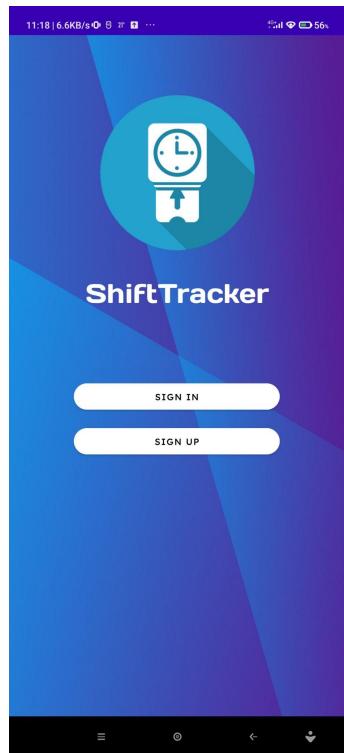
Splash Screen (pantalla de carga):

Esta se muestra cómo pantalla de carga hasta que la actividad principal ha cargado y se muestra.

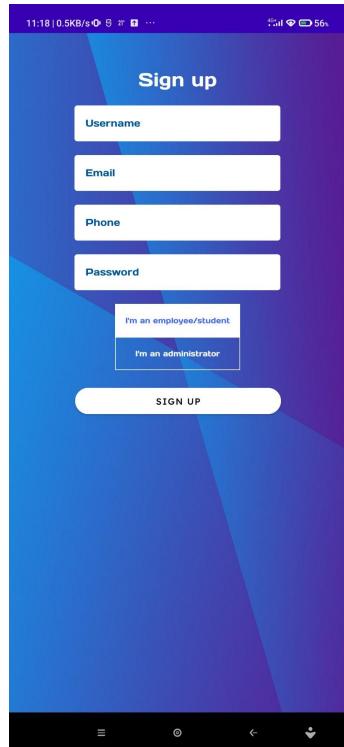


Actividad bienvenida:

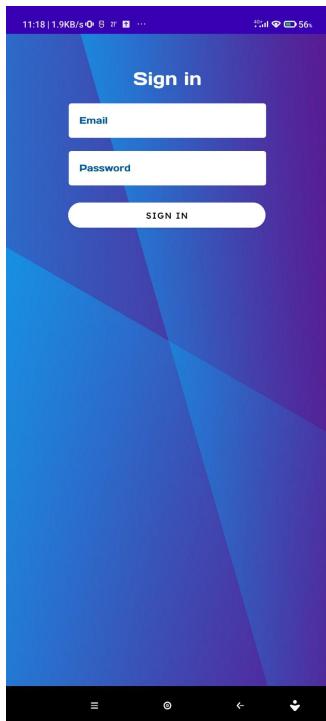
Se muestra solo cuando no hay una sesión iniciada en la aplicación.



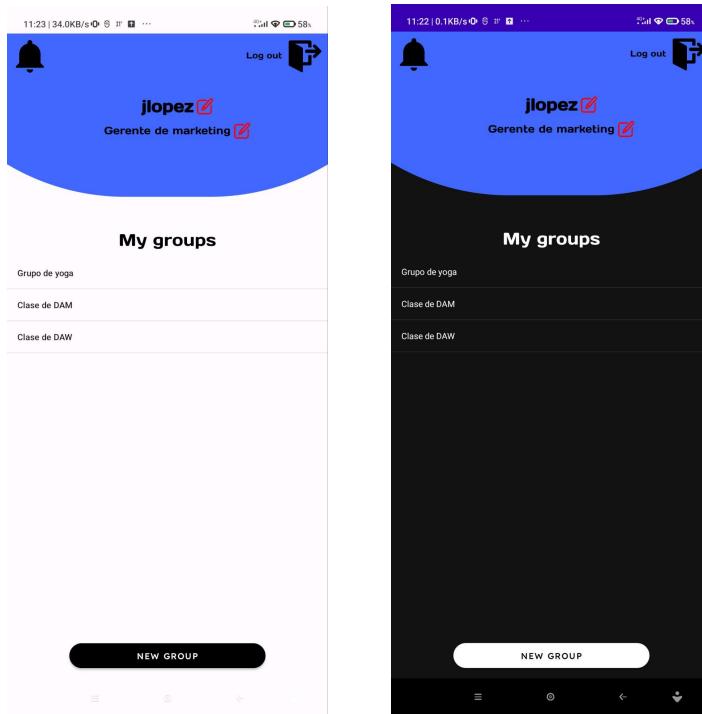
Actividad registrarse:



Actividad iniciar sesión:

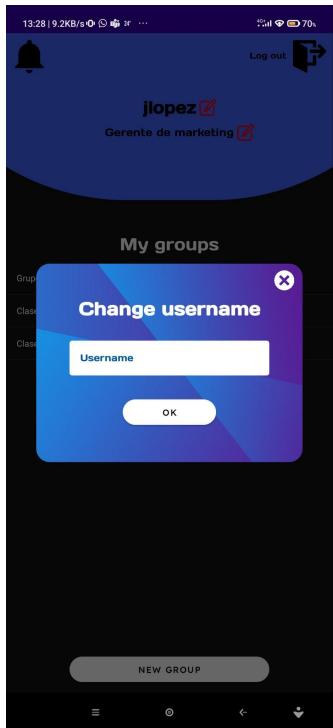


Actividad principal:



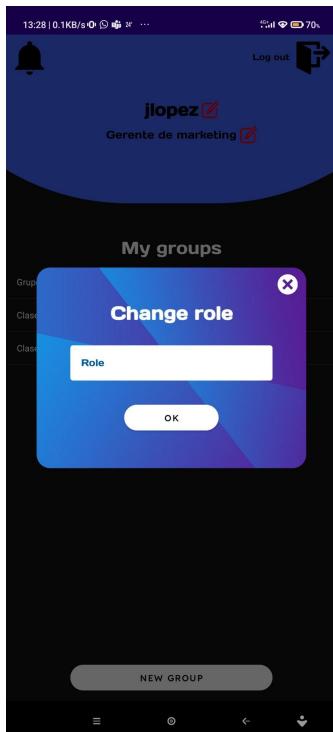
Ventana de diálogo editar nombre de usuario (Actividad principal):

Accesible pulsando el botón de editar al lado del nombre de usuario.



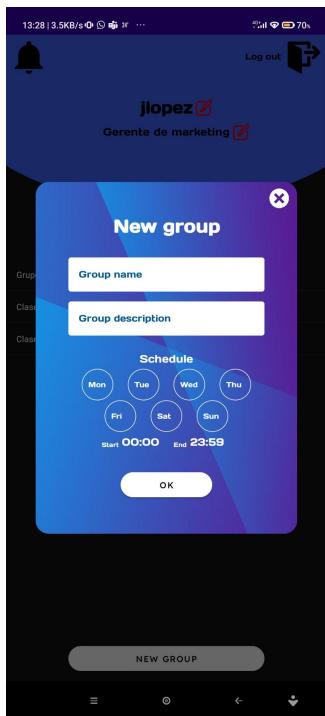
Ventana de diálogo editar cargo (Actividad principal):

Accesible pulsando el botón de editar al lado del cargo del usuario.



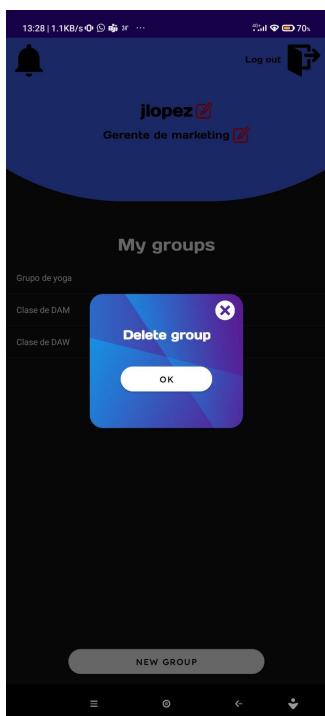
Ventana de diálogo nuevo grupo (Actividad principal):

Accesible pulsando el botón de nuevo grupo.

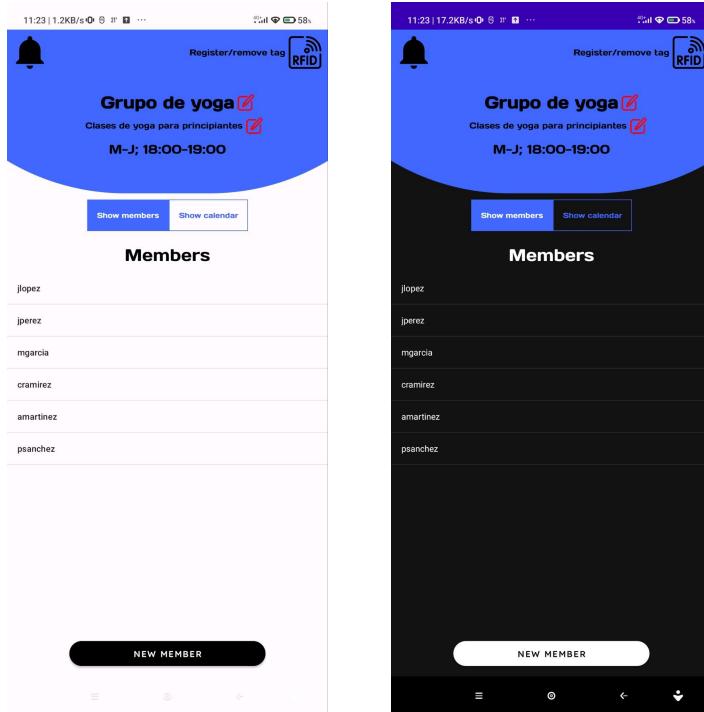


Ventana de diálogo eliminar grupo (Actividad principal):

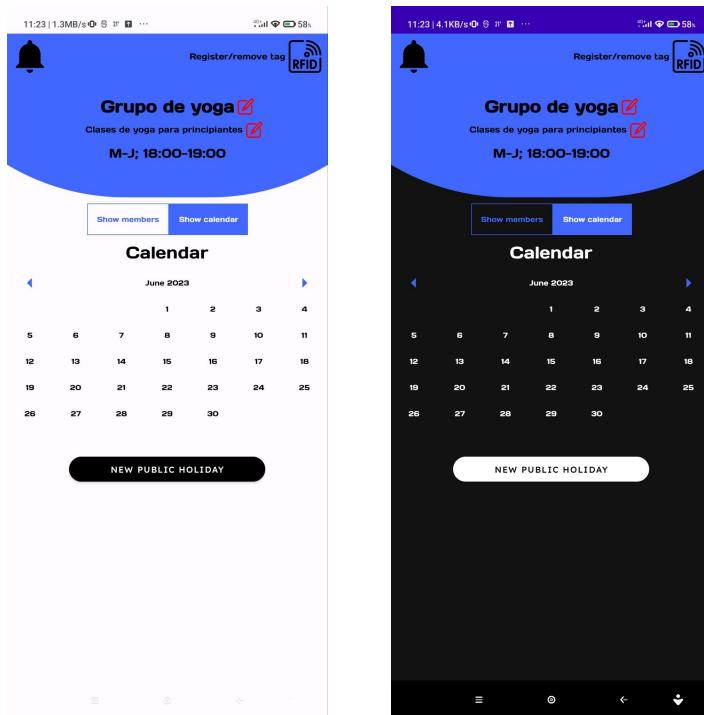
Accesible manteniendo pulsado el grupo a eliminar.



Actividad principal grupo (Fragmento/pestaña miembros):

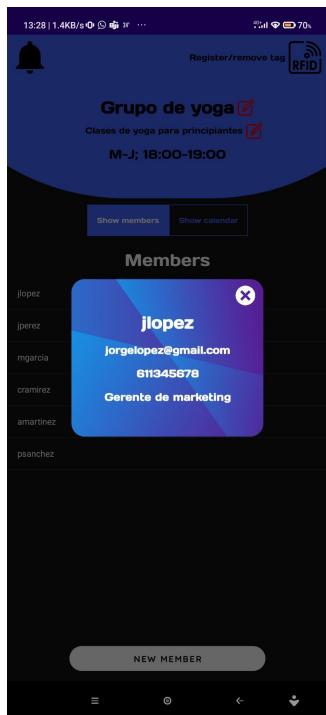


Actividad principal grupo (Fragmento/pestaña calendario):



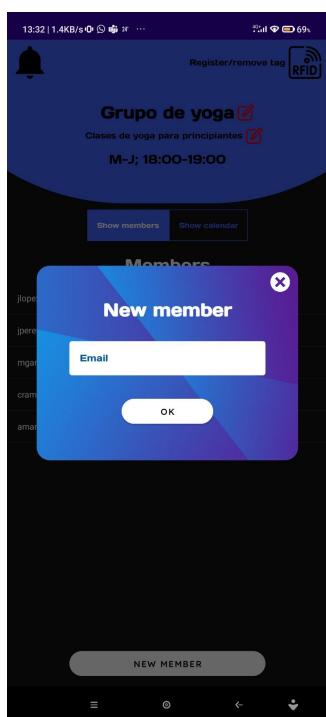
Ventana de diálogo ver miembro (Actividad principal grupo):

Accesible pulsando el miembro a ver.



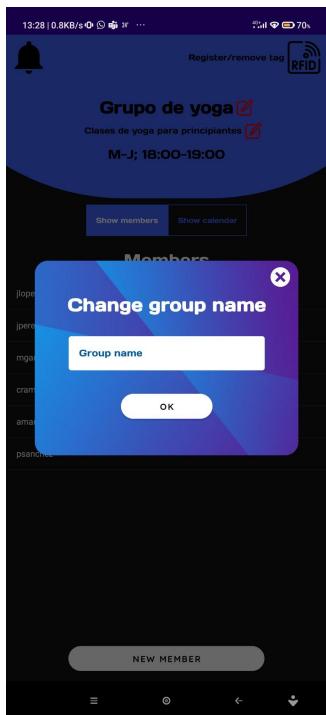
Ventana de diálogo añadir miembro (Actividad principal grupo):

Accesible pulsando el miembro a ver.



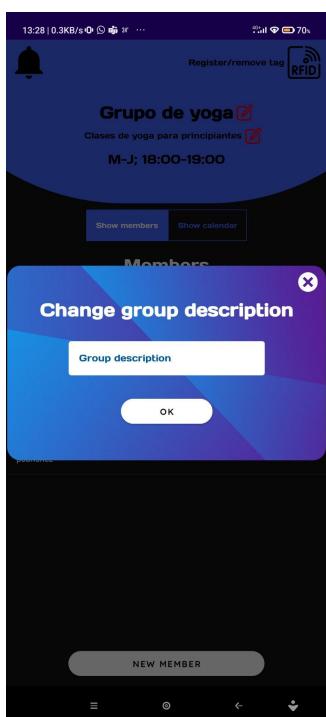
Ventana de diálogo editar nombre de grupo (Actividad principal grupo):

Accesible pulsando el botón de editar al lado del nombre de grupo.



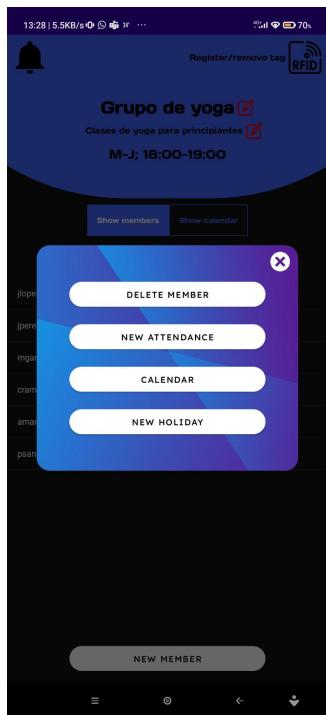
Ventana de diálogo editar descripción de grupo (Actividad principal grupo):

Accesible pulsando el botón de editar al lado de la descripción de grupo.



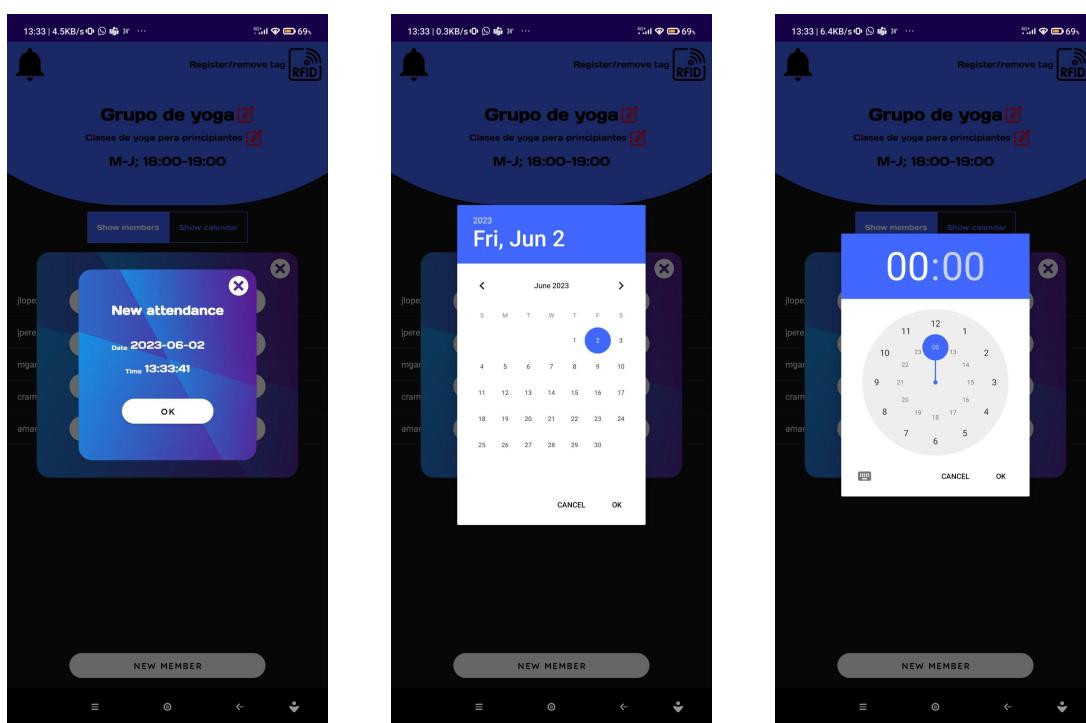
Ventana de diálogo opciones de miembros (Actividad principal grupo):

Accesible manteniendo pulsado el miembro a realizar las acciones.



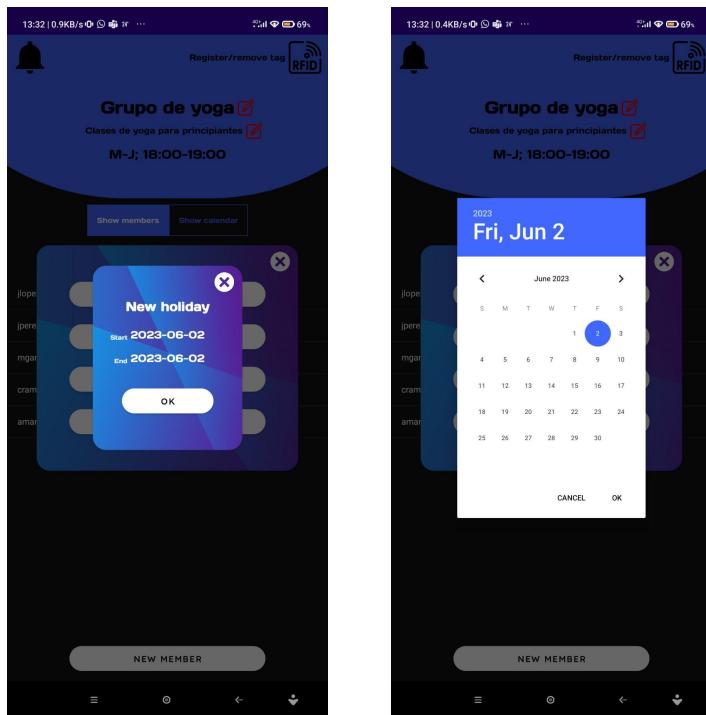
Ventana de diálogo nueva asistencia (Actividad principal grupo):

Accesible pulsando el botón de nueva asistencia.



Ventana de diálogo nuevas vacaciones (Actividad principal grupo):

Accesible pulsando el botón de nuevas vacaciones.



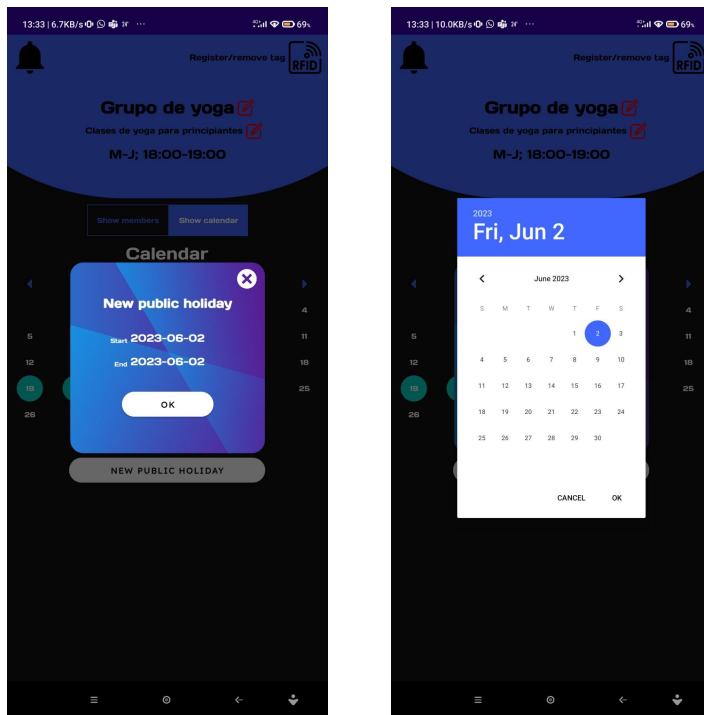
Ventana de diálogo calendario (Actividad principal grupo):

Accesible pulsando el botón de calendario.



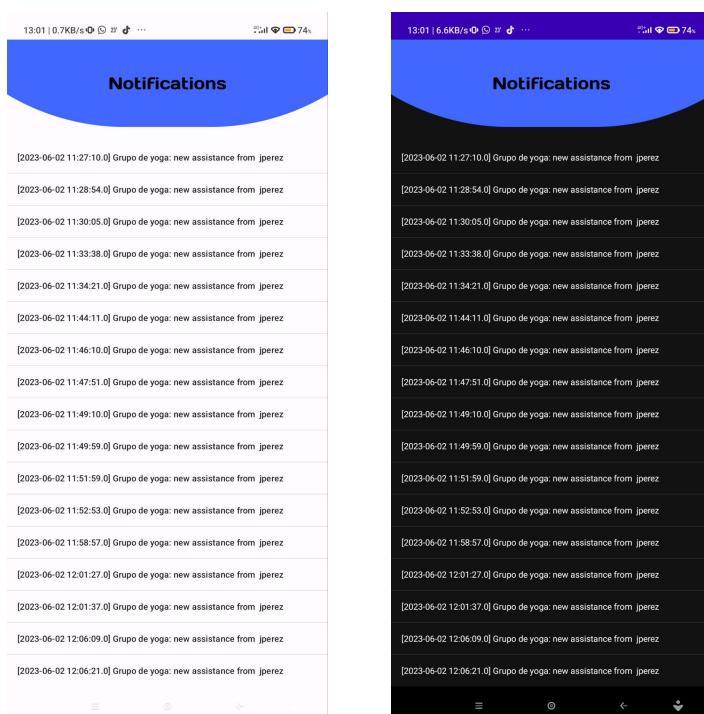
Ventana de diálogo nuevas vacaciones (Actividad principal grupo):

Accesible pulsando el botón de nuevas vacaciones.



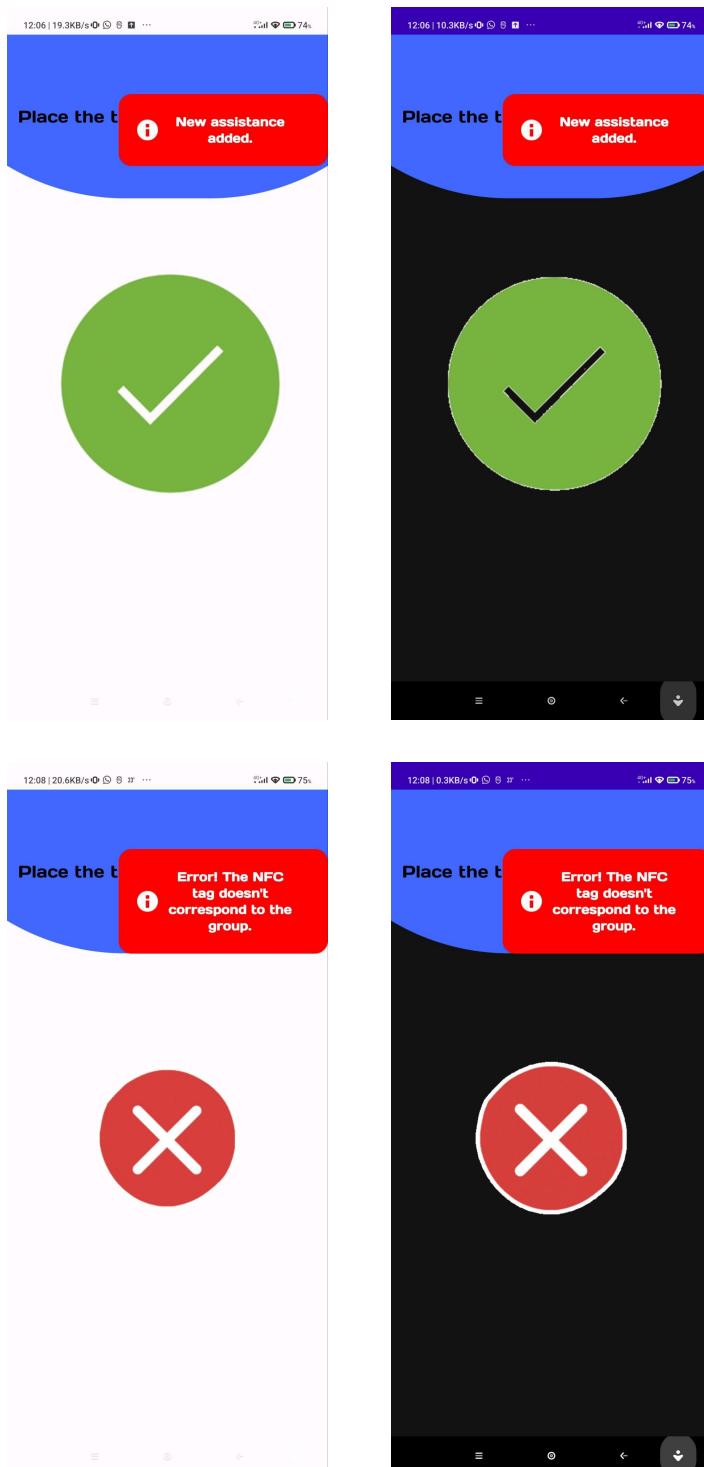
Actividad notificaciones:

Accesible pulsando el botón de campana en la actividad principal o principal grupo.



Actividad Nfc:

Accesible pulsando el botón de la etiqueta RFID en la actividad principal grupo.



RELACIÓN DE HORAS RESPECTO A LA ESTIMACIÓN

- **Primera entrega (11 de abril):** se ha cumplido correctamente la estimación de horas. La mayoría del tiempo se dedicó a planear cómo va a desarrollarse el sistema y su arquitectura y la investigación sobre la tecnología NFC a lo que se dedicó unas 15 horas aproximadamente en estos trabajos. Después, la parte de la memoria se estimó correctamente.
- **Segunda entrega (24 de abril):** la estimación de horas se cumplió, aunque no el orden de las tareas realizadas, ya que lo primero que se realizaron fueron los diagramas de bases de datos, después los de casos de uso, los de secuencia, los de clases y, por último, el de Gantt. Además, se adelantó el desarrollo del entorno cliente con el comienzo de la investigación y el desarrollo de las interfaces gráficas.
- **Tercera entrega (8 de mayo):** la mayoría del tiempo se utilizó para la investigación y el desarrollo de las interfaces gráficas. Al igual que la anterior entrega se adelantó el desarrollo. Aquí también se han comprimido más las horas en los diferentes días, pero la previsión de horas fué bien estimada. Las clases pojo se han desarrollado conforme han sido necesarias, en vez de todas a la vez, pero si se ha cumplido la intención de realizar todas las interfaces gráficas primero y después programar las funcionalidades y los sockets.
- **Cuarta entrega (22 de mayo):** ya se han dedicado aproximadamente 100 horas al proyecto. La memoria está casi finalizada y al desarrollo no le queda mucho. Ha habido que corregir algunos diagramas y ha habido que rehacer algunos como los de clases casi enteros, por lo que las horas estimadas en este han aumentado. Este diagrama ha sido dificultoso hacerlo legible para que cualquiera pueda entenderlo. A pesar de ello, no han aumentado demasiado las horas invertidas. Las horas estimadas se cumplirán aunque en menos días de los estimados, ya que se ha trabajado más horas cada día.
- **Entrega final:** finalizado el desarrollo, se han hecho comentarios explicando la funcionalidad de los métodos en el propio código. Sin embargo, como la memoria tiene que corresponderse con el desarrollo, se han tenido que realizar algunas modificaciones en la memoria. El diagrama de clases, los diagramas de bases de datos, algunos casos de uso, interfaces gráficas, etcétera han tenido que ser modificados o se les han tenido que añadir datos. A todo esto, se le añade la preparación de la presentación Power Point y la defensa del proyecto. A la presentación ha tenido que dedicarse unas 3 o 4 horas, ya que ha sido necesario adaptar y simplificar los diagramas para que sean legibles en el Power Point.

Por todo esto mencionado se han necesitado unas horas extra. Finalmente, al proyecto podemos concluir que se le han dedicado unas 130 horas. Esto son aproximadamente 10 horas más de lo estimado en el diagrama de Gantt.