**Turnosync: Aplicación android para la gestión de turnos laborales**

Trabajo de Fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA



Septiembre de 2018

Autor

Julio García Valdunciel

Tutor

Iván Álvarez Navia

D. Iván Álvarez Navia profesor del Departamento de Informática y Automática de la Universidad de Salamanca.

CERTIFICAN:

Que el trabajo titulado “Turnosync: Aplicación android para la gestión de turnos laborales” ha sido realizado por D. Julio García Valdunciel, con DNI 70892288-D y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado en Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a \_\_\_\_\_\_\_de Septiembre de 2018

D. Iván Álvarez Navia

Dpto. Informática y Automática

Universidad de Salamanca

# Resumen y palabras clave

El proyecto realizado como Trabajo de Fin de Grado consiste en un sistema cliente-servidor donde el rol de cliente lo desempeña la aplicación móvil en el sistema Android. El rol de servidor que incluye la base de datos sobre la que se basa la aplicación lo representa la plataforma Firebase, propiedad de Google.

El objetivo principal de esta aplicación está orientado a ayudar en la gestión y organización dentro de un entorno de trabajo con un modelo por turnos, normalmente siendo una cobertura de 24 horas. Aunque la aplicación no está restringida a ningún entorno ni profesión, la principal fuente de inspiración para este proyecto ha sido el entorno laboral dentro de un hospital, concretamente la profesión de enfermería.

La aplicación permitirá agilizar la comunicación a la hora de establecer y modificar el calendario de turnos dentro de un grupo de trabajo que actualmente suelen estar en formato físico en papel o en una plantilla del tipo Excel.

La aplicación propone sustituir o complementar estos medios llevando este calendario a un formato digital apoyado sobre una base de datos en la nube, de forma que cualquier modificación es accesible en tiempo real por los trabajadores que actúan como usuarios. Se dispone de un calendario personal del usuario y un calendario grupal donde es posible acceder al horario del resto de usuarios del grupo.

Además de la sincronización del calendario la aplicación dispone de un sistema de solicitud de cambios de turnos entre usuarios de forma que para que se haga efectivo deben estar de acuerdo las dos partes y ser aprobado en última instancia por el coordinador.

Palabras clave:

Aplicación móvil, Android, turnos laborales, Firebase, sincronización.

# Abstract and keywords

The proyect developed as Final Degree Project is a client-server system where the client role is played by the Android mobile app. The server role that includes the database upon the app is supported on is represented by the Firebase platform, owned by Google.

The main objective of this app is oriented towards helping at the management and organization within a work environment based on a work shift model, usually it being a 24 hours coverage. Even though the app is not restricted to any environment nor profession, the main source of inspiration for this project has been the work environment within a hospital, being more specific to nursing.

This app will allow to speed up the communication when setting up and modifying the shift calendar within a work group which currently they are usually in physical format on paper or an Excel template.

This app tries to replace or complement this mediums taking this calendar to a digital format supported by a cloud database, so that any modification is accessible in real time by the workers who act as users. It provides a personal calendar as well as a group calendar where each member has access to each other’s shifts.

In addition to calendar synchronization the app also provides a shift change request system between the users so that for it to be effective it’s necessary that both parts agree and in the end it has to be approved by the manager.

Keywords:

Mobile app, Android, work shifts, Firebase, synchronization.

Tabla de contenido

[Resumen y palabras clave - 5 -](#_Toc521324786)

[Abstract and keywords - 6 -](#_Toc521324787)

[1. Introducción - 10 -](#_Toc521324788)

[1.1 Estructura de la memoria - 12 -](#_Toc521324789)

[1.2 Estructura de los anexos - 12 -](#_Toc521324790)

[2. Objetivos del proyecto - 14 -](#_Toc521324791)

[2.1 Objetivos funcionales - 14 -](#_Toc521324792)

[2.2 Objetivos personales - 16 -](#_Toc521324793)

[3. Conceptos teóricos - 18 -](#_Toc521324794)

[3.1 Arquitectura Cliente – Servidor - 18 -](#_Toc521324795)

[3.2 Android y Android SDK - 18 -](#_Toc521324796)

[3.3 Base de datos no relacional - 20 -](#_Toc521324797)

[3.4 Material Design - 20 -](#_Toc521324798)

[4. Técnicas y herramientas - 22 -](#_Toc521324799)

[4.1 Programación orientada a objetos - 22 -](#_Toc521324800)

[4.2 Android studio - 24 -](#_Toc521324801)

[4.3 Sublime Text 3 - 25 -](#_Toc521324802)

[4.4 Lenguajes de programación - 25 -](#_Toc521324803)

[4.4.1 Java - 25 -](#_Toc521324804)

[4.4.2 XML - 26 -](#_Toc521324805)

[4.4.3 JavaScript - 27 -](#_Toc521324806)

[4.5 Plataformas externas - 27 -](#_Toc521324807)

[4.5.1 Firebase - 27 -](#_Toc521324808)

[4.5.2 GitHub - 30 -](#_Toc521324809)

[4.6 Bibliotecas - 30 -](#_Toc521324810)

[4.6.1 Joda Time - 31 -](#_Toc521324811)

[4.7 Técnicas y metodologías - 31 -](#_Toc521324812)

[5. Aspectos relevantes del desarrollo. - 31 -](#_Toc521324813)

[6. TODO\* - 31 -](#_Toc521324814)

[7. Conclusiones y líneas de trabajo futuro - 31 -](#_Toc521324815)

[8. Bibliografía - 31 -](#_Toc521324816)

Tabla de ilustraciones

[Ilustración 1 - Arquitectura de capas del sistema Android - 18 -](#_Toc521324780)

[Ilustración 2 - Estilo Material Design - 20 -](#_Toc521324781)

[Ilustración 3 - Vista fragment\_mycalendar.xml - 25 -](#_Toc521324782)

[Ilustración 4 - Cloud Firestore - 28 -](#_Toc521324783)

# Introducción

Este documento recoge la memoria del Trabajo de Fin de Grado de Julio García Valdunciel, estudiantes del Grado en Ingeniería Informática en la Universidad de Salamanca.

El modelo de trabajo por turnos se basa en la necesidad de dar un servicio de normalmente 24h al día, por lo que es necesario que los trabajadores se turnen para cumplir este horario. A continuación se explicara con más detalle este modelo.

Se organizan de las siguientes formas:

* Sistema discontinuo: Es aquel que se interrumpe por la noche y en los fines de semana. Dispone de dos turnos, el de mañana y el de tarde.
* Sistema semi-continuo: Es aquel que solo se interrumpe los fines de semana. Dispone de tres turnos, siendo estos de mañana, tarde y noche dando un servicio de 24h.
* Sistema continuo: Es un servicio 24h que no se interrumpe en fin de semana. Dando un servicio completo todo el año. Estos a su vez pueden ser fijos o rotativos.
  + Fijos: Es el mismo turno a lo largo de la semana. Permite una fácil adaptación pero puede ser un inconveniente a la hora de compaginar la vida familiar si se tienen turnos de tarde.
  + Rotativos: Los turnos varían a lo largo de la semana. Esto supone una perturbación en la vida social de la persona por tener horarios irregulares.

El sistema que esta aplicación tiene más en cuenta es el sistema continuo rotativo. Supone el sistema más irregular y donde se pueden producir más cambios.

La referencia usada para desarrollar diseñar este proyecto ha sido principalmente el entorno laboral de la sección de enfermería dentro de un hospital. Por lo general para organizar los turnos la persona responsable del grupo de enfermería de una planta pone a disposición de los trabajadores una hoja de papel donde se indican los horarios de cada individuo. Si se realizan cambios muchas veces se hacen sobre la hoja ya existente y esto las hace más difíciles de leer.

Además solo se podría acceder a ella en persona o haciendo una foto y distribuirla, el mayor inconveniente vuelve a ser cuando se realizan cambios, para estar consciente de ellos es necesario tomar una nueva foto y volverla a distribuir o estar en persona.

Esta aplicación pretende agilizar el proceso de notificación de los cambios de turnos y ayudar a tener una única versión del calendario. A través de una base de datos en tiempo real los usuarios podrán estar actualizados a la última versión de forma instantánea para que no haya confusiones por no haber sido avisados.

Otro proceso que podría ser agilizado sería el de cambiar turnos entre los usuarios, al usar la aplicación como medio se podrían solicitar cambios de turno, aceptarlos y ser aprobados por el coordinador sin necesidad de contactarse directamente.

## Estructura de la memoria

El contenido y la estructura de este documento son de la siguiente forma:

1. **Introducción**: Descripción del origen del proyecto y del problema que pretende solucionar,
2. **Objetivos**: Explicación de los principales objetivos que se tratan de conseguir y mediante que funcionalidades se intentan resolver.
3. **Conceptos teóricos**: Se describen los conceptos necesarios para la correcta compresión de este proyecto.
4. **Técnicas y herramientas**: Se detallan y exponen técnicas y herramientas usadas para el desarrollo del proyecto así como bibliotecas y la justificación de su uso.
5. **Aspectos relevantes del desarrollo**: En este apartado se detallaran aquellos aspectos destacables en cada fase del desarrollo del proyecto.
6. TODO\*\* Se presentan todas las funcionales de la aplicación en detalle
7. **Conclusiones y líneas de trabajo futuro**: Se comentan las conclusiones obtenidas a partir del desarrollo del proyecto además de las mejoras que podrían llevarse a cabo a continuación.

## Estructura de los anexos

Este proyecto también dispone de cinco anexos que se detallan a continuación:

1. Anexo I: Planificación temporal
2. Anexo II: Especificación de Requisitos del Software
3. Anexo III: Especificación de Diseño
4. Anexo IV: Documentación Técnica de Programación
5. Anexo V: Manuales de usuario

# Objetivos del proyecto

Este apartado recoge de forma detallada los objetivos planteados que la aplicación debe cumplir y pretende solucionar. Se distinguen los objetivos estrictamente funcionales dados por los requisitos del software y aquellos de carácter técnicos planteados por el alumno.

## Objetivos funcionales

Los objetivos funcionales indican las funcionalidades que debe tener la aplicación para que pueda dar solución a los problemas planteados. Estos objetivos se explican con más detalles en el Anexo II – Especificación de Requisitos del software.

Puesto que se trata de una aplicación donde los usuarios interactúan con el sistema a distintos niveles uno de los objetivos es la gestión de usuarios. Estos pueden registrarse en el sistema desde la aplicación usando un email y contraseña o una cuenta de Google. Posteriormente serán capaces de iniciar sesión para acceder al resto de funcionalidades. Si el usuario ha olvidado la contraseña dispondrá de un sistema de cambio de contraseña donde proveerá su dirección de email y se le enviará un email que le permitirá introducir una nueva contraseña. Para la gestión de usuarios se usa la plataforma Firebase, concretamente el módulo de gestión de usuarios Firebase Auth.

Una vez el usuario ha iniciado sesión correctamente es capaz de crear grupos de trabajo sobre los que se centraran la mayor parte del resto de funcionalidades. Al ser el creador del grupo actuará con el rol de coordinador del grupo. El coordinaron es la figura responsable que maneja la gestión de grupos de trabajo.

Una vez creado un grupo será posible acceder al calendario. Cada usuario dispondrá de un calendario personal dentro del grupo donde únicamente se muestran los turnos asignados a este usuario organizado por meses. Además dispondrá de un calendario grupal donde es posible acceder al horario del resto de usuarios y que se organiza por semanas. Esto le permite al usuario acceder a una visualización del horario laboral propio y de sus compañeros.

El rol de coordinador permite invitar a otros usuarios introduciendo su dirección de email. Si el usuario invitado aún no ha creado una cuenta con esa dirección, se le incluye en el grupo automáticamente en el momento en que la crea. Estos tendrán el rol de usuarios asociado dentro del grupo. Será posible modificar el nombre con el que aparecen estos usuarios además de eliminarlos. De esta manera el coordinador es el responsable de la gestión de los trabajadores.

Para poder asignar turnos a los usuarios, al coordinador se le permitirá la gestión de tipos de turnos, de forma que primero deberá definir los tipos de turno desde una ventana de configuración. Se le pedirá el nombre del turno, una etiqueta identificadora junto con un color para mostrarlo en el calendario además de la hora de inicio y la hora de fin del turno. Estos tipos de turnos serán editables después de su creación así como eliminarlos.

Una vez se han incluido los trabajadores en el grupo y se han definido los tipos de turno el coordinador será capaz de asignar turnos a los usuarios directamente desde el calendario grupal siendo posible asignar el mismo turno a la vez a lo largo de la semana a un usuario. Estos turnos serán posteriormente modificables siendo capaz de cambiar el tipo de turno o transferirlo a otro usuario además de eliminarlo.

Para respetar los derechos de los trabajadores y ayudar al coordinador a organizar los turnos, es posible definir un límite de horas semanales usado para avisar al coordinador cuando las sobrepasa a la hora de asignar turnos. Además los usuarios son capaces de acceder al recuento de horas asignado en cada mes.

El principal atractivo de la aplicación es que una vez este proceso se efectúa, los cambios en el horario se reciben en el cliente del usuario en tiempo real. Esto es posible a través de la plataforma de Firebase de base de datos en tiempo real Cloud Firestore, utilizada como base de datos de la aplicación. De esta forma los usuarios sincronizan sus dispositivos y obtienen su horario instantáneamente junto con una notificación push que les avisa de estos cambios en el calendario.

Con los turnos asignados los usuarios dispondrán de un sistema de solicitud de cambios de turnos donde un usuario solicita el cambio de un turno suyo por un turno de un compañero. Este proceso se hará en tres partes siendo estas la solicitud, la aceptación del otro usuario y la aprobación del coordinador que podrá rechazar el cambio aunque los dos usuarios estén de acuerdo. En cada fase se utilizaran notificaciones para informar a los usuarios del estado de la solicitud y para requerir su intervención para resolverla.

Como se ha comentado la aplicación dispondrá de notificaciones push donde se le indicará al usuario cuando han modificado su horario y durante el proceso de cambio de turno.

## Objetivos personales

En un proyecto del tamaño del Trabajo de Fin de Grado surgen una serie de objetivos personales, especialmente si no se ha realizado algo similar en el pasado.

* Aplicar los conocimientos adquiridos durante el grado, así como la mentalidad metódica propia de una ingeniería.
* Llevar a cabo un proyecto completo de principio a fin mediante una planificación temporal a largo plazo para nuestra experiencia y dando como resultado a un producto funcional y usable.
* Aprender a desarrollar para dispositivos móviles, concretamente para el sistema operativo Android. Así como aprender un nuevo framework.
* Poner en práctica y aprender conocimientos más avanzados en el lenguaje de programación Java tratado en algunas asignaturas durante el grado. Además de usar otros lenguajes como XML(eXtensible Markup Language o Lenguaje de Marcado Extensible) y Javascript.

# Conceptos teóricos

## Arquitectura Cliente – Servidor

La aplicación se basa en una arquitectura cliente servidor donde el cliente es la propia aplicación, sin embargo para su correcto funcionamiento necesita la base de datos y el servicio de autenticación que se alojan en la plataforma Firebase, un servicio de computación en la nube propiedad de Google que es gratuito en una pequeña escala.

La arquitectura cliente – servidor consiste en una comunicación con dos roles diferenciados que son como su nombre indica son el cliente y el servidor. Aunque esta comunicación puede darse en la misma máquina cuando mejor se aprovecha es al aplicarse a una red de máquinas a través de una red local o de internet.

La parte del cliente realiza peticiones al servidor de forma que este entienda y sepa procesarlas para que después este devuelva una respuesta acorde a la solicitud.

## Android y Android SDK

Android es un sistema operativo orientado a dispositivos móviles. Está presente en multitud de dispositivos de distintas marcas y en España tuvo una cuota de mercado del 86% en el primer trimestre de 2018.

Android está basado en el kernel de Linux, lo que lo hace de código abierto y gratuito. El lenguaje de programación para desarrollar aplicaciones en Android es Java junto con XML para definir las vistas. También es posible usar Kotlin, lenguaje creado por JetBrains al igual que IntelliJ IDEA el entorno de desarrollo sobre el que se basa Android Studio que es el entorno oficial para desarrollar aplicaciones en Android.

En Ilustración 1 - Arquitectura de capas del sistema Android es posible observar la arquitectura en capas del sistema Android.

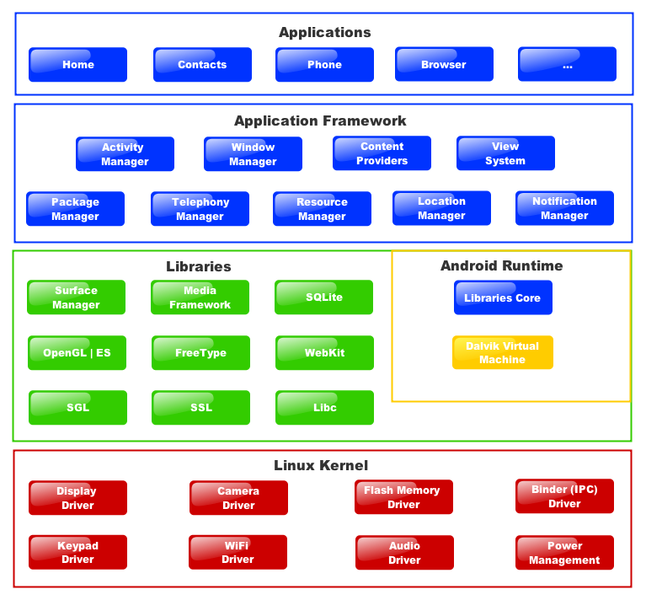


Ilustración 1 - Arquitectura de capas del sistema Android

Para poder desarrollar aplicaciones en Android es necesario obtener el Android SDK (Software Development Kit) que consiste en un conjunto de herramientas para el desarrollo de aplicaciones incluyendo un depurador, bibliotecas, un emulador para simular distintos dispositivos, documentación, ejemplos de código y tutoriales.

El SDK soporta versiones antiguas de Android para alcanzar a un mayor número de consumidores mediante la biblioteca llamada Android Support.

## Base de datos no relacional

Las bases de datos no relacionales también llamadas NoSQL se caracterizan por no tener un esquema rigido en los registros de los datos, de forma que cada registro puede tener campos diferentes a los demás. Esto significa que si un campo no es necesario en un determinado registro, este no aparece aportando claridad al no tener campos vacíos.

Escalan de forma horizontal aumentando el número de nodos (servidores) para incrementar el rendimiento del sistema. Muchos sistemas NoSQL permiten usar consultas tipo Map-Reduce, las cuales pueden ejecutarse en todos los nodos a la vez paralelizando la consulta.

Son sistemas que realizan operaciones directamente en memoria, volando los datos a disco cada cierto tiempo. De esta forma las operaciones son rápidas aunque tiene el inconveniente de que si el servidor falla y se cae se pierdan información y consistencia. Para solventar esto normalmente se requiere que otro servidor haya realizado la operación antes de validarse.

Este tipo de base de datos se expresa en JSON (JavaScript Object Notation) o puede transformarse fácilmente a objetos JSON.

En general son bases de datos con buen rendimiento y flexibilidad, ideal para usarse en tiempo real. Cloud Firebase, la base de datos usada en este proyecto es de este tipo.

## Material Design

Material Design es el estilo de diseño oficial de Android actualmente. Esto fue anunciado durante la conferencia de Google I/O de 2014 e integrado desde Android 5.0 Lollipop ofreciendo una serie de herramientas para poder implementarlo.

Este diseño se basa en objetos materiales que son piezas colocadas en un espacio y con un movimiento determinado. Las principales características distintivas son el uso de la profundidad, superficies, bordes, sombras y colores.

Las animaciones también juegan un papel principal dando una sensación de fluidez y llamando la atención hacia la información más importante.

Este proyecto ha intentado utilizar este estilo en su diseño de la interfaz de la aplicación utilizando las bibliotecas oficiales ofrecidas. En la Ilustración 2 - Estilo Material Design se puede observar un ejemplo.

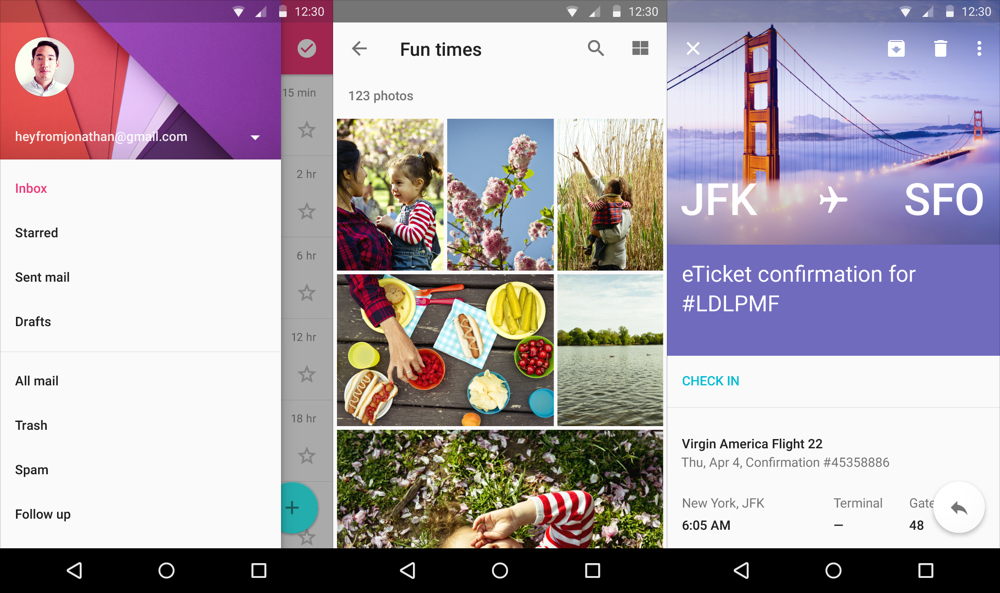


Ilustración 2 - Estilo Material Design

# Técnicas y herramientas

En este apartado se exponen todas las herramientas que han participado en el proyecto, así como lenguajes de programación usados, plataformas además de las técnicas y metodologías usadas en el desarrollo.

## Programación orientada a objetos

La programación orientada a objetos es el paradigma de programación más popular actualmente. Permite construir un sistema expresándolo de una forma cercana a la realidad a través del concepto de objetos. El lenguaje Java utilizado para desarrollar aplicaciones usando el Android SDK es un lenguaje orientado a objetos.

Un objeto es una entidad que contiene en un mismo componente datos y procedimientos de una forma lógica y cercana al mundo real. Los objetos con la misma estructura de datos y comportamiento se agrupan en una clase y se forman a partir de de esta. Se dice que un objeto es la instancia de una clase.

Algunos de los conceptos más importantes son:

* Objeto: entidad que posee un conjunto de propiedades o atributos (datos) y de funcionalidades o procedimientos (métodos). Se corresponden con objetos reales o con conceptos internos.
* Clase: Define las propiedades y comportamiento de un tipo de objeto. Se dice que los objetos son instancias de una clase y se crean a partir de estas.
* Atributos: Datos asociados a un objeto o una clase que indica las características individuales de un objeto.
* Métodos: Algoritmos asociado a un objeto o una clase que se ejecuta tras recibir un mensaje (ser llamado). Indica que es lo que puede hacer un objeto, puede modificar los atributos del objeto o llamar a otros métodos.
* Herencia: Se da entre clases y consiste en definir una clase a partir de otra, de forma que la nueva clase a definir incluye los atributos y métodos de la clase heredada.

Las características más importantes de la programación orientada a objetos son:

* Abstracción: Consiste en centrarse en las características esenciales de un objeto y ocultar los elementos que no lo son. Esto es para poder simplificar los objetos y tener un mejor control sobre la complejidad
* Encapsulación: Consiste en reunir los elementos que pertenecen a una misma identidad para formar un componente unido de forma lógica y aumentar la cohesión en el diseño.
* Principio de ocultación: Cada objeto está aislado del resto de forma que sus características internas solo pueden ser modificadas desde el mismo objeto o a través de una interfaz como interactuar con otros objetos. Esto protege las propiedades del objeto de modificaciones indeseadas por otros objetos y así evitar interacciones inesperadas.

## Programas y aplicaciones

### Android studio

Es el entorno de desarrollo utilizado para este proyecto. Es el entorno oficial para el desarrollo de aplicaciones para Android y está basado en el entorno de desarrollo IntelliJ IDEA de JetBrains. La primera versión fue publicada en diciembre de 2014 con una licencia Apache 2.0.

Este entorno tiene una serie de funcionalidades destacables:

* Soporte de la construcción basado el Gradle, un sistema de automatización para la construcción de proyectos.
* Un emulador para poder lanzar la aplicación en distintos dispositivos.
* Un editor de vistas gráfico junto al editor de textos para XML.
* Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
* Plantillas para crear diseños comunes de Android y otros componentes.
* Instant Run que permite aplicar cambios sin reconstruir un nuevo APK (Android Application Package)
* Integración con el sistema de control de versiones Git

Un proyecto en android studio se separa principalmente en las carpetas:

* **Manifests**: Contiene el fichero AndroidManifest.xml que proporciona información vital sobre la aplicación como el nombre, la clase principal Application, el icono a mostrar o el estilo visual general. También es donde se registran los permisos que necesita la aplicación.
* **Java**: Carpeta que contiene todo el árbol de paquetes que contienen el código fuente.
* **Res**: Contiene los recursos usados en la aplicación, es donde se definen las vistas usando lenguaje XML además de iconos, cadenas de texto o colores. Permite usar un recurso distinto dependiendo de diferentes factores como el lenguaje, la orientación del dispositivo o la versión de Android entre otros.

### Sublime Text 3

Se ha utilizado Sublime Text para programar en Javascript una serie de funciones ejecutadas en la nube de Firebase. Es un editor de texto que soporta multiples lenguajes de programación. Originalmente era una extensión del editor Vim y se dispone de una versión de evaluación gratuita sin fecha de caducidad que es la que se ha usado en este caso. Es desarrollado por Jon Skinner. Esta licenciado bajo copyright por Sublime HQ Pty Ltd.

### Doxygen

Es una herramienta utilizada para generar la documentación de un proyecto a partir del código fuente en diferentes lenguajes entre los que se encuentran C, C++, Objetive-C, C#, PHP, Java, Python, IDL, Fortran, VHDL y algunos otros. Fue lanzada en octubre de 1997 y desarrollada por Dimitri van Heesch. La documentación es generada en html permitiendo una navegación mediante referencias. Tiene una licencia GNU GPLv2.

### Microsoft Word

### Microsoft Project

### Visual Paradigm

## Lenguajes de programación

Para este proyecto se han usado distintos lenguajes de programación para las distintas áreas del proyecto.

### Java

El lenguaje de programación predilecto para desarrollar aplicaciones para Android usando el Android SDK es Java aunque es posible usar Kotlin. La aplicación soporta a partir de de la versión 5.0 Lollipop y por tanto la versión de Java utilizada es la 7.

Es un lenguaje de programación orientado a objetos multiplataforma altamente extendido. Se caracteriza por poder ser compilado una vez y que sea posible ejecutarse en distintas maquinas. Para esto se vale del JVM (Java Virtual Machine) que toma el bytecode generado por el compilador de Java y recompilándolo en el código nativo de la plataforma. Fue desarrollado por Sun Microsystems en 1982 y posteriormente adquirida en 2010 por Oracle.

### XML

En este proyecto se usara XML para definir todos los recursos usados por la aplicación, siendo estos las vistas, referencias a cadenas de textos, colores, iconos y otros elementos.

XML(eXtensible Markup Language) es un meta-lenguaje de marcado especificado por W3C (World Wide Web Consortium) utilizado para transportar datos y almacenar datos de forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos. Es un estándar utilizado para intercambiar datos entre diferentes plataformas que da soporte a bases de datos dando facilidades a la comunicación entre distintas plataformas.

Un ejemplo del uso de XML en la aplicación sería una de las vistas de la aplicación apreciable en Ilustración 3 - Vista fragment\_mycalendar.xml.



Ilustración 3 - Vista fragment\_mycalendar.xml

### JavaScript

En este proyecto se usa para programar una serie de funciones que se ejecutan en la nube de Firebase a partir de eventos en la base de datos.

Javascript es un lenguaje de programación orientado a la web que se puede aplicar a un documento HTML y proporcionar funcionalidad y contenidos dinámicos. Fue creado por Brendan Eich, co-fundador del proyecto Mozilla en 1995. Es un lenguaje muy flexible que permite realizar multitud de actividades desde actualizar una vista de forma dinámica a crear juegos y aplicaciones integradas basadas en base de datos.

Existen una gran cantidad de APIs (Application Programming Interface, Interfaz de programación de aplicación) para ampliar aún más la funcionalidad y permitir los desarrolladores integrarse con otros proveedores de contenido como Twitter o Facebook.

## Plataformas externas

El proyecto se ha apoyado en gran medida en la plataforma Firebase usando algunos de sus servicios que se comentarán a continuación además de usar Github como plataforma para alojar el proyecto y utilizar el control de versiones de Git.

### Firebase

Es una plataforma para el desarrollo de aplicaciones web y móviles. Soporta Android, iOS y aplicaciones web. Tiene su origen en la startup de James Tamplin y Andrew Lee den 2011 llamada Evolve. Proporcionaba una API para la integración de sistemas de chat en páginas web. Posteriormente incluyeron un sistema de arquitectura en tiempo real y fue cuando se fundó Firebase en Abril de 2012.

En ocubre de 2014 fue comprado por Google y ha integrado otros equipos como Divshot en 2015 y Tejido y Crashlytics en 2017.

Cuenta con una aplica documentación de sus servicios donde se documentan como incluir cada servicio en el proyecto y la información necesaria sobre cada una de las funciones de su API disponibles al desarrollador. También disponen de proyectos de ejemplo donde se muestran de una forma más práctica las posibilidades que ofrece. Poseen un canal de Youtube donde resuelven dudas planteadas y comentan como utilizar sus servicios además de que algunos de los propios desarrolladores se ocupan personalmente de resolver preguntas en la plataforma StackOverflow.com.

Se ha utilizado esta plataforma principalmente por ofrecer un servicio gratuito por debajo de un volumen de interacción determinado, por su servicio de base de datos en tiempo real y las facilidades que se dan al desarrollador para utilizarlas.

Además dispone de consola de control donde manejar todos los servicios usados por la aplicación como añadir y eliminar usuarios manualmente al igual que editar la base de datos, ver los registros de las funciones en la nube, enviar notificaciones manualmente y consultar los errores que experimentaron los usuarios.

Los servicios que ofrece que se han usado para desarrollar el proyecto son:

#### Firebase Auth

Se trata de un servicio de autenticación que proporciona almacenar datos esenciales de usuarios de forma segura y que admite distintas formas de autenticación como email y contraseña, teléfono, integración con Google Facebook y Twitter entre otros.

Dispone también de un sistema de verificación de direcciones de email utilizado en esta aplicación para que sea necesario confirmar que la dirección de email es válida antes de utilizar la aplicación.

Este servicio asigna un identificador único a cada usuario que será utilizado a través de toda la aplicación para distinguir y tratar con los usuarios.

#### Cloud Firestore

Se trata de una base de datos en tiempo real NoSQL flexible y escalable en la nube. Mantiene los datos sincronizados con los clientes a través de agentes de escucha en tiempo real y ofrece asistencia sin conexión a través de cachés. Permite una integración con otros servicios de Firebase como Cloud Functions.

Actualmente se encuentra en una fase beta y disponen de otro servicio de base de datos NoSQL en tiempo real llamado Realtime Database. La principal diferencia es la estructura de almacenamiento de los datos, en Realtime Database se utiliza una estructura JSON donde se debe evitar la anidación de datos, en Cloud Firestore se organizan en colecciones y documentos, una estructura similar a JSON pero que permite la anidación utilizando subcolecciones y simplificando la organización de los datos.

Dispone de restricciones de acceso de lectura y escritura usando un sistema de reglas, se ha limitado la base de datos para que los usuarios necesiten estar registrados.

En Ilustración 4 - Cloud Firestore se puede observar la estructura de la base de datos desde la consola de control.

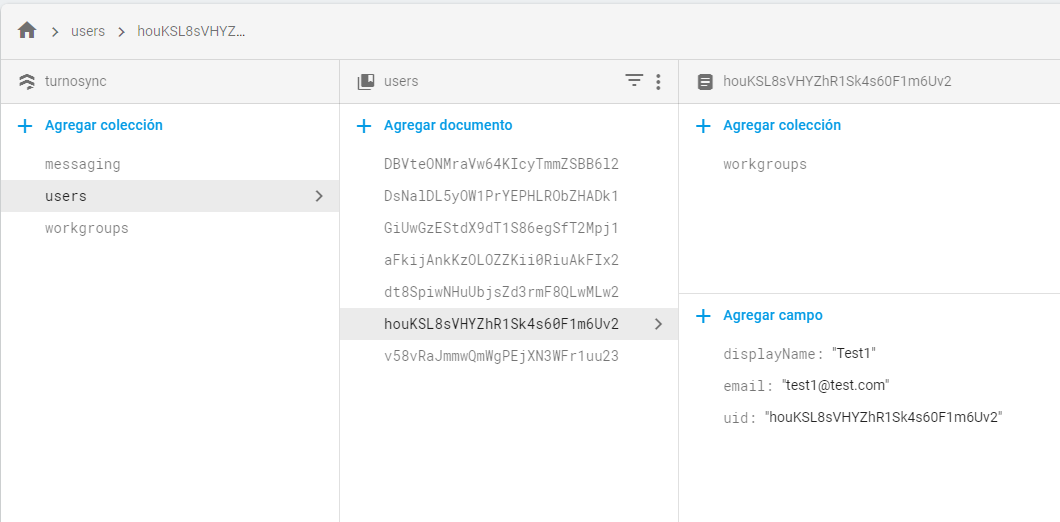


Ilustración 4 - Cloud Firestore

#### Cloud Functions

Permite ejecutar código en respuesta a eventos de otros servicios de Firebase o mediante solicitudes HTTPS. El código se almacena en la nube y se ejecuta en ella. Para desarrollar funciones es necesario inicializar el proyecto usando las herramientas de firebase del SDK de Cloud Functions. Se escribene en local y posteriormente En este proyecto se han utilizado activadores de Firebase Auth al crearse usuarios y con Cloud Firestore escribiendo en base de datos en la creación de usuarios y en la creación, borrado y actualización de datos en otras partes de la aplicación. También se ha utilizado Firebase Cloud Messaging para enviar notificaciones como respuesta a una modificación de la base de datos.

#### Firebase Cloud Messaging

Es la nueva versión de la plataforma llamada Google Cloud Messaging. Es una solución de mensajería multiplataforma que permite notificar una aplicación cliente. Se ha utilizado un sistema integrado con Cloud Firestore para enviar notificaciones cada vez que se le cambia el calendario a un usuario además de informar a los usuarios durante el proceso de cambio de turno.

### GitHub

Es una plataforma de alojamiento de proyectos que usa un sistema de versiones Git. Fue lanzado en 2008 y en junio de 2018 comprado por Microsoft. Los proyectos se almacenan de forma pública aunque con una cuenta de pago se pueden tener repositorios privados.

Github impulsa el código libre al alojar los proyectos de forma pública y facilita la colaboración entre desarrolladores. Dispone de sección de problemas que se usa normalmente para crear tickets por parte de usuarios donde se informa de los errores que puede tener un programa o aplicación y así ayudar a arreglarlos. Cuenta también con una sección de wiki donde el autor puede documentar el proyecto y otras facilidades como facilitar la elección de la licencia usada. También dispone de una sección con una tabla kanban para gestionar un equipo.

## Bibliotecas

Las bibliotecas usadas para desarrollar este proyecto han sido principalmente las que proporcionan distintos componentes de Android y las necesarias para usar los servicios de Firebase. Sin embargo para manejar fechas y horas se ha utilizado la biblioteca Joda Time que actúa como una alternativa al manejo de fechas y horas de Java 7.

### Joda Time

Es una biblioteca muy extendida de Java que pretende sustituir a las clases date y time de Java anteriores a la versión 8. En java 8 las clases referentes al tratamiento de fechas y horas se basaron en esta biblioteca.

Se ha usado esta biblioteca por la facilidad y la simplificación de su uso a la hora de tratar con fechas en la aplicación.

## Técnicas y metodologías

# Aspectos relevantes del desarrollo.

# TODO\*

# Conclusiones y líneas de trabajo futuro

# Bibliografía

* Turnos de trabajo. Emprendepyme.net – Actualizado el 21/11/2016 <https://www.emprendepyme.net/turnos-de-trabajo.html>
* ¿Qué es Android? Xatakandroid - Actualizado el 9 de febrero 2018 <https://www.xatakandroid.com/sistema-operativo/que-es-android>
* Cuota de mercado de Android. Kantar.com – Actualizado el 03/05 de mayo 2018 <https://es.kantar.com/tech/m%C3%B3vil/2018/mayo-2018-cuota-de-mercado-de-smartphones-en-espa%C3%B1a-primer-trimestre-2018/>
* Anroid SDK. Wikipedia – Actualizado el 09/10/2018 <https://es.wikipedia.org/wiki/Android_SDK>
* NoSQL. Genbeta.com – Actualizado el 28/04/2011 <https://www.genbeta.com/desarrollo/el-concepto-nosql-o-como-almacenar-tus-datos-en-una-base-de-datos-no-relacional?utm_medium=referral&utm_source=pulsenews>
* Material design – Actualizado el 09/11/2014 <https://elandroidelibre.elespanol.com/2014/11/que-es-material-design.html>
* Programación orientada a objetos. Wikipedia - Actualizado el 14/06/2018 <https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos>
* Android Studio. Developer.android.com – Actualizado el 25/04/2018 <https://developer.android.com/studio/intro/?hl=es-419>
* XML. Wikipedia – Actualizado el 20/02/2018 <https://es.wikipedia.org/wiki/Extensible_Markup_Language>
* JavaScript. Developer.mozilla.rog Actualizado 12/07/2018 <https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics>
* Firebase. Firebase.google.com – Actualizado en el 2018 <https://firebase.google.com/docs/android/setup?hl=es-419>
* GitHub.Wikipedia – Actualizado el 01/08/2018 <https://es.wikipedia.org/wiki/GitHub>
* Doxygen. Actualizado el 02/07/2018 http://www.stack.nl/~dimitri/doxygen/