SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE SAN MARCOS

Aplicación móvil “Coyucapp”

INFORME TÉCNICO DE
RESIDENCIAS PROFESIONALES

INGENIERÍA INFORMÁTICA

PRESENTA
OLIVER PIEDRA CORTES
161230046

PERIODO: 04 DE OCTUBRE DEL 2021 AL 18 DE ABRIL DE 2023

SAN MARCOS GUERRERO, MEX. ABRIL DE 2023

Agradecimientos

Quiero agradecer en primer lugar a Dios, por guiarme en el camino y fortalecerme espiritualmente para empezar un camino lleno de éxito.

Así, quiero mostrar mi gratitud a todas aquellas personas que estuvieron presentes en la realización de esta meta, de este sueño que es tan importante para mí, agradecer todas sus ayudas, sus palabras motivadoras, sus conocimientos, sus consejos y su dedicación.

Muestro mis más sinceros agradecimientos a mi asesor interno, quien con su conocimiento y su guía fue una pieza clave para que pudiera desarrollar una clave de hechos que fueron imprescindibles para cada etapa de desarrollo del trabajo.

A mis compañeros, quienes a través de tiempo fuimos fortaleciendo una amistad y creando una familia, muchas gracias por toda su colaboración, por convivir todo este tiempo conmigo, por compartir experiencias, alegrías, frustraciones, llantos, tristezas, peleas, celebraciones y múltiples factores que ayudaron a que hoy seamos como una familia, por aportarme confianza y por crecer juntos en este proyecto, muchas gracias.

Por último, quiero agradecer a la base de todo, a mi familia, en especial a mis padres, que quienes con sus consejos fueron el motor de arranque y mi constante motivación, muchas gracias por su paciencia y comprensión, y sobre todo por su amor.

¡Muchas gracias por todo!

Resumen

El presente documento pretende cubrir la demanda de una aplicación que pueda facilitar la búsqueda de negocios en Coyuca de Benítez, Gro. En su problemática Coyuca de Benítez, Gro. no cuenta con una aplicación que se dedique exclusivamente a consultar negocios y obtener información de los mismos. En Coyuca de Benítez Gro. La situación de los negocios ha decaído, parte de este problema ha sido provocado por la pandemia, aunado a esto, los negocios no cuentan con promoción digital. No obstante crear campañas publicitarias de estos negocios es muy general, es decir llega a todas partes principalmente porque no es muy específico en cuestión de la zona donde está el negocio. El proyecto consiste en buscar, ordenar y administrar los negocios de Coyuca de Benítez, Gro. mediante una Aplicación móvil, denominada "COYUCAPP". Como se menciona en el marco teórico el desarrollo de software orientado a objetos permite aplicar los conceptos de herencia, polimorfismo y encapsulación a los objetos que son una representación de la realidad. Por ejemplo, un negocio que se encuentra en Coyuca de Benítez, Gro. cuenta con Logo, nombre del negocio, dueño del negocio, descripción del negocio, Menú de productos, ubicación, número de teléfono del negocio e imágenes de sus productos más destacados los cuales serán atributos del objeto dentro del software junto a los métodos u operaciones "consultar" y "reservar". Como objetivo general se pretende desarrollar una aplicación móvil para el Sistema operativo Android, mediante la utilización del IDE Android Studio. XAMMP, API Volley y el editor de texto SublimeText, que admita la colocación de anuncios publicitarios de los negocios del municipio de Coyuca de Benítez, Gro. permitiendo a los comerciantes hacer propaganda de bajo costo y local. Tomando en cuenta los objetivos específicos también Crear diagrama UML en programa de office llamado VISIO (Diagrama relacional y la herramienta XAMMP utilizando la opción de diseñador para crear modelo relacional para posteriormente crear una base de datos, también Diseñar interfaces en el IDE Android Studio (Módulo de iniciar sesión, registro de usuario y negocio, principal, perfil de negocio, favoritos, Creadores y políticas, información de negocio) con el fin de darle una vista intuitiva para el usuario.

Introducción

En el presente documento estarán diversos factores como objetivos que ayudarán a conocer mejor el desarrollo de la aplicación móvil “Coyucapp” y la raíz del por qué se desarrolla, recalcando que actualmente Coyuca de Benítez, Gro. no cuenta con una aplicación en la cual pueda alojar negocios virtuales, gratuitos y fomentar que negocios son de buena calidad. ¿De qué manera podemos hacer que los usuarios puedan tener un espacio virtual para sus negocios? Esta aplicación responde a la pregunta. Ya que la función principal y de valor agregado es que los negocios locales puedan alojarse en la aplicación, así los consumidores locales y turistas puedan apreciar los negocios que existen en Coyuca de Benítez, Gro.

Índice

Agradecimientos 2
Resumen 3
Introducción 4
Objetivo General 6
Objetivos específicos 6
Justificación 7
Capítulo 1 Fundamento Teórico 8
1.1 Marco referencial 8
1.2 Marco histórico 8
1.3 Marco teórico 9
1.4 Herramientas para el desarrollo 11
Capítulo 2 Metodología 13
2.1 Procedimiento y descripción detallada de las actividades Utilizando Metodología Scrum. 13
2.2 Fase Planificación: Backlog de producto 13
2.3 Fase Ejecución: Sprint 15
Capítulo 3 Resultados 16
3.1 Diagrama relacional 16
3.2 Diagrama E-R 16
3.3 Base de datos 17
3.4 Módulo principal 17
3.5 Módulo de información de negocios 22
3.6 Módulo de perfil 25
3.7 Módulo de favoritos 29
3.8 Módulo de registro e iniciar sesión 31
3.9 Scrips en PHP 35
Capítulo 4 Privacidad y uso de datos 43
4.1 Términos y condiciones 43
4.2 Política y privacidad del usuario 44
Conclusión y recomendaciones 45
Competencias desarrolladas 46
Fuentes de información 47
Anexos 48

Objetivo General

Desarrollar una aplicación móvil para el sistema operativo Android, mediante la utilización del IDE Android Studio. XAMMP, API Volley, y el editor de texto SublimeText, que admita la colocación de anuncios publicitarios de los negocios del municipio de Coyuca de Benítez, Gro. permitiendo a los comerciantes hacer publicidad de bajo costo y local.

Objetivos específicos

- Crear diagrama UML en programa de office llamado VISIO (Diagrama relacional y la herramienta XAMMP utilizando la opción de diseñador para crear modelo relacional para posteriormente crear una base de datos.
- Diseñar interfaces en el IDE Android Studio (Módulo de iniciar sesión, registro de usuario y negocio, principal, perfil de negocio, favoritos, Creadores y políticas, información de negocio) con el fin de darle una vista intuitiva para el usuario.
- Desarrollar Base de datos en el software de XAMPP que permita gestionar la base de datos.
- Desarrollar y Diseñar Interfaz de la aplicación móvil en el IDE Android Studio para sistemás operativos Android.
- Utilizar la Api Volley para la interacción de scrips de PHP que se crearan en el editor de texto SublimeText.
- Implementar la aplicación móvil Coyucapp.

Justificación

En Coyuca de Benítez, Gro. la situación de los negocios ha decaído, y no solo por el motivo de la pandemia. Los negocios para poder sustentarse de ganancias necesitan hacer publicidad, utilizando plataformás como Facebook e Instagram, no obstante crear campañas publicitarias de estos negocios es muy general, es decir llega a todas partes principalmente porque no es muy específico en cuestión de la zona donde está el negocio. Con las redes sociales para llegar a más personas y haciendo que cada usuario solamente vea por un minúsculo momento esa publicidad. Entendiendo que negocios pequeños y medianos son aquellos que quieren crecer, por ende, es necesario una aplicación móvil el cual ayude a darse a conocer en su área local y sus alrededores.

Actualmente Coyuca de Benítez, Gro. no cuenta con una aplicación en la cual pueda alojar negocios virtuales gratuitos y fomentar que negocios son de buena calidad. ¿De qué manera podemos hacer que los usuarios puedan tener un espacio virtual para sus negocios? Esta aplicación responde a la pregunta. Ya que la función principal y de valor agregado es que los negocios locales puedan alojarse en la aplicación, así los consumidores locales y turistas puedan apreciar los negocios que existen en Coyuca de Benítez, Gro. la inversión del desarrollo de la app móvil se obtendrá mediante banners de publicidad, Siempre existirá una versión gratuita de Coyucapp para que los usuarios que se registren tanto como negocio o usuario será de manera gratuita y su vez que tengan un beneficio social. Algo importante es que impactara en la economía local, a su vez a las localidades colindantes con Coyuca de Benítez, Gro.

Capítulo 1 Fundamento Teórico

1.1 Marco referencial

Lo que se describe a continuación está compuesto por el marco histórico y teórico. El marco histórico relaciona la importancia de los conceptos y tecnologías usados en cada la aplicación móvil. Dentro del marco teórico se agrupan los conceptos y teorías utilizados para el desarrollo del argumento del presente proyecto, así mismo, el marco conceptual recoge los conceptos propios sustentados en bases de datos especializadas y su interpretación utilizados dentro del tema de investigación.

1.2 Marco histórico

La era del computador personal ha terminado y la era de los dispositivos móviles comienza. (Steve Jobs, 2010).

En muy pocos años los teléfonos móviles han llegado a convertirse en un dispositivo esencial para particulares o empresas con posibilidad de elección entre miles de modelos que los diferentes fabricantes siguen innovando intentando destacar sobre los de la competencia. En 1997, Nokia lanzó el primer Smartphone, el Nokia 9000i. El teléfono podía abrirse de manera horizontal, mostrando una pantalla Autor Gonzalo de Lucas 3 panorámica y un teclado tipo máquina de escribir. Podía recibir y enviar faxes, SMS y emails, pudiendo acceder a Internet a través de mensajes SMS. La innovación no es una opción, sino una obligación para las empresas en un contexto en el que los modos de comunicarnos, informarnos, pensar y trabajar han cambiado. Y hoy en día, una empresa que se considere a sí misma como innovadora y actualizada no puede dejar de tener en cuenta ciertos ejes: la móvilidad, el cloud computing, y el modelo de comunicación impuesto por las redes sociales son algunos de los elementos que dan forma al actual ecosistema tecnológico en el que los ejecutivos deben desenvolverse. (Gonzalo de Lucas, 2017).

En la actualidad, uno de los aspectos más notorios es utilizar aplicaciones móviles, tecnología que hasta la fecha sigue siendo manipulada por millones de usuarios, mediante esto se sabe, que ha tenido un gran impacto a nivel mundial. No obstante, al igual que las personas la tecnología de debe adaptarse mediante las necesidades del ser humano. Por ejemplo, aplicaciones móviles para ver videos, chatear, descargar y cargar datos de la nube, etc… un sinfín de aplicaciones para sus respectivas necesidades. Aplicaciones como MI NEGOCIO “Mi negocio es una aplicación para la administración total de tu negocio.”. Sacado de

https://play.google.com/store/apps/details?id=com.segb_d3v3l0p.minegocio&hl=es_MX&gl=USn

1.3 Marco teórico

En el desarrollo del marco teórico se aplican teorías que desde el punto de vista profesional que permiten la elaboración del contenido del proyecto, a continuación, se muestra la siguiente información:

- El desarrollo de software orientado a objetos permite aplicar los conceptos de herencia, polimorfismo y encapsulación a los objetos que son una representación de la realidad. Por ejemplo, un negocio que se encuentra en Coyuca de Benítez, Gro. cuenta con Logo, nombre del negocio, dueño del negocio, descripción del negocio, Menú de productos, ubicación, número de teléfono del negocio e imágenes de sus productos más destacados los cuales serán atributos del objeto dentro del software junto a los métodos u operaciones “consultar” y “reservar”.

- Según www.lucidchart.com Un diagrama entidad-relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los diagramás ER se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemás de información empresarial, educación e investigación. También conocidos como los ERD o modelos ER, emplean un conjunto definido de símbolos, tales como rectángulos, diamantes, óvalos y líneas de conexión para representar la interconexión de entidades, relaciones y sus atributos. Son un reflejo de la estructura gramatical y emplean entidades como sustantivos y relaciones como verbos. También se puede adjuntar departe de www.Oracle.com una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila en una tabla es un registro con una ID única, llamada clave. Las columnas de la tabla contienen los atributos de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.

- Para la creación de los módulos que compone la aplicación móvil Coyucapp tenemos al IDE Android studio que según la respuesta en su página oficial www.developer.android.com es Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA.

- El uso de Metodología Scrum, esta metodología está compuesta por cuatro fases que incluyen las actividades relacionadas con la ingeniería de software: En el desarrollo de la fase de planificación, las personas interesadas buscan establecer el enfoque del sistema que se va a desarrollar, se determina las funcionalidades, de igual forma se establece el orden cronológico de cada actividad para cumplir

el objetivo final que es desarrollar la aplicación móvil mediante reuniones en fecha ya seleccionadas. La siguiente fase es el desarrollo el cual deben garantizar que no se generen cambios de último momento que puedan afectar los objetivos del mismo. Además, se asegura el cumplimiento de los plazos establecidos para su término. La tercera fase es Revisión, al final del desarrollo del intervalo, es posible analizar y evaluar los resultados. Si es necesario, todo el equipo colaborará para saber qué aspectos necesitan ser cambiados. Por último, tenemos la fase de retroalimentación, en dicha fase las personas que utilizarán directamente lo que se desea lograr; es decir, los clientes potenciales. Las lecciones aprendidas durante esta etapa permitirán que el siguiente sprint pueda ser mucho más efectivo y ágil.

1.4 Herramientas para el desarrollo

Para el desarrollo de la aplicación móvil se utilizan las siguientes herramientas describiendo cada uno de estos:

XAMMP

1.4.1 xammp

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

Apache Friends es un proyecto para promocionar el servidor de web Apache y es el hogar del proyecto XAMPP. XAMPP es una distribución de Apache fácil de instalar que contiene MariaDB, PHP y Perl. Según la página oficial de XAMMP. (Apache Friends, 2021).

Microsoft Visio

1.4.2 Visio

Microsoft Visio es un software para dibujar una variedad de diagramás. Entre ellos se incluyen diagramás de flujo, organigramás, planos de construcción, planos de planta, diagramás de flujo de datos, diagramás de flujo de procesos, modelado de procesos de negocios, diagramás de carriles, mapas 3D y mucho más. Visio se puede usar en una variedad de escenarios para crear diagramás de aspecto profesional. Dado el costo de Visio, un diagrama dibujado a mano puede cubrir bien tus necesidades o podrías optar por las funciones básicas de diagramás en MS Word. Según la página oficial de Lucidchart. (Lucid Software Inc,2021).

PHP

1.4.3 PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Según la página oficial de PHP. (THE PHP GROUP, 2021).

Android studio

1.4.4 IDE Android studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial que se usa en el desarrollo de apps para Android. Basado en el potente editor de código y las herramientas para desarrolladores de IntelliJ IDEA, Android Studio ofrece aún más funciones que mejoran tu productividad cuando compilas 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 código y recursos a la app en ejecución sin reiniciarla.
- Variedad de marcos de trabajo y herramientas de prueba.
- Herramientas de Lint para identificar problemás de rendimiento, usabilidad y compatibilidad de versiones, entre otros.
- Compatibilidad con C++ y NDK.

Según la documentación oficial de Andorid studio. (Google Developers, 2021).

SublimeText

1.4.5 SublimeText

Sublime Text es un editor de texto para escribir código en casi cualquier formato de archivo. Está especialmente pensado para escribir sin distracciones. Es decir, que visualmente ofrece un entorno oscuro donde las líneas de código que escribas resaltarán para que puedas centrarte exclusivamente en ellas. De una forma sencilla podemos decir que Sublime Text, es un editor de texto ligero, pensado desde un inicio en la velocidad, Siendo uno de los editores de texto más rápido y fácil de usar. Además de la velocidad se tiene más de 1000 de plugins adicionales y todos de código abierto, con una comunidad de desarrolladores que día a día contribuyen desarrollando nuevos plugins los cuales proveen de más funcionalidad a este programa. Según la página de Codedonostia. (Codedonostia,2022).

Capítulo 2 Metodología

2.1 Procedimiento y descripción detallada de las actividades Utilizando Metodología Scrum.

Aplicando la metodología Scrum es un enfoque ágil para la gestión de proyectos que se enfoca en la entrega continua de valor al cliente. El proceso consta de tres fases principales: planificación, desarrollo y resultado.

2.2 Fase Planificación: Backlog de producto

El equipo Scrum se reúne para planificar el sprint (iteración) en la que se trabajarán las funcionalidades de la aplicación móvil. Se identifican los elementos del backlog del producto que se abordarán durante el sprint.

2.2.1 Diseño de diagramás UML:

Programa VISIO- Se realizo diagramás entidad relación y diagrama relacional, se utilizó el programa Visio posteriormente estos diagramás se utilizaron para crear la base de datos.

2.2.2 Diseño de la base de datos:

Software XAMMP -Se creó una base de datos utilizando el software XAMPP para alojar la información en un servidor local y el tipo de base de datos será MYSQL.

2.2.3 Diseño de interfaz:

El diseño se creó en el IDE Android Studio, utilizando elementos del mismo para plasmar la idea y el proceso del diseño de la aplicación móvil. De igual forma para ver la conexión entre los módulos y sus aspectos de la vista.

2.2.4 Diseño y desarrollo los siguientes módulos:

Módulo de iniciar sesión- En este apartado se diseñó el módulo para que el usuario pueda iniciar sesión de esta manera el usuario podrá tener una vista con la cual pueda verse reflejada el módulo y así posteriormente con el back-end tenga el desarrollo que se necesitara.

Módulo de registro de usuario y negocio- Este Modulo tiene la particularidad de tener a los dos tipos de usuario, es decir tendrá la vista con opción de elegir entre el usuario o el negocio dependiendo del tipo de usuario en registro.

Modulo principal- Tendrá una vista para el usuario siendo esta la primera al cargar la aplicación móvil. Esta tendrá categorías de negocios, buscador de negocios y buscador por área.

Módulo de perfil de negocio- Esta será una vista para los usuarios registrados como negocios en la cual podrán editar datos como nombre de negocio, descripción y horario, productos o servicios que tenga el negocio, foto de negocio, 3 fotos de sus productos, número de teléfono y ubicación.

Módulo de Favoritos- Esta será una vista para que el usuario pueda ver los negocios que Agregaron a favoritos.

Módulo de Información de negocio- La vista en diseño será para presentar todo lo que tiene el módulo de perfil de negocio con la restricción de editar los datos para los usuarios en general. Diseñada exclusiva para ver información del negocio. Aquí se utiliza el número de teléfono para llamar y enviar mensaje al WhatsApp del negocio y tendrá una forma de calificación que será los favoritos indicados en la parte del back-end.

2.2.5 Desarrollo del back-end:

Se uso el editor de texto SublimeText generando scripts de PHP para hacer consultas correspondientes como editar, insertar, consultar y actualizar datos. Se utiliza el IDE Android Studio, utilizando elementos del mismo para plasmar la idea y el desarrollo de la aplicación móvil. En el desarrollo del back-end se harán las pruebas con la metodología prueba y error.

2.2.6 Se desarrollará el back-end en los siguientes módulos:

Módulo de Iniciar sesión-Se aplico restricciones entre ellas, el usuario inicia sesión con correo electrónico o datos como nombre de usuario y contraseña para tener opciones de ingreso a su cuenta respectivamente del tipo de usuario registrado. Se utilizo scrips de y la Api de Volley para consultar datos a la base de datos

Módulo de registro de usuario y negocio: El registro requiere datos como, Correo electrónico, nombre de usuario, contraseña, tipo de usuario. También requisitos de aceptar términos y condiciones de la aplicación móvil. Se utilizo las scripts de PHP y la Api de Volley para insertar y consultar datos a la base de datos

Modulo principal- En la parte del back-end el módulo principal será la primera vista y la principal para ver negocios, los usuarios solo pueden buscar en indagar sin editar la vista. Puede obtener información de los negocios que existen con respectivas condicionales utilizando las scrips de PHP y la Api de Volley para así mismo consultar datos de la Base de datos.

Módulo de perfil del negocio- El usuario podrá acceder solo si está registrado como negocio. Se utilizo las scrips de PHP y la Api de Volley para así mismo consultar, actualizar e insertar datos de la Base de datos.

Módulo de favoritos- Aquí se mostrará como información estructurados de cada negocio que el usuario agrego a sus favoritos. Se utilizo las scrips de PHP y la Api de Volley para así mismo consultar datos de la Base de datos.

Módulo de información del negocio- Los usuarios en general pueden acceder a ver la información proporcionada por los datos del negocio. Se utilizo las scrips de PHP y la Api de Volley para así mismo consultar datos de la Base de datos.

2.3 Fase Ejecución: Sprint

El equipo se reúne todos los días durante unos 15 minutos para revisar el progreso del sprint, identificar los impedimentos y ajustar el plan en consecuencia.

2.3.1 Sprint: Diseño UML

Comienzo: 30 de agosto del 2021

Finalización: 14 de septiembre del 2021

Fecha de reuniones

10 de septiembre del 2021. Hora: 3:00pm-3:30pm

14 de septiembre del 2021. Hora: 3:00pm-3:30pm

2.3.2 Sprint: Diseño de la base de datos

Comienzo: 15 de septiembre del 2021

Finalización: 14 de octubre del 2021

Fecha de reuniones

13 de octubre del 2021. Hora: 3:00pm-3:30pm

2.3.3 Sprint: Diseño de interfaz y desarrollo (Front-End)

Comienzo: 17 de octubre del 2021

Finalización: 30 de noviembre del 2021

Fecha de reuniones

17 de noviembre del 2021. Hora: 3:00pm-3:30pm

30 de noviembre del 2021. Hora: 3:00pm-3:30pm

2.3.4 Sprint: Desarrollo de Back-End.

Comienzo: 1 de diciembre del 2021

Finalización: - de enero del 2021

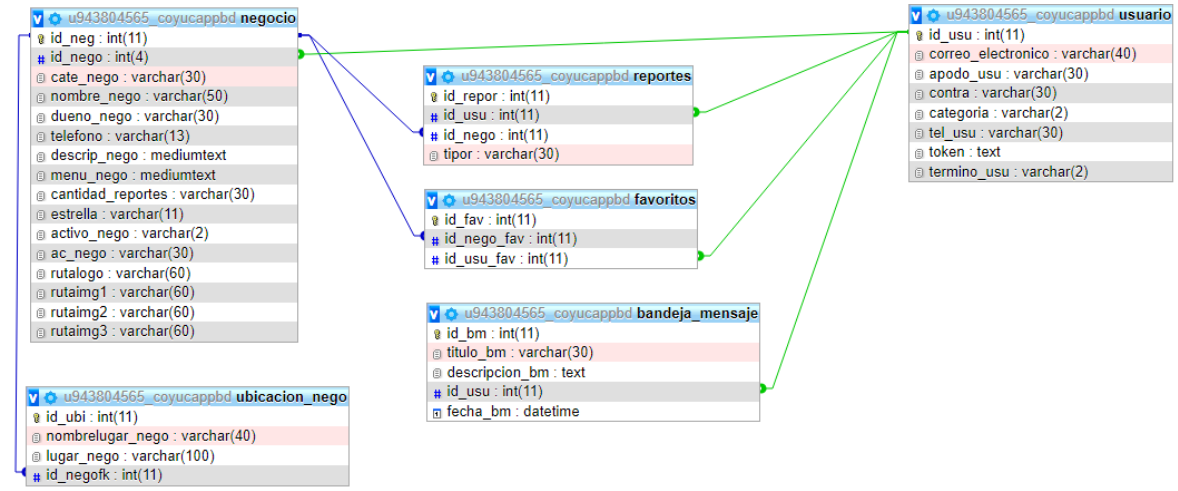
Fecha de reuniones

20 de diciembre del 2021. Hora: 3:00pm-3:30pm

Capítulo 3 Resultados

Los resultados obtenidos siguen el mismo orden en que fueron recogidos y como se presentan en el diseño de la metodología Scrum.

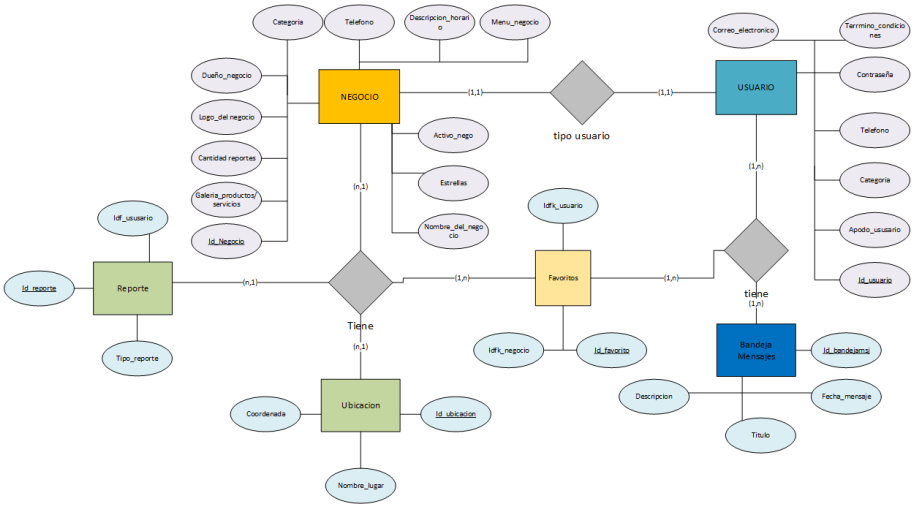
3.1 Diagrama relacional



3.1 Diagrama relacional

Diagrama relacional con sus respectivas relaciones con cada tabla mediante identificadores foráneos

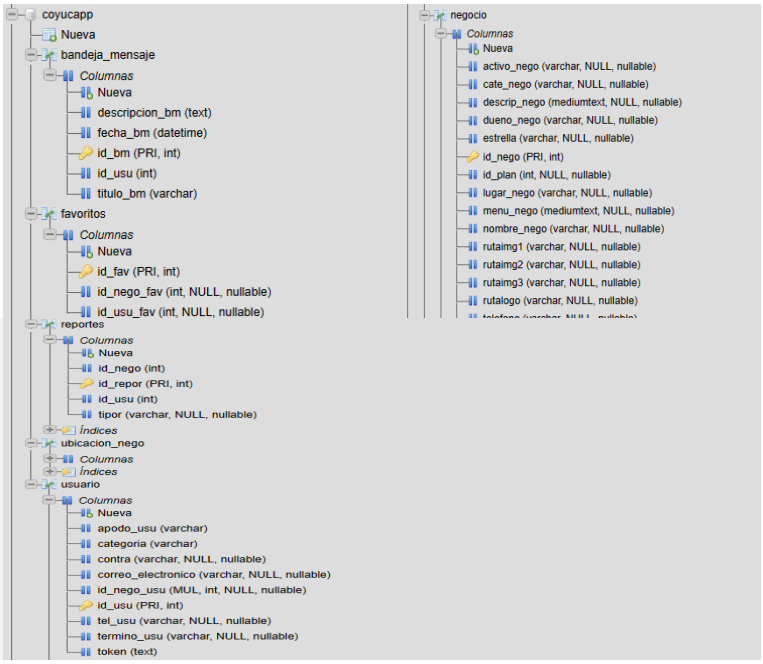
3.2 Diagrama E-R



3.2 Diagrama E-R

Diagrama E-R para establecer atributos que construyen a la base de datos como se aprecia se tiene 6 entidades.

3.3 Base de datos



3.3 Base de datos completa

Así quedo la base de datos después del desarrollo de los diagramas UML. Asimilando gran parte de los atributos que contemplaba el diagrama E-R y relacional.

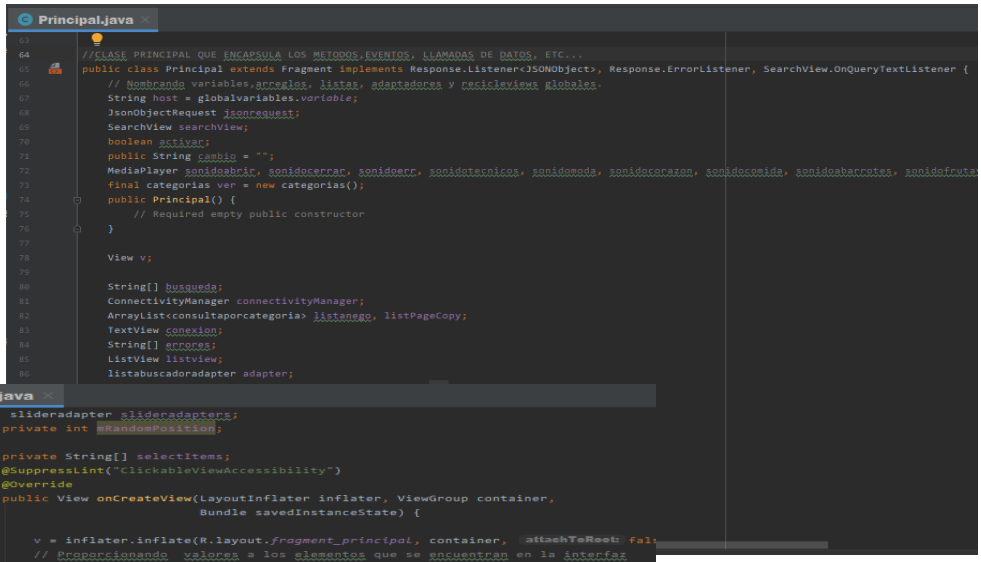
3.4 Módulo principal

3.4.1 Principal

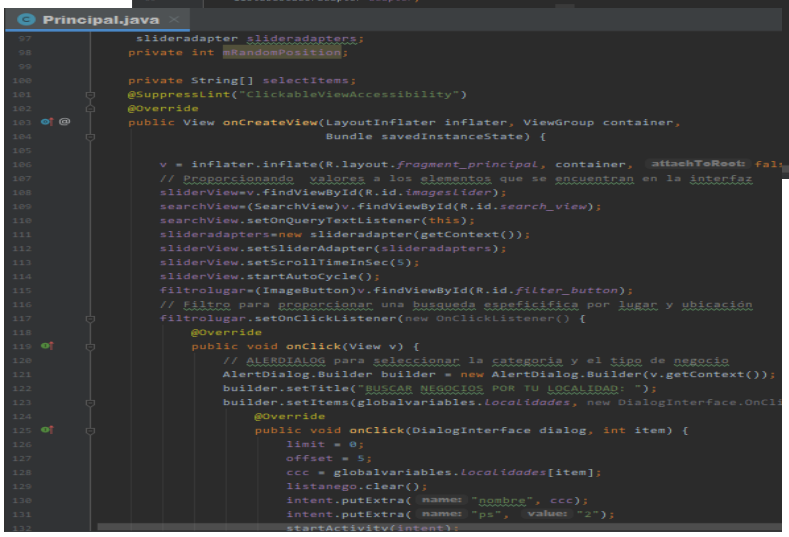
3.4.2 Principal segunda

La parte principal se observan la búsqueda por categorías, zona y por escrito. La función al dar clic entre estas funcionalidades abre el apartado de varios negocios.

A continuación, se muestra imágenes del código implementado desde el back-end para las funciones y eventos de los elementos, capturando los más relevantes utilizando el IDE Android Studio, evidentemente en base al módulo principal.

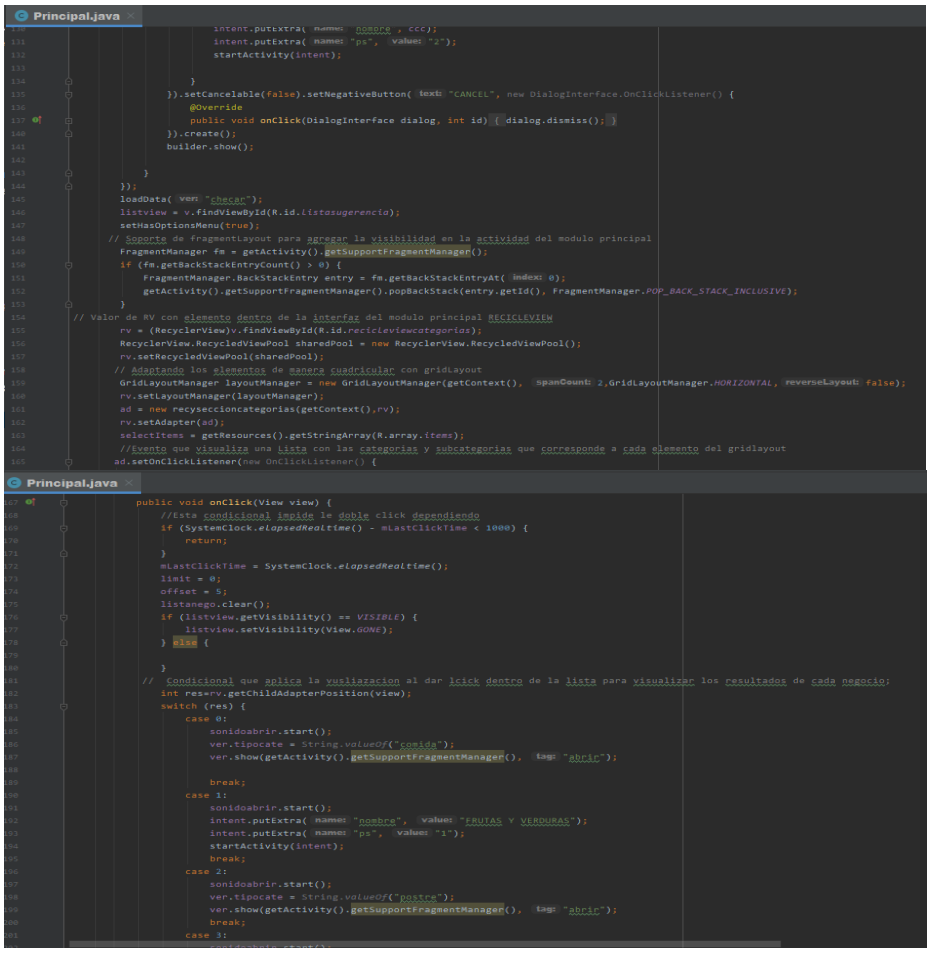


3.4.3 Inicializando variables



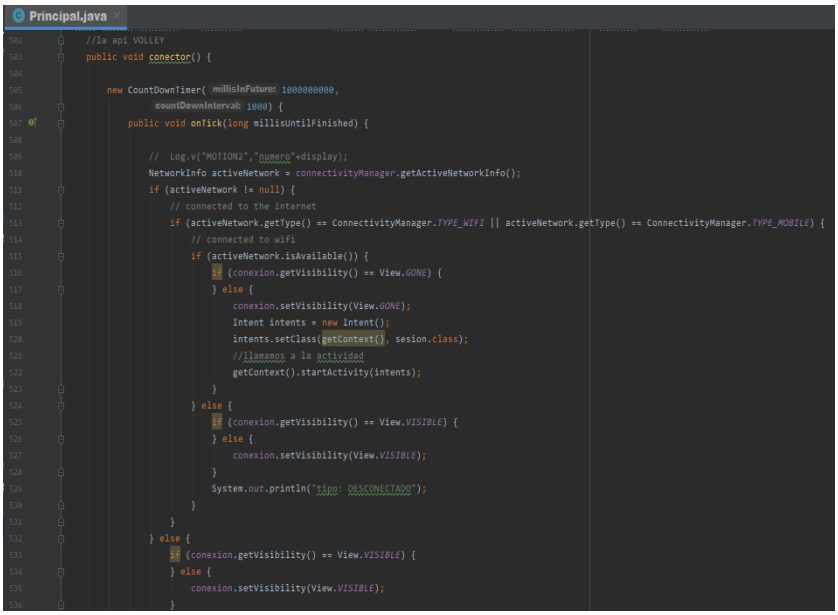
3.4.4 Inicializando variables

En la imagen se muestra el método principal, la cual crea e infla los elementos de la interfaz para conectarlos con el código del front-end.



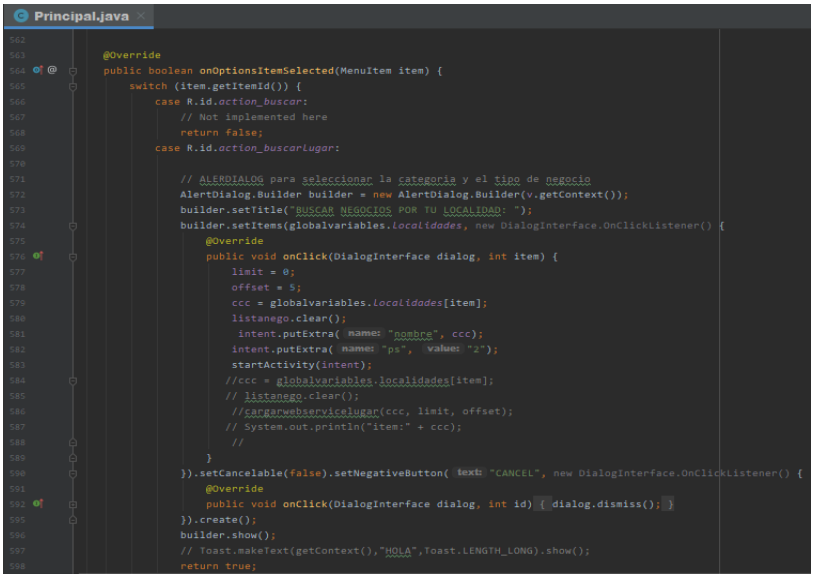
3.4.5 Condicionales de categorias

Esta es la segunda parta del módulo principal donde se observa la utilización del fragmentLayout mostrado en la activity principal, al igual que condicionales para establecer los eventos que se plasman al seleccionar un elemento para ver la categoría de los productos. También se aplica la parte del RecycleView para llenarlo con los datos establecidos previamente para la vista del usuario.



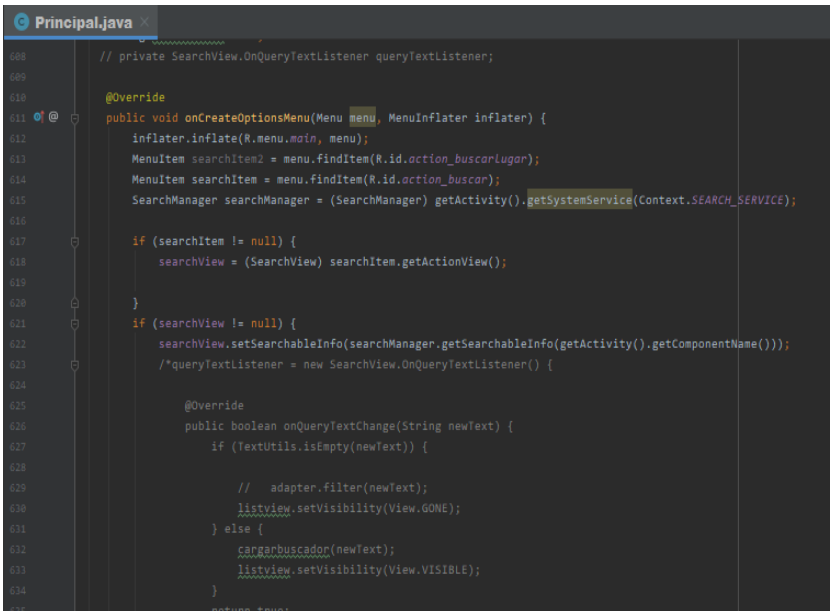
3.4.5 Conexión a internet

El código postrado verifica la conexión mediante un método llamado conector. El cual le indica cuando existe conexión a internet para evidentemente verificar los recursos utilizados por la Api VOLLEY



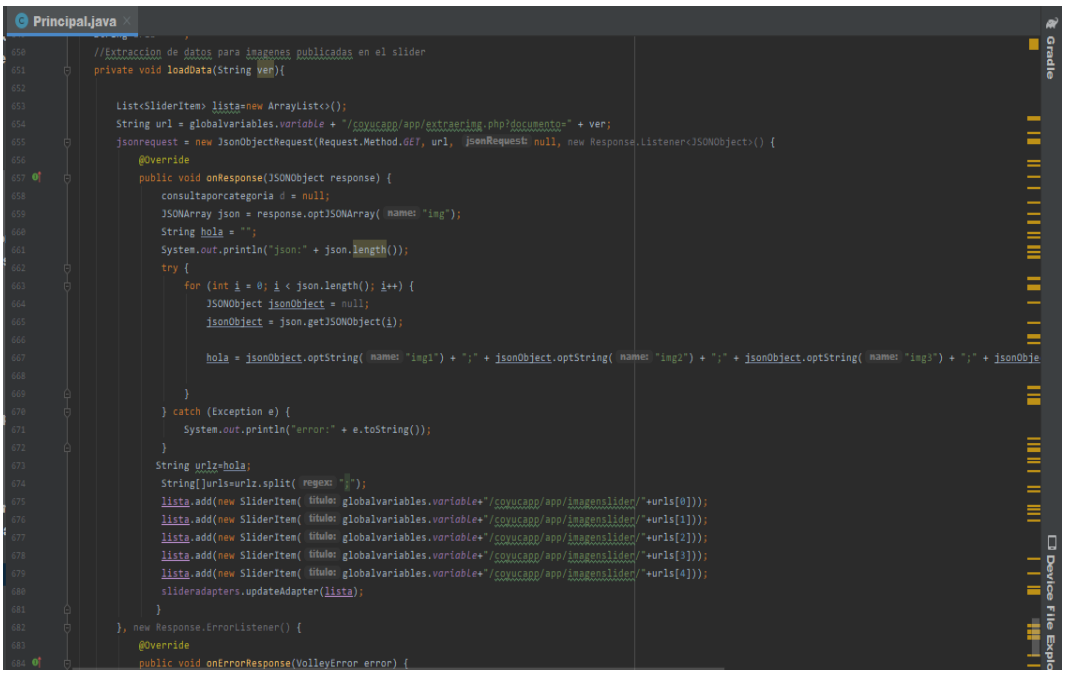
3.4.6 Valores de un clic

En esta etapa del módulo principal, este verifica si proporciono valor seleccionado con un Clic para después realizar o visualizar los datos del negocio.



3.4.7 Busqueda inmediata

La opción del OncreateOpcionMenu es para que el objeto al buscar dentro del buscador este se lanza, garantizando la búsqueda inmediata, mediante lo agregado de manera Alfanumérica por el usuario.



3.4.8 Slider

Aquí se solo un buscador mediante URL para buscar imágenes y plasmarlas en slider.

3.5 Módulo de información de negocios

3.5.1 Cabecera de información de negocios

3.5.1 Cuerpo de información de negocios

3.5.3 Ubicación de negocios

Como se mencionó el módulo de información de negocios visualiza la información básica del negocio como su ubicación y utilización de su número de teléfono para llamadas y mensajes.

```
public class vermasnego extends DialogFragment implements Response.Listener<JSONObject>, Response.ErrorListener {
    View v;
    String idd;
    Principal p;
    private Principal.OnFragmentInteractionListener listener;
    public TextView nombre,reportar;
    ImageView cerrar;
    Animation animation;
    public TextView menu,descripcion;
    public String no_telefono,estre;
    String idvistaprevia,idvistaprevia1,idvistaprevia2,idvistaprevia3;
    LottieAnimationView progressBar,progressBar2,progressBar3,progressBar4,llamar,whasa;
    String host=globalvariables.variable;
    MediaPlayer sonidocerrar,sonidoquitar,sonidoagregar,sonidoerr;
    CircleImageView logo,img1,img2,img3;
    public ImageView imgfavoritono,imgfavoritosi;
    JsonObjectRequest jsonObjectRequest;
    Activity actividad;
String lugar;
   String cambio;
    private static final  String TAG="abrir";
    String identificador = null;
    LottieAnimationView verubi; final mapanegocio mapa=new mapanegocio();

    View vv;
    View ver;
    @Override
```

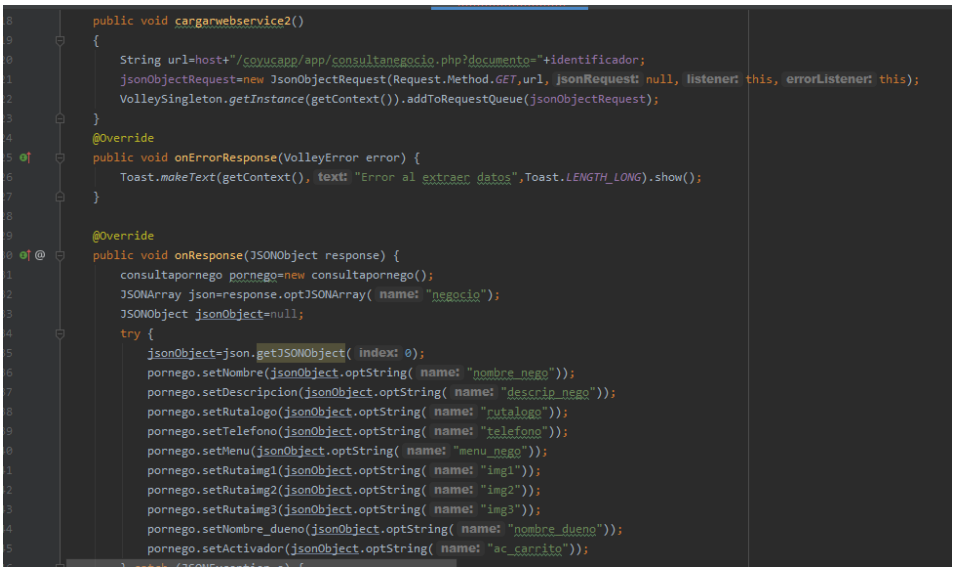
3.5.4 Iniciando variables

Como en la mayoría de fragments y activitys se nombran las variables a utilizar y consecuente se les asigna el valor que se necesita para cada variable.

```
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        vv=  inflater.inflate(R.layout.vermasnego, container);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            if(Build.VERSION.SDK_INT <= Build.VERSION_CODES.M)
            {

            }
            else
            {
                getDialog().requestWindowFeature(Window.FEATURE_NO_TITLE);
                getDialog().getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));
            }
        }
        return  vv;
    }

    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
    }
    String id_usuario="";
    public void cargarpreferencias()
    {
        SharedPreferences preferences=getContext().getSharedPreferences( name: "credencial", Context.MODE_PRIVATE);
        String user=preferences.getString( key: "user", defValue: "Invitado");
        id_usuario=preferences.getString( key: "id_usu", defValue: "Eres invitado");

```

3.5.5 Inflando vista de información de negocios

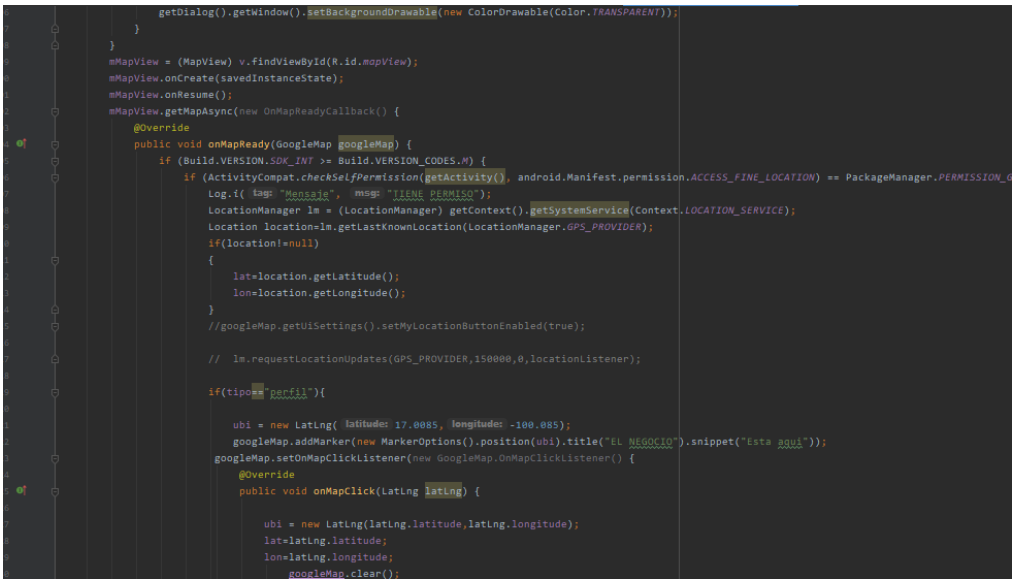
Este método OncreateView se muestra en la gran mayoría de las activitys y fragments. Aquí se infla la vista con la interfaz de usuario.



```
public void cargarwebservice2()
{
    String url=host+"/coyucapp/app/consultanegocio.php?documento="+identificador;
    jsonObjectRequest=new JsonObjectRequest(Request.Method.GET,url, jsonRequest: null, listener: this, errorListener: this);
    VolleySingleton.getInstance(getContext()).addToRequestQueue(jsonObjectRequest);
}
@Override
public void onErrorResponse(VolleyError error) {
    Toast.makeText(getContext(), text: "Error al extraer datos",Toast.LENGTH_LONG).show();
}

@Override
public void onResponse(JSONObject response) {
    consultapornego pornego=new consultapornego();
    JSONArray json=response.optJSONArray( name: "negocio");
    JSONObject jsonObject=null;
    try {
        jsonObject=json.getJSONObject( index: 0);
        pornego.setNombre(jsonObject.optString( name: "nombre_nego"));
        pornego.setDescripcion(jsonObject.optString( name: "descrip_nego"));
        pornego.setRutalogo(jsonObject.optString( name: "rutalogo"));
        pornego.setTelefono(jsonObject.optString( name: "telefono"));
        pornego.setMenu(jsonObject.optString( name: "menu_nego"));
        pornego.setRutaimg1(jsonObject.optString( name: "img1"));
        pornego.setRutaimg2(jsonObject.optString( name: "img2"));
        pornego.setRutaimg3(jsonObject.optString( name: "img3"));
        pornego.setNombre_dueno(jsonObject.optString( name: "nombre_dueno"));
        pornego.setActivador(jsonObject.optString( name: "ac_carrito"));
```

3.5.6 Cargawebservice

Se consta que en la mayoría de las activityes y fragments tendrán llamado a un método llamado "cargarwebservice" este indica la conexión de que aplica la api Volley con las scrips de PHP. Como se muestra en la imagen se aprecia la obtención de datos mediante un arreglo Json.



```
            getDialog().getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));
        }
    }
    mMapView = (MapView) v.findViewById(R.id.mapView);
    mMapView.onCreate(savedInstanceState);
    mMapView.onResume();
    mMapView.getMapAsync(new OnMapReadyCallback() {
        @Override
        public void onMapReady(GoogleMap googleMap) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                if (ActivityCompat.checkSelfPermission(getActivity(), android.Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_G
                    Log.i( tag: "Mensaje", msg: "TIENE PERMISO");
                    LocationManager lm = (LocationManager) getContext().getSystemService(Context.LOCATION_SERVICE);
                    Location location=lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
                    if(location!=null)
                    {
                        lat=location.getLatitude();
                        lon=location.getLongitude();
                    }
                    //googleMap.getUiSettings().setMyLocationButtonEnabled(true);

                    //  lm.requestLocationUpdates(GPS_PROVIDER,150000,0,locationListener);

                    if(tipo=="perfil"){

                        ubi = new LatLng( latitude: 17.0085, longitude: -100.085);
                        googleMap.addMarker(new MarkerOptions().position(ubi).title("EL NEGOCIO").snippet("Esta aqui"));
                     googleMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
                        @Override
                        public void onMapClick(LatLng latLng) {

                            ubi = new LatLng(latLng.latitude,latLng.longitude);
                            lat=latLng.latitude;
                            lon=latLng.longitude;
                                googleMap.clear();
```

3.5.7 Servicio externo

Para mostrar el negocio dentro de un mapa es evidente que se utilizan servicios externos para ello se utiliza Google maps indicando y dando permiso al acceso de la información de ubicación en base a geolocalización. Se mostro que se obtiene la latitud y longitud dependiendo de cómo el usuario registro los datos en su negocio, ya aquí se aprecia condicionales para admitir dicha idea en la interfaz.

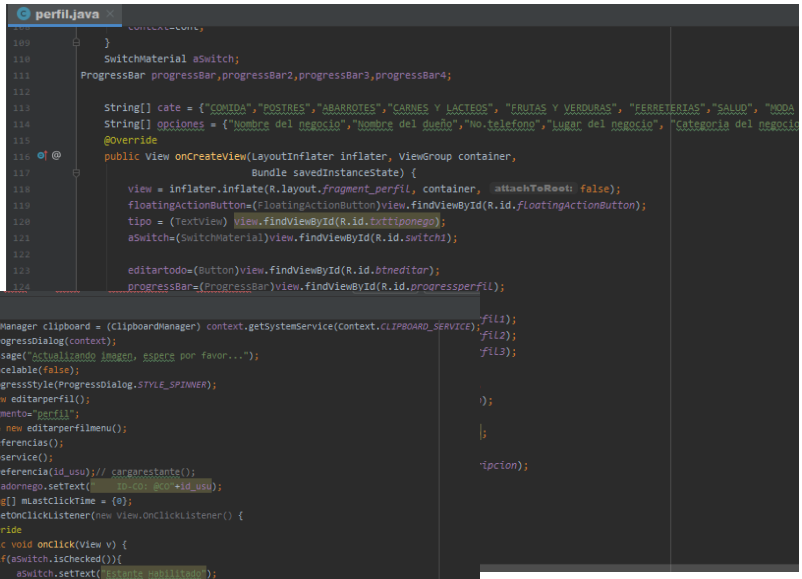
3.6 Módulo de perfil

3.6.1 Cabecera de perfil

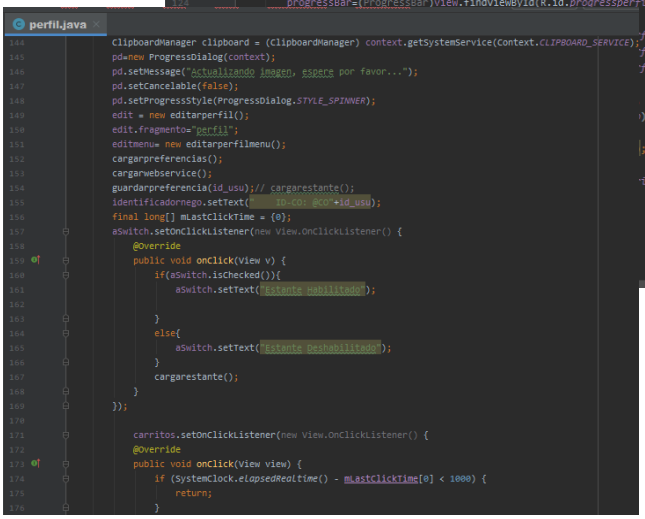
3.6.2 Cuerpo de perfil

3.6.3 Sección de ubicación

En la parte de perfil se observa las siguientes imágenes de las cuales, muestra los datos necesarios para que el negocio pueda visualizarse en la aplicación móvil Coyuca, así mismo la entrada de imágenes y ubicación necesarias para indicar donde está el negocio.

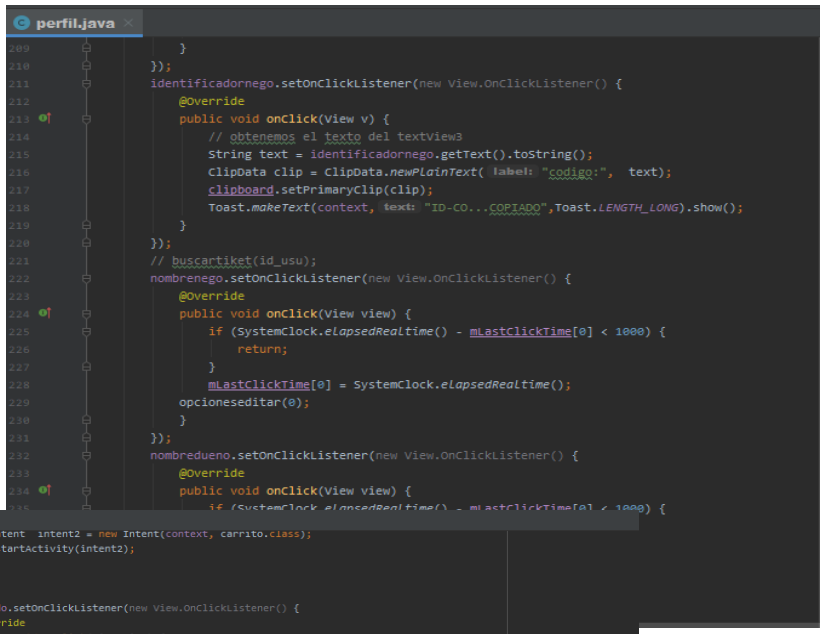


3.6.4 Variables y arreglos

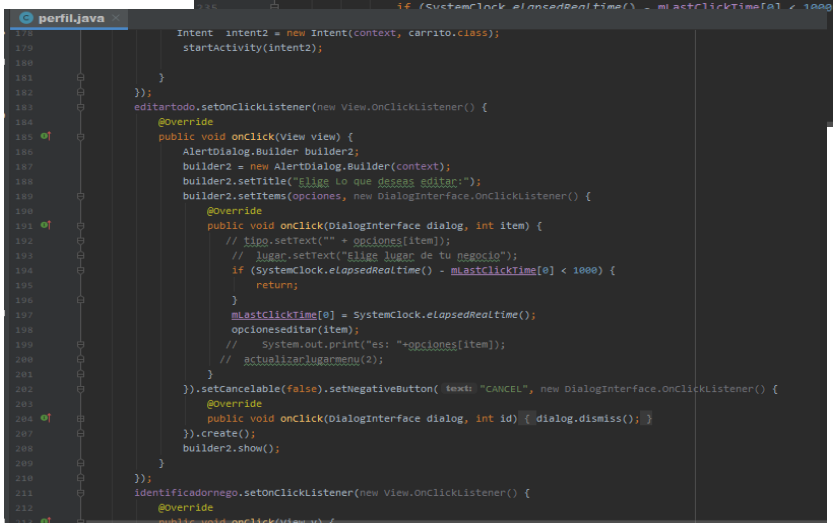


3.6.5 Inicializando objetos y variables

En la interfaz del módulo de perfil es normal tener muchas variables, asignándoles valores como se muestra a continuación, eso incluye la conexión entre frontend y backend para así poder realizar dichos eventos como los clics que se aprecia en la imagen adjunta y los métodos, esto ya es algo básico para todos módulos.



3.6.6 Eventos OnclickListener



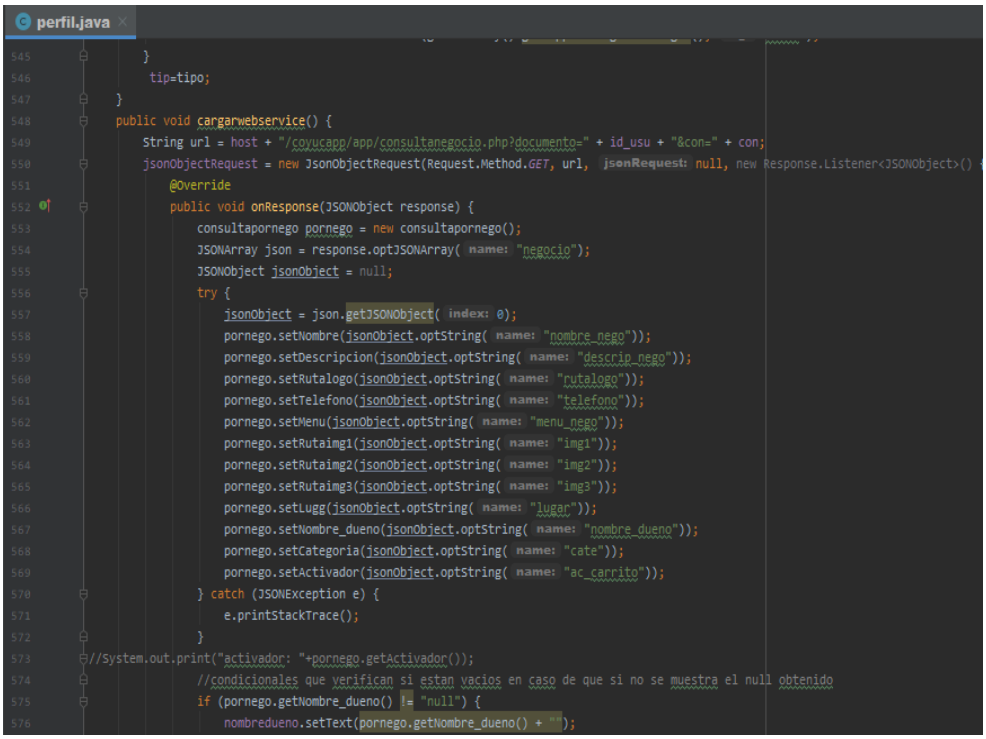
3.6.7 Editar por evento clic

Para editar los valores de cada textview o en cuyo caso imageView se utilizan los eventos OnclickListener. Uno de los eventos más utilizados en el desarrollo móvil para ello se indica el cambio mediante los clics que se les proporciona a los elementos, recordando que esto es orientado a objetos.

```
perfil.java
        });
        tel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (SystemClock.elapsedRealtime() - mLastClickTime[0] < 1000) {
                    return;
                }
                mLastClickTime[0] = SystemClock.elapsedRealtime();
                opcioneseditar(2);
            }
        });
        lugar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (SystemClock.elapsedRealtime() - mLastClickTime[0] < 1000) {
                    return;
                }
                mLastClickTime[0] = SystemClock.elapsedRealtime();
                opcioneseditar(3);

            }
        });
        tiponegocio.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (SystemClock.elapsedRealtime() - mLastClickTime[0] < 1000) {
                    return;
                }
                mLastClickTime[0] = SystemClock.elapsedRealtime();
              opcioneseditar(4);
                //----------------------------------------------------termino de ALERTDIALOG---------
            }
```

3.6.8 editar tipo de negocio, lugar, etc…

Indica los eventos para editar los objetos que el usuario desea.



```
perfil.java
    }
     tip=tipo;
}
public void cargarwebservice() {
    String url = host + "/covucapp/app/consultanegocio.php?documento=" + id_usu + "&con=" + con;
    jsonObjectRequest = new JsonObjectRequest(Request.Method.GET, url,  jsonRequest: null, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            consultapornego pornego = new consultapornego();
            JSONArray json = response.optJSONArray( name: "negocio");
            JSONObject jsonObject = null;
            try {
                jsonObject = json.getJSONObject( index: 0);
                pornego.setNombre(jsonObject.optString( name: "nombre_nego"));
                pornego.setDescripcion(jsonObject.optString( name: "descrip_nego"));
                pornego.setRutalogo(jsonObject.optString( name: "rutalogo"));
                pornego.setTelefono(jsonObject.optString( name: "telefono"));
                pornego.setMenu(jsonObject.optString( name: "menu_nego"));
                pornego.setRutaimg1(jsonObject.optString( name: "img1"));
                pornego.setRutaimg2(jsonObject.optString( name: "img2"));
                pornego.setRutaimg3(jsonObject.optString( name: "img3"));
                pornego.setLugg(jsonObject.optString( name: "lugar"));
                pornego.setNombre_dueno(jsonObject.optString( name: "nombre_dueno"));
                pornego.setCategoria(jsonObject.optString( name: "cate"));
                pornego.setActivador(jsonObject.optString( name: "ac_carrito"));
            } catch (JSONException e) {
                e.printStackTrace();
            }
//System.out.print("activador: "+pornego.getActivador());
            //condicionales que verifican si estan vacios en caso de que si no se muestra el null obtenido
            if (pornego.getNombre_dueno() != "null") {
                nombredueno.setText(pornego.getNombre_dueno() + "");
```

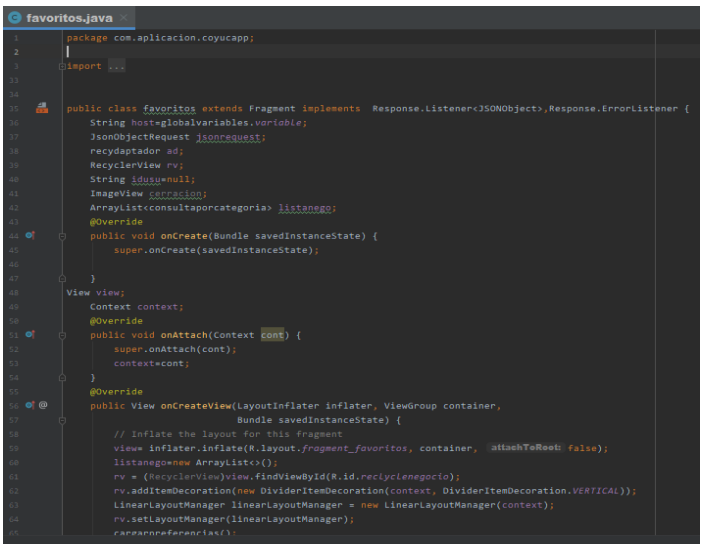
3.6.9 CargarwebService del perfil

Obtiene los datos mediante la api Volley esto incluye todo los datos y la ubicación del negocio que el usuario agrego.

3.7 Módulo de favoritos

3.7.1 Favoritos

Este módulo ya mencionado, muestra los negocios agregados por el usuario registrado en la aplicación móvil Coyucapp.



3.7.2 Backend de favoritos

En el módulo de favoritos de igual manera se infla la vista del fragment para visualizar los datos y ejecutar eventos.

favoritos.java

```
@Override
public void onErrorResponse(VolleyError error) {
    Toast.makeText(getContext(), text: "No tiene favoritos",Toast.LENGTH_LONG).show();
}
private long mLastClickTime = 0;
@Override
public void onResponse(JSONObject response) {
    listanego.clear();
    consultaporcategoria d=null;
    JSONArray json=response.optJSONArray( name: "negocio");
    try{
        for(int i=0;i<json.length();i++)
        {
            d=new consultaporcategoria();
            JSONObject jsonObject=null;
            jsonObject=json.getJSONObject(i);
            d.setId(jsonObject.optString( name: "id_nego"));
            d.setCategoria(jsonObject.optString( name: "cate_nego"));
            d.setNombre(jsonObject.optString( name: "nombre_nego"));
            d.setRutalogo(jsonObject.optString( name: "rutalogo"));
            d.setLocalidad(jsonObject.optString( name: "nombrelugar_nego"));
            d.setLugar(jsonObject.optString( name: "lugar"));
            d.setEstrella(jsonObject.optString( name: "estrella"));
            listanego.add(d);

        }
        ad=new recydaptador(context,listanego);
       final vermasnego ver= new vermasnego();
        ad.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //Esta condicional impide le doble click dependiendo
                if (SystemClock.elapsedRealtime() - mLastClickTime < 1000){
                    return;
                }
```

3.7.2 Rellena recycleview

Obtiene los datos y los muestra en un recycleview para su posterior uso.

favoritos.java

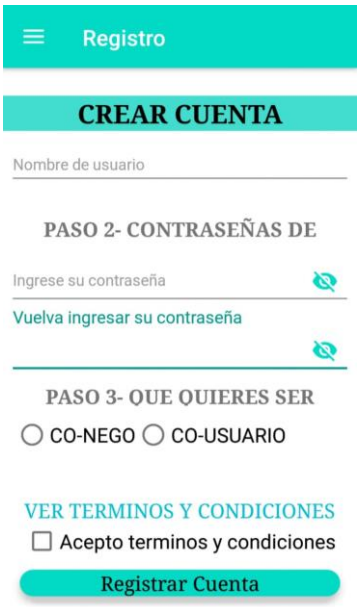
```
}
public void cargarpreferencias()
{
    SharedPreferences preferences=getActivity().getSharedPreferences( name: "credencial", Context.MODE_PRIVATE);
    String user=preferences.getString( key: "user", defValue: "invitado");
    idusu=preferences.getString( key: "id_usu", defValue: "eres invitado");
}

@Override
public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == 2) // 1 is an arbitrary number, can be any int
    { // This is the return result of your DialogFragment

        if(resultCode == 2) // 1 is an arbitrary number, can be any int
        { // Now do what you need to do after the dialog dismisses.
            Bundle bundle=data.getExtras();
            if(bundle!=null){
                int b=bundle.getInt( key: "ids");
                String ca=bundle.getString( key: "cambio");
                String estrellas=bundle.getString( key: "estrellas");
                System.out.println("datos: "+b);
                if(ca=="2" && estrellas!=null){

                    recydaptador.NumerosViewHorder holder=(recydaptador.NumerosViewHorder)rv.findViewHolderForAdapterPosition(b);
                    holder.estrellas.setText("\u0B3D\uDCC8"+estrellas+" favs.");
                }
                if(ca=="1")
                {
                    recydaptador.NumerosViewHorder holder=(recydaptador.NumerosViewHorder)rv.findViewHolderForAdapterPosition(b);
                    holder.estrellas.setText(estrellas+" favs.");
                }
```

3.7.3 Asignando valores

Aquí asigna valores y verifica la cantidad de favoritos que tienen los negocios, por ejemplo, obtiene datos desde la api Volley como se menciona a lo largo del proyecto.

3.8 Módulo de registro e iniciar sesión

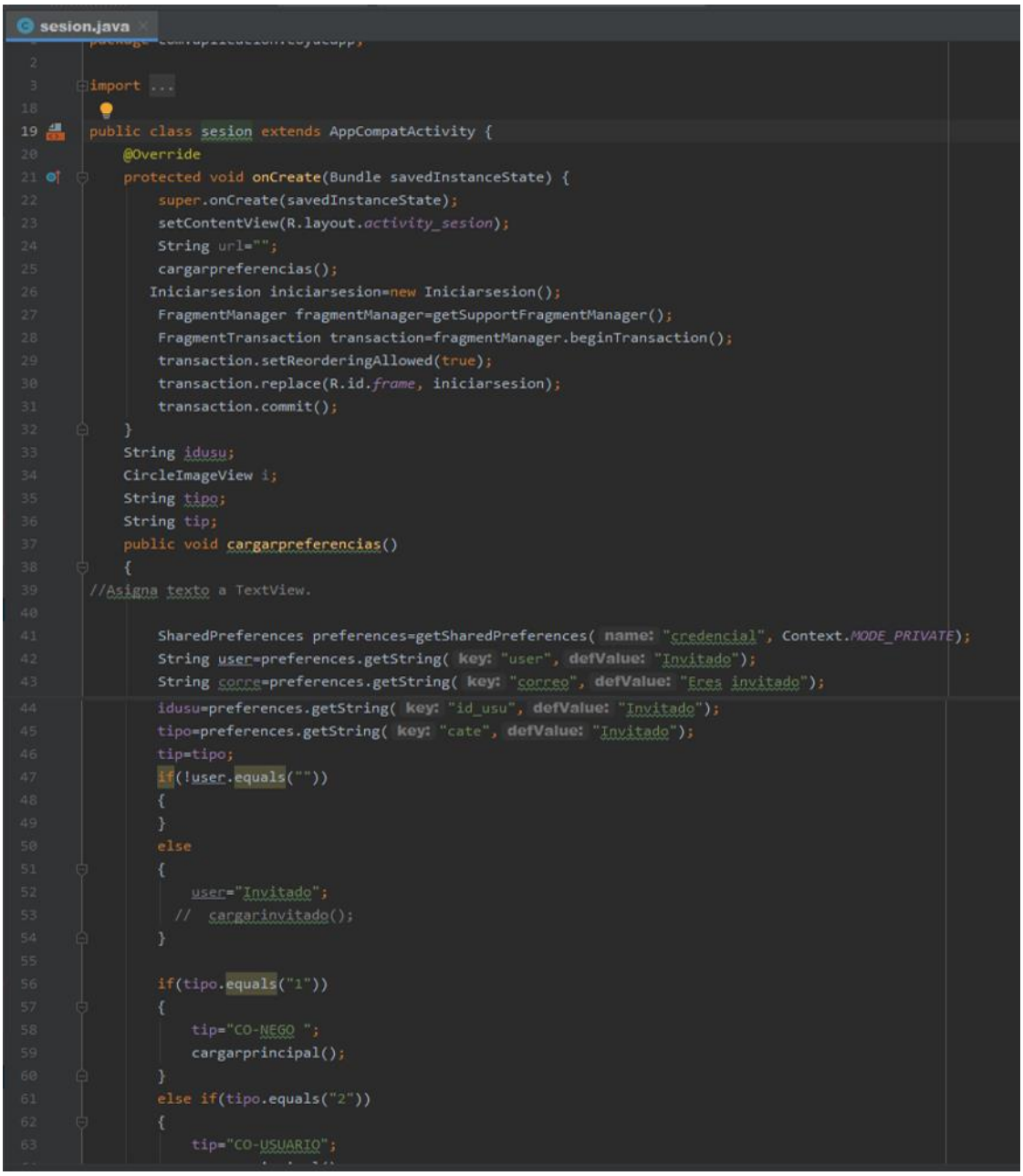
3.8.1 Iniciar sesión



3.8.2 Registro de cuenta

En el módulo de registro e iniciar sesión se muestran las siguientes imágenes, en iniciar sesión se tiene como restricción solo nombre de usuario contraseña o correo electrónico.

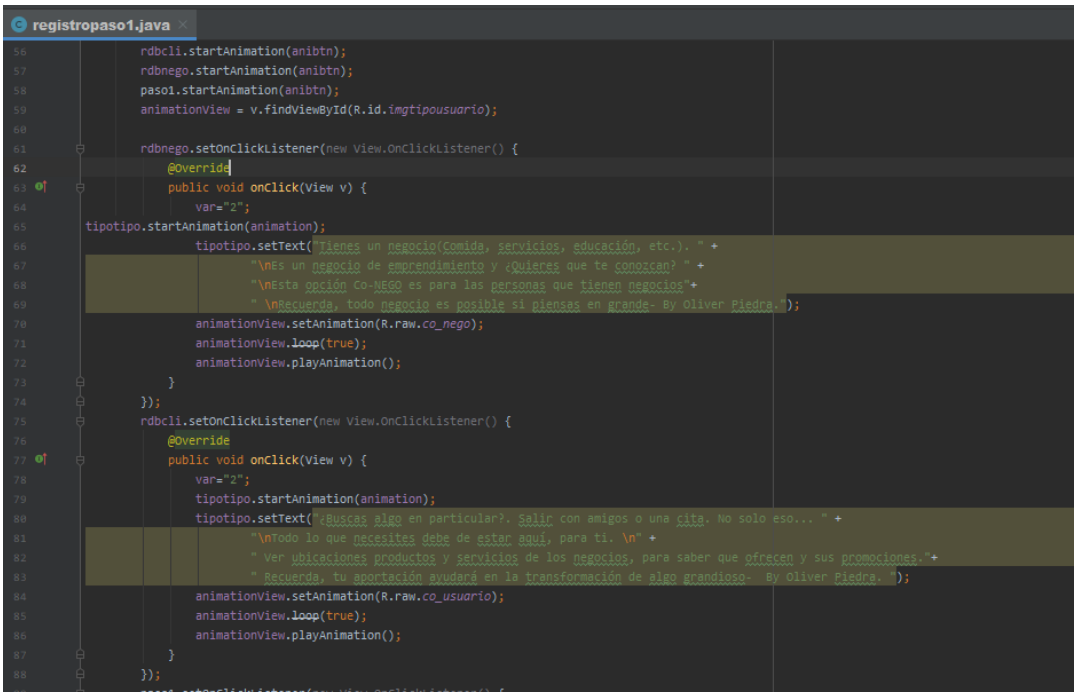
En registro se tienen los siguientes datos, agregar nombre de usuario, contraseña y tipo de usuario.



```java
import ...

public class sesion extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sesion);
        String url="";
        cargarpreferencias();
       Iniciarsesion iniciarsesion=new Iniciarsesion();
        FragmentManager fragmentManager=getSupportFragmentManager();
        FragmentTransaction transaction=fragmentManager.beginTransaction();
        transaction.setReorderingAllowed(true);
        transaction.replace(R.id.frame, iniciarsesion);
        transaction.commit();
    }
    String idusu;
    CircleImageView i;
    String tipo;
    String tip;
    public void cargarpreferencias()
    {
//Asigna texto a TextView.

        SharedPreferences preferences=getSharedPreferences( name: "credencial", Context.MODE_PRIVATE);
        String user=preferences.getString( key: "user", defValue: "Invitado");
        String corre=preferences.getString( key: "correo", defValue: "Eres invitado");
        idusu=preferences.getString( key: "id_usu", defValue: "Invitado");
        tipo=preferences.getString( key: "cate", defValue: "Invitado");
        tip=tipo;
        if(!user.equals(""))
        {
        }
        else
        {
            user="Invitado";
          //  cargarinvitado();
        }

        if(tipo.equals("1"))
        {
            tip="CO-NEGO ";
            cargarprincipal();
        }
        else if(tipo.equals("2"))
        {
            tip="CO-USUARIO";
```

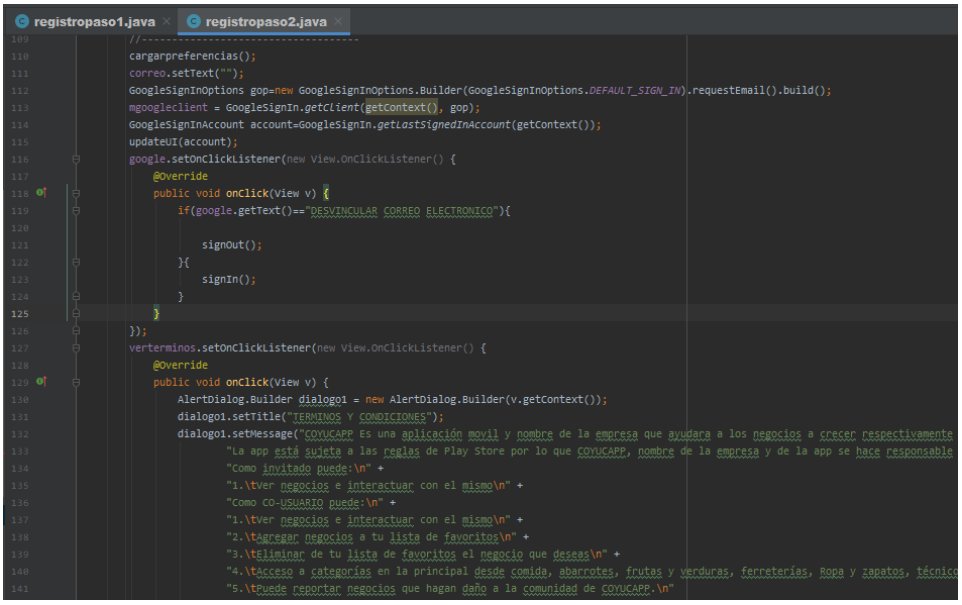
3.8.3 Verificación de inicio

Este es uno de los más importantes ya que verifica si los datos agregados al iniciar sesión son los correctos haciendo muchas condiciones para llegar al punto de dejar entrar al usuario o no. Se utilizo preferencias para obtener temporalmente los datos y así eventualmente cada acceso que tenga puede acceder sin agregar nuevamente los datos.



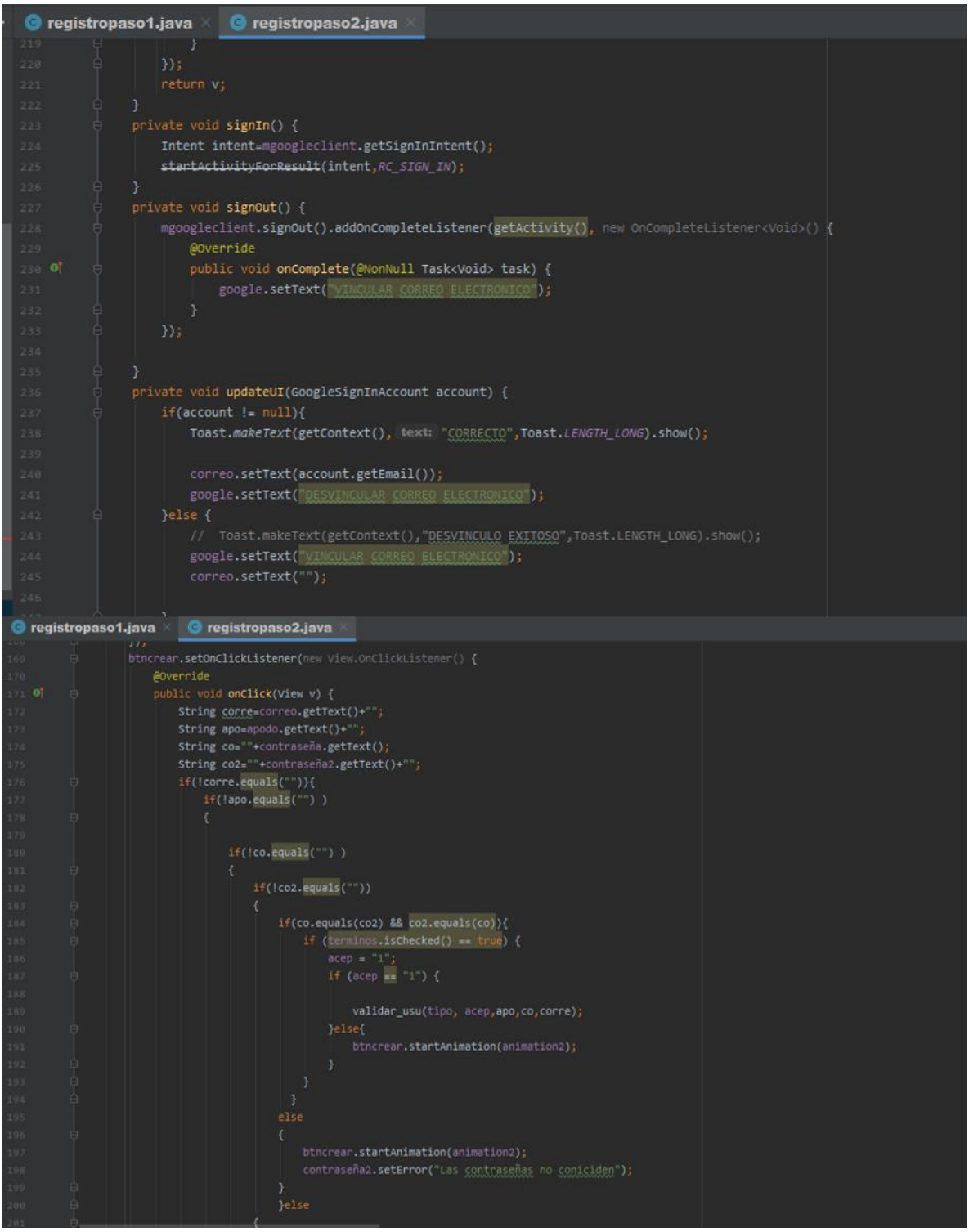
3.8.4 Registro de cuenta 1

El registro se divido en dos fragments agregados en una sola activity para verificar los datos de acceso y el tipo de usuario. Mencionando que al radioButton se le clasifica el tipo de usuario.



3.8.5 Registro de cuenta 2

Aquí se mostró la segunda parte donde verifica el correo si existe dentro de los registros o no. Evidentemente es un protocolo sumamente importante por la duplicación de datos.



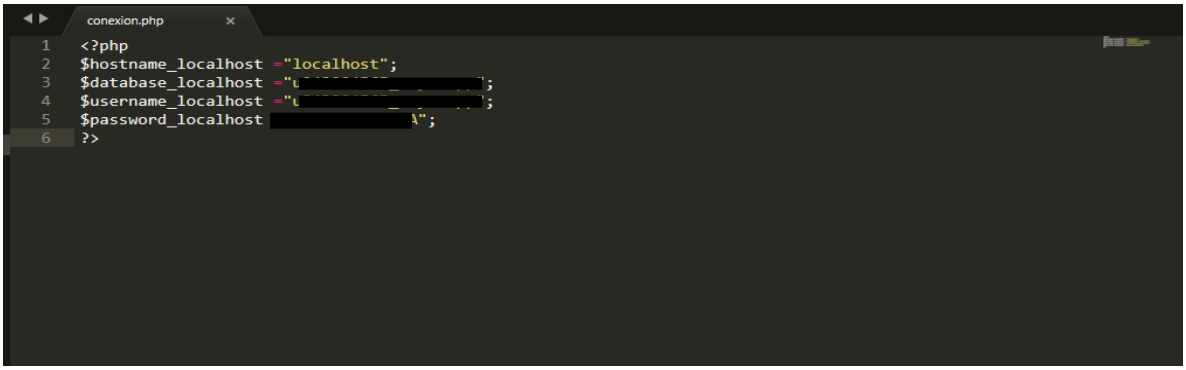
```
registropaso1.java | registropaso2.java

        }
    });
    return v;
}
private void signIn() {
    Intent intent=mgoogleclient.getSignInIntent();
    startActivityForResult(intent,RC_SIGN_IN);
}
private void signOut() {
    mgoogleclient.signOut().addOnCompleteListener(getActivity(), new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            google.setText("VINCULAR CORREO ELECTRONICO");
        }
    });

}
private void updateUI(GoogleSignInAccount account) {
    if(account != null){
        Toast.makeText(getContext(), text: "CORRECTO",Toast.LENGTH_LONG).show();

        correo.setText(account.getEmail());
        google.setText("DESVINCULAR CORREO ELECTRONICO");
    }else {
        //  Toast.makeText(getContext(),"DESVINCULO EXITOSO",Toast.LENGTH_LONG).show();
        google.setText("VINCULAR CORREO ELECTRONICO");
        correo.setText("");

registropaso1.java | registropaso2.java

btncrear.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String corre=correo.getText()+"";
        String apo=apodo.getText()+"";
        String co=""+contraseña.getText();
        String co2=""+contraseña2.getText()+"";
        if(!corre.equals("")){
            if(!apo.equals("") )
            {

                if(!co.equals("") )
                {
                    if(!co2.equals(""))
                    {
                        if(co.equals(co2) && co2.equals(co)){
                            if (terminos.isChecked() == true) {
                                acep = "1";
                                if (acep == "1") {

                                    validar_usu(tipo, acep,apo,co,corre);
                                }else{
                                    btncrear.startAnimation(animation2);
                                }
                            }
                        }
                    else
                    {
                        btncrear.startAnimation(animation2);
                        contraseña2.setError("Las contraseñas no coniciden");
                    }
                    }else
```

3..8.6 Verificación de registro

Por otro lado, verificar es de suma importancia así que se utiliza muchas condicionales para dar lugar al registro y entrada del usuario validando todos los datos necesarios para el acceso.

3.9 Scrips en PHP

Para la conexión con la base de datos en la nube se utilizó scrips de PHP donde se desarrollo en SublimeText el cual se aprecia en las siguientes imágenes el uso de cada una de estas y el por qué se creó la conexión con la aplicación movil mediante la Api Volley



3.9.1 Conexión a base de datos

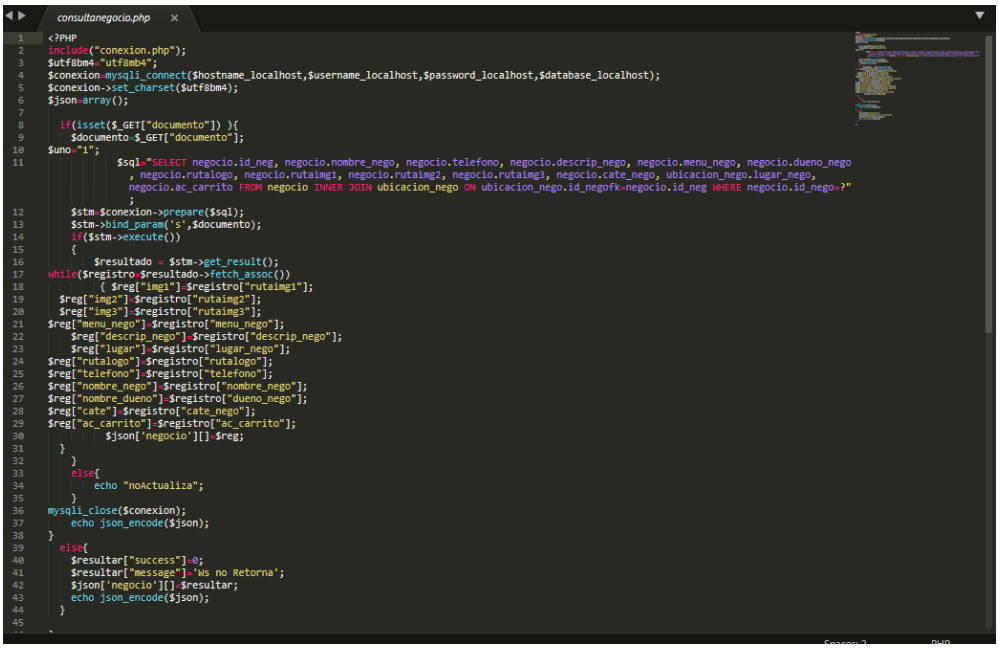
En este archivo se encuentran los datos más importantes que son la contraseña, nombre de base de datos y el nombre de usuario. Cabe recalcar que por privacidad no se muestran, ya que son datos importantes.

consultabusqueda.php

```php
<?PHP
include("conexion.php");

$json=array();
$json2=array();
$clean=array();
$sql=array();;
$conexion = mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_localhost);

  if(isset($_GET["documento"])){
        $documento=$_GET["documento"];
       $limit=$_GET["lim"];
        $offset=$_GET["off"];
        //LIMIT ? OFFSET ?
        $uno="1";
$sql="SELECT negocio.id_nego,negocio.cate_nego,negocio.nombre_nego,negocio.rutalogo,negocio.estrella,ubicacion_nego.lugar_nego,
  ubicacion_nego.nombrelugar_nego FROM negocio INNER JOIN ubicacion_nego ON ubicacion_nego.id_negofk=negocio.id_neg WHERE negocio.cate_nego
  =? AND negocio.ac_nego=? AND negocio.activo_nego=? ORDER BY negocio.estrella DESC LIMIT ?,?";
    $stm=$conexion->prepare($sql);
    $stm->bind_param('sssss',$documento,$uno,$uno,$limit,$offset);
    if($stm->execute())
    {
        $resultado = $stm->get_result();
while($registro=$resultado->fetch_assoc())
        {
    $reg["id_nego"]=$registro["id_nego"];
    $reg["cate_nego"]=$registro["cate_nego"];
$reg["nombre_nego"]=$registro["nombre_nego"];
$reg["nombrelugar_nego"]=$registro["nombrelugar_nego"];
$reg["rutalogo"]=$registro["rutalogo"];
$reg["lugar"]=$registro["lugar_nego"];
$reg["estrella"]=$registro["estrella"];
            $json['negocio'][]=$reg;
  }
    }
    else{
        echo "noActualiza";
    }

    mysqli_close($conexion);
        echo json_encode($json);
}
//-------------------------------------------
  if(isset($_GET["id_usuario"]))
  {

      $idusu=$_GET["id_usuario"];
      $ac="1";
```

3.9.2 Datos con inner join

El modulo principal tiende a llenar el recycleview en cuyo caso, se utilizó este archivo para extraer los datos y como se puede observar al utilizar el INNER JOIN en la consulta solo ocupamos pocos datos ya que recordemos, que obtener todos los datos de un solo negocio y visualizar conlleva al procesamiento de datos lentos. por ende, cada vez que se manda a llamar este archivo solo obtiene 7 datos por las columna en cada fila.



```php
<?PHP
include("conexion.php");
$utf8bm4="utf8mb4";
$conexion=mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_localhost);
$conexion->set_charset($utf8bm4);
$json=array();

  if(isset($_GET["documento"]) ){
    $documento=$_GET["documento"];
$uno="1";
            $sql="SELECT negocio.id_neg, negocio.nombre_nego, negocio.telefono, negocio.descrip_nego, negocio.menu_nego, negocio.dueno_nego
              , negocio.rutalogo, negocio.rutaimg1, negocio.rutaimg2, negocio.rutaimg3, negocio.cate_nego, ubicacion_nego.lugar_nego,
             negocio.ac_carrito FROM negocio INNER JOIN ubicacion_nego ON ubicacion_nego.id_negofk=negocio.id_neg WHERE negocio.id_nego=?"
             ;
    $stm=$conexion->prepare($sql);
    $stm->bind_param('s',$documento);
    if($stm->execute())
    {
        $resultado = $stm->get_result();
while($registro=$resultado->fetch_assoc())
        { $reg["img1"]=$registro["rutaimg1"];
  $reg["img2"]=$registro["rutaimg2"];
  $reg["img3"]=$registro["rutaimg3"];
$reg["menu_nego"]=$registro["menu_nego"];
    $reg["descrip_nego"]=$registro["descrip_nego"];
    $reg["lugar"]=$registro["lugar_nego"];
$reg["rutalogo"]=$registro["rutalogo"];
$reg["telefono"]=$registro["telefono"];
$reg["nombre_nego"]=$registro["nombre_nego"];
$reg["nombre_dueno"]=$registro["dueno_nego"];
$reg["cate"]=$registro["cate_nego"];
$reg["ac_carrito"]=$registro["ac_carrito"];
            $json['negocio'][]=$reg;
  }
    }
    else{
        echo "noActualiza";
    }
mysqli_close($conexion);
    echo json_encode($json);
}
  else{
    $resultar["success"]=0;
    $resultar["message"]='Ws no Retorna';
    $json['negocio'][]=$resultar;
    echo json_encode($json);
  }
```

3.9.3 Selección de negocios o negocio

Este archivo muestra la consulta para seleccionar a los negocios que posteriormente se llama con el Api Volley. A preciando claramente que la consulta al extraer los datos necesarios se almacena en un arreglo el cual se adjunta al json. Este mismo se utilizo de igual forma para mostrar la información del negocio del usuario visualizándose en el módulo de perfil y en el módulo de información de negocios.

consultausuario.php

```php
<?PHP
include("conexion.php");
$utf8bm4="utf8mb4";
$conexion=mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_localhost);
$conexion->set_charset($utf8bm4);
$json=array();

    if(isset($_GET["documento"]) ){
        $documento=$_GET["documento"];

            $sql="SELECT apodo_usu,correo_electronico,contra,tel_usu FROM usuario WHERE id_usu=? ";
    $stm=$conexion->prepare($sql);
    $stm->bind_param('s',$documento);
    if($stm->execute())
    {
        $resultado = $stm->get_result();
while($registro=$resultado->fetch_assoc())
         { $reg["usuario"]=$registro["apodo_usu"];
  $reg["correo"]=$registro["correo_electronico"];
  $reg["contra"]=$registro["contra"];
$reg["tel"]=$registro["tel_usu"];
          $json['usuario'][]=$reg;
  }
    }
    else{
        echo "noActualiza";
    }
mysqli_close($conexion);
        echo json_encode($json);
}
    else{
        $resultar["success"]=0;
        $resultar["message"]='Ws no Retorna';
        $json['usuario'][]=$resultar;
        echo json_encode($json);
    }
```

3.9.1 Consulta de selección de usuario

Este archivo muestra la consulta para seleccionar al usuario que posteriormente se llama con la Api Volley. A preciando claramente que la consulta al extraer los datos necesarios se almacena en un arreglo el cual se adjunta al json.

consultabandejamsj.php

```php
<?PHP
include("conexion.php");
$conexion = mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_localhost);

$json=array();
    if(isset($_GET["id_usu"]))
    {
            $id_usu=$_GET["id_usu"];
            $sql="SELECT titulo_bm,descripcion_bm,fecha_bm FROM bandeja_mensaje
        WHERE id_usu=? ORDER BY fecha_bm DESC";
    $stm=$conexion->prepare($sql);
    $stm->bind_param('s',$id_usu);
    if($stm->execute())
    {
        $resultado = $stm->get_result();
while($registro=$resultado->fetch_assoc())
        {
     $reg["titulo_bm"]=$registro["titulo_bm"];
     $reg["descripcion_bm"]=$registro["descripcion_bm"];
     $reg["fecha_bm"]=$registro["fecha_bm"];
     $json['bandejamensaje'][]=$reg;
        }
    }
    else{
        echo "noActualiza";
    }
        mysqli_close($conexion);
        echo json_encode($json);
    }
        else
        {
$resultar["success"]=0;
        $resultar["message"]='Ws no Retorna';
        $json['negocio'][]=$resultar;
        echo json_encode($json);
        }
?>
```

3.9.4 Bandeja de mensajes

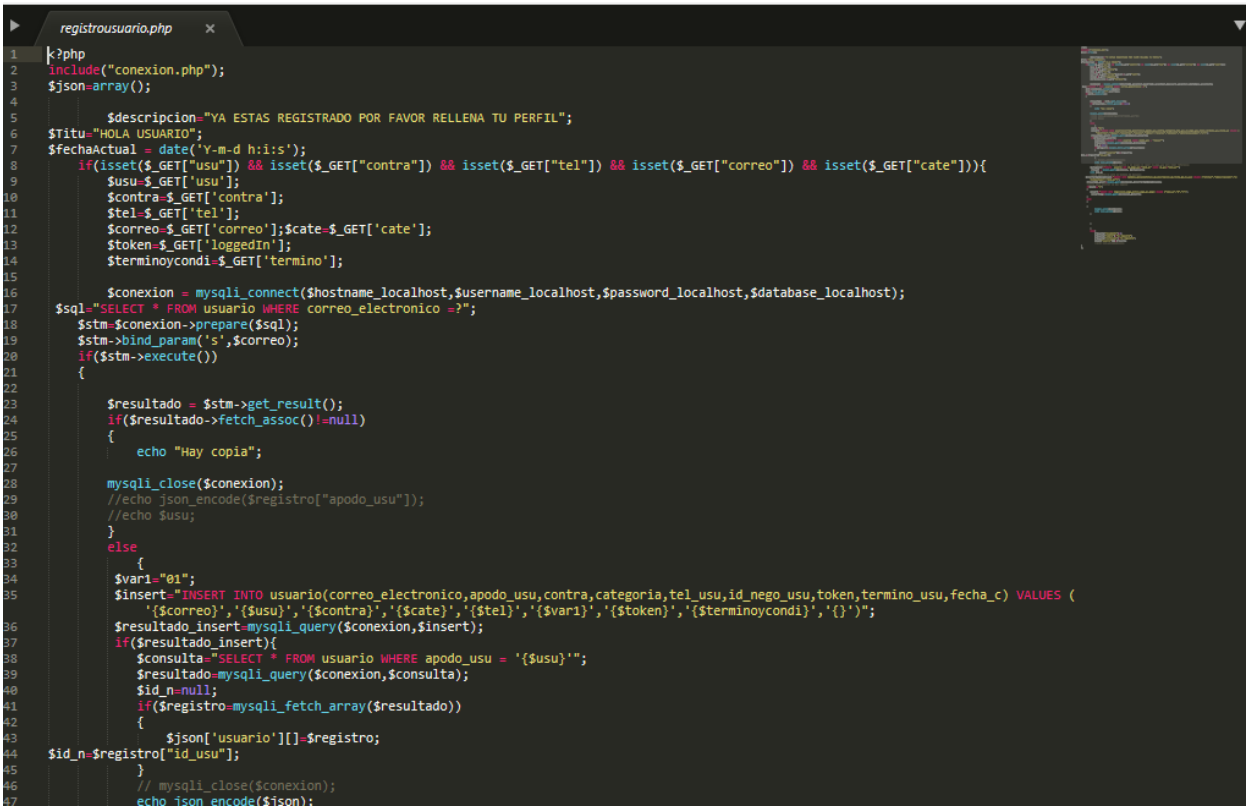
Para extraer datos de la bandeja de mensaje se utilizó el siguiente scrip de PHP el cual indica obtener título, id, descripción, y la fecha del mensaje (Los mensajes solamente se envían directamente a otros intermediarios, pero se almacenas en la base de datos de igual forma no se incluyó una interfaz exclusiva para mensajes en los módulos).

updatefavoritos.php

```php
<?PHP
include("conexion.php");
    $idnego=$_POST['idnego'];
    $idusu=$_POST['idusu'];
    $opcion=$_POST['opcion'];
    $pp=0;
    $conexion = mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_localhost);
    if($opcion=="1")
    {    $sql="SELECT * FROM favoritos WHERE id_nego_fav =? AND id_usu_fav=?";
    $stm=$conexion->prepare($sql);
    $stm->bind_param('ss',$idnego,$idusu);
    if($stm->execute())
    {
        $resultado = $stm->get_result();
        if($resultado->fetch_assoc()!=null)
        {
         echo "existe";
        }
        else
        {
            echo "No existe";
        }
            mysqli_close($conexion);
    }
    else
    {
        echo "No entro";
    }
    }
    else if($opcion=="2")
    {
        $sql="SELECT * FROM favoritos WHERE id_nego_fav =? AND id_usu_fav=?";
    $stm=$conexion->prepare($sql);
    $stm->bind_param('ss',$idnego,$idusu);
    if($stm->execute())
    {

        $resultadox= $stm->get_result();

        if( $resultadox->fetch_assoc()!=null){
            echo "existe";
        }
        else
        {

            $pp=0;
             $insertar="INSERT INTO favoritos(id_nego_fav,id_usu_fav) VALUES(?,?) ";
    $stm2=$conexion->prepare($insertar);
    $stm2->bind_param('ss',$idnego,$idusu);
    if($stm2->execute())
    {

 $seleccionarestrella="SELECT estrella FROM negocio WHERE id_nego =?";
 $stm3=$conexion->prepare($seleccionarestrella);
    $stm3->bind_param('s',$idnego);
     if($stm3->execute())
    {      $resul = $stm3->get_result();
        while($regis=$resul->fetch_assoc())
         {
         $r['es']=$regis['estrella'];
         }
          $pp=$r['es']+1;
         $actualizar="UPDATE negocio SET  estrella=? WHERE id_nego=?";
         $stm4=$conexion->prepare($actualizar);
    $stm4->bind_param('ss',$pp,$idnego);
     if($stm4->execute())
    {
          echo "exito bebe";

    }
    else
    {
        echo "no entro stm4";
    }

        }

        else
    {
        echo "no entro stm3";
    }
    }

    }
    mysqli_close($conexion);
}
        else{
        echo "No entro";
    mysqli_close($conexion);
    }

    }


else{
    echo "error dase de datos no relcionada no concuerda con datos xxxx-xx-xx-xasd%^&";
```

3.9.5 Condiciones para favoritos

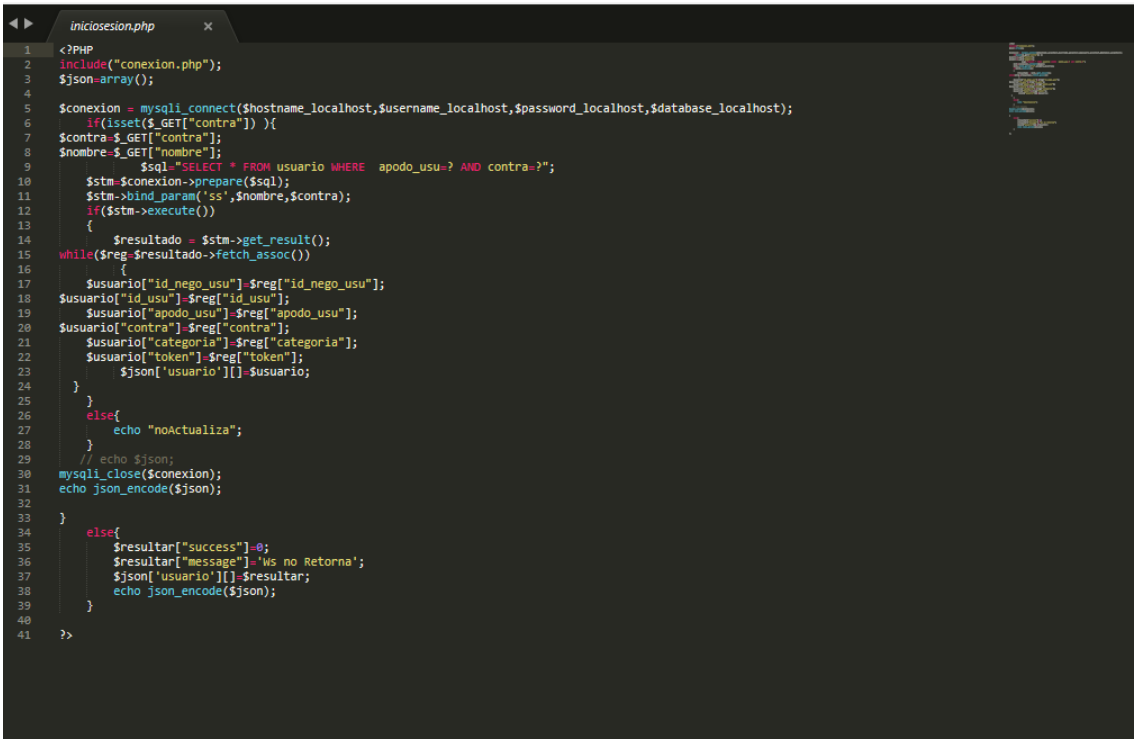
Para que un usuario pueda tener favoritos se hace estas condicionales, uno de los archivos con más líneas de código para la verificación de datos aplicando la selección y actualización de datos.



```php
registrousuario.php
<?php
include("conexion.php");
$json=array();

        $descripcion="YA ESTAS REGISTRADO POR FAVOR RELLENA TU PERFIL";
$Titu="HOLA USUARIO";
$fechaActual = date('Y-m-d h:i:s');
    if(isset($_GET["usu"]) && isset($_GET["contra"]) && isset($_GET["tel"]) && isset($_GET["correo"]) && isset($_GET["cate"])){
        $usu=$_GET['usu'];
        $contra=$_GET['contra'];
        $tel=$_GET['tel'];
        $correo=$_GET['correo'];$cate=$_GET['cate'];
        $token=$_GET['loggedIn'];
        $terminoycondi=$_GET['termino'];

        $conexion = mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_localhost);
 $sql="SELECT * FROM usuario WHERE correo_electronico =?";
    $stm=$conexion->prepare($sql);
    $stm->bind_param('s',$correo);
    if($stm->execute())
    {

        $resultado = $stm->get_result();
        if($resultado->fetch_assoc()!=null)
        {
            echo "Hay copia";

        mysqli_close($conexion);
        //echo json_encode($registro["apodo_usu"]);
        //echo $usu;
        }
        else
            {
         $var1="01";
         $insert="INSERT INTO usuario(correo_electronico,apodo_usu,contra,categoria,tel_usu,id_nego_usu,token,termino_usu,fecha_c) VALUES (
             '{$correo}','{$usu}','{$contra}','{$cate}','{$tel}','{$var1}','{$token}','{$terminoycondi}','{}')";
         $resultado_insert=mysqli_query($conexion,$insert);
         if($resultado_insert){
            $consulta="SELECT * FROM usuario WHERE apodo_usu = '{$usu}'";
            $resultado=mysqli_query($conexion,$consulta);
            $id_n=null;
            if($registro=mysqli_fetch_array($resultado))
            {
                $json['usuario'][]=$registro;
$id_n=$registro["id_usu"];
            }
            // mysqli_close($conexion);
            echo json_encode($json);
```

3.9.6 Verificación de registro

Para el registro de usuario se verifica si existe un usuario con los datos similares a los agregados por el mismo. La verificación consta de contraseña, nombre de usuario y correo electrónico. En dado caso que cumpla los requisitos puede acceder y crear una cuenta insertado a la base de datos.



```php
<?PHP
include("conexion.php");
$json=array();

$conexion = mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_localhost);
    if(isset($_GET["contra"]) ){
$contra=$_GET["contra"];
$nombre=$_GET["nombre"];
            $sql="SELECT * FROM usuario WHERE  apodo_usu=? AND contra=?";
    $stm=$conexion->prepare($sql);
    $stm->bind_param('ss',$nombre,$contra);
    if($stm->execute())
    {
        $resultado = $stm->get_result();
while($reg=$resultado->fetch_assoc())
         {
    $usuario["id_nego_usu"]=$reg["id_nego_usu"];
$usuario["id_usu"]=$reg["id_usu"];
    $usuario["apodo_usu"]=$reg["apodo_usu"];
$usuario["contra"]=$reg["contra"];
    $usuario["categoria"]=$reg["categoria"];
    $usuario["token"]=$reg["token"];
         $json['usuario'][]=$usuario;
  }
    }
    else{
        echo "noActualiza";
    }
   // echo $json;
mysqli_close($conexion);
echo json_encode($json);

}
    else{
        $resultar["success"]=0;
        $resultar["message"]='Ws no Retorna';
        $json['usuario'][]=$resultar;
        echo json_encode($json);
    }

?>
```

3.9.7 Iniciar sesión

Al iniciar sesión los datos se envían al archivo “iniciarsesion.php” el cual verifica los datos correspondientes para enviar devuelta datos necesarios y verificar si existe o no dentro de la base de datos con SELECT siempre y cuando exista el usuario.

Capítulo 4 Privacidad y uso de datos

4.1 Términos y condiciones

COYUCAPP Es una aplicación móvil y nombre de la empresa que ayudara a los negocios a crecer respectivamente en la zona establecida por el negocio, esta tendrá términos y condiciones que ayuden a la comunidad de COYUCAPP a crecer y así poder ser una empresa sincera, sin violencia y un lugar virtual para que todos puedan utilizar la información de manera que tenga una conexión entre los usuarios. La app está sujeta a las reglas de Play Store por lo que COYUCAPP, nombre de la empresa y de la app se hace responsable administrando los usuarios tóxicos para la comunidad.

Como invistado puede:

- Ver negocios e interactuar con el mismo.

Como CO-USUARIO puede:

- Ver negocios e interactuar con el mismo.
- Agregar negocios a tu lista de favoritos.
- Eliminar de tu lista de favoritos el negocio que deseas.
- Acceso a categorías en la principal desde comida, abarrotes, frutas y verduras, ferreterías, Ropa y zapatos, técnicos.
- Puede reportar negocios que hagan daño a la comunidad de COYUCAPP.

Como CO-NEGO puede:

- Modificar el perfil de su propio negocio.
- Los usuarios te podrán agregar en sus favoritos
- La posibilidad de estar en primera fila siempre y cuando tengas más favoritos que otros negocios.
- Te podrán enviar mensaje a la plataforma WhatsApp y también te podrán llamar al número teléfono que agregaste en tu negocio.
- Puede reportar negocios que hagan daño a la comunidad de COYUCAPP.
- Ver negocios e interactuar con el mismo.
- Agregar negocios a tu lista de favoritos.
- Eliminar de tu lista de favoritos el negocio que deseas.
- Podrás tener acceso al servicio de pedidos.

- Tanto como los invistados, los usuarios y los negocios que la utilizan se les prohíbe:

1. Agregar contenido sexual, violento.
2. Agregar contenido que incite al odio.
3. Utilizar el número de teléfono para extorciones.
4. Reportar inadecuadamente un negocio que tiene contenido que no perjudique a la comunidad.

El uso de la marca COYUCAPP es exclusivo y está prohibido hacer referencia de la marca sin consentimiento.
La aplicación móvil COYUCAPP y nombre de la empresa espera que tengas el éxito que necesitas.

4.2 Política y privacidad del usuario

¿Qué pasa con tus datos?

Realmente tus datos son solamente para garantizar una información detallada de los Co-nego a su vez los Co-usuarios puedes obtener dicha información de los Co-nego como el número de teléfono, nombre de los Co-nego, Ubicación, id del usuario. Es para que toda la comunidad pueda tener información de los negocios al alcance sin necesidad de ir o navegar en otros sitios que no son exclusivos de dicho servicio como es este dar información al usuario de lo que se encuentra a su al rededor o más lejos. Los datos del usuario, siendo estos el correo electrónico, Número de teléfono, nombre de usuario y la ubicación en tal caso del Co-nego o Co-usuario no se entregarán a terceros y se centrarán en una base de datos especial para los usuarios de la aplicación móvil coyucapp. Es de suma importancia garantizar privacidad, ya que estos a su vez son para fines informativos y la comunidad de coyucapp está más conectada para facilitar la información para los usuarios registrados en COYUCAPP. Para hacerlo más sencillo lo mostramos desglosado:

1.- Los usuarios registrados en COYUCAPP se registrarán con correo electrónico, el número de teléfono es opcional, pero necesario para que los usuarios puedan comunicarse.

2.- La ubicación es con la finalidad de ubicar Co-negos proporcionando dicha viabilidad y seguridad al encontrar físicamente el lugar.

3.- Los datos están en una base de datos exclusiva de COYUCAPP, estos mismos no se ofrecen a terceros solo se comparte entre los usuarios registrados en COYUCAPP.

4.-Tratamos los datos que recogemos para enviarte comunicaciones relacionadas con nuestros productos y servicios, incluyendo sobre actualizaciones de productos y cambios en nuestras políticas

y condiciones. Si te parece bien que lo hagamos, también podremos enviarte comunicaciones de marketing de vez en cuando.

5.-Tratamos los datos que compartes con nosotros o que nosotros recogemos con el fin de realizar estudios, encuestas y pruebas de productos y resolver problemás, lo que nos ayuda a proporcionar y mejorar nuestros servicios.

6.-Tratamos los datos que recogemos para medir y analizar la eficacia de nuestros productos y servicios y para comprender mejor cómo los utilizas con el fin de mejorarlos.

El correo electrónico es usado para fines de identificación de usuarios. El propósito de utilizar dicha información es para establecer la garantía de que solo se agregan cuentas reales y a su vez se autentica dentro de los servidores de COYUCAPP para verificar si el usuario esta registrado en la aplicación móvil COYUCAPP.

Conclusión y recomendaciones

Así mismo esta aplicación móvil denominada Coyucapp ayuda y facilita la consulta de información de diversos negocios para que los usuarios puedan utilizar en beneficio propio, cabe recalcar que los negocios que se incluyan en la aplicación móvil Coyucapp tiene un lugar exclusivo para que estos mismos puedan agregar o editar su negocio virtual. Recordemos que también los negocios son principalmente de la zona Coyuca de Benítez, Gro y sus alrededores. Evidentemente la aplicación móvil ha concluido, al tener una aplicación móvil con estas funcionalidades los usuarios tendrán mejor visualización de lo que se puede crear en tiempos de pandemia y de manera teórica observando los resultados del front-end, es intuitiva y simplista, sumando estas prácticas se crea Coyucapp, en cuestión de Diagramás UML tenemos la entidad- relación y la relacional, donde se aprecia que al desarrollar la base de datos cumple con los requerimientos necesarios para enlazar tablas con otras, dado que las consultas en la base de datos sean rápidas y efectivas, tomando en cuenta que es un servidor externo y no depende de otros factores como la dependencia de firebase. Como recomendación es necesario aplicar plantillas para el front-end para el desarrollo de una aplicación móvil por razones como tiempo y estética, de igual forma aplicar la metodología scrum en un escenario donde solo hay un solo desarrollador es más tedioso, ya que este tipo de proyectos es más dedicado a equipos para dividir fácilmente las actividades y generar módulos establecidos en la aplicación móvil Coyucapp. Por último, en el transcurso del desarrollo de la residencia, de manera externa de los resultados obtenidos, mediante los objetivos disponibles en el presente documento, se

invitó a Coyucapp a participar en los stand de emprendedores de Coyuca de Benítez, Gro. Donde mostramos a la mayoría de las personas que era y el porqué de su existencia, así mismo se participó en un taller de diseño grafico para mejorar el logo desarrollado en el proceso de la residencia exclusivo de la aplicación Coyucapp. En el caso de subir la aplicación a la Play Store, considero que no fue difícil ya que anteriormente ya se había comprado una licencia de desarrollador de aproximadamente $25 dólares, fue obtenida en el año 2020 no obstante esto no se incluye en los anexos ya que los objetivos como bien se sabe son diferente. La residencia ayudo mucho en el desarrollo de la aplicación móvil Coyucapp, principalmente por la experiencia laboral y el desempeño en base a la actitud obtenida. Gracias a esto también se pudo incluir en los anexos actualizaciones del mes de febrero, por el apoyo que se obtiene de la comunidad dentro de la aplicación móvil Coyucapp es decir, usuarios registrados.

Competencias desarrolladas

- Aplicar conocimiento en forma independiente e innovadora en la búsqueda de soluciones informáticas, con responsabilidad y compromiso social.
- Identificar oportunidades para mejorar el desempeño de las organizaciones a través del uso eficiente y eficaz de soluciones informáticas.
- Liderar procesos de incorporación, adaptación, transferencia y producción de soluciones
- Aplicar metodologías de investigación en la búsqueda, fundamentación y elaboración de soluciones informáticas.
- Asimilar los cambios tecnológicos y sociales emergentes.

Fuentes de información

Android Developers. (2020). Android Studio. https://developer.android.com/studio

Lucidchart. (2019). Lucidchart.www.lucidchart.com

Oracle Developers. (2020) Oracle.www.Oracle.com

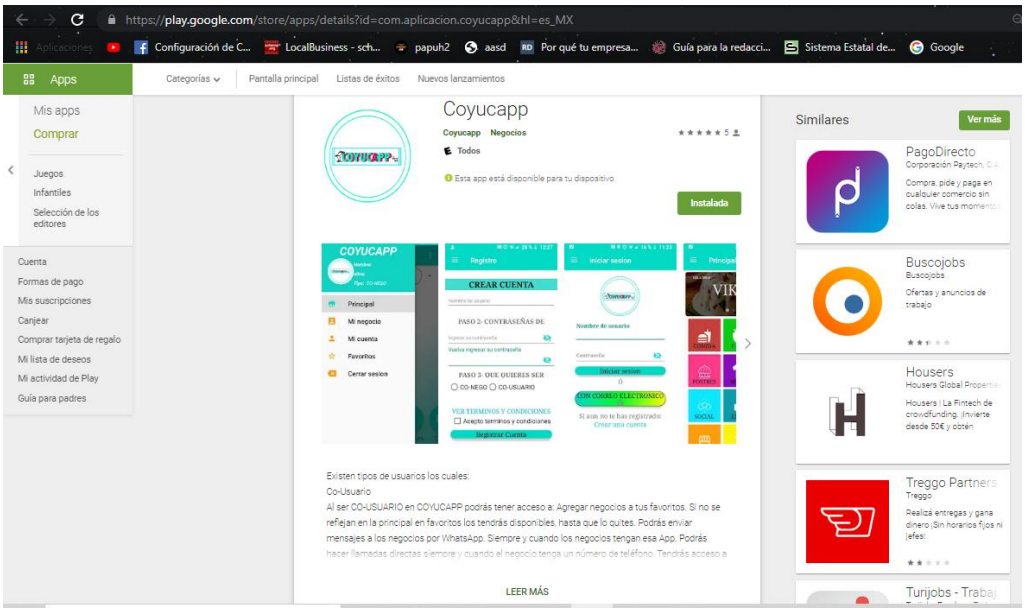
Google Play. (2020). Condiciones de servicio de Google play. https://play.google.com/intl/es_es/about/play-terms/

Grupo Winecta. (2019). historia de las aplicaciones móviles. https://www.appandweb.es/blog/historia-aplicaciones-móviles/

Minegocio. (2018). Aplicación móvil minegocio.https://play.google.com/store/apps/details?id=com.segb_d3v3l0p.minegocio&hl=es_MX&gl=USn

Codedonostia. (2022). https://www.codedonostia.com/

Anexos

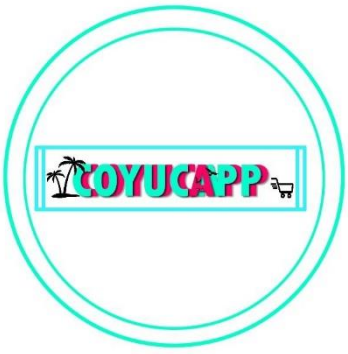


Anexo 1 Coyucapp en la Play Store Disponible.

Anexo 2 Participación del stand de emprendedores del día de muertos 2021

Anexo 3 Participación de Coyucapp en el evento del día de muertos para stand de emprendedores (Reportaje de VO). 2021

Anexo 4 Primeros Negocios registrados en Coyucapp.

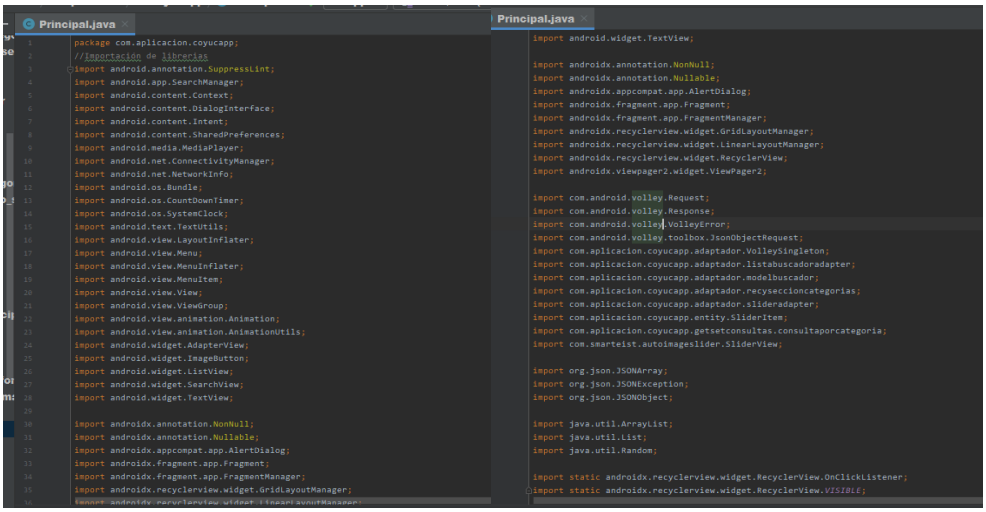


Anexo 5 Logo de Coyucapp

Anexo 6 Participación de Coyucapp en taller de diseño gráfico de parte del Ayuntamiento con la dirección de atención y emprendimiento de la juventud de Coyuca de Benítez, Gro.2021.

Anexo 7 Actualización del inicio de la aplicación móvil Coyucapp por el mes de febrero.

Anexo 8 Actualización de los favs.. de la aplicación móvil coyucapp por el mes de febrero 2022



IAnexo 9 Importación de librerías utilizadas.