

Modelamiento de requisitos específicos para el mejoramiento de comprensión al momento de desarrollar módulo GoU (conductor)

Iván David Chavarro Durán A¹, Nicolas Montagut Chaparro A², Andrés Alejandro Ayure Flores A³

Universidad de Cundinamarca Extensión Facatativá, 1,2: Estudiantes de la Universidad de Cundinamarca, 3: Profesor de la Universidad de Cundinamarca y director del proyecto

Resumen. Actualmente, el transporte público es el medio de transporte elegido por la mayoría de las personas para desplazarse desde su lugar de vivienda hasta su lugar de trabajo o estudio, las TIC's han incursionado en diferentes temas y diferentes ámbitos sociales, entre ellos en el ámbito del transporte público, insertando un nuevo tipo de economía colaborativa [1], brindándole a una gran cantidad de personas la posibilidad de trabajar con horarios flexibles y con herramientas propias como son sus vehículos.

La mayoría de estas aplicaciones se basan en tener una forma de ingreso adicional a la manera tradicional, se caracterizan por ser una aplicación donde el usuario a priori de subirse al vehículo, realiza una búsqueda del lugar al que requiere ir y, el aplicativo en manera de subasta, da la carrera al conductor que sea más rápido en aceptarla, de esta manera está brindando un servicio de movilidad personalizado y puerta a puerta, pero algo costoso sobre el valor del transporte público tradicional.

I. Introducción

La movilidad y la tecnología en los últimos años han logrado acoplarse de tal manera que la mayoría de los aplicativos más importantes de esta década se enfocan en movilidad y transporte particular privado, donde destacan por brindar comodidad, seguridad y eficacia al momento de prestar el

GoU es el aplicativo móvil exclusivo, personal y seguro para la Universidad de Cundinamarca, a diferencia de las aplicaciones de transporte colaborativo como por ejemplo (Uber, Beat, Cabify, etc), permite a la gran cantidad de personas que diariamente se dirigen a su institución en su vehículo propio la oportunidad de viajar acompañados por otra persona de la misma institución que será beneficiada con el transporte exclusivo, seguro y cómodo.

Esta aplicación busca ayudar al personal de la Universidad de Cundinamarca que diariamente debe madrugar a coger un transporte público e indisponerse en él durante el viaje; GoU evitará la incomodidad del “pasito atrás que aún hay espacio” de los transportes públicos y dará la confianza a ambas partes (conductor y pasajero), de saber que la persona que están llevando o el que conduce, es una fracción de la Universidad de Cundinamarca.

Palabras Clave: Carpooling, transporte compartido, Universidad de Cundinamarca, transporte.

servicio. Las aplicaciones móviles enfocadas principalmente a una institución o grupo de personas específicas no son muy vistas, ya que, a pesar de ser rentable, pocos optan por otorgar confianza a dicha exclusividad.

El transporte público, a pesar de ser necesario cuenta con demasiados problemas como

infraestructura, servicio y seguridad, entrando en contexto con el transporte público en el municipio de Facatativá a los diferentes municipios aledaños, representado en un estudio donde se toman en cuenta 8 rutas presentes en la movilidad pública urbana del municipio, donde con una serie de marcadores como la densidad vial urbana, las redes de transporte público y su demanda en zonas de influencia se detalla el flujo en las vías que conectan a Facatativá, además de otros marcadores que ayudan a identificar fallas y problemas en el sistema de transporte público urbano ayudando a la toma de decisiones para mejorar la eficiencia del servicio en el municipio [2]

Uno de los factores que más afectan el transporte urbano e intermunicipal es la congestión del tráfico en varias zonas y horarios, lo cual hace necesario una alternativa al momento de fijar un destino, ya que los atascos de tránsito, el clima y la demanda de los servicios de transporte tienden a afectar a los usuarios al intentar desplazarse de un punto a otro [3]

Las personas hoy en día buscan mejores opciones a la hora de transportarse a sus destinos, como lo es el trabajo, la universidad, incluso donde familiares que en la mayoría de los casos no se encuentran en el municipio donde residen. Dado este problema la opción principal a tomar en cuenta son los medios de transporte público, como el bus intermunicipal o los Taxis, aunque no tan a menudo también toman en cuenta el transporte particular ya sea propio o por medio de aplicaciones en sus dispositivos móviles.

II. Metodologías

A. Metodología de investigación.

Para este proyecto se ocupa la metodología mixta, debido a que, por medio de cuestionarios, se revisará la necesidad de la aplicación para el ámbito universitario, basado en sus resultados cuantitativos se definirá si la aplicación es viable, dado caso que no lo sea, se reajustarán sus planteamientos de forma que por nuevos cálculos cuantitativos se pueda conocer la cantidad de personas beneficiadas al usar GoU.

B. Metodología de desarrollo.

Para este proyecto es Scrum, debido a que habrá transparencia al momento de manipular el código ya que se comunicarán los cambios realizados entre los desarrolladores del proyecto, igualmente se definirán trabajos para cada uno de los miembros del equipo, lo cual permitirá inspeccionar el avance conseguido mediante los sprints; igualmente se busca incluir una sinergia entre SCRUM con XP para tener estándares al momento de codificar, realizando programación dirigida por testeos, teniendo integraciones permanentes de las diferentes ramas que se utilizarán en git a lo largo del proyecto, para que al momento de realizar pruebas unitarias se pueda identificar los errores y así corregirlos rápidamente.

III. Problemática

El transporte público siempre ha sido un medio de transporte indispensable, pero a su vez muy criticado, ya sea por su servicio, las rutas, la seguridad e incluso por los mismos usuarios que incursionan en él diariamente.

En la mayoría de los casos los usuarios del servicio de transporte público optan por alternativas diferentes como adquirir un vehículo, ya sea bicicleta, motocicleta o automóvil o, simplemente buscar otro servicio que les facilite su movilidad a diario. En este caso las aplicaciones de transporte que se encuentran en el mercado como Uber, Didi, Picap, Cabify, etc....son empresas que prestan un servicio a personas que decidan ser conductores, permitiéndoles trabajar con su vehículo y, a usuarios brindando una cómoda y eficaz forma de transportarse.

En Cundinamarca, más del 50% de los viajes a instituciones educativas o empresas son realizados por personas que viven en municipios y se movilizan hacia Bogotá o de personas que viven en Bogotá y se desplazan hacia municipios, Cundinamarca tiene el 28% de la movilidad vehicular a nivel nacional[4], lo que provoca que haya congestión vial diariamente, las aplicaciones de carpooling, permite que la cantidad de personas que necesitan transporte a diario, obtienen la opción

de transporte público, inclinándose por una opción que les brinde seguridad y comodidad.

Abordando el caso de las universidades, la mayoría de los entes que las componen, utilizan el servicio público o privado para movilizarse a las instituciones y aquí es donde se complica la movilidad, ya no solo los integrantes de universidades se transportan, también están los trabajadores y personas del común, lo cual al juntarse todos en horarios similares y con la poca capacidad con la que cuentan estos vehículos prestadores de servicio, ocasionan que el sistema colapse. Aquí es donde entran las aplicaciones de movilidad privada ofreciendo una alternativa eficaz para evitar dichos congestionamientos.

Actualmente, la Universidad de Cundinamarca, no cuenta un servicio de transporte urbano similar al público, prestado para y por los integrantes de la Universidad, si tuviese una aplicación de transporte compartido exclusivo de la Universidad, que dé a los integrantes de la institución educativa una opción de excelente servicio, brindando comodidad con vehículos particulares de la comunidad, donde se le asegure al usuario un asiento (algo que el transporte público masivo no asegura), horarios flexibles, del mismo modo permita tener un horario más amplio al de servicio público, ya que hay horario nocturno que va hasta las 9:30 pm, hora de la última ruta para viaje intermunicipal, entonces. **¿Cómo un aplicativo de movilidad compartida, exclusivo para la Universidad de Cundinamarca, puede facilitar y mejorar el transporte para la comunidad universitaria?** [5]

IV. Modelado

Basado en los once (11) requerimientos funcionales[5], se desarrollaron tres diferentes tipos de diagramas que serán de utilidad al momento de dar a entender la funcionalidad interna del aplicativo y su uso guiado hacia el usuario final.

En este artículo se presentarán los diagramas más relevantes para el funcionamiento del aplicativo móvil GoU conductor.

En un inicio se implementó el diagramado de casos de uso, basándose en los requerimientos funcionales (RF) los cuales se encuentran en el libro de proyecto de grado, previamente planteados en diagramación UML (Unified Modeling Language) buscando especificar los roles y las

actividades que pueden ejecutar con el fin de tener claridad del sistema [5]

A continuación, en la figura 1, se puede entender que existen tres actores, el Usuario (conductor), la aplicación (IU/Back) y la Base de datos (BDD) donde el aplicativo solicitará al usuario que le suministre datos de la cuenta, personales y del vehículo con el que desea ser registrado. El aplicativo hará la validación de toda la información ingresada por usuario, al ser validada la enviará a la base de datos creando un espacio para el nuevo usuario.

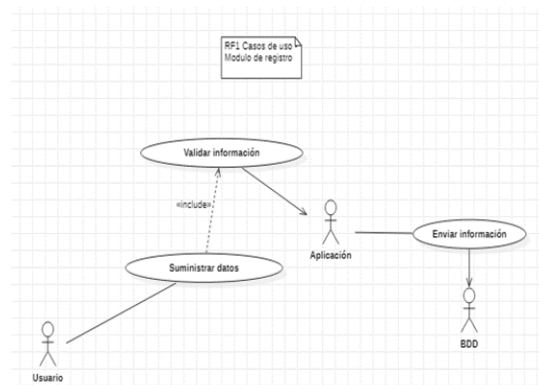


Figura 1. Casos de uso Modulo de registro. [5]

Para continuar, con la figura 2 se puede entender que hay dos actores involucrados, el usuario del módulo, es decir el conductor y la BDD (Base de Datos), donde el aplicativo solicitará al conductor mediante una interfaz los datos necesarios para generar una ruta, como el punto de inicio, destino, fecha, hora, cupos, entre otros datos que lleguen a ser necesarios para la creación de la ruta.

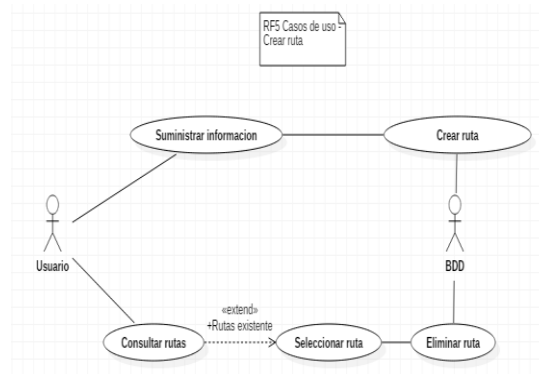


Figura 2. Caso de uso Crear ruta. [5]

Continuando con la explicación del modelado en la figura 2, se puede observar la interacción entre el usuario, la aplicación y la BDD (Base de Datos)

dado que el aplicativo mostrará un listado de rutas realizadas con su respectiva información al usuario, mostrará datos individuales como: punto inicial, destino, hora de partida, fecha, calificación, cupos, cobro, etc.

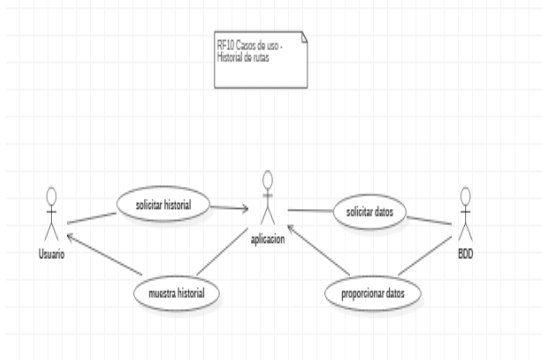


Figura 3. Caso de uso historial de rutas. [5]

Destacando el modelado de la figura 3, se puede detallar la intervención de los tres actores anteriores, usuario del aplicativo móvil (Conductor), aplicación y BDD (Base de Datos), dando a entender que el conductor podrá tener una conversación con los usuarios de la ruta para facilitar la comunicación entre ellos.

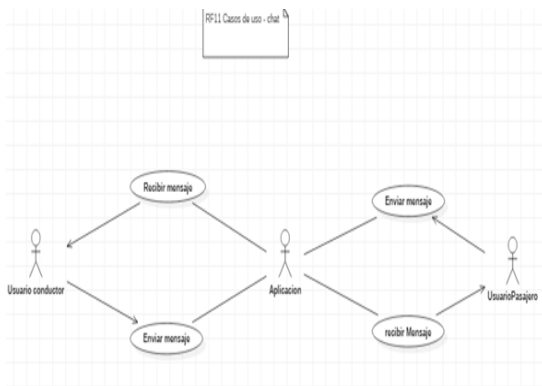


Figura 4. Caso de uso chat. [5]

De igual manera, se implementó el diagramado de secuencia, basándose en los (RF) requerimientos funcionales del Libro de proyecto de grado plantados en diagramación UML (Unified Modeling Language) con el fin de mostrar la interacción del conjunto de objetos de la aplicación a través del tiempo [5].

Dando inicio al modelado de requerimientos en la figura 5, se ven reflejados cuatro actores, donde el usuario que interactúa con la aplicación, es decir el

conductor, tiene que diligenciar el formulario, con los datos indispensables en una correcta creación de la ruta, la interfaz gráfica es la que proporciona toda la ayuda visual intuitiva que facilitara el proceso de creación de la ruta. en donde el usuario cargara los datos solicitados, el controlador se encargará de recibir y validar los datos ingresados, que los campos estén correctamente diligenciados, no se encuentren en blanco y cumplan con los estándares previamente solicitados y, la base de datos recibirá los datos enviados por el controlador, guardándolos y creando un espacio con los datos recibidos generando una nueva ruta, en caso de haber coincidencias en hora y fecha negará la creación de la ruta.

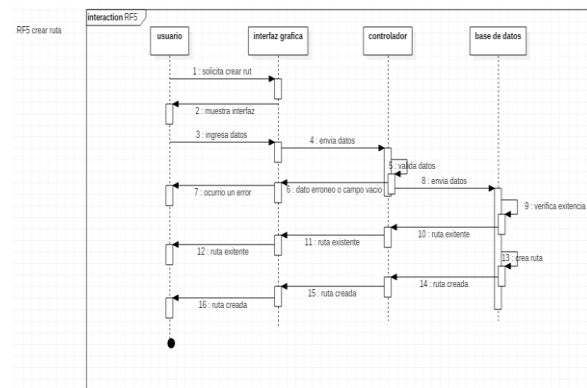


Figura 5. Diagrama de secuencia crear ruta. [5]

Al igual que en la figura anterior, la figura 6 tiene los mismos 4 actores donde el conductor solicita y visualiza el historial de rutas, la interfaz gráfica se encargará de mostrar en pantalla al usuario el historial de las rutas realizadas en forma de scroll panel donde podrá ver la información correspondiente a cada ruta realizada, en controlador se encargará llamar de la base de datos la información correspondiente al historial y la base de datos recibirá la petición y proporcionará los datos requeridos de este módulo

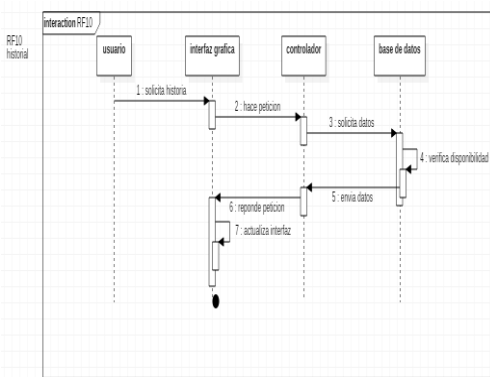


Figura 6. Diagrama de secuencia historial de rutas. [5]

Determinando la funcionalidad propuesta del aplicativo descrita como “chat”, la figura 7 contiene los mismos 4 actores, donde el conductor se encarga de interactuar con la interfaz gráfica enviando y visualizando los mensajes, la interfaz gráfica mostrará la sección de “chat” donde se verán los mensajes enviados y recibidos de la conversación elegida por el usuario, además se desplegara automáticamente el teclado del dispositivo para una correcta conversación, el controlador se encargará de enviar y recibir los mensajes y, la base de datos recibirá los mensajes entrantes y salientes dividiéndolos para así mismo enviarlos a los controladores implicados en el “chat”.

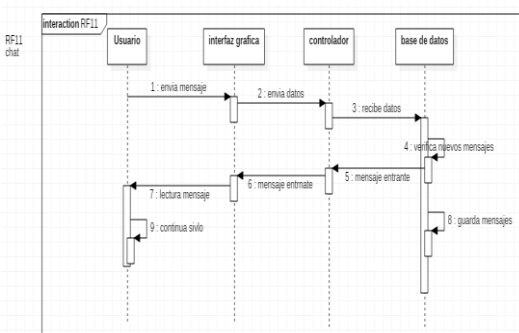


Figura 7. Diagrama de secuencia chat. [5]

De igual manera, se implementó el diagramado de actividades, basándose en los requerimientos funcionales (RF) del Libro de proyecto de grado levantado previamente y plantados en diagramación UML (Unified Modeling Language) con el fin de mostrar el flujo de las actividades ejecutadas por el sistema. [5]

Al observar la figura 8, se puede apreciar el desglose del requerimiento funcional (RF) 5 en diferentes actividades guiadas por un orden específico, donde conductor solicita la interfaz de crear ruta, seguido la interfaz se muestra al conductor para que el conductor llene el formulario de crear ruta y el controlador valide que los campos estén llenos; si no están los campos llenos, el controlador notifica a la interfaz, pero si están los datos llenos, la base de datos valida los datos recibidos con los almacenados. Si no coinciden con ninguna ruta activa, la base de datos crear el registro de la nueva ruta y notifica la interfaz, en este punto puede existir otra respuesta derivada de ese condicional ya que, si los datos coinciden con alguna ruta activa, la base de datos niega la creación de la ruta y notifica la interfaz, por último, el controlador junto con la interfaz informa al conductor que los datos ingresados no pueden ser almacenados porque hay una ruta activa con las mismas credenciales.

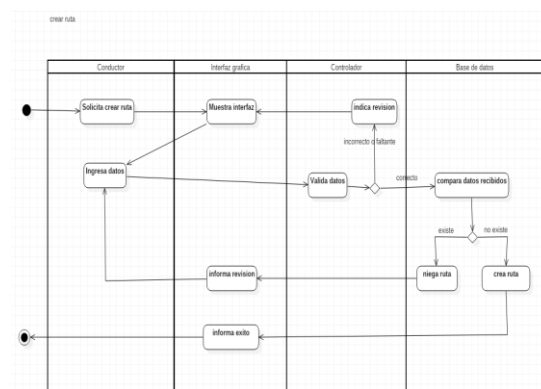


Figura 8. Diagrama de actividad crear ruta. [5]

Después del requerimiento funcional (RF) 5 en la figura 9, se observa el desglose del (RF) 11 donde el conductor ingresa al módulo de chat para que se le muestre la interfaz, seguido a esto el conductor ingresa el mensaje para comunicarse con el pasajero, este mensaje llega al controlador que se encarga de gestionar el mensaje para enviarlo al pasajero, durante este proceso, la base de datos empieza a guardar el chat para retornar el mensaje capturado por el controlador y quedar en espera de la respuesta del pasajero para repetir el proceso anterior pero con el conductor, este punto de mensaje quedará en bucle hasta que se le solicite finalizar la conversación

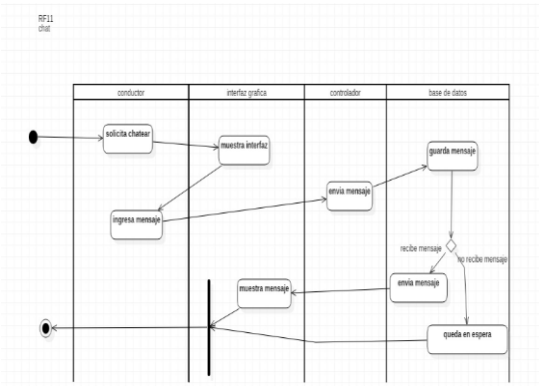


Figura 9. Diagrama de actividad chat. [5]

V. Desarrollo del software

Ya teniendo mayor claridad sobre los requerimientos funcionales, modelado a continuación se evidencia parte del funcionamiento de la aplicación, habiendo realizado un diagrama de clases para tener presente la cantidad de entidades, métodos y atributos que se consumirán a lo largo del desarrollo[5], se inició el desarrollo del aplicativo móvil GoU conductor.

```

package com.example.gouconductor;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class ActivitySplash extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_splash);

        //ANIMACIONES
        Animation animacion1 = AnimationUtils.loadAnimation(this, R.anim.animacionintro_persona);
        Animation animacion2 = AnimationUtils.loadAnimation(this, R.anim.animacionintro_oculo);

        //TEXTOS
        TextView txt = findViewById(R.id.txt);
        //LOGO
        ImageView logoconductor = findViewById(R.id.logoconductor);

        txt.setText(animacion1);
        logoconductor.setVisibility(animacion2);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(getApplicationContext(), ActivityPrincipal.class);
                startActivity(intent);
                finish();
            }
        }, TimeUnit.SECONDS);
    }
}
    
```

Figura 10. Activity Splash. [6]

Continuando con la figura 10 se puede visualizar la clase inicial del aplicativo, nombrada como Activity_Splash la que se encarga de realizar la animación inicial identificando el aplicativo con el logotipo de la marca como se puede apreciar en la (Figura 11).



Figura 11. Activity Splash XML. [6]

```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Método para botón Iniciar sesion
        public void iniciar(View view) {
            //Toast.makeText(this, "hahaha", Toast.LENGTH_SHORT).show();

            Intent intent = new Intent(getApplicationContext(), MenuPrincipal.class); //This indica que se quiere pasar
            // del activity actual, al activity login
            startActivity(intent); // Indica que se inicia el segundo activity
        }
    }
}
    
```

Figura 12. Main Activity. [6]

En la figura 12, 13, se puede visualizar la clase que principal de la aplicación, lo que hasta el momento hará que el conductor pueda elegir el apartado de menú principal, en este artículo se mostrarán los avances del apartado de menú principal (figura 18) con varias funcionalidades activas hasta el momento, lo cual permite a los desarrolladores del módulo, tener un esqueleto solido sobre el cual trabajar.

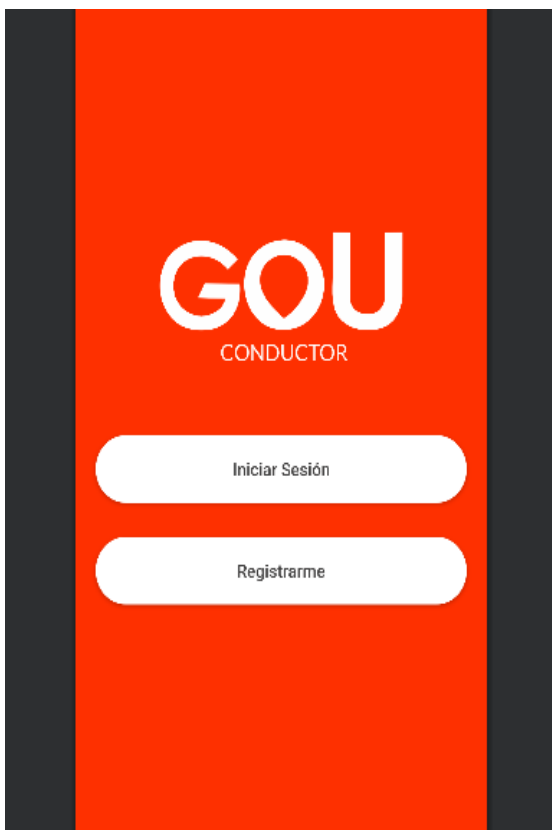


Figura 13. Main Activity XML. [6]



Figura 14. Login Activity. [6]

Esta figura presenta el módulo de iniciar sesión el cual permite el acceso a los usuarios (conductores) a su cuenta solicitando las credenciales

previamente ingresadas al momento de hacer el registro en la aplicación, además de esto, cuenta con un enlace que será utilizado para la recuperación de contraseña a usuarios que olviden sus credenciales.



Figura 15. Recover password. [6]

La figura 15 da una vista al módulo recuperar contraseña realizado con el lenguaje de etiquetas de hipertexto (HTML), Hojas de estilo en cascada (Css), JavaScript y Php como backend utilizando un hosting web gratuito.

Figura 16. Qualify Activity. [6]

Esta figura representa el módulo de calificación de usuarios, donde se puede expresar la recepción de los conductores sobre el comportamiento de los usuarios involucrados en una de sus rutas respectivamente.



Figura 17. Menú Principal. [6]

El menú principal (Figura 17) es el esqueleto principal desde el cual se puede acceder a las diferentes funcionalidades con las que cuenta el aplicativo hasta la fecha, como la calificación de usuario, el historial de viajes y el menú de configuraciones (figura 18) del cual se pueden destacar los módulos, actualización de datos personales, cambio de contraseña y la desactivación temporal de la cuenta.

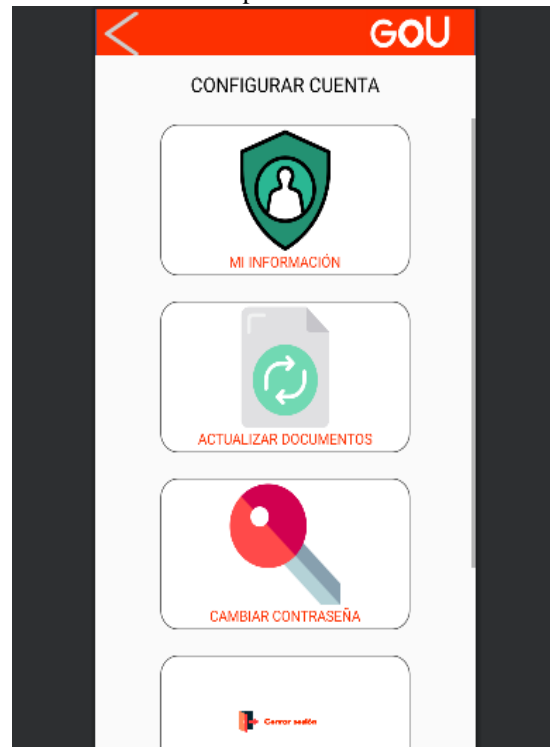


Figura 18. Configuración Activity. [6]

VI. Conclusiones

El modelamiento como técnica de planificación basado en requisitos específicos de un proyecto resulta muy útil ya que este desglosa cada requerimiento y sus actores, facilitando al desarrollador la tarea de comprender el funcionamiento esperado y así lograr generar módulos lo más cercanos a la idea original al momento de plantear el proyecto.

Para finalizar se puede decir que satisfactoriamente este artículo presenta los módulos, modelado, y figuras suficientes para dar al lector un acercamiento directo al proyecto GoU respectivo al módulo Conductor.

VII. Referencias

- [1] M. De Lourdes, V. Arango, J. A. Acevedo Martínez, and A. Ruíz Martínez, “Las TIC en el Transporte Público Urbano y su impacto en la reducción de la marginación en las colonias Los Ángeles y Nazareno del Municipio de Xoxocotlan, Oaxaca,” pp. 17–18.
- [2] C. C. Ordoñez Castillo, “Sistema de indicadores acerca de la cobertura y accesibilidad del transporte público urbano en el municipio de Facatativá,” 2018.
- [3] L. Vencataya, S. Pudaruth, G. Dirpal, and V. Narain, “Assessing the Causes & Impacts of Traffic Congestion on the Society, Economy and Individual: A Case of Mauritius as an Emerging Economy,” *Stud. Bus. Econ.*, vol. 13, no. 3, pp. 230–242, 2018, doi: 10.2478/sbe-2018-0045.
- [4] G. B. N, G. L. J, V. N, V. J, and N. A, “INTEGRACIÓN REGIONAL EN MOVILIDAD,” 2021.
- [5] N. Montagut and I. Chavarro, “Libro proyecto de Grado.” 2021.
- [6] I. Chavarro and N. Montagut, “GoU,” 2021, [Online]. Available: <https://github.com/IvanChavarro/GoU.git>.