



**“PROTOTIPO DE APLICACIÓN MÓVIL DE SIMULACROS PREPARATORIOS DE
LAS PRUEBAS DE ESTADO SABER PRO y TyT - preSaber”**

Juan Carlos Higuita Jaramillo

**Corporación Universitaria Minuto de Dios
Rectoría Antioquia y Chocó
Sede Bello
Tecnología en Informática
Mayo de 2020**

**“PROTOTIPO DE APLICACIÓN MÓVIL DE SIMULACROS PREPARATORIOS DE
LAS PRUEBAS DE ESTADO SABER PRO y TyT - preSaber”**

Juan Carlos Higueta Jaramillo

**Trabajo de Grado presentado como requisito para optar al título de Tecnólogo en
Informática**

**Asesor(a)
Andrea Viviana Yanza Hurtado**

**Corporación Universitaria Minuto de Dios
Rectoría Antioquia y Chocó
Sede Bello
Tecnología en Informática
Mayo de 2020**

Resumen

Este proyecto busca llenar un vacío que se ha observado en la preparación de los estudiantes de la Institución Universitaria Minuto de Dios sede Bello, en adelante UNIMINUTO Bello, para las pruebas de estado Saber Pro ya que la mayoría de citaciones a exámenes preparatorios tipo simulacro, se realizan en un lapso de tiempo muy corto o no tienen buena acogida en los estudiantes debido al cruce con los horarios de estudio, trabajo y/o descanso. La universidad, en la actualidad, realiza la citación mediante un correo electrónico y en algunos casos no se conocen los resultados, por lo que una APP se hace necesaria con el fin de brindar una mejor preparación, ayudar a tomar medidas de acción y mejorar el conocimiento en los estudiantes.

Este proyecto ofrece un prototipo de aplicación móvil desarrollada en plataforma Android Studio, el cual permitirá un contacto directo entre las personas que intervienen en dicho test (Alumnos – Profesores – Área Académica) al momento de gestionar exámenes preparatorios para las pruebas de estado.

Palabras clave:

Aplicación móvil, Saber Pro, Android Studio, Kotlin

Abstract

This project seeks to fill a gap that has been observed in the preparation of the students of the Minuto de Dios University Institution, Bello headquarters, hereinafter UNIMINUTO Bello, for the Saber Pro state tests since most citations to mock preparatory exams, They are carried out in a very short period of time or are not well received by students due to the intersection with study, work and / or rest schedules. The university currently makes the appointment by email and in some cases the results are not known, so an APP is necessary in order to provide better preparation, help take action steps and improve the knowledge in students.

This project offers a prototype of a mobile application developed on the Android Studio platform, which will allow direct contact between the people involved in said test (Students - Teachers - Academic Area) when managing preparatory exams for the state tests.

Keywords:

Mobile app, Saber Pro, Android Studio, Kotlin

Tabla de contenido

Capítulo 1. Descripción del proyecto	9
Planteamiento el problema	9
Antecedentes	9
Justificación	10
Objetivos	11
Objetivo general	11
Alcance	12
Capítulo 2. Marco Teórico	13
Capítulo 3. Desarrollo de la propuesta	18
Metodología	18
Cronograma	19
Presupuesto	20
Capítulo 4. Resultados	21
Arquitectura de la aplicación	21
Capítulo 5. Conclusiones y Recomendaciones	34
Lista de referencias	35

Lista de tablas

Tabla 1 Cronograma	20
Tabla 2 Presupuesto	20

Lista de figuras

Ilustración 1 - Arquitectura	21
Ilustración 2 - Tabla Answer	22
Ilustración 3 - Tabla Image	22
Ilustración 4 - Tabla Module	23
Ilustración 5 - Tabla Question	23
Ilustración 6 - Tabla Result	24
Ilustración 7 - Tabla User	24
Ilustración 8 - Android Manifest	25
Ilustración 9 - Archivos PHP	26
Ilustración 10 – Librería format gson	27
Ilustración 11 – Pregunta en el simulador	28
Ilustración 12 – Campos requeridos a Google ID	29
Ilustración 13 – Librería firebas	30
Ilustración 14 – Datos importados de Google ID	31
Ilustración 15 – actividades principales	32

Introducción

En el ámbito estudiantil colombiano resulta inimaginable el día a día sin nuestro teléfono móvil y más para consumir servicios a través de sus diferentes aplicaciones. Debido a la aceptación por parte del consumidor final, cada vez más personas utilizan los múltiples servicios que se encuentran en las diferentes APP las cuales se pueden utilizar en los diversos equipos de telefonía celular. Esto también nos demuestra la buena aceptación y lo cambiante que pueden ser los diferentes métodos de estudio en la actualidad.

Para ello es importante unificar estos servicios – APP y teléfonos celulares – en las metodologías de estudio, ya que tendría cada vez más relevancia. Este proyecto, en este sentido, puede realizar un aporte a la preparación del estudiante y su enfoque con las Pruebas de Estado Saber Pro.

El documento esta conformado por 5 capítulos. En el primero se hace la descripción del proyecto, el segundo describe el marco teórico, el tercero se presenta el desarrollo de la propuesta de prototipo de aplicación, el cuarto nos muestra los resultados que se obtuvieron y por último, el quinto, muestra las conclusiones y recomendaciones a las que llegó el desarrollador.

Capítulo 1. Descripción del proyecto

Planteamiento el problema

En los últimos años la educación ha buscado diferentes alternativas como forma de aprendizaje, no solo se enfoca en ir a un salón de clase para aprender sobre un tema en específico sino en utilizar diferentes medios tecnológicos que ofrece el mercado actual y así mejorar el conocimiento y las destrezas en las personas.

Por lo anterior, este proyecto se ha creado con la intención de buscar un espacio de aprendizaje orientado a la preparación académica con un enfoque hacia las pruebas de estado que se realizan cada semestre en el territorio nacional.

Por esta razón se plantea desarrollar un prototipo de aplicación móvil la cual le permita a los estudiantes de Uniminuto Bello realizar simulacros preparatorios relacionados con las pruebas de estado Saber Pro, en la cual el estudiante selecciona la cantidad de preguntas que desea resolver (5, 20, 40, entre otras opciones).

Antecedentes

Internacional.

Duolingo: Es una aplicación móvil que le permite a las personas realizar diferentes test relacionados con el estudio de idiomas, tiene una forma de calificación. La empresa fue creada por el profesor guatemalteco Luis von Ahn.

Nacional.

ICFES Móvil: Es una aplicación móvil desarrollada por el ICFES, ofrece preguntas de las 5 áreas que se evalúan en las pruebas de estado. El ICFES es una entidad autónoma la cual se encuentra adscrita al Ministerio de Educación Nacional de Colombia.

Justificación

En el **Congreso Colombia 4.0** que se realizó en la ciudad de Bogotá, año 2016, y con la participación de varios expertos en la materia intervino Michael Mandel, Ph D de la Universidad de Harvard, manifestó que "El sector de las apps tiene un gran potencial en el país a largo plazo", además, "Los únicos insumos necesarios para el desarrollo de aplicaciones móviles son una mano de obra cualificada y buenas conexiones de telecomunicaciones, tanto a nivel nacional como internacional". (Revista Dinero, 2016)

A pesar de este buen pronóstico, en la actualidad solo se conoce la APP “El ICFES tiene un preicfes” la cual fue lanzada por el Instituto Colombiano de Educación Superior – Icfes – a inicios del mes de febrero del año 2019 y durante su lanzamiento la Directora General manifestó que “Es importante reiterar que lo que queremos desde el Icfes, es mostrar la forma cómo los estudiantes pueden acercarse a estos procesos”. (Canal Trece, 2020)

Existente otra aplicación como “Pruebas Saber” la cual fue desarrollada por CubiDevs o simuladores web como el de la Universidad Nacional el cual hace referencia a su examen de admisiones.

Objetivos

Objetivo general

Desarrollar un prototipo de aplicación móvil que permita realizar simulacros relacionadas con las pruebas de estado, SABER PRO y TyT, orientada a los estudiantes de la comunidad educativa UNIMINUTO Bello.

Objetivos específicos

1. Identificar las características de las Pruebas Saber Pro y TyT en cuanto a las competencias que se evalúan, tipos de preguntas y demás aspectos relevantes para la construcción del prototipo.
2. Realizar el análisis y diseño del prototipo de la aplicación (BD - arquitectura - patrones - lenguaje de programación).
3. Implementar un prototipo funcional de la aplicación según el diseño propuesto.

Alcance

La ejecución de este proyecto llega hasta lograr un prototipo funcional que permita la instalación del mismo mediante un archivo APK en los diferentes equipos de telefonía celular de los participantes del proyecto.

Proporcionar una APP que le permita a la universidad Uniminuto sede Bello mejorar sus planes de acción con miras a la formación para las pruebas de Estado Saber Pro.

Capítulo 2. Marco Teórico

En nuestro país, (Mineducación)

El Ministerio de Educación Nacional divide la educación virtual o a distancia en 3 generaciones:

- La primera generación se caracteriza por la utilización de una sola tecnología y la poca comunicación entre el profesor y el estudiante.
- La segunda generación introdujo otras tecnologías y una mayor posibilidad de interacción entre el docente y el estudiante. Además del texto impreso, el estudiante recibe casetes de audio o video, programas radiales y cuenta con el apoyo de un tutor (no siempre es el profesor del curso) al que puede contactar por correo, por teléfono o personalmente en las visitas esporádicas que éste hace a la sede educativa. En algunos casos cada sede tiene un tutor de planta para apoyar a los estudiantes.
- Por último, la tercera generación de la educación a distancia se caracteriza por la utilización de tecnologías más sofisticadas y por la interacción directa entre el profesor del curso y sus alumnos. Mediante el computador conectado a una red telemática, el correo electrónico, los grupos de discusión y otras herramientas que ofrecen estas redes, el profesor interactúa personalmente con los estudiantes para orientar los procesos de aprendizaje y resolver, en cualquier momento y de forma más

rápida, las inquietudes de los aprendices. A esta última generación de la educación a distancia se la denomina "educación virtual" o "educación en línea".

Un artículo publicado por la Universidad de Guadalajara, manifiestan que:

“Los simuladores constituyen un procedimiento tanto para la formación de conceptos y construcción de conocimientos, en general, como para la aplicación de éstos a nuevos contextos, a los que, por diversas razones, el estudiante no puede acceder desde el contexto metodológico donde se desarrolla su aprendizaje. De hecho, buena parte de la ciencia de frontera se basa cada vez más en el paradigma de la simulación, más que en el experimento en sí. Mediante los simuladores se puede, por ejemplo, desarrollar experimentos de química en el laboratorio de informática con mayor seguridad.

Previo al inicio del tema se definen los simuladores usados en educación como programas que contienen un modelo de algún aspecto del mundo y que permite al estudiante cambiar ciertos parámetros o variables de entrada, ejecutar o correr el modelo y desplegar los resultados (Escamilla, 2000). Hoy en día, las actuales tecnologías han cambiado al aparecer nuevos soportes, como el magnético y el óptico; la información ahora es digitalizada: se pasa del lápiz y el papel al teclado y la pantalla y, aún más, a la simulación (Rosario, 2005).

Aunque las investigaciones sobre simulación son todavía muy escasas, se pueden encontrar experiencias que desarrollan procesos de enseñanza-aprendizaje con simuladores; mediante la integración de las tecnologías de telecomunicaciones por computadora con instrumentación virtual se han desarrollado laboratorios de física disponibles para ingeniería y

accesibles a través de la red en tiempo real, lo cual asegura una rica experiencia de aprendizaje para el estudiante. Ellos toman en cuenta las limitaciones reales de los laboratorios, tales como el aprovechamiento de tiempo, los costos de instrumentación y los gastos de operación, la falta de personal, y la disponibilidad de laboratorio en horario diferente al de oficina (Macías, 2007).

En el mismo sentido, se puede encontrar que, en el área de la medicina, el crecimiento de la simulación ha sido significativo, ya que, tras una larga gestación, los últimos avances han puesto a disposición tecnologías que permiten la reproducción de eventos clínicos con suficiente fidelidad, para permitir la participación de los alumnos en una forma realista y significativa. Por otra parte, la importancia del trabajo en equipo interprofesional y de los enfoques de aprendizaje y la atención de la salud puede promoverse mediante el uso de ambientes simulados (Bradley, 2005).” (Contreras Gelves, García Torres, & Ramírez Montoya, 2010)

Para el desarrollo del prototipo de aplicación móvil se utilizaron las siguientes tecnologías:

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle
- Un emulador rápido y cargado de funciones

- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android
- Aplicación de cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra
- Variedad de marcos de trabajo y herramientas de prueba
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros
- Compatibilidad con C++ y NDK
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine (Developers)

Kotlin es un lenguaje de programación estáticamente tipado, es decir, es la máquina virtual la que infiere el tipo a las variables y, por tanto, no hay que especificarlas. Es un lenguaje que corre bajo la Máquina Virtual de Java, por lo que tiene el mismo rendimiento que Java, sin penalizar en ningún momento en este aspecto.

Está diseñado para interoperar con Java, por lo tanto podemos tener módulos programados en Java y otros módulos desarrollados en Kotlin. Estos módulos se podrían comunicar perfectamente sin problemas, simplemente tendrían que salvar el escalón que existe entre el desarrollo en Java y las premisas que tiene el desarrollo en Kotlin. (Lopez Jurado, 2018)

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. (Crockford)

Retrofit es un cliente REST para Android y Java, desarrollada por Square, muy simple y fácil de aprender. Permite hacer peticiones GET, POST, PUT, PATCH, DELETE y HEAD, gestionar diferentes tipos de parámetros y parsear automáticamente la respuesta a un POJO. (Sandoval Mondragón, 2017)

Capítulo 3. Desarrollo de la propuesta

Metodología

Para la realización de este prototipo se va a adaptar el modelo “Mobile-D”, según las necesidades del proyecto, el cual se basa en 5 fases de desarrollo: Exploración, Iniciación, Producción, Estabilización y Pruebas del Sistema, cada una compuesta de una serie de etapas, tareas y prácticas asociadas.

EXPLORACIÓN: Se define el alcance, se realiza la planificación inicial del proyecto y los requisitos del prototipo.

INICIACIÓN: Se preparan los recursos físicos, tecnológicos y de entrenamiento requeridos para el desarrollo del prototipo. Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

PRODUCCIÓN: Se hacen iteraciones de 15 días hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se desarrolla e integra con el código existente. El desarrollador tiene a su cargo realizar las pruebas unitarias de funcionamiento.

ESTABILIZACIÓN Y PRUEBA DEL SISTEMA: Se realizan acciones de integración para asegurar el correcto funcionamiento, es especialmente importante en proyectos multi-equipos. En la segunda, se hacen las validaciones necesarias para entregar una versión estable y plenamente funcional. Dado el alcance y las condiciones del proyecto, estas fases no se tendrán en cuenta.

Cronograma

Actividad	Fecha
Conocer el entorno de desarrollo de Android Studio y realizar pruebas de desarrollo en la plataforma web.	17-feb-2020
Realizar el levantamiento de requerimientos de la aplicación móvil.	24-feb-2020
Iniciar el diseño, en una hoja de papel, sobre las vistas y el código de colores que llevará la aplicación móvil.	02-mar-2020
Desarrollar la BD de la aplicación móvil.	09-mar-2020
Desarrollar aplicaciones de prueba para estructurar la versión final de la versión Beta que se va a entregar.	23-mar-2020
Iniciar el desarrollo de la aplicación móvil, con cada una de sus vistas y sus direccionamientos internos por cada una de las vistas.	30-mar-2020
Conocer acerca del lenguaje de programación Kotlin, sus funciones y como se debe compilar en Android Studio.	13-abr-2020
Estudiar el uso de las librerías Retrofit, servicios RESTful, inicio de sesión con Google ID.	27-abr-2020
Desarrollar la aplicación móvil en Android Studio.	04-may-2020
Conectar la aplicación móvil con la BD mediante objetos JSON.	18-may-2020
Realizar pruebas con la aplicación móvil.	25-may-2020

Tabla 1 Cronograma

Presupuesto

Recurso Humano				
Encargado	Rol	Valor hora	# Horas proyecto	Valor total
Juan Carlos Higueta Jaramillo	Desarrollador	\$ 10.000	84	\$ 840.000
Rodrigo Castrillo Restrepo	Asesor	\$ 25.000	30	\$ 750.000
SUBTOTAL			114	\$ 1'590.000
Recursos físicos				
Tipo	# Meses	Valor mensual	Valor total	
Internet	4	\$ 50.000	\$ 2000.000	
Servicios públicos	4	\$ 45.000	\$ 180.000	
Transporte	4	\$ 15.000	\$ 60.000	
SUBTOTAL			\$ 440.000	
Recursos software y hardware				
Tipo	Cantidad	Valor unitario	Valor total	
Equipo de cómputo	1	\$ 1'700.000	\$ 1'700.000	
SUBTOTAL	\$ 1'700.000			

Tabla 2 Presupuesto

Valor total del proyecto: \$ 3'730.00

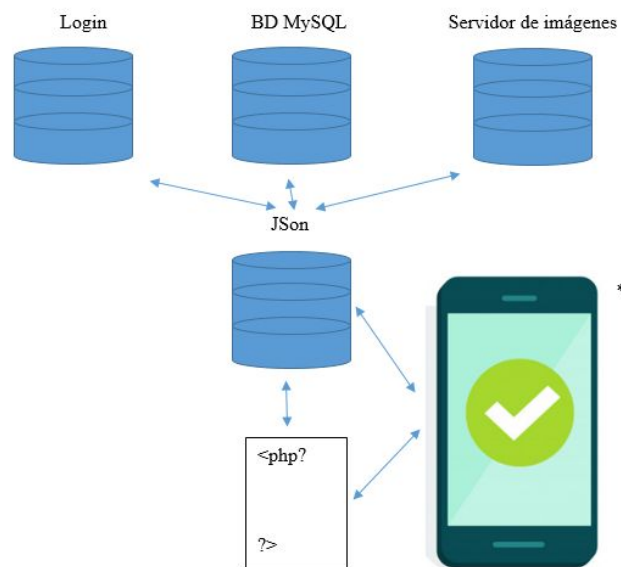
Capítulo 4. Resultados

El presente proyecto tuvo como finalidad diseñar y desarrollar un prototipo de aplicación móvil que permitirá realizar simulacros preparatorios a los estudiantes de Uniminuto Bello relacionados con las pruebas de estado Saber Pro.

Arquitectura de la aplicación

La aplicación móvil se implementará bajo la Arquitectura de microservicios o también llamada MSA (por sus siglas en inglés), permitiendo el acceso a la información que se encuentra almacenada en diferentes servidores, esta se gestionará mediante una API. (Microservices Grupo 10, 2016)

A continuación se muestra el modelo de la arquitectura:



* Imagen tomada de https://image.freepik.com/vector-gratis/telefono-movil-o-celular-marca-verificacion-pantalla-dibujos-animados-ilustracion-plana-vector_101884-381.jpg

Ilustración 1 - Arquitectura

Este proyecto requirió implementar una Base de Datos con la creación de unas tablas en formato PHP

Las imágenes describen los nombres de los campos que contienen cada una de las tablas, tipo de campo, clave primaria, clave foránea.

- Tabla answer: Esta tabla contiene las respuestas de las preguntas, se relaciona con la tabla *question* mediante un ID.



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1 Id 	int(10)			No	Ninguna
<input type="checkbox"/>	2 IdQuestion 	int(10)			No	Ninguna
<input type="checkbox"/>	3 Answer	varchar(200)	utf8_unicode_ci		No	Ninguna

Ilustración 2 - Tabla Answer

- Tabla image: Esta tabla contiene las imágenes que llevan algunas preguntas, las imágenes se encuentran ubicadas en una página web (<https://es.imgbb.com/>) y se gestiona mediante una URL.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Co
<input type="checkbox"/> 1	Id	int(10)			No	Ninguna	
<input type="checkbox"/> 2	Name	varchar(25)	utf8_unicode_ci		No	Ninguna	
<input type="checkbox"/> 3	Link	varchar(70)	utf8_unicode_ci		No	Ninguna	

Ilustración 3 - Tabla Image

- Tabla module: Esta tabla contiene los módulos a los que pertenece cada una de las preguntas.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Co
<input type="checkbox"/> 1	Id	int(10)			No	Ninguna	
<input type="checkbox"/> 2	Module	varchar(25)	utf8_unicode_ci		No	Ninguna	

Ilustración 4 - Tabla Module

- Tabla question: Esta tabla contiene las preguntas que debe responder el estudiante, contiene unos campos que se relacionan con las tablas module, image y answer mediante un ID.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Com
<input type="checkbox"/> 1	Id 🔑	int(10)			No	Ninguna	
<input type="checkbox"/> 2	IdModule 🔑	int(10)			No	Ninguna	
<input type="checkbox"/> 3	IdImage 🔑	int(10)			Sí	NULL	
<input type="checkbox"/> 4	Question	text	utf8_unicode_ci		No		

Ilustración 5 - Tabla Question

- Tabla result: Esta tabla contiene el resultado de las preguntas que han respondido los estudiantes. Se relaciona con la tabla *answer* y *user* mediante un ID.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Com
<input type="checkbox"/> 1	Id 🔑	int(10)			No	Ninguna	
<input type="checkbox"/> 2	IdUser 🔑	int(10)			No	Ninguna	
<input type="checkbox"/> 3	IdQuestion 🔑	int(10)			No	Ninguna	
<input type="checkbox"/> 4	IdAnswer 🔑	int(10)			No	Ninguna	
<input type="checkbox"/> 5	Date 🔑	date			No	Ninguna	

Ilustración 6 - Tabla Result

- Tabla user: Esta tabla contiene los datos que se exportan desde Google ID, con previa autorización del usuario.

The screenshot shows a database management interface with the following details:

- Server: localhost
- Database: id12478353_saberpro
- Table: user
- Views: Estructura de tabla (selected), Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/> 1	id	int(10)			No	Ninguna
<input type="checkbox"/> 2	idToken	varchar(80)	utf8_unicode_ci		No	Ninguna
<input type="checkbox"/> 3	givenName	varchar(40)	utf8_unicode_ci		No	Ninguna
<input type="checkbox"/> 4	familyName	varchar(40)	utf8_unicode_ci		No	Ninguna
<input type="checkbox"/> 5	displayName	varchar(40)	utf8_unicode_ci		No	Ninguna
<input type="checkbox"/> 6	email	varchar(60)	utf8_unicode_ci		No	Ninguna
<input type="checkbox"/> 7	photoURL	varchar(60)	utf8_unicode_ci		No	Ninguna
<input type="checkbox"/> 8	password	varchar(20)	utf8_unicode_ci		No	Ninguna

Ilustración 7 - Tabla User

Para permitir la interacción del prototipo de aplicación móvil con la BD se agregaron los permisos de verificación del estado de conexión de red y de acceso a Internet en el archivo *AndroidManifest.xml* de la carpeta Manifest del código Android, se creó una cuenta gratuita en un hosting web que permitiera la gestión de la información mediante archivos PHP

Las imágenes describen lo realizado:

- En el archivo ubicado en la carpeta Manifest se agregaron las siguientes líneas de código

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
```

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.unimuto.simulacro.saber.pro.activities">
4
5     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
6     <uses-permission android:name="android.permission.INTERNET" />
7
8     <application
9         android:allowBackup="true"
10        android:icon="@mipmap/ic_launcher"
11        android:label="UniminutoSaberPro"
12        android:roundIcon="@mipmap/ic_launcher_round"
13        android:supportsRtl="true"
14        android:theme="@style/AppTheme">
15        <activity android:name=".HomeActivity"></activity>
16        <activity android:name=".QuestionActivity"
17            android:screenOrientation="fullSensor"/>
18        <activity
19            android:name=".SplashActivity"
20            android:label="UniminutoSaberPro"
21            android:screenOrientation="fullSensor">
22            <intent-filter>
23                <action android:name="android.intent.action.MAIN" />
24
25                <category android:name="android.intent.category.LAUNCHER" />
26            </intent-filter>
27        </activity>
28        <activity
29            android:name=".LoginActivity"
30            android:label="Login" />
31        <activity android:name=".MainActivity" />
32    </application>
33 </manifest>

```

Ilustración 8 - Android Manifest

- El hosting gratuito que se utilizó para éste fin fue www.000webhost.com y en el cual se crearón una serie de archivos en formato PHP los cuales permiten gestionar la información consignada en la BD

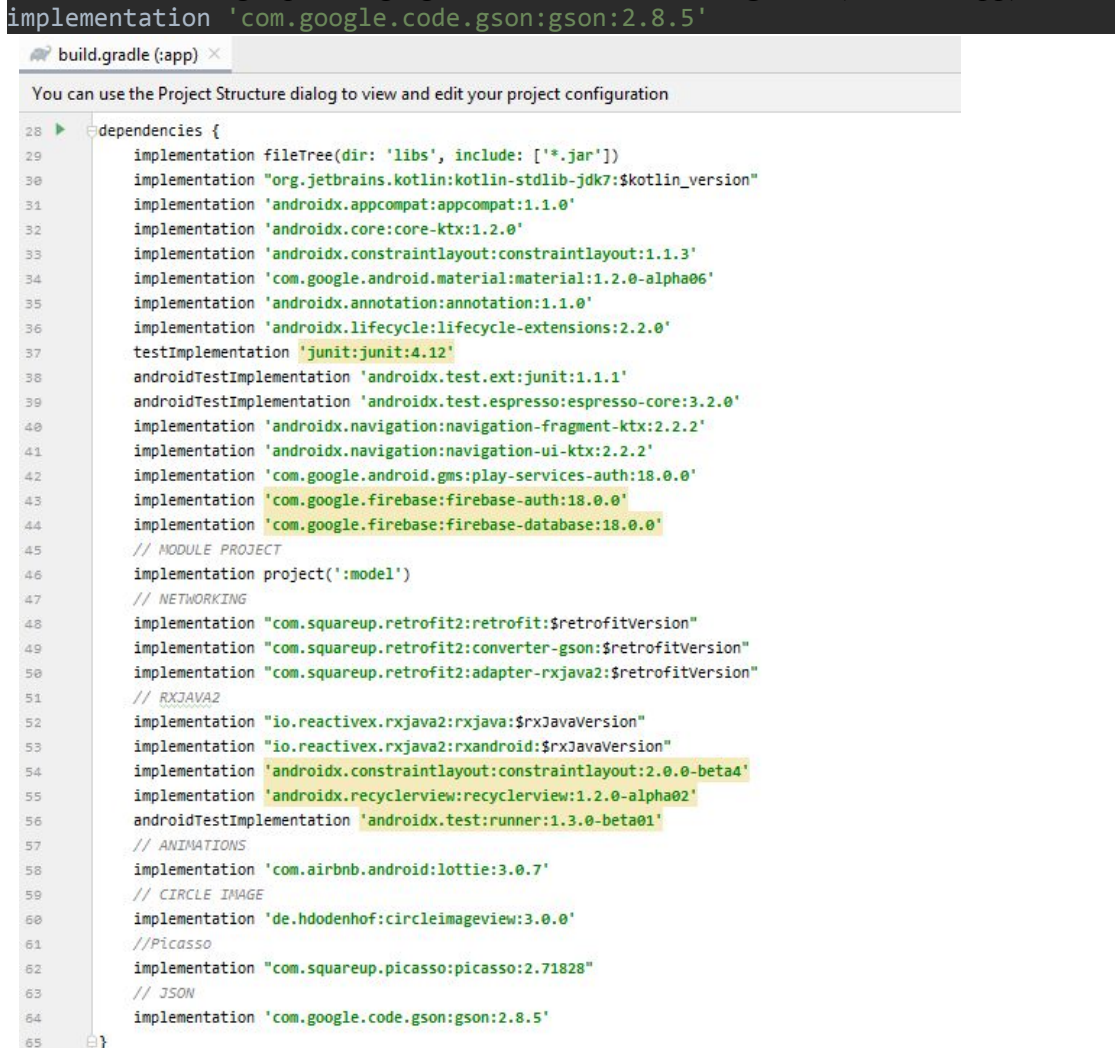
The screenshot shows a web hosting control panel interface. On the left, a file manager displays a directory structure with files: `index.php` (4.4 KB, 2020-05-21 02:22:00) and `utils.php` (0.9 KB, 2020-05-21 00:36:00). The `index.php` file is selected. On the right, a code editor titled 'Editar archivo' shows the PHP code for `/public_html/index.php`. The code includes database connection logic and a query to retrieve question data from a database table.

La respuesta del servidor remoto a la plataforma Android Studio fue retornada mediante objetos en formato JSON, para lo cual se requirió agregar líneas de código en el archivo build.gradle (Module: app) las cuales permiten cambiar la presentación de la información en el prototipo de aplicación móvil

Las imágenes describen lo realizado:






- Las líneas de código que se agregaron en el archivo *build.gradle (Module: app)*

```
implementation 'com.google.code.gson:gson:2.8.5'
```



```
dependencies {
28  implementation fileTree(dir: 'libs', include: ['*.jar'])
29  implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
30  implementation "androidx.appcompat:appcompat:1.1.0"
31  implementation "androidx.core:core-ktx:1.2.0"
32  implementation "androidx.constraintlayout:constraintlayout:1.1.3"
33  implementation "com.google.android.material:material:1.2.0-alpha06"
34  implementation "androidx.annotation:annotation:1.1.0"
35  implementation "androidx.lifecycle:lifecycle-extensions:2.2.0"
36  testImplementation "junit:junit:4.12"
37  androidTestImplementation "androidx.test.ext:junit:1.1.1"
38  androidTestImplementation "androidx.test.espresso:espresso-core:3.2.0"
39  implementation "androidx.navigation:navigation-fragment-ktx:2.2.2"
40  implementation "androidx.navigation:navigation-ui-ktx:2.2.2"
41  implementation "com.google.android.gms:play-services-auth:18.0.0"
42  implementation "com.google.firebase:firebase-auth:18.0.0"
43  implementation "com.google.firebase:firebase-database:18.0.0"
44  // MODULE PROJECT
45  implementation project(':model')
46  // NETWORKING
47  implementation "com.squareup.retrofit2:retrofit:$retrofitVersion"
48  implementation "com.squareup.retrofit2:converter-gson:$retrofitVersion"
49  implementation "com.squareup.retrofit2:adapter-rxjava2:$retrofitVersion"
50  // RXJAVA2
51  implementation "io.reactivex.rxjava2:rxjava:$rxJavaVersion"
52  implementation "io.reactivex.rxjava2:rxandroid:$rxJavaVersion"
53  implementation "androidx.constraintlayout:constraintlayout:2.0.0-beta4"
54  implementation "androidx.recyclerview:recyclerview:1.2.0-alpha02"
55  androidTestImplementation "androidx.test:runner:1.3.0-beta01"
56  // ANIMATIONS
57  implementation "com.airbnb.android:lottie:3.0.7"
58  // CIRCLE IMAGE
59  implementation "de.hdodenhof:circleimageview:3.0.0"
60  // Picasso
61  implementation "com.squareup.picasso:picasso:2.71828"
62  // JSON
63  implementation "com.google.code.gson:gson:2.8.5"
64
65 }
```

- El resultado final fue

#quedateencasa     74%  10:24 a.m.

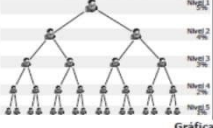
Simulacro saber pro

Razonamiento Cuantitativo

Un vendedor con una red de 5 niveles quiere duplicar sus ingresos mensuales. Para ello necesita

Una empresa tiene un sistema de ventas por redes, en el cual cada vendedor (Nivel 1) recibe comisiones por sus ventas, por las de vendedores que él haya inscrito (Nivel 2) y por las ventas de aquellos que fueron inscritos por las personas que inscribió (Niveles 3, 4 y 5).

Comisiones de un vendedor con una red de cinco Niveles



Tomado y adaptado de: <http://exceloienter-internet-marketing.com/mbm.html>

Pregunta de única selección

- que cada nivel duplique sus ventas.
- duplicar sus ventas.
- que cada vendedor del nivel 5 duplique sus ventas.
- duplicar el número de vendedores.

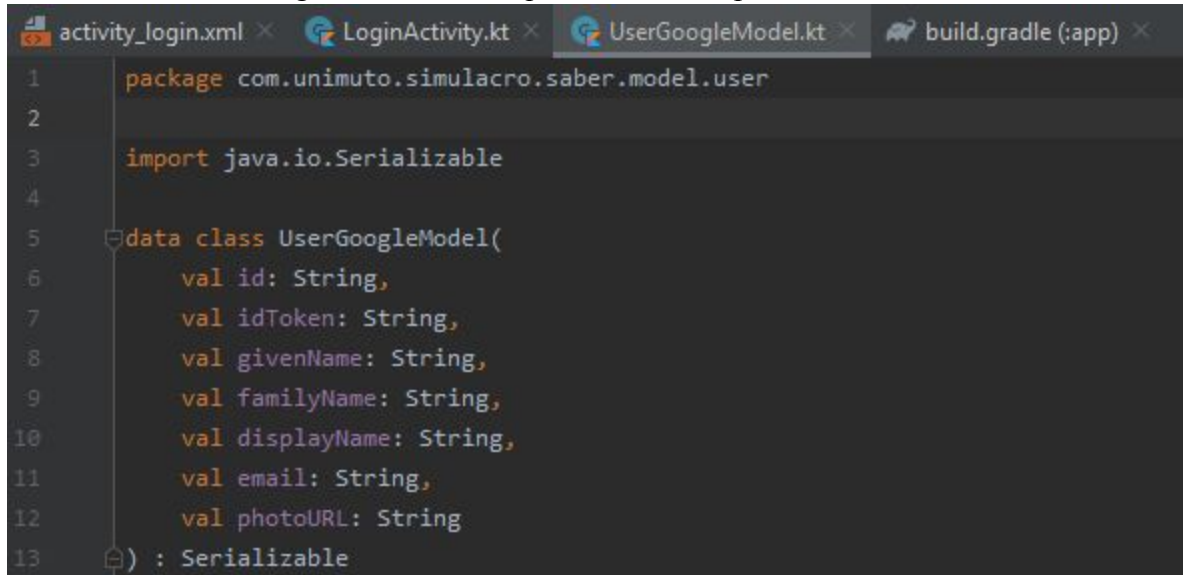
< Atrás 1/5 Siguiente >

Ilustración 11 – Pregunta en el simulador

Para el método de inicio de sesión se utilizó el login de Google ID el cual trae una información que se encuentra ubicada en los servidores de Google, con previa autorización, se toma la foto, el nombre y la dirección de correo electrónico

Las imágenes describen lo realizado:

- Se creó una vista de *Kotlin file / class* dentro de la carpeta raíz *model* -> *user* la cual hace la solicitud a Google con los datos que se desean importar



```
1 package com.unimuto.simulacro.saber.model.user
2
3 import java.io.Serializable
4
5 data class UserGoogleModel(
6     val id: String,
7     val idToken: String,
8     val givenName: String,
9     val familyName: String,
10    val displayName: String,
11    val email: String,
12    val photoURL: String
13 ) : Serializable
```

Ilustración 12 – Campos requeridos a Google ID

- Las líneas de código que se agregaron en el archivo *build.gradle (Module: app)* son:

```
implementation 'com.google.firebase:firebase-auth:18.0.0'
implementation 'com.google.firebase:firebase-database:18.0.0'
```

```
build.gradle (:app) ×
You can use the Project Structure dialog to view and edit your project configuration

28 dependencies {
29     implementation fileTree(dir: 'libs', include: ['*.jar'])
30     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
31     implementation 'androidx.appcompat:appcompat:1.1.0'
32     implementation 'androidx.core:core-ktx:1.2.0'
33     implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
34     implementation 'com.google.android.material:material:1.2.0-alpha06'
35     implementation 'androidx.annotation:annotation:1.1.0'
36     implementation 'androidx.lifecycle:lifecycle-extensions:2.2.0'
37     testImplementation 'junit:junit:4.12'
38     androidTestImplementation 'androidx.test.ext:junit:1.1.1'
39     androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
40     implementation 'androidx.navigation:navigation-fragment-ktx:2.2.2'
41     implementation 'androidx.navigation:navigation-ui-ktx:2.2.2'
42     implementation 'com.google.android.gms:play-services-auth:18.0.0'
43     implementation 'com.google.firebase:firebase-auth:18.0.0'
44     implementation 'com.google.firebase:firebase-database:18.0.0'
45     // MODULE PROJECT
46     implementation project(':model')
47     // NETWORKING
48     implementation "com.squareup.retrofit2:retrofit:$retrofitVersion"
49     implementation "com.squareup.retrofit2:converter-gson:$retrofitVersion"
50     implementation "com.squareup.retrofit2:adapter-rxjava2:$retrofitVersion"
51     // RXJAVA2
52     implementation "io.reactivex.rxjava2:rxjava:$rxJavaVersion"
53     implementation "io.reactivex.rxjava2:rxandroid:$rxJavaVersion"
54     implementation 'androidx.constraintlayout:constraintlayout:2.0.0-beta4'
55     implementation 'androidx.recyclerview:recyclerview:1.2.0-alpha02'
56     androidTestImplementation 'androidx.test:runner:1.3.0-beta01'
57     // ANIMATIONS
58     implementation 'com.airbnb.android:lottie:3.0.7'
59     // CIRCLE IMAGE
60     implementation 'de.hdodenhof:circleimageview:3.0.0'
61     // Picasso
62     implementation "com.squareup.picasso:picasso:2.71828"
63     // JSON
64     implementation 'com.google.code.gson:gson:2.8.5'
65 }
```

Ilustración 13 – Librería firebase

- Esta es la forma como retorna la información



Ilustración 14 – Datos importados de Google ID

La estructura del proyecto en Android Studio tiene 5 actividades principales:

- HomeActivity: Contiene la información del usuario que se registró mediante Google ID. Foto, nombre, correo.
- LoginActivity: Contiene el código fuente para el registro en el prototipo de aplicación mediante Google ID.
- MainActivity: Contiene la actividad principal en la que se muestran las preguntas

- **QuestionActivity:** Es la vista secundaria en la que se muestran las preguntas. De manera fija contiene el título de la vista, los botones Atrás y Siguiente, y la cantidad de preguntas que ha respondido de las que selecciono.
- **SplashActivity:** Contiene la animación con la que se inicia la aplicación.

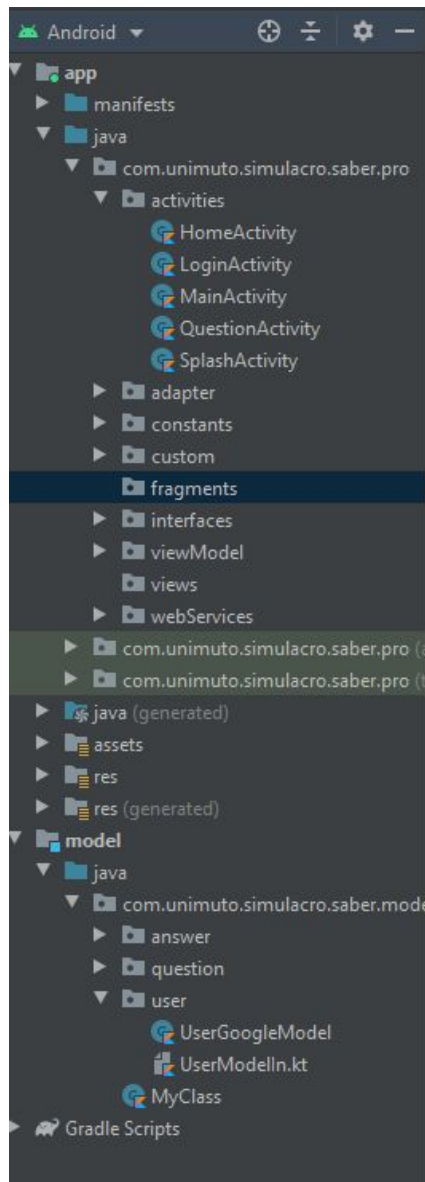


Ilustración 15 – actividades principales

Capítulo 5. Conclusiones y Recomendaciones

Conclusiones

- El levantamiento de los requerimientos fue fundamental para identificar cada una de las actividades que se iban a desarrollar en el prototipo de aplicación móvil.
- Durante el proceso de desarrollo del prototipo de la aplicación móvil en Android Studio fue necesario aprender sobre el “encarpetamiento” de las diferentes vistas y así tener un código más limpio, utilizar librerías Retrofit, hacer uso de arquitectura de software como RESTful API para concatenar los diferentes datos que se encontraban en cada una de las tablas de la BD y que se encontraban relacionadas.
- El atributo ScrollView es muy útil cuando se utiliza demasiada información en la vista ya que le permite al usuario desplazar la pantalla hasta llegar al final de la misma.

Recomendaciones

- Es indispensable tener un teléfono celular conectado para estar descargando el código ya que en algunas ocasiones se dificulta encontrar un error en la sintaxis del código debido a la cantidad de actividades que permite Android Studio.
- Fue muy útil realizar el ejercicio de dibujo acerca de la aplicación ya que da pautas de cómo deben ir las vistas finales.

Lista de referencias

- Cabezas, L., & González, F. (2018). *Desarrollo Web con PHP y MySQL Edición 2018*. Anaya Multimedia.
- Canal Trece. (21 de 02 de 2020). *Canal Trece*. Obtenido de <https://canaltrece.com.co/noticias/simulador-icfes-gratis-descargar-pc-examen-pruebas/>
- Crockford, D. (s.f.). *JSON.org*. Obtenido de <https://www.json.org/json-es.html>
- Darmawikarta, D. (2014). SQL for MySQL. Brainy Software.
- Developers. (s.f.). *Developers*. Obtenido de <https://developer.android.com/studio/intro?hl=es-419>
- García Holgado, A. &. (2015). *Estudio sobre la evolución de las soluciones tecnológicas para dar soporte a la gestión de la información. INFORME TÉCNICO, 8*.
- Lopez Jurado, F. (01 de 06 de 2018). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-kotlin/>
- Microservices Grupo 10. (07 de 11 de 2016). *Medium*. Obtenido de <https://medium.com/@grupo10.msa/arquitectura-de-microservicios-5ce65f70f980>
- Mineducación. (s.f.). *Mineducación*. Obtenido de https://www.mineducacion.gov.co/1759/w3-article-196492.html?_noredirect=1
- Padilla Beltran, J. E., Vega Rojas, P. L., & Rincón Caballero, D. A. (2014). Tendencias y dificultades para el uso de las TIC en la educación superior. *Ciencias de la educación, 10*(1). Obtenido de <http://www.scielo.org.co/pdf/entra/v10n1/v10n1a17.pdf>
- PHP. (s.f.). *PHP*. Obtenido de <https://www.php.net/manual/es/intro-what-is.php>
- Revista Dinero. (13 de 10 de 2016). *Revista Dinero*. Obtenido de <https://www.dinero.com/emprendimiento/articulo/congreso-colombia-40-y-cifras-del-mercado-de-las-aplicaciones-moviles/234972>
- Sánchez Jiménez, M. Á. (2018). Origen y Evolución de Internet y su Desarrollo Como Entorno de Interacción Social a Través de los Medios Sociales Digitales. *Revista Contribuciones a Las Ciencia Sociales*. Obtenido de <https://www.eumed.net/rev/cccss/2018/03/medios-sociales-digitales.html>
- Sandoval Mondragón, S. (09 de 05 de 2017). *Blog fixter*. Obtenido de <http://blog.fixter.org/retrofit-el-mejor-cliente-rest/>
- Tecnologías información. (s.f.). *Tecnologías información*. Obtenido de <https://www.tecnologias-informacion.com/basesdedatos.html#>