**I.E.S Politécnico Jesús Marín**

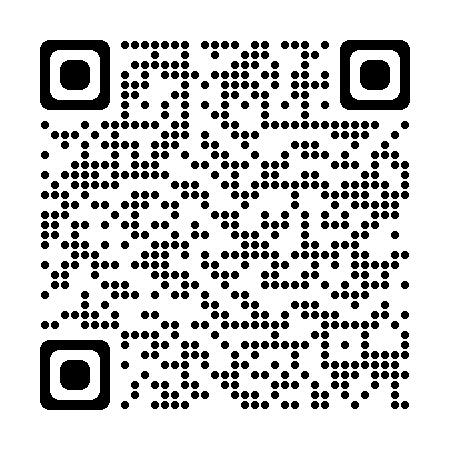
**DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**PROYECTO INTEGRADO**

Imagen que contiene dibujo

Descripción generada automáticamente

**STOCK**



**Departamento de Informática**

**Manual Técnico**

**Autor: D. Francisco Javier Zambrana Alarcón**

**ÍNDICE DE CONTENIDOS**

[**1 - Sobre éste proyecto**](#_nq0qhyoq4yjx) **3**

[1.1 - Control de versiones](#_rur0ht75osnc) 3

[1.2 - Licencia de uso](#_gqwjpia4w65r) 3

[**2 - Análisis del problema**](#_9qqdpa8pygkx) **3**

[2.1 - Introducción al problema](#_gpn9hsf0g51m) 3

[2.2 - Antecedentes](#_s8ywqgvt10r3) 3

[2.3 - Objetivos](#_8zwuwrlglmn7) 3

[2.4 - Requisitos](#_immbdcdy373h) 3

[2.4.1 - Funcionales](#_lqp8nreaojo4) 3

[2.4.2 - No funcionales](#_hyginiswjagz) 3

[2.5 - Recursos](#_3x7ty6r8xzcx) 3

[2.5.1 - Software](#_naj6cd1uwzx5) 3

[2.5.2 - Hardware](#_fb97ejy2epeq) 3

[**3 - Diseño de la solución software**](#_k06mzeo3hc41) **4**

[3.1 - Modelados](#_7q25992glvzh) 4

[3.1.1 - Casos de uso](#_48eeqmqqmsoo) 4

[3.1.2 - [ Interacción ]](#_c6pb4btxkzax) 4

[3.1.3 - [ Estado ]](#_c6pb4btxkzax) 4

[3.1.4 - [ Actividad ]](#_c6pb4btxkzax) 4

[3.2 - Base de datos](#_2wqcqskzd6yd) 4

[3.2.1 - Diseño Conceptual (ER)](#_8utgc2tqzb0) 4

[3.2.2 - Diseño lógico (tablas normalizadas)](#_8utgc2tqzb0) 4

[3.3 - Prototipado gráfico](#_u833q5srhqde) 4

[3.3.1 - [ Tablets / Smartphones ]](#_3hokoomf4ake) 4

[**4 - Implementación**](#_8p5gepj7iuu6) **4**

[4.1 - Codificación](#_3xqwqgkx4jto) 4

[4.1.1 - [ Usabilidad ]](#_ba8vimrg29ny) 4

[4.1.2 - Backend](#_vnafx7527xoa) 4

[4.1.3 - Frontend](#_userf15m80ds) 4

[4.2 - [ Pruebas ]](#_udmqodhaovl0) 4

[**5 - Documentación**](#_z6g6wwwyfw1w) **5**

[5.1 - Empaquetado / Distribución](#_t8fzvkglpyu) 5

[5.2 - Instalación](#_svl4kk7mx05w) 5

[5.3 - Manual de Usuario / Referencia](#_z779aftbhtke) 5

[**6 - Conclusiones**](#_nzhrwm20wdri) **6**

[**7 - Bibliografía**](#_iw4ugw9ge1ib) **7**

# 1 - Sobre este proyecto

## - Control de versiones

El control de versiones se ha llevado a cabo a través de un plugin de la plataforma GitHub aplicado a Android Studio, con este plugin cada cierto tiempo he realizado un commit del proyecto y posteriormente un push para que todo quedase subido a mi directorio:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En el siguiente pantallazo muestro los commit con los cambios que ha ido teniendo la aplicación hasta llegar a la versión definitiva:

Texto

Descripción generada automáticamente

Enlace a la dirección donde se encuentra el repositorio de Github de la aplicación (tambien se puede acceder a través del código QR de la portada) <https://github.com/zKisko/Stock>

Interfaz de usuario gráfica

Descripción generada automáticamente

# 2 - Análisis del problema

## 2.1 - Introducción al problema

El problema surge a raíz de la necesidad de tener controlado todo lo que es el mobiliario y el hardware de la oficina, y por eso, para tener un control más exhaustivo se ha pensado en crear una aplicación para poder llevar un conteo del estocaje qué hay en la oficina de la empresa.

## 2.2 – Antecedentes

No existen antecedentes de ninguna aplicación en la empresa que controle este stock, de ahí y tambien por el aumento del teletrabajo en la empresa, lo que hace mas dificil tener el control total sobre todo lo que tiene la empresa, pudiendo tener más control en cuanto a lo que hay en la oficina físicamente pero no tanto en las “oficinas virtuales”.

## 2.3 – Objetivos

El objetivo es poder tener controlado de manera sencilla el stock de la oficina para así poder reponer los componentes que se rompan, extravíen o no se hayan proporcionado para así completar los puestos de trabajo que no tengan todas las herramientas suficientes.

## 2.4 - Requisitos

### 2.4.1 – Funcionales

Los requisitos funcionales en un principio, sin perjuicio de que se puedan implementar algunos más en un futuro, serán:

* Login: es la primera pantalla con la que nos encontramos, en ella el usuario se podrá logar, si es que cuenta con un usuario y contraseña o bien podrá registrarse obteniendo las credenciales para poder logarse posteriormente.
* Crear los puestos de trabajo: el usuario podrá crear los puestos de trabajo que necesite para llegar al número de puestos que tenga la oficina.
* Consultar en una vista general los puestos creados.
* Modificar puesto de trabajo: el usuario podrá modificar los elementos del puesto de trabajo.
* Editar el puesto de trabajo, tanto como cambiar el nombre como para añadir elementos
* Eliminar puesto de trabajo: el usuario podrá eliminar cualquier puesto de trabajo.
* Una vez creados los puestos de trabajo, dentro de ellos el usuario podrá:
  + - Crear, consultar un elemento del puesto de trabajo.
    - Consultar los puestos de cada puesto de trabajo.
    - Modificar los elementos creados con anterioridad.
    - Eliminar por completo cualquier elemento creado.

### 2.4.2 - No funcionales

Los requisitos no funcionales en un principio serán los siguientes:

* La aplicación será creada para un sistema operativo Android pudiéndose ejecutar en cualquier dispositivo que cuente con esta tecnología bien sean smartphones, tablets e incluso Smart TV´s.
* Los lenguajes empleados para programar la aplicación serán; Java para la parte lógica de la aplicación, XML para la parte visual en las diferentes pantallas con las que va a contar la aplicación para su uso, y sentencias SQL para la creación de la base de datos encargada de contener los datos de los puestos de trabajo y de los elementos que compongan dichos puestos de trabajo.
* El manejo de la aplicación será eminentemente táctil, aunque pudiéndose ejecutar en dispositivos como tablets o en Smart TV´s se procurará la interacción con otros dispositivos cómo podían ser ratones, teclados y mandos a distancia.
* Se procurará que sea una aplicación intuitiva y con respuesta rápida a las interacciones del usuario, y que no consuma muchos recursos del sistema.
* La aplicación tendrá un diseño agradable inspirado en el logotipo de la empresa y en sus colores corporativos.

## 2.5 - Recursos

### 2.5.1 – Software

El software empleado para esta aplicación es:

Android Studio Dolphin | 2021.3.1 Patch 1

Build #AI-213.7172.25.2113.9123335, built on September 30, 2022

Runtime version: 11.0.13+0-b1751.21-8125866 amd64

VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.

GC: G1 Young Generation, G1 Old Generation

Non-Bundled Plugins:

com.github.b3er.idea.plugins.arc.browser (0.32)

El SDK API Level 33, Android 12 y Builds Tool 32 que incluye el emulador de Android, y el emulador utilizado para probar el emulador será un Nexus 5X con API 32.

Apache Commons Collections 3.2.2.

Android Studio ha sido el software empleado para la práctica totalidad de la programación de la aplicación tanto para la parte lógica como para la visual, ya que es el software creado por la compañía Google para diseñar sus aplicaciones, además tambien lo he elegido ya que la programación de la aplicación se ha hecho en Java y XML, lenguajes con los cuales me podía sentir más cómodo ya que han sido los más utilizados durante los dos años cursados de DAM.

Ademas de Android Studio, usada de manera principal, han servido de apoyo otras aplicaciones como:

* Visual Studio Code: es un editor de código fuente, esta aplicación que no había usado antes, en este caso usándolo como apoyo a Android Studio ya que permite trabajar con diferentes lenguajes de programación en el mismo entorno.
* Figma: otra aplicación que no había usado anteriormente ha sido muy interesante aprender a usarla, al menos lo bastante para realizar un diseño ya que me sirvió mucho para saber lo que quería y como podría llegar a conseguirlo, aunque no se pueda hacer exactamente el mismo diseño en Android Studio, si ha sido bastante orientativo:

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

* DB Browser for SQLite: tampoco había usado está aplicación, su función entre otras es la de poder ver, editar, eliminar bases de datos creadas en SQLite. Me ha permitido comprobar la creación de una de las bases de datos de nuestra aplicación y como los datos se insertaban bien en las tablas y permanecían.

### 2.5.2 – Hardware

El hardware utilizado para la programación de la aplicación es un procesador AMD de ryzen 5 3600 montado en una placa base Gygabite, con una memoria RAM DDRIII de 16 gigas, tarjeta gráfica Nvidia 1050 4Gb de RAM y un sistema operativo Windows 10 10.0. Tambien en contado con dos monitores de 24 pulgadas, teclado y ratón inalámbricos.

La aplicación tambien ha sido probada en un dispositivo movil Huawei P20 pro con sistema operativo EMUI 12.

# 3 – Diseño de la solución software

## 3.1 – Modelados

### 3.1.1 – Casos de uso

Para el diseño del diagrama de casos de uso he utilizado la aplicación web <Https://lucid.app>:

Diagrama

Descripción generada automáticamente

|  |  |
| --- | --- |
| Caso de uso 1 | Pantalla inicial de Login |
| Actor | Usuario |
| Propósito | El usuario puede logarse |
| Descripción | Identificarse como usuario para poder iniciar la aplicación |

|  |  |
| --- | --- |
| Caso de uso 2 | Pantalla de Registro de usuario |
| Actor | Usuario |
| Propósito | El usuario puede registrarse |
| Descripción | Hacer un registro con las credenciales suficientes para poder completar un login |

|  |  |
| --- | --- |
| Caso de uso 3 | Listado de Puesto de trabajo |
| Actor | Usuario |
| Propósito | Interfaz principal de nuestra aplicación |
| Descripción | El usuario podrá crear, modificar, eliminar o consultar puestos de trabajo |

|  |  |
| --- | --- |
| Caso de uso 4 | Listado de elementos del puesto de trabajo |
| Actor | Usuario |
| Propósito | Interfaz en la que se muestran los elementos del puesto de trabajo |
| Descripción | El usuario podrá crear, modificar, eliminar o consultar elementos del puesto de trabajo |

|  |  |
| --- | --- |
| Caso de uso 5 | Creación de puesto de trabajo |
| Actor | Usuario |
| Propósito | Proceso para crear los puestos de trabajo |
| Descripción | El propósito de este uso es el de la creación de los puestos de trabajo de nuestra empresa |

|  |  |
| --- | --- |
| Caso de uso 6 | Modificación de puesto de trabajo |
| Actor | Usuario |
| Propósito | Proceso para la modificación de los puestos de trabajo |
| Descripción | Modificar los puestos de trabajo creados con anterioridad |

|  |  |
| --- | --- |
| Caso de uso 7 | Eliminación de puestos de trabajo |
| Actor | Usuario |
| Propósito | Elimina puestos de trabajo |
| Descripción | El usuario utilizará este caso de uso para eliminar los puestos de trabajo |

|  |  |
| --- | --- |
| Caso de uso 8 | Creación de elementos del puesto de trabajo |
| Actor | Usuario |
| Propósito | Crear los elementos de nuestro stock |
| Descripción | El usuario creará los elementos que componen cada puesto de trabajo de nuestro stock |

|  |  |
| --- | --- |
| Caso de uso 9 | Edición de elementos del puesto de trabajo |
| Actor | Usuario |
| Propósito | Editar los elementos del puesto de trabajo |
| Descripción | El usuario podrá editar los elementos de cada puesto de trabajo que han sido creados con anterioridad |

|  |  |
| --- | --- |
| Caso de uso 10 | Eliminación de elementos del puesto de trabajo |
| Actor | Usuario |
| Propósito | Eliminar los elementos |
| Descripción | Con esta opción el usuario puede eliminar los elementos de cada puesto de trabajo |

|  |  |
| --- | --- |
| Caso de uso 11 | Consulta de elementos del puesto de trabajo |
| Actor | Usuario |
| Propósito | Consultar los elementos |
| Descripción | El usuario podrá consultar los elementos que han sido creados anteriormente para cada puesto de trabajo |

## 3.2 – Base de datos

Para nuestra aplicación he decidido implementar dos bases de datos diferentes para diferenciar dos procesos, el de autentificación de usuario y el de guardado de nuestros puestos de trabajo y elementos.

La base de datos utilizada para el registro y la autenticación de los usuarios en nuestra aplicación ha sido la base de datos de la plataforma Google Firebase en su modalidad Firebase Authentication.

Firebase Authentication es un sistema de autenticación seguro y que nos ofrece múltiples opciones a la hora de identificar y de registrarse de nuestros usuarios, como son identificación por email, por cuenta de Google(@gmail), Facebook, Twitter, Github, Yahoo! etcétera.

He elegido Firebase Authentication para la autenticación y el registro de los usuarios de nuestra aplicación también porque la plataforma nos ofrece una administración de los usuarios total, pudiendo eliminar los usuarios, modificar cualquier campo introducido, la actividad en tiempo real de cada usuario, o reenviar contraseña directamente desde la interfaz web, entre otras.

En la elección también ha sido importante su integración en nuestro IDE Android Studio, a través de un plugin por el medio del cual, siguiendo los pasos que nos indican el asistente, podremos hacer desde la creación de la base de datos hasta la integración en nuestros archivos Java.

A continuación, mostraré el proceso llevado a cabo para integrar Firebase Authentication en nuestra aplicación y lo que nos ofrece el entorno web de Firebase la hora de administrar nuestros usuarios:

El asistente del plugin de Firebase de nuestro IDE Android Studio nos va haciendo una guía paso a paso de cómo realizar la autenticación a través de Firebase Authentication:

Texto

Descripción generada automáticamente

Lo primero que hice fue crear nuestro proyecto en la web:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteUna captura de pantalla de una computadora

Descripción generada automáticamente

El siguiente paso fue conectar nuestra aplicación con Firebase:

Logotipo

Descripción generada automáticamente con confianza baja

Y después añadí el SDK de Firebase Authentication a nuestra aplicación.

Una vez en la consola de Firebase, fui a la pestaña Authentication y una vez allí seguí los pasos para crear nuestra autenticación a través de email:

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Despues de haber habilitado el método de autenticación, creé la base de datos en la consola de Firebase:

Primer paso fue comenzar la base de datos en modo prueba:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

En el segundo paso hay que elegir la ubicación de nuestra nube en este caso se recomienda elegir la de Estados Unidos:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al pulsar el botón de habilitar en la pantalla anterior, vemos como en la siguiente pantalla ya me aparecía el nombre de nuestra base de datos y un mensaje de que ya había sido creada:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez que ya tenía la base de datos creada en la consola de Firebase Authentication, implementé las tres clases .java de nuestra aplicación en Android Studio:

LoginActivity: javaclass que valida o invalida el logeo para acceder a nuestra aplicación. MainActivity: javaclass que controla que la identificación sea verdadera.

RegisterActivity: javaclass que si no tenemos login recoge los datos de registro para crear el usuario para poder logarse en nuestra aplicación.

Como dije anteriormente, uno de los motivos importantes por el que elegí Firebase Authentication es que nos permite una administración total de los usuarios de nuestra aplicación desde la consola de Firebase, en los siguientes pantallazos se puede ver cómo podemos controlar que se ha ingresado correctamente el usuario a través de su correo electrónico y desde ahí podremos borrarlo, podemos mandarle una, podemos inhabilitar la cuenta para que no se pueda volver a usar, también podemos mandar un correo para él restablecimiento de contraseña, etc.:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

He utilizado SQLite para la creación de la base de datos encargada de contener los datos de los puestos de trabajo y de los elementos que compongan dichos puestos de trabajo, a su vez también se han implementado las tablas categoría y producto para los elementos, la cual nos sirven para llenar los spinner con el nombre del producto y la marca:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Para cada tabla se ha hecho una precarga de datos (mockData) para que al iniciar la aplicación se muestren elementos con los que interactuar:

Tabla

Descripción generada automáticamente con confianza media Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Imagen que contiene Gráfico

Descripción generada automáticamente Tabla

Descripción generada automáticamente

### 3.2.1 – Diseño Conceptual (ER)

Diagrama

Descripción generada automáticamente

## 3.3 – Prototipado gráfico

### 3.3.1 – [ Tablets / Smartphones ]

Estos son los Layouts utilizado, que forman el FRONTEND, para cada pantalla de nuestra aplicación:

Splashscreen y Login:

SPLASHCREEN LOGIN REGISTRO

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Pantalla principal para los equipos (Workstation):

ACTIVITY\_WORKSTATION CONTENT\_WORTKSTATION LIST\_WORKSTATION

Cuadrado

Descripción generada automáticamente con confianza baja  Forma

Descripción generada automáticamente con confianza media

FRAGMENT\_ADD\_EDIT\_WORKSTATION FRAGMENT\_WORKSTATION

Forma, Cuadrado

Descripción generada automáticamente Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

Pantallas para los elementos (Ítems)

ACTIVITY\_ITEMS CONTENT\_ITEMS ACTIVITY\_ADD\_EDIT\_ITEMSForma

Descripción generada automáticamente Captura de pantalla de computadora

Descripción generada automáticamente Forma

Descripción generada automáticamente

ADD\_EDIT\_ITEMS\_CONTAINER CONTENT\_ITEM\_DETAIL FRAGMENT\_ITEM

Forma

Descripción generada automáticamente Forma, Cuadrado

Descripción generada automáticamente Tabla

Descripción generada automáticamente con confianza media

FRAGMENT\_ITEMS\_DETAIL LIST\_ITEM\_ITEMS FRAGMENT\_ADD\_EDIT\_ITEM

Texto

Descripción generada automáticamente Forma, Cuadrado

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## 4 – Implementación

## 4.1 – Codificación

[**4.1.2 - Backend**](#_vnafx7527xoa)

El backend de la aplicación está formado por clases estructuradas en carpetas realizadas principalmente en lenguaje Java, salvo para las sentencias SQL de la base de datos

El código de nuestra aplicación va a ser entregado en un formato digital aparte de la memoria, de todas formas, voy a explicar un poco cómo se ha estructurado dicho código:

En la carpeta DB se incluyen todas las clases que actúan para la creación de las tablas en la base de datos:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

La carpeta ítems contiene todas las clases y los métodos correspondiente a las acciones que hacen los layout de la pantalla elementos y editar elementos, las cuales entre otras son la carga de la base de datos de los elementos creados, la eliminación de elementos creados, la creación de elementos, la modificación y guardado de la modificación de los elementos creados, etc.:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

En la carpeta login estructuramos la pantalla inicial de nuestra aplicación, la cual corresponde a la clase LoginActivity, donde se crean las variables email y contraseña, se verifican con la base de datos Firebase y si son correctas abre la actividad main de nuestra aplicación que es Workstation activity, la clase RegisterActivity hace algo parecido en cuanto a iniciar las variables necesarias para el registro controlar las que sean correctas y comunicar con la base de datos Firebase con la diferencia que en esta pantalla no inicia la aplicación sino que te vuelve a mandar a la página login activity para realizar el login correctamente:

Texto

Descripción generada automáticamente con confianza media

En la carpeta recursos tenemos la clase Java CommonVar, en ella se agrupan de forma pública variables comunes de la base de datos para poder usarla en otras clases:

Texto

Descripción generada automáticamente con confianza media

La clase ProductosCursorAdapter es la que nos ayuda a crear las vistas de la tabla Productos y movernos por ellas:

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

En la clase de Workstation contamos con una estructura bastante parecida a la anteriormente descrita para los ítems, pero algo menos compleja ya que en las Workstation se realizan menos acciones que para los ítems:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

La parte dedicada al FRONTEND está compuesta por los diferentes layouts (explicados más específicamente tanto en el apartado prototipado grafico punto 3.3.1, como en el manual de usuario 5.3) dichos layouts están programado en el lenguaje XML:

Texto

Descripción generada automáticamente con confianza baja

## 4.2 – [ Pruebas ]

Después de codificar gran parte de la aplicación y habiendo obtenido pruebas positivas, entiendo que, por una actualización del IDE, la aplicación deja de iniciarse, empiezo a hacer pruebas hasta encontrar una versión del IDE estable que me vuelva a dar seguridad de que la carga no fallará y empiezo a hacer pruebas:

Dejo el AndroidManifest.xml solamente con la actividad principal para poder lanzar la aplicación y de ahí ir limpiando los errores:



Comento el If del OnCreate, ya que en el debug nos indica un fallo en el ItemsActivity (que es nuestro Main) y el resultado es que ya lanza nuestra aplicación, aunque de momento sigue vacía y nuestro botón no hace nada:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Descomento el IF y vuelvo a hacer un debug para seguir intentando encontrar el error que hace que no cargue la aplicación, haciendo el debug me doy cuenta de que el fallo está en la clase ItemCursorAdapter, en concreto a la línea 51 y 52, estas líneas pertenecen a la biblioteca Glide de Android, que es la encargada de descargar y guardar en memoria cache las imágenes que se desean mostrar en nuestra aplicación:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



Comento las líneas de código que corresponden a la aplicación que hemos hecho de la librería Glide en nuestra javaclass ItemCursorAdapter y vuelvo a hacer un debug.

Lanzamos la aplicación y tanto en el debug como en el emulador ya vemos que no nos aparece el error que nos estaba dando la librería Glide, lanza la aplicación y ya nos aparece un Ítem que hemos cargado previamente en la base de datos y que el botón de añadir funciona:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Detectado el error en la librería Glide, de momento no vamos a utilizarla, si me da tiempo a poder resolver del todo este problema e insertar imágenes para nuestros Items, en un futuro intentaré incluirlas.

Otro error que nos surgía al cargar la aplicación era que al pulsar el objeto creado como ejemplo la base de datos para visualizar el monitor al pulsarlo la aplicación pues se cerraba, haciendo el debug nos aparece error en el auto content\_items\_detail:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Según el mensaje que nos da la consola de debug el error es que no existe o que no se encuentra ninguna vista llamada ContentItemDetail para el fragment ItemsDetailFragment.

Abro el layout ContentItemDetail y compruebo que hay un warning que nos dice que el ActivityItemsDetail actualmente no incluye este layout:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Vamos al layout ActivityItemsDetail y comprobamos que el layout incluido no es el que debería ser, Y esto es lo que nos da el error y lo modificamos:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Logotipo, nombre de la empresa

Descripción generada automáticamente

Con este cambio ya podemos comprobar lanzando la aplicación que al pulsar el elemento Monitor que hemos precargado en la base de datos, para poder comprobar la usabilidad de la aplicación, ya Funciona correctamente y se puede acceder a los detalles del elemento, y también podemos ver en la parte superior derecha los iconos que hemos incluido para editar y borrar:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Otro error que nos da en la vista principal es que solo aparece 1 de los elementos precargados en la base de datos:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Vamos al layout del activity\_items para ver cuál puede ser el fallo y vemos que en este layout se incluye el content\_items qué es el contenedor de todos los elementos que se van a mostrar en la actividad principal:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En el layout content\_items vemos que está desarrollado como un NestedScrollView, es por esto por lo que solo aparece el primer elemento ya que no hace bien el scroll que necesitamos, para que se visualicen todos los elementos precargados en la base de datos, lo cambio por un RelativeLayout y con esto solucionamos el problema del scroll y ya se ven todos los elementos de la base de datos:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Cuando ya tenía definida la vista donde se reflejan los puestos de trabajo y a través de ella se podía acceder a los elementos de cada puesto de trabajo para modificar dichos elementos nos ha surgido un error en el CursorAdapter, lo cual está creando un conflicto en nuestra Javaclass ItemsFragment:

Texto

Descripción generada automáticamente

Al pulsar en los equipos nos deberías de llevar a la vista elementos y es lo que hace, pero nos obliga a hacer un cierre de la aplicación debido al error:

Captura de pantalla de un celular

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Este fallo lo hemos solucionado de la siguiente manera:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

La variable la habíamos iniciado con un valor que usa Android en muchos de sus clases y estaba dando conflicto nano solucionado renombrándola con un subrayado antes de id (\_id) y ahora sí ya podemos volver a ver todas las acciones que hace nuestra esta aplicación

Captura de pantalla de un celular

Descripción generada automáticamenteCaptura de pantalla de un celular

Descripción generada automáticamenteCaptura de pantalla de un celular

Descripción generada automáticamente

# 5 – Documentación

## 5.1 – Empaquetado / Distribución

***Empaquetado:***

Para el empaquetado para la posterior distribución de nuestra aplicacion vamos a crear nuestro archivo .apk a través del IDE Android Studio, siguiendo los siguientes pasos:

1er. Paso:

Abrimos a la opción Build en la parte superior del IDE y pulsamos en la opción Generate Signed Bundle / APK:

Captura de pantalla con letras y números

Descripción generada automáticamente

2do. Paso:

En la siguiente pantalla el IDE nos da dos opciones para el empaquetado de la aplicación, elegiremos la opción de APK y le daremos a la opción de next:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

3er. Paso:

Relleno todos los campos que nos requieren, Key store path, creamos también el password del Key store path, el alias de la Key y también el password que va a tener nuestra Key, cuando se hayan rellenado todos los datos se pulsará el botón next:

Captura de pantalla de un celular

Descripción generada automáticamente

4to. Paso:

Creamos nuestro Key store path entrando en la opción create new y rellenando lo campos Y le damos a ok

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

5to. Paso:

En la siguiente pantalla nos van a dar dos opciones para la construcción de nuestro fichero .apk, escogeremos la opción reléase y le damos al botón Finish para terminar la creación del ejecutable de la aplicación:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

6to. Paso:

Después de realizar los pasos anteriores, Android Studio hace una compilación de nuestros archivos y nos informa de que nuestro .apk ha sido creado con éxito:

Texto

Descripción generada automáticamente

7mo. Paso:

El ejecutable de nuestra aplicación (.apk) se ha creado en la dirección de la carpeta que indicamos más arriba, los seleccionamos, lo renombramos con el nombre de nuestra aplicación y ya está listo para ser instalado en cualquier dispositivo Android.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

***Distribución:***

La distribución en un principio, y sin perjuicio de medios de distribución futuros como pueden ser Google Play o cualquier repositorio de aplicaciones Android on-line, al ser una aplicación pensada para la empresa en la que he hecho las prácticas se va a hacer directamente mandándole el ejecutable por el medio que ellos prefieran, también se le facilitará la clave es para que puedan acceder a la base de datos en él Firebase Authentication para que puedan manejar los usuarios registrados, además, el fichero .apk va ser incluido en mi repositorio de GitHub para que lo pueda descargar todo aquel al que yo le facilite el enlace.

## 5.2 – Instalación

La instalación se hace a través del ejecutable que hemos creado en el apartado anterior y cual hemos renombrado como Stock, Para la instalación nos el sistema al no ser una aplicación que se ha descargado desde Google Play nos lanzará un mensaje diciendo que si confiamos en la instalación de la aplicación ya que es de origen desconocido le daremos a que sí y el mismo sistema nos llevará a un apartado para que le demos permiso a la aplicación para terminar su instalación una vez dado este permiso se termina la instalación de la aplicación en nuestro sistema androide y ya puede empezar a utilizarse.

## 5.3 – Manual de Usuario / Referencia

Para comenzar a usar nuestra aplicación iremos al menú de nuestro dispositivo donde se encuentran instaladas todas las aplicaciones y pulsaremos en el icono de nuestra aplicación que hemos instalado previamente:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al presionar el icono se abrirá la pantalla donde se tendrá que, o bien logar utilizando el email y la contraseña con los que previamente se ha registrado y pulsando posteriormente el botón acceder, si aún no cuenta con un registro de usuario, debajo del botón acceder tendrá una opción qué es “Regístrate aquí” la cual abre la pantalla de registro:

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Una vez realizado el login accederemos a la pantalla principal de nuestra aplicación donde podremos agregar equipos con el botón que aparece en el margen derecho abajo de la pantalla y haciendo un pulsado largo podremos editar los puestos ya creados, además en la parte superior derecha contamos con un botón de información que nos aclara lo que se puede hacer en esta pantalla principal:

Diagrama

Descripción generada automáticamente

Botón de información

Haciendo clic en el icono

equipo podrás consulta,

editar, eliminar o añadir

los elementos del puesto

de trabajo.

Haciendo un pulsado largo

podrás editar el nombre

o eliminar el puesto.

Pulsa en este botón para añadir

puestos de trabajo.

Una vez dentro de la pantalla del puesto de trabajo creado se podrá consultar los elementos sí ya están creados, editarlos, eliminarlos pulsando sobre ellos, o bien crearlos pulsando el botón que se encuentra en la parte inferior derecha si no existiera, en la parte superior derecha podrás encontrar también un botón de información sobre esta pantalla, en la parte superior izquierda podrás encontrar una flecha para poder regresar a la pantalla anterior:

Flecha de retroceso Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja botón para obtener información

Haz clic en el nombre

del elemento para poder

editarlo o eliminarlo

Botón para agregar elementos

Al pulsar en el botón para agregar nuevos elementos se desplegará una pantalla donde podrás añadir los campos necesarios para la creación de dichos elementos, una vez añadido todos los campos hay que clicar el botón inferior derecho para que se guarde todos los campos y se termine la creación del elemento:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez creado un elemento, pulsando en él, nos aparecerá una pantalla para poder eliminarlo o editarlo, para eliminarlo pulsaremos el icono con forma de cubo de basura, y para editarlo pulsaremos el icono en forma de lápiz, el cual nos desplegará otra pantalla con los campos editables para poder modificarlos, una vez modificados pulsaremos el botón derecho inferior para guardar los cambios, esto nos devolverá a la pantalla del elemento y podremos comprobar que se han guardado correctamente los cambios, para volver a la pantalla de los elementos del puestos de trabajo pulsaremos en el botón superior izquierdo de retroceso en forma de flecha:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteFlecha de Flecha de retroceso retroceso Pulsa los spinners para seleccionar categoría y producto

Pulsa el icono Pulsa para

del lápiz para editar el

editar Pulsa el icono de campo

elementos de la papelera para borrar elementos

Cuando hayas editado los campos pulsa el botón para guardar los cambios

# 6 – Conclusiones

Mi conclusión sobre el proyecto es que ha sido muy motivador y satisfactorio, ya que he podido utilizar muchas de las materias impartidas durante los dos años de curso, también me ha permitido aprender de forma autodidacta el uso de bastantes herramientas así como programas que pienso que me serán de mucha utilidad en la nueva etapa laboral que quiero afrontar a partir de la obtención del título de grado superior de desarrollo de aplicaciones multiplataforma, ya que el desarrollo de aplicaciones móviles es un mercado que está en auge.

He comprobado que es importante dedicarle tiempo a la planificación y al diseño tanto de las estructuras como de los diseños.

***Posibles mejoras:***

A pesar de que pienso que se ha conseguido de manera satisfactoria el objetivo plasmado en la propuesta de anteproyecto de la aplicación, soy consciente de que se le puede sacar más partido y hacer modificaciones que la mejoren en un futuro (seguiré trabajando en ella ya que en mi primera aplicación y tengo mucha ilusión y muchas ganas de seguir mejorándola).

Algunas de esas mejoras podrían ser las siguientes:

Uno de los aspectos a incluir en sucesivas versiones y para hacer la aplicación más accesible (pero menos controlable, por eso no se ha hecho) es incluir la recuperación de contraseña desde la pantalla de login, en nuestra aplicación se ha pensado que esa recuperación de contraseña se haga solicitándosela al administrador de la base de datos Firebase Authentication para mayor control.

Crear búsqueda en la pantalla de Workstation, donde el resultado será un listado de Workstation que contenga el nombre del puesto de trabajo y los elementos que coincidan con la búsqueda.

Cuando no existe ningun elemento en la lista debería de mostrar un mensaje del tipo:

“no existen elementos”, igualmente en la pantalla de equipos.

Al final no pude resolver el problema de la librería Glide por falta de tiempo para poder mostrar una imagen de nuestros ítems, esto sería una mejora que podríamos incluir en esta aplicación para hacerla más atractiva visualmente.

# 

# 7 – Bibliografía

Para la realización del proyecto he tenido que consultar en bastantes sitios web tanto en sitios oficiales para la descarga de programas que necesitaba como para las dudas sobre cómo hacer lo que quería en dichos programas, además de vídeos tutoriales de YouTube, también he consultado en páginas web del tipo blogs para resolver problemas o aprender cómo hacer lo que tenía pensado para la aplicación.

Por otro lado, también he visitado webs de utilidades para el diseño y las interfaces, tanto como para sacar iconos como para hacer los diseños y poder tener una guía para plasmarlos en Android Studio y para los diagramas del uso y de la base de datos.

Adjunto las direcciones consultadas por ordenadas por modalidad:

[Download Android Studio & App Tools - Android Developers](https://developer.android.com/studio)

<https://developer.android.com/studio>

<https://developer.android.com/guide/>

<https://developer.android.com/reference/>

[Firebase console (google.com)](https://console.firebase.google.com/)

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

[Google Developers](https://developers.google.com/)

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

[Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/)

<https://stackoverflow.com/>

<https://www.youtube.com/> Canales de desarrolladores

<https://www.youtube.com/@pildorasinformaticas>

<https://www.youtube.com/@LaGeekipediaDeErnesto>

<https://www.youtube.com/@codigosdeprogramacion>

<https://www.youtube.com/@CristianDavidHenao>

<https://www.youtube.com/@mouredev>

<https://www.youtube.com/@Developeru>

<https://www.youtube.com/@AndroidCoding>

<https://www.youtube.com/@pluartz9595>

<https://www.youtube.com/@Firebase>

<https://www.youtube.com/@FiveCodSoftWare>

<https://github.com/>

The home for all developers

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

<https://gitlab.com/>

<https://m2.material.io/>

Material es un sistema adaptable de guidelines, components, and tools que respaldan las mejores prácticas de diseño de interfaces de usuario. Respaldado por código abierto, Material agiliza la colaboración entre diseñadores y desarrolladores, y ayuda a los equipos a crear rápidamente productos hermosos.

***Blogs:***

<https://pillsfromtheweb.blogspot.com/>

Tutorials and pills from the world of technology, and DevOps!

<https://devexperto.com/>

DEVEXPERTO POR ANTONIO LEIVA

<https://codigonautas.com/>

Descubre miles de códigos para ayudarte a aprender desarrollo web y móvil

<https://appgametutoriales.com/>

Tutoriales principalmente enfocados a la creación de aplicaciones móviles y vídeo Juegos.

<https://qastack.mx/programming>

Preguntas y respuestas para programadores profesionales y entusiastas.

<https://www.flipandroid.com/>

<https://tecnofaq.com/>

<https://www.adictosaltrabajo.com/2012/08/30/inicio-android-2/#032>

Primeros pasos en Android

[Herramientas para desarrollo de software y gestión de proyectos (atlassian.com)](https://www.atlassian.com/es)

<https://www.atlassian.com/es>

<https://elbauldelprogramador.com/>

<https://medium.com/@alexmendozaventura/android-primeros-pasos-con-glide-f26a9db303ee>

Android — Primeros pasos con Glide

***Buscadores***:

<https://es.search.yahoo.com/>

<https://www.bing.com/>

<https://es.wikipedia.org/wiki/>

***Utilidades:***

[Material Symbols and Icons - Google Fonts](https://fonts.google.com/icons)

<https://fonts.google.com/icons>

<https://www.flaticon.com/>

Free access 8.8M+ icons in PNG

<https://www.figma.com/>

<https://to-do.live.com/>

[EdrawMax Online | All-in-One Diagram Tool](https://www.edrawmax.com/online/es/)

<https://www.edrawsoft.com/>