

APLICACIÓN PARA LA COMUNICACIÓN ENTRE CENTROS EDUCATIVOS Y FAMILIAS

Anexo IV: Documentación técnica



VNiVERSIDAD
D SALAMANCA

Septiembre de 2023

Trabajo de fin de Grado

Tutor:

María Navarro Cáceres

Alumno:

Raúl Rodríguez Villanueva

Tabla de Contenido

| | |
|---|---|
| 1. INTRODUCCIÓN..... | 4 |
| 2. ESTRUCTURA DEL SISTEMA | 4 |
| 2.1. PAQUETE MODEL..... | 4 |
| 2.2. PAQUETE VIEW | 4 |
| 2.3. PAQUETE VIEWMODEL..... | 5 |
| 2.4. PAQUETE ADAPTER..... | 5 |
| 2.5. PAQUETE UTILS..... | 5 |
| 3. ESPECIFICACIÓN DE LA DOCUMENTACIÓN | 6 |
| 4. REFERENCIAS..... | 6 |

Tabla de Imágenes

| | |
|--|---|
| Imagen 1. HTML generado por JavaDoc..... | 6 |
|--|---|

1. INTRODUCCIÓN

En este anexo se va a recoger la documentación técnica de nuestro código software creado para el desarrollo de la aplicación móvil. La división de la estructura del código es la misma que la propuesta en el análisis del diseño.

Se va a dividir el anexo en los siguientes apartados:

- Estructura del sistema
- Especificación de la documentación

2. ESTRUCTURA DEL SISTEMA

Antes de comenzar con la estructura del sistema, podemos hablar de algunas características relevantes de nuestro proyecto, que pueden ayudarnos a entender mejor el mismo:

- El lenguaje utilizado para el desarrollo del código es Java, con una parte de interfaz gráfica en XML.
- El entorno de trabajo es Android Studio.
- El SDK de desarrollo es Android 33.
- La base de datos utilizada es Firebase Firestore, la cual es hosteada por servidores de Google.
- Todo lo relacionado con los usuarios se ha realizado mediante Firebase Auth.
- Para la serialización de objetos se ha utilizado la interfaz Parcelable.

Nuestro código va a estar dividido en diferentes paquetes dentro de uno principal. La subdivisión seguida es la que ya se ha comentado anteriormente, siguiendo el patrón de diseño MVVM (Model-View-ViewModel).

2.1. PAQUETE MODEL

En este paquete se encuentran las clases con el modelo de la aplicación:

- Actividad.java
- Exam.java
- Resultado.java
- User.java

2.2. PAQUETE VIEW

En el paquete de la vista se encuentran las clases Java de nuestra vista, aunque también tenemos las clases XML relacionadas con nuestra interfaz. Estas clases de la vista son cada una de las actividades que se han creado dentro de la aplicación, dentro de las cuales se van a inicializar los componentes de cada una de ellas:

- AddActivityActivity.java
- AddExamActivity.java
- AddResActivity.java

- CalendarActivity.java
- ContactActivity.java
- EducamosActivity.java
- ExamsActivity.java
- HomeActivity.java
- LoginActivity.java
- MainActivity.java
- NewsActivity.java
- RegisterActivity.java
- ResultsActivity.java
- SettingsActivity.java

2.3. PAQUETE VIEWMODEL

El paquete del ViewModel va a recoger todas las clases que harán de “controlador”, intermediando entre el modelo y la vista, siendo el encargado de almacenar en la base de datos todo lo que le proporciona el usuario a través de la interfaz gráfica:

- AddActivityViewModel.java
- AddExamViewModel.java
- AddResViewModel.java
- CalendarViewModel.java
- ExamsViewModel.java
- LoginViewModel.java
- RegisterViewModel.java
- ResultsViewModel.java
- SettingsViewModel.java

2.4. PAQUETE ADAPTER

Este paquete Adapter va a contener las clases que se van a encargar de presentar en cada uno de los RecyclerView de cada actividad, el contenido correspondiente, presentándolos uno a uno en función del número de ellos que se encuentren en la base de datos, dándoles un formato específico asociado con una interfaz XML:

- ActividadAdapter.java
- ExamAdapter.java
- ResultAdapter.java

2.5. PAQUETE UTILS

Este paquete de Utils solamente va a contener una clase, la cual se encarga de realizar diferentes funciones relacionadas con el calendario y con objetos Calendar:

- CalendarUtils.java

3. ESPECIFICACIÓN DE LA DOCUMENTACIÓN

Para generar la documentación técnica de nuestro proyecto he utilizado la herramienta JavaDoc, la cual crea documentación legible a partir de los comentarios hechos en el código, especificando para que es cada una de las clases, los valores de las funciones, lo que realizan, etc.

Esta documentación es muy útil de cara al futuro, ya que cualquiera que lo vea puede tener una primera idea de lo que se lleva a cabo dentro de cada uno de las clases de cada paquete.

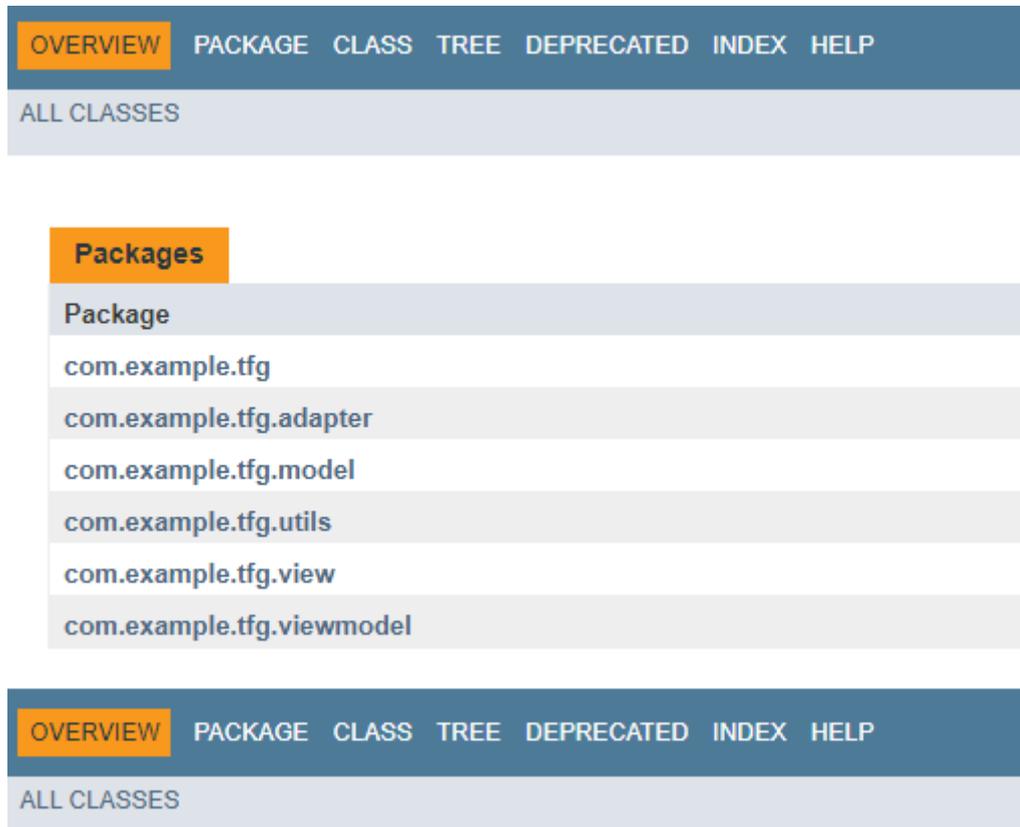


Imagen 1. HTML generado por JavaDoc

Este es el HTML generado, a partir del cual se puede ir navegando para conocer en detalle todo el código. Este HTML generado por JavaDoc se encuentra dentro de la carpeta de JavaDoc dentro de nuestro proyecto.

4. REFERENCIAS

Documentación de Firebase. (2023). Obtenido de <https://firebase.google.com/?hl=es>

Java. (2023). Obtenido de <https://www.java.com/es/>

Tutorial Javadoc. (2015). Obtenido de <https://stackoverflow.com/questions/29162820/how-to-create-javadoc-using-android-studio-without-r-and-buildconfig>