Facultad de Economía y Negocios
Escuela de Ingeniería Informática Empresarial

SISTEMA DE GESTIÓN DE RESERVA DE MOTELES PARA “CREAPP SPA”

Autores: Favio Alejandro Ortega Burgos
Héctor Joaquín Tobar Rodríguez

Prof. Guía: Jorge Bozo

Proyecto de memoria para optar al título de INGENIERO INFORMÁTICO EMPRESARIAL.

TALCA – CHILE
2022

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su unidad de procesos técnicos certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.

Talca, 2023

CONTENIDO

ÍNDICE DE IMÁGENES

RESUMEN

En los tiempos que transcurren resulta imposible imaginar rubros en los en los cuales las tecnologías de información no estén presente, de una u otra manera poseen un carácter indispensable a la hora de mejorar o crear un servicio o proceso. En base a esta conjetura y a sus propios análisis, a la empresa "CREAPP SPA" le surgió la idea de optimizar el proceso de gestión de reservas de moteles, esto en base a una aplicación que actúa como **intermediario** entre las partes interesadas. En base a la necesidad de esta empresa, se desarrolló un sistema móvil de gestión de reservas para moteles, el cual permite añadir moteles, habitaciones y ubicaciones por parte del administrador. Y realizar la reserva desde el punto de vista del cliente del motel. Con la finalidad de agilizar el proceso de reserva.

Para realizar el desarrollo de esta aplicación, se utilizó la metodología Proceso Racional Unificado (RUP en sus siglas en ingles), con la cual se realizaron cinco iteraciones, partiendo por la fase inicial que comprendía levantar los requerimientos. Como producto de la aplicación de esta metodología, se obtuvo un sistema fiable de carácter escalable esto gracias a su arquitectura de desarrollo comprendida en MVC (Modelo-Vista-Controlador)

Esta aplicación fue desarrollada en el lenguaje de programación Kotlin para el Frontend, NodeJs para el Backend y SQL, con el gestor de base de datos PGadmin. También se implementó Firebase para almacenar todas las imágenes generadas por la aplicación.

En cuanto a los resultados, se entregó a la empresa una aplicación con una arquitectura correctamente diseñada dada la factibilidad con la cual es escalable el desarrollo de la aplicación, entregando la facilidad de añadir eficazmente módulos que "CREAPP SPA" estime conveniente

Palabras clave: RUP, Patrón de arquitectura de software, Frontend, Backend, Kotlin, Nodejs, postgreSQL, Firebase, API REST, Google Maps, Retrofit, Camara, Git.

ABSTRACT

In these times it is impossible to imagine areas in which information technologies are not present, in one way or another they play an indispensable rol at the time of improving or creating a service or process. Based on this conjecture and its own analysis, the company "CREAPP SPA" came up with the idea of optimizing the motel reservation management process, based on an application that acts as an intermediary between the interested parties. Based on the need of this company, a mobile reservation management system for motels was developed, which allows adding motels, rooms and locations by the administrator. And make the reservation from the point of view of the motel customer. In order to streamline the booking process.

To develop this application, the Rational Unified Process (RUP) methodology was used, with which five iterations were carried out, starting with the initial phase that included the requirements. As a result of the application of this methodology, a reliable and scalable system was obtained thanks to its development architecture comprised in MVC (Model-View-Controller).

This application was developed in the Kotlin programming language for the Frontend, NodeJs for the Backend and postgreSQL, with the PGadmin 4 database manager. Firebase was also implemented to store all the images generated by the application.

As for the results, we delivered to the company an application with a properly designed architecture, thinking on a scalable development, which gives the facility for CREAPP's owners to add modules properly if they deems appropriate. This facilitates the integration of new developers who can be hired to continue with the development and refinement of the app.

Keywords: RUP, Software Architecture Pattern, Frontend, Backend, Kotlin, Nodejs, postgreSQL, Firebase, API REST, Google Maps, Retrofit, camera, Git.

INTRODUCCIÓN

Es indiscutible el rol que hoy en día juegan las tecnologías de información en las empresas, ya que según Correa & Diaz (2018) si se busca excelencia operacional, de gestión o mejorar la capacidad de desempeño, es primordial cortar con herramientas de sistemas de información o gestión, dado que entregan una ventaja competitiva en el mercado. Por tales motivos, es imposible frenar la digitalización de algunos procesos de las empresas, entregarles dinamismo y comodidad a los procesos es crucial para aumentar el flujo de clientes, como se ha visto en otro tipo de mercados, como por ejemplo el de UBER que al convertirse en un intermediario a través de una app, en el proceso de transportar gente alcanzo el éxito empresarial. Este caso se quiere extrapolar a la gestión de reservas de moteles. Dando la posibilidad de que de forma intermediaria, al alcance de la mano, este la opción de solicitar una reserva y ver la disponibilidad de que "X" motel tiene, mediante una app de smartphones.

Destacando que algunos autores como Watson, McCarthy y Rowley (2013) definen las aplicaciones móviles como vehículos de marketing por excelencia. Otros autores como Fang (2019) indican que son herramientas cruciales para atraer negocios y clientes sobre la marcha.

Dado estos criterios la empresa desarrolladora y comercializadora de software CREAPP SPA (contraparte) ubicada en la ciudad de Talca, tiene como principal misión crear aplicaciones nativas para smartphones.

En base a lo indicado en los párrafos anteriores, el gestor de proyectos de CREAPP SPA, tiene como proyecto en carpeta digitalizar y agilizar la gestión de reserva de moteles, de forma cómoda, sencilla y más económica. Ya que según las solicitudes que ha recibido CREAPP SPA por parte de moteles de la zona, concluyen que la instalación de sistemas de gestión ya sea de tipo intermediario como Booking o uno concreto hecho especialmente para un motel, cobran comisiones muy altas o salen muy costosos de implementar, es por esto que ellos desean ofrecer una opción más económica en el mercado. En efecto, un sistema de gestión de reservas de moteles convertido en una app, más allá de su función principal, que es actuar como intermediario entre el cliente y el motel en la gestión de reservas ofreciendo una alternativa más económica, también impulsaría el rubro de moteles en base a lo mencionado, respecto a la influencia de las apps en contextos de marketing y atracción de clientes.

Dado el carácter aplicativo del proyecto, en conjunto con la tesis se desarrollará e implementará en una fase inicial un sistema que gestione la reserva de moteles a lo largo de Chile. Enfocando su desarrollo en una aplicación para plataformas móviles como Android, ya que, aun sabiendo que la mayoría de las aplicaciones hoy en día son de carácter responsiva, la experiencia de uso puede ser aún más confortante cuando una aplicación se desarrolla para entornos concretos.

Respecto a su implementación en fase inicial, se espera que la aplicación cumpla con los requerimientos funcionales, es decir, en una fase preliminar o fase beta. Además se encuentre en la plataforma Google play lista para ser descargada y utilizada.

Los objetivos que se han definido para el desarrollo de esta tesis son:

OBJETIVO GENERAL

- Desarrollar una aplicación móvil que permita facilitar y agilizar la gestión de reserva de moteles.

OBJETIVOS ESPECÍFICOS

- Identificar los requerimientos funcionales, no funcionales y sus respectivas especificaciones para el sistema aplicando métodos de captura de requerimientos.
- Implementar un algoritmo que posibilite la interacción de forma intermediaria entre el cliente y el motel, resguardando el anonimato de clientes mediante registro de usuario y "contrato" de confidencialidad.
- Evaluar el funcionamiento del sistema mediante métricas y pruebas de uso para medir el impacto del sistema.
- Demostrar el funcionamiento de la app en su primera versión de desarrollo, cumpliendo con los siguientes requerimientos:
    1. Registrar moteles con sus respectivos servicios.
    2. Registrar perfiles de usuario
    3. Permitir al usuario poder reservar una habitación de un motel de la zona mediante la aplicación móvil.

Este documento de tesis estará compuesto principalmente por cuatro capítulos estructurados de forma secuencial por ende en principio se desarrollará el marco teórico,

teniendo como finalidad dar una imagen teórica más general acerca del contexto del trabajo que se está realizando, como segundo capítulo se introducirá respecto a las metodologías y se profundizara en torno a la que se llevara a cabo en el proyecto, en cuanto al tercer capítulo se realizara la presentación y análisis de resultado relacionado con el desarrollo del sistema de creación de reservas, finalmente y como último capítulo, se recabaran las conclusiones pertinentes al desarrollo del proyecto.

CAPÍTULO 1 – MARCO TEÓRICO

En esta primera instancia correspondiente al marco teórico del documento, en donde se nutrirá al lector con las definiciones pertinentes para el correcto entendimiento de este proyecto de tesis aplicado, dada la alta cantidad de tecnicismos en la materia, se debe considerar el requerimiento de implementar un sistema mediante una app móvil. Dicho esto, los conceptos analizados a continuación son fundamentales para el desarrollo sin obstáculos de la lectura, por lo que el conocimiento de dichos conceptos son primordiales y se darán definiciones concisas y breves.

1.1 SISTEMAS DE INFORMACIÓN

Inicialmente para conocer el significado de sistemas de información, es necesario descomponer esta frase en las palabras “sistemas” e “información” y establecer en un principio definiciones por separado para eventualmente analizar el resultado al unificar ambas palabras.

Para la palabra “sistemas” existen diversas fuentes que le han otorgado significado, partiendo desde la Real Academia Española que nos dice que sistema es el conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí. Hasta fuentes como IEEE Journals and Magazine, que definen sistema como “un arreglo de partes o elementos que juntos exhiben un comportamiento o significado que los constituyentes individuales no tienen”. Podemos entender que son partes racionales (o no) que interactúan entre si.

Un motor de automóvil está compuesto por diversas piezas y engranajes que en conjunto logran un objetivo mayor que es el funcionamiento del motor. Esto sería un sistema. La radio del automóvil compuesta de cables, botones, placa, es también un sistema. Ahora, estos sistemas mencionados, son sólo parte de un sistema más grande que es el automóvil en sí, el que se

compone de diversas partes, como el motor o la radio, los que pasarían a ser subsistemas al formar parte de un sistema tan grande como es el automóvil.

El alcance de la palabra sistema es mucho mayor al mencionado, y existen libros completos sobre ello en el que se mencionan principios y condiciones, por ejemplo, introducción a la teoría general de sistemas (Johansen, 1993)

Con la idea formada de este concepto podemos analizar el concepto "información" cuya definición radica según Adami (2016) desde un punto de vista matemático, lo que permite hacer predicciones con una mejor precisión (a quien tiene poderío de esta información).

Aunque parezca simplista la definición capta muy bien el concepto, haciendo referencia a la frase, y con el objetivo de nutrir la definición matemática de la información, se sostiene que según Adami (2016) la información que se percibe como inútil en realidad es "entropía", es decir solo incertidumbre sin datos concisos, o también dicho "información inútil"

En base al concepto de incertidumbre es que otros autores como Chiavenato (2007) indican que existen diferentes puntos de vista para el termino información pero que en síntesis todas las definiciones coinciden en el mismo punto, reducir la incertidumbre
Si se hace hincapié en el término "datos concisos", que se mencionó anteriormente, se puede llevar la definición al punto de vista informático, que hace mención al procesamiento de los datos, es decir, lo que se obtiene al tomar cierto conjunto de datos y otorgarle un significado, entonces sin importar el contexto esto unificación de los datos reduce la incertidumbre y se transforma en información.

Al llevar el concepto de información al ámbito empresarial, es imposible negar la importancia que tiene la información, ya que, con un buen uso de esta, se pueden agilizar o mejorar procesos, tomar decisiones estratégicas convenientes, realizar estudios y análisis de mercado asertivos, y un sinfín de aplicaciones con las cuales reduce la incertidumbre de la empresa. En el presente caso se enlazará información para agilizar un proceso de reservas, tomando la información de parte de clientes y de los moteles. Sabiendo cuando, donde y como reservar. Acortando tiempos de espera y entregándole un valor agregado al rubro.

Por lo tanto, sistemas de información se puede comprender como la información o grupos de información en sinergia, interactuando y complementándose para alcanzar un objetivo. Para complementar y reforzar la definición entregada, Hernández (2006) dice que un sistema de información no es únicamente un conjunto de programas y equipos informáticos los cuales se utilizan en la gestión diaria de la actividad productiva; su perspectiva se ha ampliado, evolucionando a lo largo del tiempo y de considerarse como una mera herramienta que disminuía la burocracia y facilitaba las transacciones ha pasado a considerarse un arma estratégica que permite a la organización lograr una ventaja competitiva sostenible.

Otros autores también centran su definición en la totalidad de las ventajas y no solo se centran en la agilización de procesos. Mencionan por ejemplo la capacidad de análisis de los sistemas de información para la toma de decisiones y análisis de problemas.

Según Laudon y Laudon (2004) Como se citó en Fernández (2006) definen los sistemas de información como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores a analizar problemas, a visualizar asuntos complejos y a crear productos nuevos.

1.2 SISTEMAS DE GESTIÓN

Un sistema de gestión según Malagón (2018) se define principalmente como lo que les permite a las organizaciones establecer un mejor desempeño de manera metódica, en base a una herramienta, y que surge dada la necesidad de estandarizar un proceso y adquirir parte de las condiciones necesarias para competir.

Antes de introducir el concepto de sistemas de gestión normalizados, que es lo que precede los sistemas de gestión según un punto de vista esquemático o dicho en otros términos “formalizado”, es necesario definir el concepto norma, que según Malagón (2018) la normalización tiene por objetivo establecer en base a actividades, ante diferentes problemas, métodos que se pueden repetir como un uso común con la finalidad de obtener orden en diferentes procesos o aspectos de la organización según el contexto necesario.

Según Deming (2021) los sistemas de gestión se definen como normalizados cuando el conjunto de elementos está interrelacionado en una organización, es decir, interactúan entre si con la finalidad de establecer políticas y procesos para concretar los objetivos pactados.

Para aplicar un sistema de gestión normalizado, normalmente se realiza en base a el "Circulo de Deming". Este se compone mediante pasos secuenciales que tienen que ser correctamente ejecutados para permitir instaurar los procesos y procedimientos con sus respectivas mejoras necesarias. En base a los conceptos: Planificar, Hacer, verificar y actuar, como se dan a entender mejor en la siguiente imagen en base a sus definiciones.

Ilustración 1: Círculo de Deming.

Fuente: (adaptado de ISO 45001, 2021)

Para mejor entendimiento del concepto y posterior entendimiento de este proyecto, se debe considerar al sistema de gestión normalizado, como una herramienta formal que permite de manera metódica estructurar métodos con el fin de planificar, hacer, actuar y controlar bajo

ciertas normas estándares secuenciales de carácter global, con la finalidad de establecer mejores procesos y medirlo mediante los objetivos pactados.

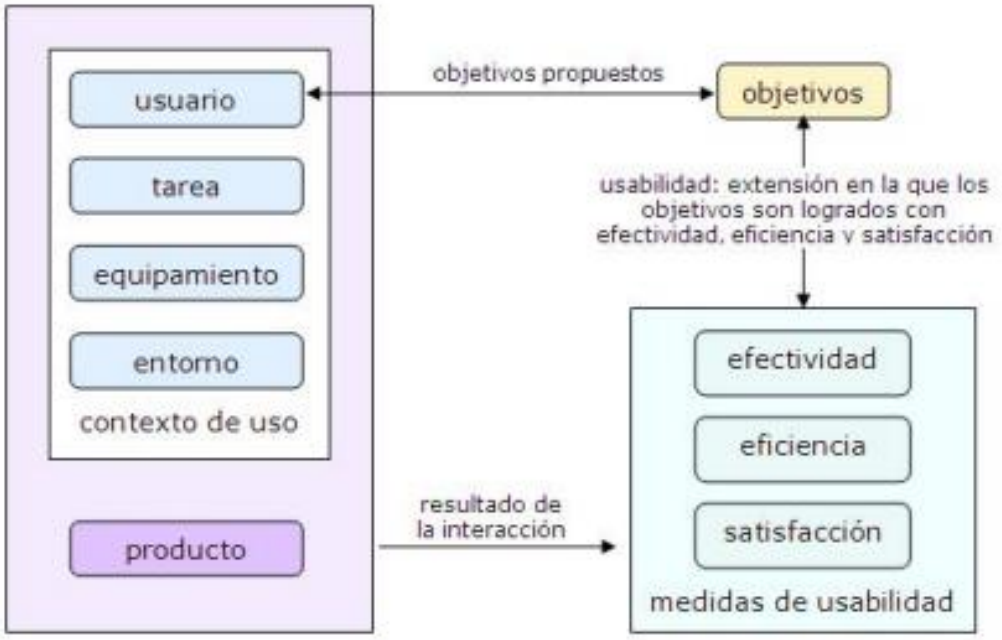
1.3 APLICACIONES MÓVILES

El término aplicación móvil según Zydney y Warner (2016) se relaciona a todos los softwares que fueron desarrollados para ejecutarse en dispositivos móviles en formato app, es decir, un archivo empaquetado ejecutable, que contiene a su disposición en formato "launcher" todo lo necesario para que el software sea ejecutable desde la pantalla de menú o inicio del móvil. Para poder desarrollar este tipo de apps se tiene considerar ciertos aspectos, como por ejemplo las restricciones que tiene cada dispositivo en consideración a su hardware, refiriéndose a dimensiones, procesador, almacenamiento, entre otros.

Cabe destacar que existen principalmente dos categorías de aplicaciones móviles, las webs y las nativas. La primera refiere a aplicaciones que, si o si tienen que estar alojadas en servidores, estas son accedidas por medio de navegadores, por este motivo es fácil lograr su implementación e integración dada la versatilidad que entrega el navegador, aun así, que sea fácil de implementar no significa que saque todo el provecho del dispositivo, ya que por ejemplo no se puede usar la cámara en un entorno web. En cuanto a las aplicaciones nativas, éstas son desarrolladas para un entorno y/o dispositivo concreto, lo que en efecto aumenta las posibles funcionalidades que la aplicación puede entregar en relación al hardware del dispositivo, sacándole un provecho considerablemente mayor, esto se traduce, normalmente, en una mejor experiencia de uso. Por ejemplo, se relaciona con cortos tiempos de carga o como se mencionó anteriormente, menos limitantes en relación al hardware del dispositivo, como lo es el uso de la cámara, gps, entre otros. Aun así, uno de los inconvenientes que surgen en las aplicaciones nativas es en relación a los costos y tiempos de desarrollo, ya que, el desarrollo al estar orientado a plataformas específicas, cada una necesitaría su propio desarrollo.

La siguiente ilustración representa al framework de usabilidad. Esta figura describe el prototipado de un buen producto móvil, el que debe satisfacer los objetivos principales mediante el uso intuitivo o fácil de la aplicación.

Ilustración 2: Framework de usabilidad.



Fuente: (adaptado de Enríquez & Casas, 2014)

El término usabilidad es lo que hace factible la correcta interacción, o la interacción exitosa entre el usuario y la aplicación móvil. Según los autores Enríquez y Casas (2014), al definir la usabilidad utilizan conceptos como efectividad, eficiencia y satisfacción al momento de que el usuario logra sus objetivos mediante este uso exitoso de la aplicación.
Dichas palabras se clasifican como medidas de usabilidad, las que permiten evaluar el grado de usabilidad.

El grado de aceptación que logra tener una aplicación móvil o software por parte de la comunidad recae en lo que cada usuario considere relévate e importante. Aun así, la calidad del producto es sinónimo de aceptación, según el contexto de la ingeniera de software.

1.4 INGENIERÍA DE SOFTWARE

Según Wang (2007) se define que la ingeniería de software como una disciplina que estudia tanto los enfoques, metodologías y el contexto que implica el desarrollo del software.

Según Maida & Pacienzia (2015) se refiere a esta como una rama de la informática, que conjuga un uso razonable de principios o marcos de ingeniería con la finalidad de obtener soluciones a distintas problemáticas según el contexto requerido de la contraparte u organización, esta tiene que ser económicamente factible y que se adapte a las necesidades reales de la empresa. Es importante mencionar que según Wang (2007) la ingeniería de software no solo se limita al desarrollo "duro" del software, sino también a su contexto de desarrollo, refiriéndose a su equipo de trabajo, y las practicas que estos utilizan para su desarrollo. Esperando un alineamiento adecuado en el equipo de trabajo.

En base a diferentes definiciones investigadas como la de Wang (2007). Se infiere que la ingeniería de software es el proceso de fabricar o construir software desde la perspectiva de la ingeniería, es decir, mediante el uso de procesos y metodologías con el objetivo de producir software de calidad, que satisfaga las necesidades de quien desea construirlo.

A continuación, se mencionan dos procesos trascendentales para el desarrollo. La captura de requerimientos y el modelado del software. Dichos procesos existen en diferentes etapas y son previas al desarrollo.

1.4.1 CAPTURA DE REQUERIMIENTOS

Al momento de la construcción de un software, el objetivo es satisfacer los requerimientos. Un profesional capacitado, entregará el software que le solicitan, con las características que le solicitan. Dichas características se clasifican como requerimientos, y son entregadas en una etapa previa a la construcción del software por quien desea que el software sea desarrollado.

Capturar los requisitos correctamente, garantizará que las metas propuestas sobre el desarrollo, sean cumplidas de manera exitosa (IBM, año). Por lo tanto, es un proceso importante y fundamental que debe ser empleado cuidadosamente si queremos lograr el éxito.

Si logramos detallar los requisitos o requerimientos, podemos optimizar el desarrollo mediante el cumplimiento de objetivos específicos y por ende acelerar el proceso completo de desarrollo sin fallos, es por eso que se debe ser meticuloso al momento de gestionar los requisitos. Todo el proyecto girará en torno a dichos requisitos y si no logramos entender lo que nos solicitan, entregaremos un producto que satisface otro objetivo y no el solicitado, por lo tanto sería un software inservible.

Para la compañía IBM, tener una buena gestión de requisitos tiene diferentes ventajas, ya que, esta nos evitará retrasos en los plazos del proyecto, sobrecostos del mismo y además, tendremos un producto de calidad que cumpla con las necesidades del cliente y con los requisitos de seguridad (IBM, año).

Según Sommerville (2011) existen distintos tipos de requerimientos para el desarrollo de un software y/o sistema, los cuales son:

- Requerimientos no funcionales: acá se comprenden todos los requerimientos que no refieren a las funciones específicas del sistema. Se relaciona normalmente con propiedades "emergentes" del sistema, como fiabilidad, capacidades de almacenamientos y tiempos de respuesta. En ocasiones se define las restricciones que debe tener el sistema en el contexto de implementación, como capacidades de dispositivos entre otros.

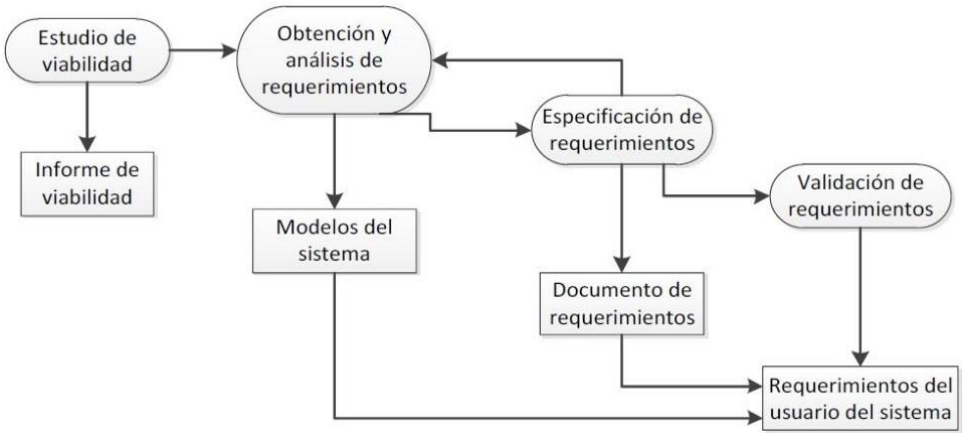
- Requerimientos funcionales: enuncia que debe proveer el sistema, que debe realizar y como tiene que proceder en base a entradas particulares y específicas. En ocasiones también se detalla que no hace el sistema.

- Requerimientos del stakeholder (usuario): estos normalmente son entregados en un lenguaje "natural". En base a diagramas, en un contexto de lo que esperan los usuarios del sistema, y las pertinentes restricciones con las que se espera. Tal caso sea lo más cómodo y simplificado posible.

- Requerimientos del software y/o sistema: se refiere a descripciones que precisan más detalles de las funciones, servicios y restricciones operacionales del sistema. Es decir, un documento que define detalladamente lo que se implementara. Este puede formar parte del contrato del desarrollo del software y puede contemplar o no requerimientos funcionales y no funcionales.

A continuación, se presenta un esquema referente al proceso de requerimientos, teniendo como eje central la "obtención y análisis de requerimientos" y convergiendo en los "requerimientos del usuario del sistema", y como inicio de proceso el estudio de viabilidad de la toma de requerimientos.

Ilustración 3: Proceso de Ingeniería de requerimientos.



Fuente: (adaptado de Aguilar, 2014)

1.4.2 MODELADO DEL SOFTWARE

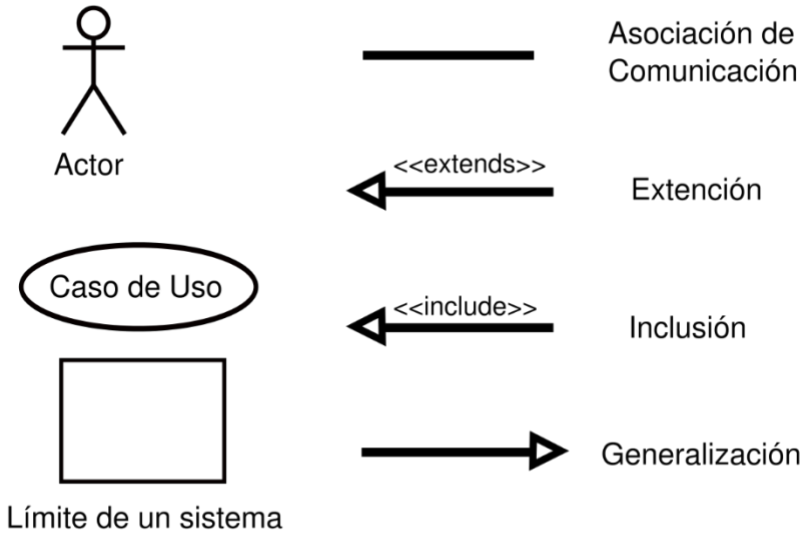
Antes de mencionar la etapa de modelado de software, es necesario mencionar lo que "introduce" esta etapa, que es referente al diseño de software, que por su parte está la antecede la etapa de "Captura de requerimientos" de la cual se detalló anteriormente. Respecto al diseño de software, Según Sommerville (2005) esta etapa corresponde al cómo se aborda y que hace cada requerimiento de la etapa anterior, es decir, como cada módulo da solución a cada requerimiento definido en base a los siguientes procesos. En primer lugar, está la actividad de

dividir requerimientos, con la finalidad de organizarlos de tal forma que sean abordables, posteriormente se **identifican los subsistemas o módulos del software** y así diferenciar las funcionalidades que otorgara el software, después se **asignan los requerimientos a los módulos o subsistemas** con la finalidad de agrupar requerimientos en base a su funcionalidad, el penúltimo proceso refiere a **especificar la funcionalidad de los módulos o subsistemas,** detallando cada función con la que contara dicho subsistema y finalmente se **definen las interfaces del módulo o subsistema,** en ocasiones cuando ya se definen estas interfaces se desarrollan subsistemas en paralelo en base al contexto de requerimientos.

Ya introducido el concepto de diseño de software, se puede explicar el modelado de software. Según Sommerville (2011) esta etapa se define como una forma para dar representación en base a una notación gráfica, y que normalmente hoy en día se realiza en base a UML que traducida sus siglas al español significa Lenguaje de Modelado Unificado. Cabe destacar que según Sommerville (2011) normalmente se consideran más importantes cinco tipos de diagrama dentro de todos los existentes del UML, estos son los diagramas de actividad, casos de uso, de secuencias, de estado y finalmente de clase. Cada uno de ellos con su respectiva importancia en el proceso de **derivar los requerimientos de un sistema**.

En la siguiente ilustración se representa la simbología de uno de los tipos de diagramas mencionados, siendo este específicamente el de Casos de uso. Cabe mencionar que la figura del actor hace referencia al usuario del software y/o sistema, la figura ovalada refiere al caso de uso en particular, siendo esta una acción o tarea a realizar, como por ejemplo lo puede ser “Enviar un recordatorio”. La figura cuadrada refiere al límite del sistema del diagrama, es decir contempla todos los casos de uso que se necesitan.

Ilustración 4: Simbología de diagramas de caso de uso.



Fuente: (adaptado de Flaaut, 2016)

En la siguiente ilustración se representan las explicaciones de la simbología relacional de diagramas de casos de uso.

Ilustración 5: Explicación de relaciones de casos de uso.

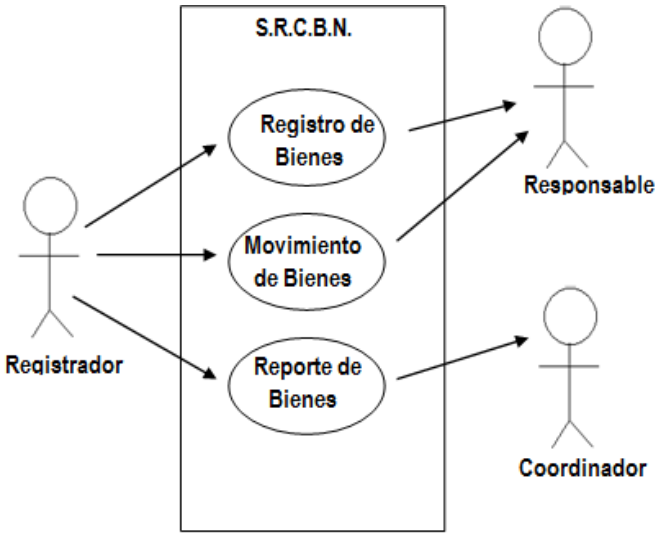
Relación	Símbolo	Significado
Comunica	———	Para conectar un actor con un caso de uso se utiliza una línea sin puntas de flecha.
Incluye	<< Incluye >> <---------	Un caso de uso contiene un comportamiento común para más de un caso de uso. La flecha apunta al caso de uso común.
Extiende	<< Extiende >> --------->	Un caso de uso distinto maneja las excepciones del caso de uso básico. La flecha apunta del caso de uso extendido al básico.
Generaliza	———▷	Una "cosa" de UML es más general que otra "cosa". La flecha apunta a la "cosa" general.

Fuente: (adaptado de Kendall & Kendall, 2011)

Finalmente, con el fin de contextualizar las imágenes anteriores, en la siguiente ilustración se muestra un ejemplo de un diagrama de caso uso con un formato simplificado, perteneciente a un sistema de registro de bienes de Caracas, Venezuela, cada bien debe ser

ubicado en un área en la cual un actor responsable asume el resguardo del mismo. Este sistema constantemente debe generar informes a los coordinadores sobre el estado y la situación de los bienes

Ilustración 6: Ejemplo de caso de uso.



Fuente: (adaptado de Alfonzo, 2013)

En síntesis, un diagrama de casos de uso es clave a lo hora del desarrollo de un software, dado que es una herramienta trascendental a la hora de definir requerimientos y explicar como se quiere que funcione un software. Dando la posibilidad de en el desarrollo corregir requerimientos dependiendo de las especificaciones que entre el feedback de diagramas de casos de uso.

1.5 FRAMEWORKS Y ENTORNOS DE DESARROLLO

En los siguientes puntos introduciremos y explicaremos los términos correspondientes a frameworks y entornos de desarrollo, esto dado la importancia que tienen hoy en día el desarrollo de un sistema y/o software, debido a sus respectivas cualidades permiten que el desarrollo sea, en la mayoría de las ocasiones, más sencillo.

1.5.1 FRAMEWORKS

En simples palabras un framework puede ser, un programa, un marco de trabajo o en ocasiones una guía, todo depende al contexto al que se asocie la definición, para la utilidad del actual proyecto compete asociarlo al desarrollo de software. Indicando que según Rodríguez (2021) es una estructura tecnológica que sirve como guía para el desarrollo y organización de un software. En otras palabras, se define como un esquema con conceptos estructurados, con módulos predeterminados que normalmente sirven como una "base" para un desarrollo, y así no empezar desde cero, ya que, se cuenta con un desarrollo previo que sirve como referencia si el contexto de software es similar. En síntesis, un framework sirve para simplificar procesos de desarrollo, disminuyendo dichos periodos. Aun así en ocasiones la utilización de un framework resulta contraproducente, normalmente esto ocurre en las primeras veces que utilizan, dado que al tener conocimiento de su funcionamiento y/o estructura el desarrollo suele ser más pausado.

En cuanto a las ventajas respecto al uso de frameworks, cabe destacar que el desarrollo en términos económicos, normalmente resulta más atractivo, dado que los tiempos de desarrollo son más acotados y en efecto, menos esfuerzo. Otra ventaja importante a mencionar es que normalmente reduce la cantidad de errores dadas sus funcionalidades integradas y preestablecidas, ya que son "partes de código" que han sido probadas y testeadas anteriormente por diversos desarrolladores. Sin embargo, es importante mencionar que también deben cosiderarse ciertas desventajas en el uso de estas herramientas. Por ejemplo, la falta de optimización del software, que a su vez provoca requerir un hardware más potente para el óptimo funcionamiento del programa, esto debido al código "innecesario" que se genera durante el desarrollo, ya que, al venir empaquetadas las funciones, arrastran funciones que no se requieren para el pertinente desarrollo, sin embargo, estarán generando código inutilizable.

Respecto a los distintos tipos de frameworks, según Rodríguez (2021) normalmente se clasifican en tres tipos, siendo estos denominados como de Caja Blanca, Negra y Gris. El primero está orientado a objetos, y en efecto, basado en herencias. Se necesita un amplio conocimiento del framework en cuestión dadas las jerarquías de herencias. El segundo se caracteriza por ser más fácil de utilizar, pero más difícil de desarrollar, dado que al definir interfaces soporta su extensibilidad, es decir, los componentes pueden quedar enlazados a través

de la creación de objetos. El tercero es una combinación de los anteriores, pero se "encierra" la información interna. Normalmente este incluye la mayoría de las clases a utilizar, siendo el desarrollador el responsable de combinarlas y elegir las apropiadas.

Un framework que se utilizará, será express, este es un framework de nodejs. Dicho framework tiene una amplia usabilidad, y para efectos de este proyecto, usaremos el sistema de enrutamiento y middlewares. De esta manera se hace posible la comunicación entre el frontend Kotlin desarrollado en Android Studio, con el backend node.js desarrollado en Visual Studio Code.

1.5.2 ENTORNOS DE DESARROLLO

Un entorno de desarrollo o más conocido como IDE (por su sigla en inglés Integrated Development Environment, que significa Entorno de Desarrollo Integrado), según Salazar et al. (2011) es todo lo necesario para poder desarrollar software o aplicaciones. Se refiere a todas aquellas herramientas que forman parte del ecosistema en donde el desarrollador está trabajando.

La función del IDE es permitirle al programador integrar aquellas herramientas necesarias para el correcto desarrollo o bien algunos IDE vienen con herramientas ya integradas, y esto se debe a los tipos de IDE que existen.

Ejemplos de características que posee un Entorno Virtual Integrado (IDE) según Salazar et al (2011) son:

- Un editor de código fuente, en donde se escribe y desarrolla el código ya que el IDE proporciona herramientas para autocompletar código y así optimizar el tiempo de escritura. Algunos IDE son compatibles con varios lenguajes de programación y permiten resaltar la sintaxis del lenguaje con el que se está trabajando, así como auto rellenado y auto ordenado del código.
- Un compilador, que permite el empaquetado de los datos para la compilación de estos de manera local y así poder realizar un testeo del código que se está trabajando.
- Un depurador, que básicamente permite probar el programa que se está desarrollando e identifica y resalta los problemas presentes en el código.

Existen distintos tipos de IDE, ejemplos de estos son IntelliJ, Ecplise, Visual Studio Code, BlueJ, Codenvy, y muchos más todos con diversas características y herramientas para facilitar el desarrollo.

Dado el desarrollo a construir, sobre el que habla el presente documento, el IDE que se adoptará para el desarrollo de la parte visual o Frontend será Android Studio dado que según developer.andoid (2021) es el IDE oficial para el desarrollo de apps en Android y es un entorno unificado que permite desarrollar para todos los dispositivos Android.

Por otra parte, para el desarrollo del Backend, se utilizará el IDE Visual Studio Code. mediante este IDE se efectuará la implementación de node.js un entorno de código abierto para la capa del servidor, que se define según su desarrollador nodejs.org (2022) como un entorno de ejecución de JavaScript orientado a eventos asíncronos, el cual está diseñado para el desarrollo de aplicaciones escalables, permitiendo establecer y gestionar múltiples conexiones al mismo tiempo.

1.6 EXPLORANDO SISTEMAS DE GESTIÓN DE RESERVAS

Al momento de indagar por sistemas de gestión de reservas en el gran mundo del internet, nos encontramos con una amplia gama de opciones si de realizar reservas se trata. No obstante, antes de revisar ejemplos, se debe mencionar que, con sistema de gestión de reserva, se habla de un sistema enfocado a eso: La gestión de la reserva. No se refiere a los otros 4 sistemas que forman parte del funcionamiento sistemático de hoteles, moteles, o lugares en donde se deben gestionar reservas de hospedaje.

Según Rondón (2019), existen cinco tipos de sistemas que actúan como herramientas auxiliares, ya sea de hoteles, moteles, hostales o cualquier otro similar. Estos serían:

1. Los sistemas PMS (Property Management System), según Oracle (2022), permiten controlar diversas áreas de un hospedaje como tareas de recepción y reserva, check-in y check-outs registrados, asignación automática de habitaciones, proporcionar la gestión de precios y efectuar boletas o facturaciones. Además, permite la interconectividad con los sistemas auxiliares que se comentarán a continuación.

2. CMS (Channel Management System): Que se encarga de actualizar el inventario de habitaciones disponibles en los canales de venta en los que el hospedaje figure, para no tener que hacerlo manualmente.
3. Motor de reservas online, que es mediante donde el usuario efectúa la reserva.
4. CRM (Customer Relationship Management), que se encarga de adjuntar datos de los clientes y permite usarlos de diferentes maneras, por ejemplo, para el envío de correos o segmentar perfiles.
5. Sistemas RMS (Revenue Management System), que permite a los administradores tomar decisiones relacionadas a los cobros, tomando en cuenta temporadas altas, bajas y conveniencias.

El sistema a desarrollar en el presente proyecto, se basa en el motor de reservas online, mediante el cual el usuario efectuará las reservas.

Según el motor de búsqueda de GetApp (2022), algunos de los sistemas de gestión de reservas existentes son:

- eZee Reservation, es un motor de reserva online y administrador de canales (CMS), que permite gestionar reservas directas sin cobrar comisiones.
- Hotelogix, es un software de gestión hotelera para pequeñas y medianas empresas que se basa en la nube y ofrece inventario de habitaciones disponibles.
- MotelNow, es un motor de reserva online de habitaciones para moteles, muy útil pero que se limita a dos regiones del país de Chile.
- Booqable, es un software de alquiler en línea y en la tienda que permite optimizar el negocio de alquileres. No se especializa en el rubro de moteles, pero sirve como motor de reserva online.
- FareHarbor, es un software de reserva en general, apto para dispositivos móviles y que permite que los usuarios reserven los servicios ofrecidos a través de la app.
- Setmore, es un software de programación en línea para pequeñas empresas.
- Xola, es una plataforma de reservas, marketing y distribución para empresas que ofrecen actividades y experiencias.

CAPÍTULO 2 – METODOLOGÍA

La metodología que se utilizó para el desarrollo del sistema móvil es RUP, por sus siglas, Rational Unified Process, que significa proceso unificado Racional, clasificada como metodología ágil, que indica y estructura el proceso de desarrollo de software, para llevarlo a cabo de manera ordenada, implementando buenas técnicas de ingeniería de software. Según Jacobson et al. (2000), es un proceso de desarrollo de software capaz de soportar el ciclo de vida del desarrollo en su totalidad.

La metodología RUP cuenta con las fases de inicio, elaboración, construcción y transición. Se definirán a continuación las fases de la metodología, con el objetivo de que el lector pueda comprender los contextos en los que se llevó a cabo los procesos durante el desarrollo.

1.1 Fase de inicio: Durante esta fase inicial, lo imprescindible es determinar qué se espera del producto final y así poder determinar el alcance del proyecto, y los riesgos asociados. En el libro "El proceso unificado de desarrollo de software" de (Jacobson, 2000), se recomienda responder a tres preguntas durante la fase inicial:

1.1.1 ¿Cuáles son las principales funciones para sus usuarios más importantes?

Asimismo, Jacobson et al. (2000), señalan que la respuesta de esa pregunta está en los casos de uso más críticos que están contenidos en un modelo de casos de uso simplificado.

1.1,2 ¿Cómo podría ser la arquitectura del sistema?

En esta etapa del proyecto de desarrollo, la arquitectura que se diseña es provisional y estará siempre sujeta a cambios futuros.

1.1,3 ¿Cuál es el plan de proyecto y cuánto costará desarrollar el producto?

Esta pregunta se responde planificando detalladamente la fase posterior a la de inicio que sería la de elaboración, además de calcular un aproximado del costo total del proyecto y deducir los riesgos asociados.

1.2 Fase de Elaboración: La fase de elaboración requiere del desarrollo detallado con foco en dos aspectos:

a) Elaboración de Casos de uso del sistema.

b) Diseño de la arquitectura del sistema.

Se debe desarrollar en esta etapa, modelos estables de casos de uso para una arquitectura sólida. Una vez terminado este proceso, se finaliza la etapa de elaboración con un estimado del costo de desarrollo e implementación.

1.3 Fase de Construcción: En esta fase es el momento de dar inicio a la construcción o elaboración de los sistemas para la posterior integración de las partes del sistema. Si se cumple con los requerimientos base, se puede unificar todo y continuar a transición para elaborar una nueva versión.

1.4 Fase de Transición: En esta fase se toma el sistema diseñado y codificado previamente, se corrigen errores, se evalúan deficiencias y se elabora una versión beta para usos y pruebas. Dichos errores los autores Jacobson et al. (2000), señalan que se recomienda clasificarlos en los que son urgente y requieren repararlo y sacar una nueva versión, y en los errores o deficiencias que pueden esperar hasta la próxima actualización.

La siguiente figura es una representación del RUP en dos dimensiones en la que se observan los flujos de trabajo fundamentales y las fases con sus respectivas iteraciones.

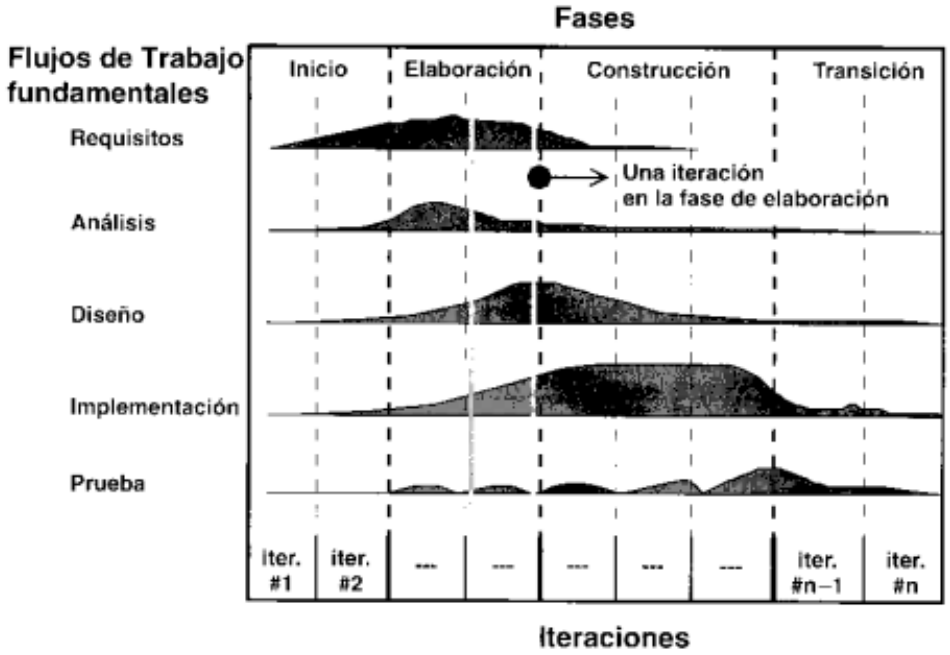
Los flujos de trabajo son cinco:

- Requisitos.
- Análisis.
- Diseño.
- Implementación.
- Prueba.

Mientras que las fases, como se revisó previamente son cuatro, en donde cada una de ellas se subdivide en iteraciones. Normalmente cada iteración recorre los cinco flujos de trabajo y cada fase puede tener tantas iteraciones se estimen necesarias. Las olas de cada flujo o “curvas” como lo llaman Jacobson et al. (2000), representan el alcance del flujo de trabajo en cada una de las fases y mencionan que son solo aproximaciones, y recomiendan no considerarlas muy literalmente.

A modo de un mejor entendimiento para la siguiente figura, Jacobson et al. (2000) concluyen de la figura que los cinco flujos de trabajo, tienen lugar en las cuatro fases.

Ilustración 7: Representación del RUP en dos dimensiones.



Fuente: (adaptado de Jacobson, Booch y Rumbaugh, 2000)

2.1 Fases e iteraciones del proyecto

En este apartado se buscó contextualizar las fases que se han mencionado en el desarrollo de este documento sobre la metodología RUP, enfocándolo en el desarrollo del presente proyecto.

2.1.1 Fase de inicio

En lo que transcurrió esta fase, se buscó analizar y comprender la petición de la contraparte "CREA APP SPA", en base al entendimiento de esta propuesta y las respectivas necesidades por parte de la empresa, se propuso un modelo de producto que asegurara satisfacer la respectiva petición por parte de la empresa de crear un sistema de gestión de reservas de moteles orientada a móviles. Con la finalidad de concretar resultados en esta etapa se agendaron variadas reuniones con la contraparte, dichas reuniones estaban pactadas con la finalidad de reunir la información necesaria para entender las principales funcionalidades que requería el producto para poder así determinar los objetivos y los requerimientos del sistema en cuestión,

para posteriormente realizar un boceto de caso de uso que introdujera técnicamente la compresión estructural de la aplicación.

2.1.2 Fase de elaboración

En esta etapa se buscó definir y desarrollar los orígenes por los cuales se desarrollaría la aplicación, esto en base a los requerimientos tanto funcionales como no funcionales que se reunieron en la primera fase del proyecto.

Con la finalidad de formular de correcta forma la línea base de la arquitectura de la aplicación, fue necesario la creación de la arquitectura de la base de datos, que fue realizado en base a la hilar todos los procesos conceptuales mencionados en la primera fase de desarrollo, más a los que refieren en esta fase.

Es decir, tomando los requerimientos del sistema y los bocetos de caso de uso, se concretó un caso de uso con el cual se crearon dos modelos entidad relación, el primero sin atributos y otro con atributos con el fin de detallar y conceptualizar lo que se desarrolló posteriormente.

Finalmente, y en base a lo anterior, se prosiguió a concretar la realización del modelo relacional lo cual permitió facilitar el análisis y manejo óptimo de los datos que se esperaría manejaría el sistema. Por otro lado, con la finalidad de comprender y graficar de forma más global de la arquitectura de la herramienta se creó un diagrama de paquetes, dado que según Jacobson (2006) facilita la comprensión del sistema en detalle junto a sus demás componentes.

2.1.3 Fase de construcción

En base a la metodología elegida, en esta fase se realizaron cinco iteraciones las cuales se expusieron en el plan de trabajo. En concreto, en esta etapa se dio paso a la realización de la codificación del sistema. Para términos prácticos estas cinco etapas se describen a continuación:

- Iteración 1: En esta iteración se abocó a desarrollar la base de datos, la cual se utilizó como "pilar" para la construcción del sistema, dado a ser el sustento de la arquitectura. Para la realización de la bbdd se utilizó el gestor de base de datos phpMyAdmin.
- Iteración 2: En esta iteración se destinó principalmente a la creación de Mock up's de las vistas de la aplicación, con la finalidad de comprender una idea más visual de cómo se vería posteriormente la aplicación. En conjunto con lo anteriormente mencionado, se realizaron reuniones con la contraparte en las que se obtuvieron un feedback sobre los Mock up's realizados.
- Iteración 3: En lo que competía a esta tercera iteración, se instalaron las herramientas y entorno desarrollo que se utilizaron, además de realizar sistemática y secuencialmente retroalimentaciones conceptuales de las herramientas que se utilizaron para el desarrollo de la aplicación.
- Iteración 4: En esta iteración se procedió a desarrollar el código de la aplicación, contextualizando la arquitectura definida anteriormente en base a los requisitos que entregó la contraparte, también se trabajó en la creación de las interfaces con las cuales constaría el sistema.
- Iteración 5: En esta iteración se investigó, desarrolló e implementó un algoritmo que funciona de intermediario en la interacción cliente-motel, con la finalidad de resguardar los datos del cliente, como se menciona en los objetivos específicos del desarrollo de este libro.

2.1.4 Fase de transición

Ya finalizada la fase o etapa de construcción, se generaron testeos de la primera versión desarrollada, con la contraparte teniendo acceso a todas las funcionalidades que comprende la aplicación. En base al feedback entregado por la contraparte, se realizaron modificaciones puntuales. A su vez se realizaron diferentes testeos del sistema, con la finalidad de corregir ciertos errores y funcionalidades de la primera versión funcional entregada a la contraparte.

CAPÍTULO 3 - PRESENTACIÓN Y ANÁLISIS DE LOS RESULTADOS

El capítulo actual, tiene como objetivo demostrar lo que se consiguió en base a efectuar la metodología descrita en el capítulo anterior, la cual corresponde a la solicitud por parte de "CREA APP SPA" de desarrollar un sistema de reservas para moteles orientada a móviles.

Mencionado lo anterior, en consiguiente se describir dichas fases con sus respectivos resultados:

Fase 1. Inicio

En esta fase, en base a las reuniones realizadas en conjunto con la empresa "CREAPP SPA". Se conceptualizó la solicitud de desarrollar una aplicación móvil que gestione la reserva de moteles, de acuerdo con su necesidad empresarial. Para esto, se trabajó en las diversas reuniones con la finalidad de entregar un objetivo general que contextualizara el requerimiento global de la empresa. A su vez y con el fin de especificar este objetivo general , se desprendieron objetivos específicos, con los cuales se podía comprender más fácilmente de qué forma se concretaría el objetivo general.

Objetivo General

- Desarrollar una aplicación móvil que permita facilitar y agilizar la gestión de reserva de moteles.

Objetivos Específicos

- Identificar los requerimientos funcionales, no funcionales y sus respectivas especificaciones para el sistema aplicando métodos de captura de requerimientos.
- Implementar un algoritmo que posibilite la interacción de forma intermediaria entre el cliente y el motel, resguardando el anonimato de clientes mediante registro de usuario y "contrato" de confidencialidad.
- Evaluar el funcionamiento del sistema mediante métricas y pruebas de uso para medir el impacto del sistema.
- Demostrar el funcionamiento de la app en su primera versión de desarrollo, cumpliendo con los siguientes requerimientos:
    - Registrar moteles con sus respectivos servicios.
    - Registrar perfiles de usuario
    - Permitir al usuario poder reservar una habitación de un motel de la zona mediante la aplicación móvil

En base a los objetivos definidos y a las reuniones con "CREAPP SPA" se definieron los requerimientos tanto funcionales como no funcionales, los cuales se presentan a continuación en las siguientes tablas:

Tabla 1 - Requerimientos Funcionales (Elaboración propia)

Requerimientos funcionales	Explicación
• Almacenar los datos del motel	Al momento de que el usuario "motel" cree el motel los datos se deben almacenar correctamente
• Almacenar los datos del cliente	Crear perfil del cliente para poder gestionar la reserva, con un id de identificación.

• Gestionar la reserva de habitaciones disponibles	Mostrar en pantalla las habitaciones disponibles con sus respectivos precios y fotos.
• Gestionar la cancelación de la reserva realizada	Tener la posibilidad de cancelar la reserva.
• Indicar el periodo que se desea reservar en horas	mostrar los lapsos de tiempo en los cuales se puede reservar la habitación.
• Guardar ubicación satelital del motel.	Que la aplicación almacene la ubicación satelital del motel

Tabla 2 - Requerimientos No Funcionales (Elaboración propia)

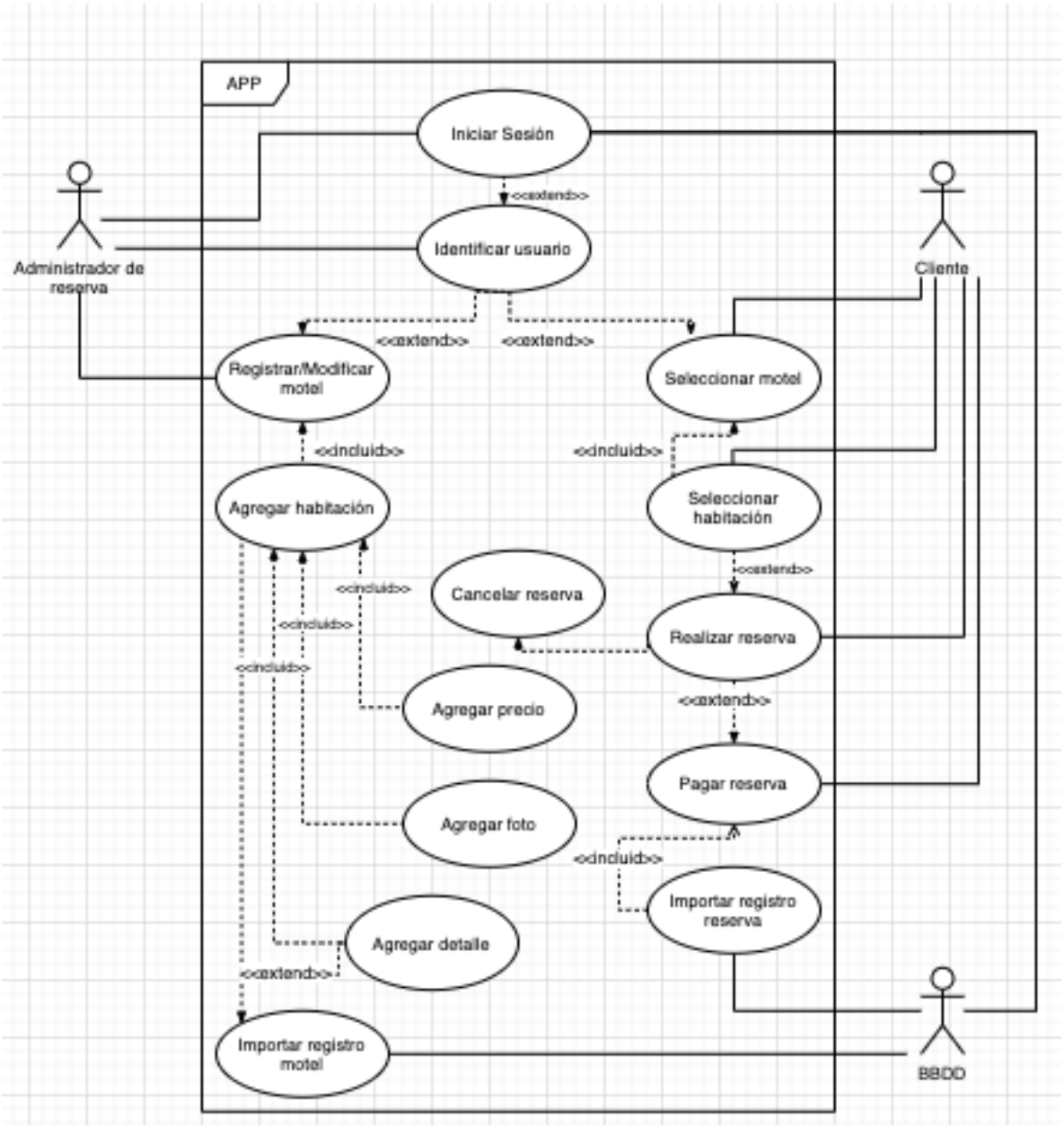
Requerimientos no funcionales	Explicación
• Interfaz gráfica agradable	Que la app tenga una interfaz con un código de colores idóneo según el contexto de la aplicación.
• el sistema debe encontrarse en situación funcional el 98% del tiempo	El sistema debe presentar una disponibilidad horaria casi completa.
• De fácil uso	Que sea intuitivo a lo hora de usar, interpretar e ingresar algún dato o información solicitada.
• Acceso completo a usuario “Motel”	El usuario admin o “motel” tendrá la posibilidad de subir contenido respecto a sus habitaciones y moteles, como por ejemplo fotos de las habitaciones.
• Tiempos de respuesta razonables	Contar con tiempos de respuestas acotados que ayuden a una mejor experiencia de uso.

• Entorno de desarrollo mayormente en Android Studio en el lenguaje Kotlin , base de datos PostgreSQL Y Node JS	Android Studio en su versión 2021.2, Kotlin 1.5 en adelante, PGadmin 4 y Node JS 16.17

Una vez definidos los requerimientos tanto funcionales como no funcionales, se realizó el paso correspondiente a la elaboración de los casos de uso, ya que, dada la existencia de los requerimientos, la elaboración de esos diagramas resulta más sencilla. Tanto como en el boceto como en el diagrama final se ejemplificaron los atributos que el sistema debiera contener, con sus respectivos límites. Destacando en concordancia que el primer boceto tendrá un enfoque más simplificado,

A continuación, se presenta el primer boceto de uno de los diagramas de casos de uso, perteneciente al modelado de la app, en este modelo se puede aprecia la presencia de tres actores los cuales se identifican como “Administrador de reserva”, “Cliente” y “BBDD”. En este diagrama se puede observar según los diferentes casos de uso que la aplicación tendrá una interfaz subdividida, una enfocada al cliente y al proceso de reserva y otra enfocada al administrador del motel y al proceso gestionar las habitaciones y las reserva

Ilustración 8: Diagrama de casos de uso – UML (boceto)



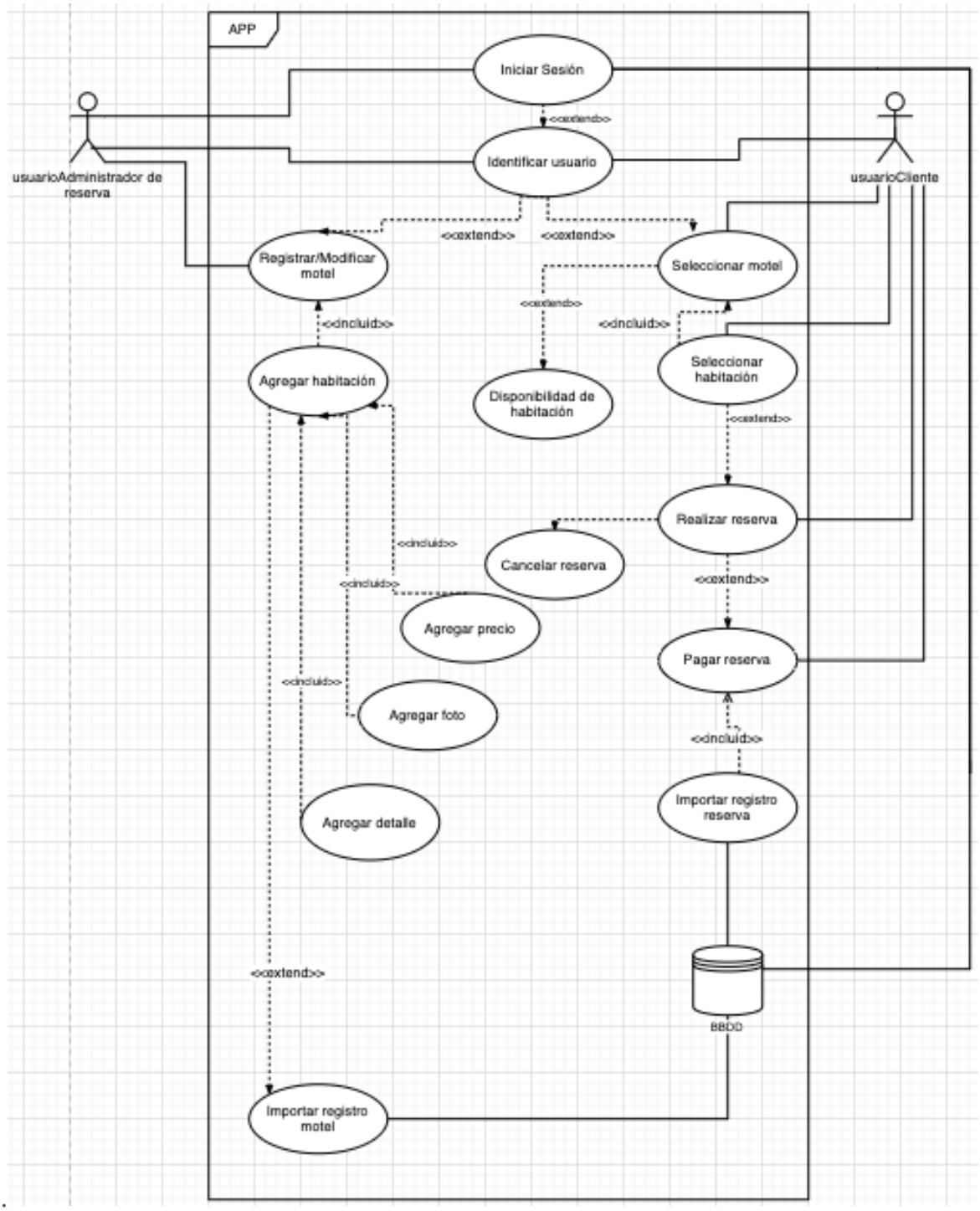
Fuente: Elaboración propia mediante draw.io

Fase 2. Elaboración

En la primera instancia se elaboró el boceto del caso de uso, en base a esto se elaboró el diagrama final con el cual se logró identificarlas las funcionalidades que tendrá este sistema, estas actividades se grafican en óvalos, también se puede observar que los usuarios se representan por actores, que son dibujados con una "caricatura simple" de una persona, con el objetivo de lograr identificarlos y representarlos, esto se puede observar en las Ilustraciones 8 y 9 respectivamente. La versión final de caso de uso, ya no está simplificada, detallando que la base de datos no se ejemplifica como un actor, como si se hace en el primer boceto.

También cabe agregar que las interacciones entre los actores y los casos de uso están representadas por flechas, al igual que sus dependencias con flechas directas o puntuadas, estas últimas se utilizan para relacionar los óvalos, los cuales según su dependencia se clasifican como: "extend" o "incluid", haciendo referencia a la dependencia que tienen uno del otro, es decir si se extiendo o si depende directamente.

Ilustración 9: Diagrama de casos de uso v2



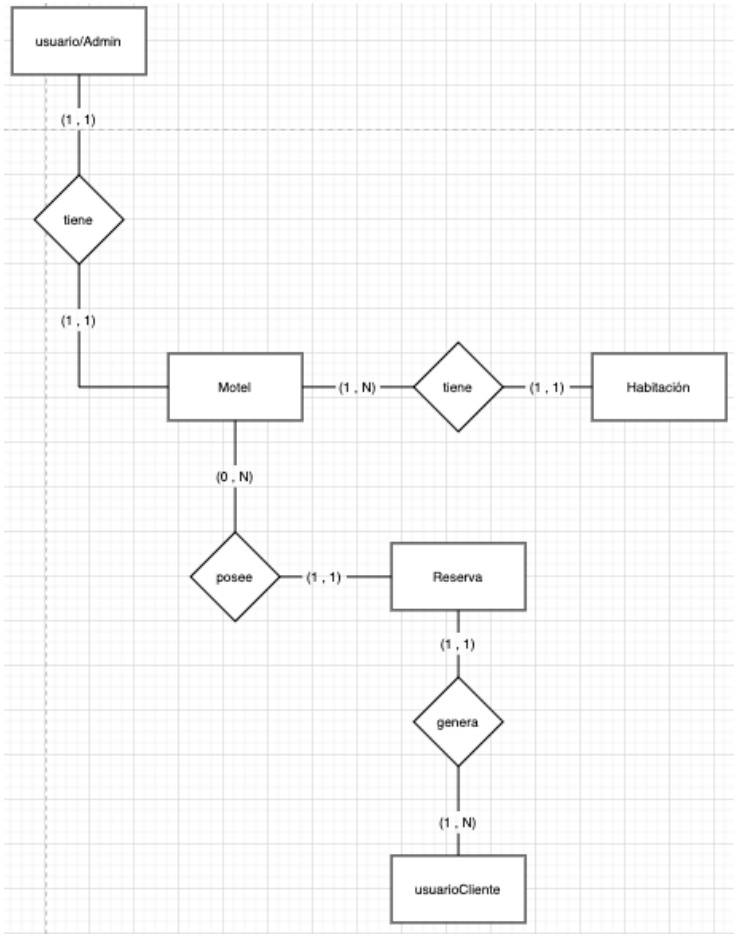
Fuente: Elaboración propia mediante draw.io

Posterior a la elaboración de los casos de uso y siguiendo el plan de trabajo, se elaboró el modelo de entidades y relaciones para dar forma y estructura a la arquitectura que se convertirá posteriormente en las bases del software.

Las entidades se representan con un rectángulo y las relaciones con un rombo. Los números entre paréntesis representan la cardinalidad y grado de participaciones de A en B.

El siguiente modelo es una primera versión. A modo de ejemplo, la relación Motel posee Reserva, se debe leer como: 1 motel posee 0 o muchas reservas (N), mientras que una reserva solo puede ser poseída por 1 motel.

Ilustración 10: Modelo entidad relación v1.

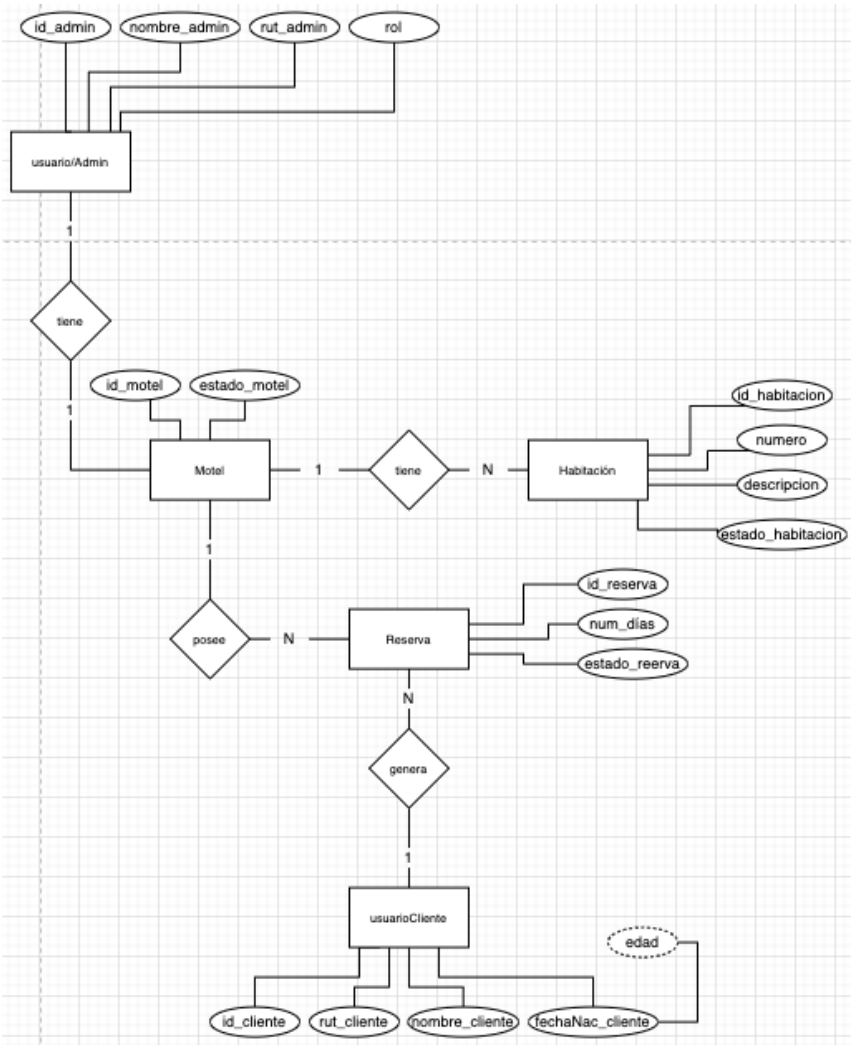


Fuente: Elaboración propia mediante draw.io

Posterior a la elaboración del MER v1, se elaboró el MER v2 de manera más detallada para dar forma y estructura a la arquitectura que se convertirá posteriormente en la el modelo relacional, el que muestra el modelado de las tablas de la base de datos.

Las entidades se representan con un rectángulo y las relaciones con un rombo. Los números entre paréntesis representan la cardinalidad y grado de participaciones de A en B.
El siguiente modelo es una versión más detallada. A modo de ejemplo, la relación Motel posee Reserva, se debe leer como: 1 motel puede poseer muchas reservas (N), mientras que una reserva solo puede ser poseída por 1 motel.

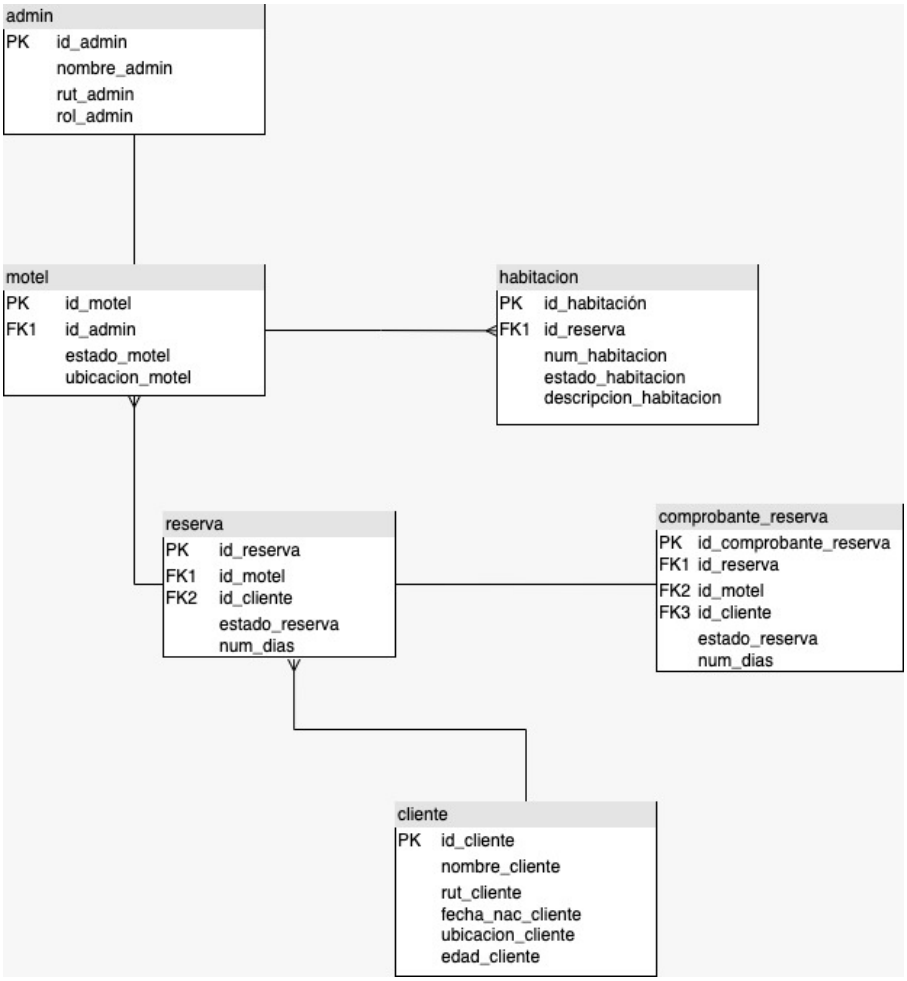
Ilustración 11: Modelo entidad relación – MER v2.



Fuente: Elaboración propia mediante draw.io.

Posterior a la creación de los modelos entidad relación, y en base al mismo, se creó el modelo relacional, el cual se puede observar en la Ilustración 12, con sus respectivas relaciones atributos y entidades con los cuales cuenta la estructura de la base de datos. Como se ha evidenciado y siguiendo la estructura del plan de trabajo, a la fecha se llevaba realizado los requerimientos funcionales, no funcionales, bocetos de casos de uso y modelos finales, modelo entidad relación y el respectivo modelo relacional. Enunciado esto posteriormente se continuo con la presentación de prototipos que ejemplifican gráficamente la arquitectura del sistema, para concluir con la fase de inicio del desarrollo del sistema

Ilustración 12: Modelo relacional.

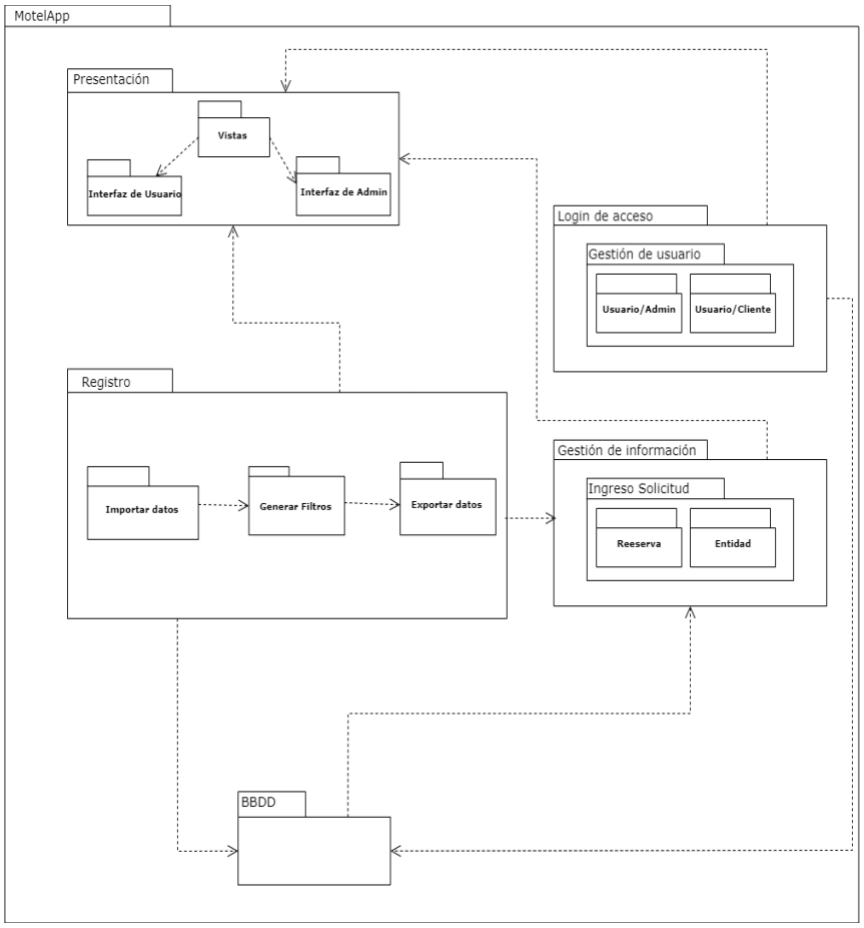


Fuente: Elaboración propia mediante draw.io

Ya elaborado el modelo entidad relación, se diagramó mediante el lenguaje unificado de modelado un diagrama de paquetes, con la finalidad de graficar la arquitectura de la herramienta.

En el diagrama que se presenta a continuación, en la Ilustración 13, engloba como paquete general “Motelapp” ya que contiene los principales elementos del software, dentro de este paquete existen paquetes más pequeños o agrupaciones de elementos, los cuales están relacionados con otros grupos de carpetas o paquetes diferentes, existiendo relaciones que explican funcionalidades distintas dentro del sistema. La finalidad con la cual se construyó este diagrama es poder tener una vista más concisa de la organización los distintos elementos que componen el software.

Ilustración 13: Diagrama de paquetes de arquitectura de la herramienta.



Fuente: Elaboración propia mediante draw.io

Fase 3. Construcción

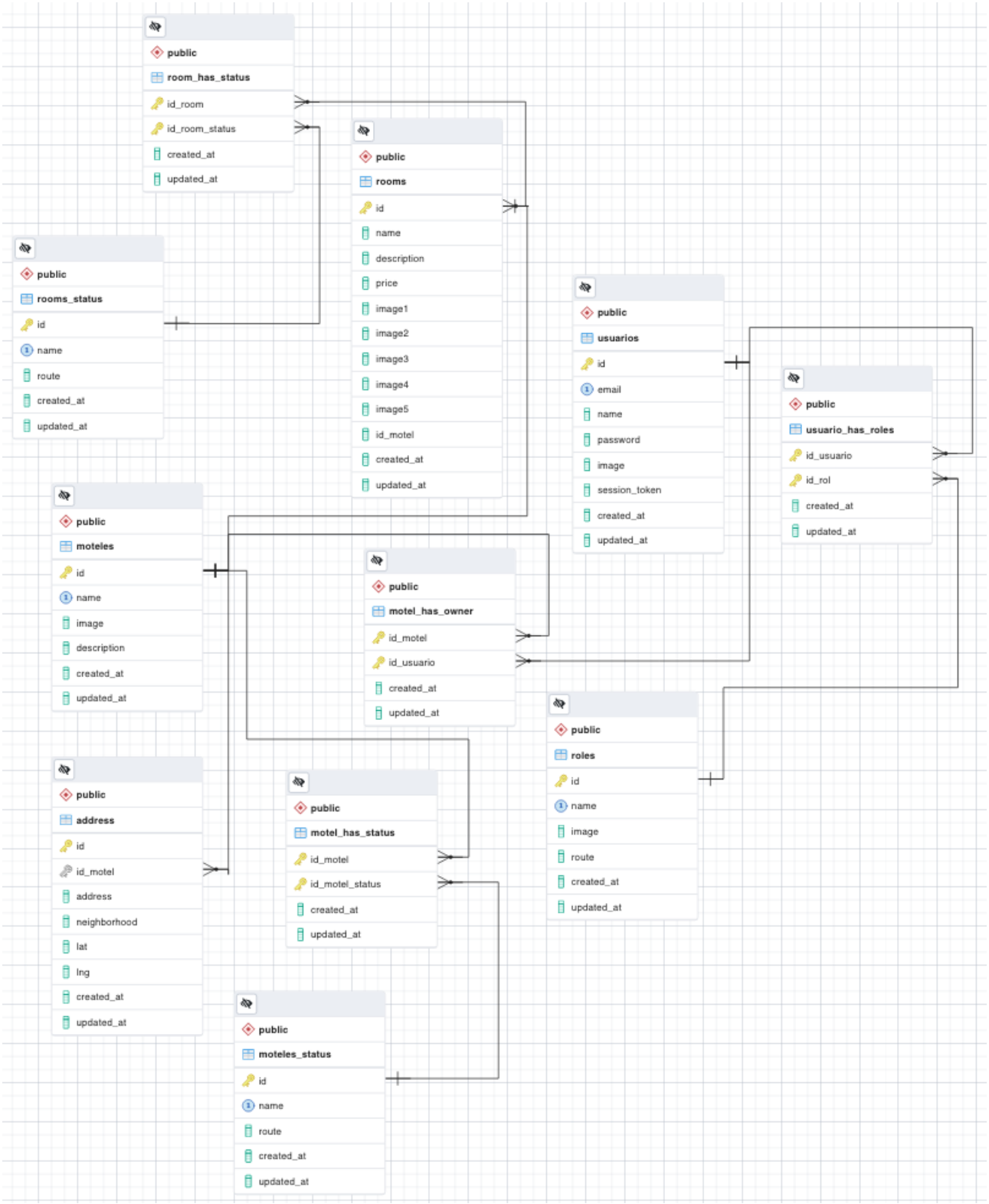
Esta fase consistió en el desarrollo del sistema de gestión de reserva de moteles, la cual se realizó mediante iteraciones, como se detalla en el plan de trabajo.

Iteración 1:

La primera iteración correspondió al desarrollo de las tablas del modelo relacional realizado en la fase anterior, correspondiente a la elaboración. En base a este modelo se creó la estructura de bbdd la cual posteriormente se implementaría en el sistema de reserva de moteles.

Como se logra observar en la Ilustración 14, las tablas se conectan a través de claves foráneas como por ejemplo lo es “id_reserva” en la tabla “habitación”. Dado que esta información se encuentra relacionada se logra comprender de forma más simplificada y ordenada, los cuales son puntos cruciales para poder tener un buen manejo de los datos en las iteraciones posteriores del desarrollo del proyecto.

Ilustración 14: Estructura BBDD.

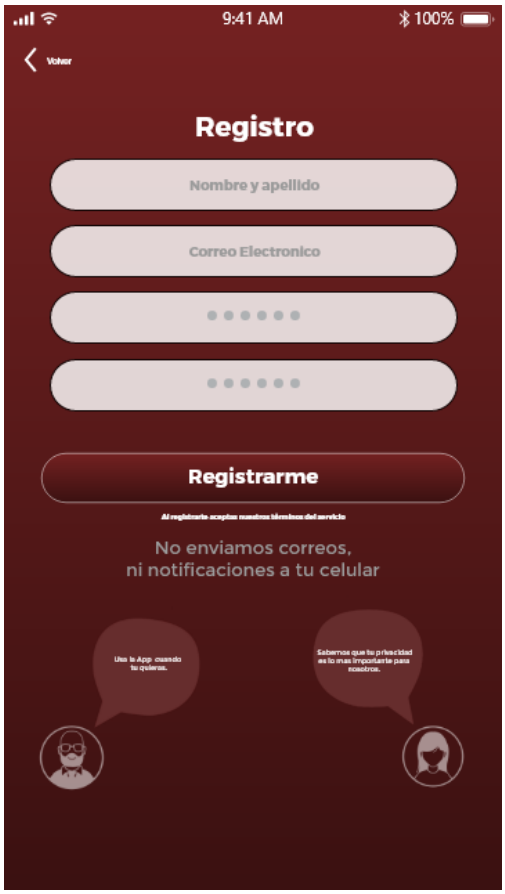


Fuente: Elaboración propia

Iteración 2:

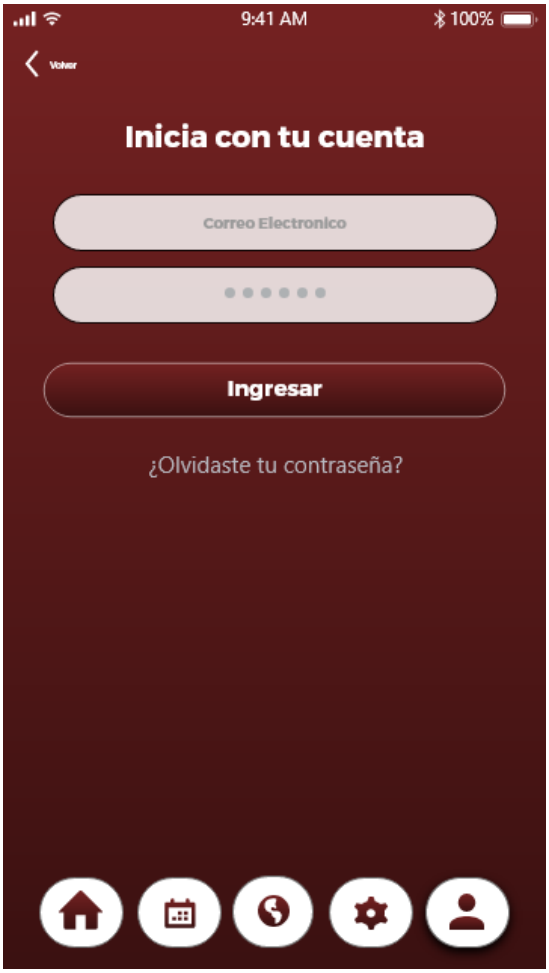
En esta iteración se dio paso a la creación de diferentes maquetas, con la finalidad de contextualizar una idea visual inicial de lo que se desarrolló, dado esto se presentan diferentes “mock ups” de cómo se pretendía ver la app en un inicio: cabe destacar que aún que sean solo maquetas, para el desarrollo de estas en esta instancia ya se debía comprender como sería la arquitectura de la herramienta, ya que las funcionalidades que entregaría la aplicación iban de la mano con la comprensión de la conceptual de la base de datos que se realizó en la primera iteración del proyecto.

Ilustración 15: Mock up de registro de cuenta.



Fuente: Elaboración propia

Ilustración 16: Mock up de inicio de cuenta.



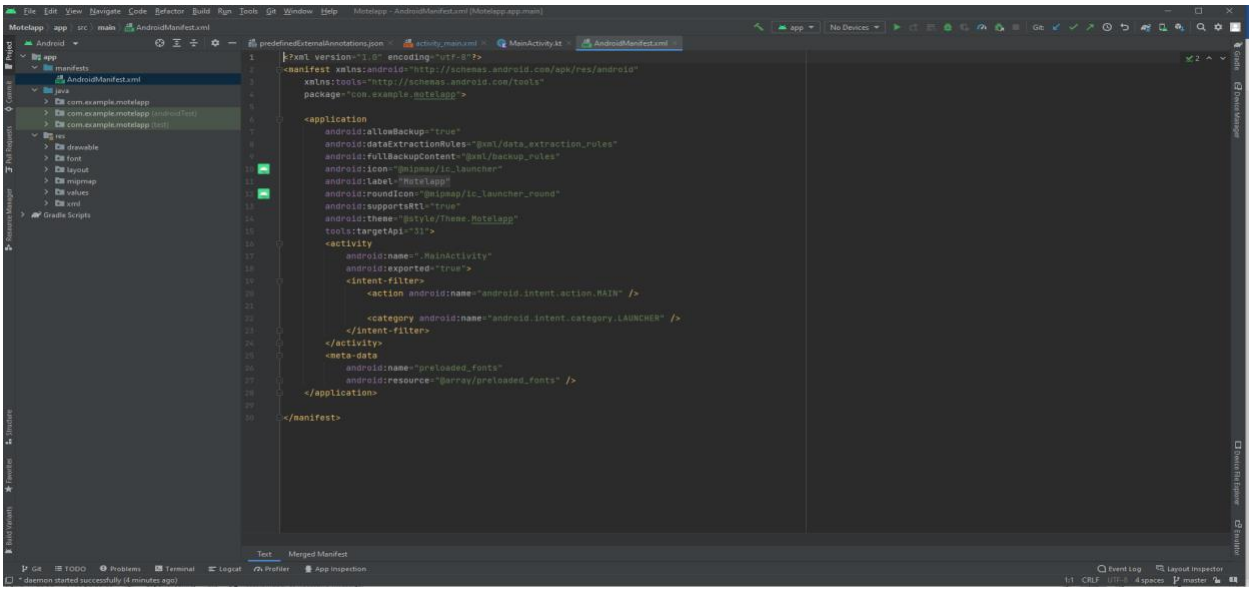
Fuente: Elaboración propia

Los prototipos o mocks ups presentados anteriormente fueron creados en base al feedback de la contraparte del proyecto **(los mock ups restantes se presentan en el ANEXO 1),** pese a no ser el resultado final del mismo, la idea principal de la contraparte era mantener el código de colores para la entrega final sin importar cambios en la interfaz que se podrían estimaran conveniente. En base a esta conjetura se ejercieron cambios en la composición de la interfaz en términos generales, debido a factores implicados a facilitar el desarrollo de la aplicación.

Iteración 3:

En esta iteración se dio paso a la instalación del entorno de desarrollo con el cual se creó la aplicación, basándose en los mock ups y arquitectura presentada en las iteraciones anteriores. Este sistema fue desarrollado mediante el entorno de desarrollo de Android Studio y en lenguaje de programación Kotlin, en la imagen que se presentara a continuación se puede evidenciar una captura de pantalla del entorno de desarrollo ya instalado.

Ilustración 17: Captura de pantalla de Android Studio



Fuente: Elaboración propia

También es importante destacar para el contexto de esta iteración, para poder desarrollar la aplicación en conjunto se utilizó el control de versiones Git, con la finalidad optimizar el trabajo en equipo a lo que el código del sistema competía, además de desarrollar un proyecto escalable.

Gracias a esta herramienta se pudo crear un repositorio, el cual se traduce en la app y que se llamó producción. A la rama o espacio de trabajo de producción, se le hizo una copia, lo que dio como resultado una rama idéntica nombrada la rama testing.

Con la rama de producción y la de testing creadas, se efectuaron dos copias más que se desprendían de testing, y por lo tanto, eran copias idénticas de producción o la app como tal.

Cada desarrollador tomó una de estas dos últimas ramas creadas y lo asignó como su espacio local, su rama propia de trabajo. De esta forma cada desarrollador pudo programar de manera local en su equipo o computador, sin afectar en producción, que es donde se encuentra el código principal y donde se aloja la app en funcionamiento.

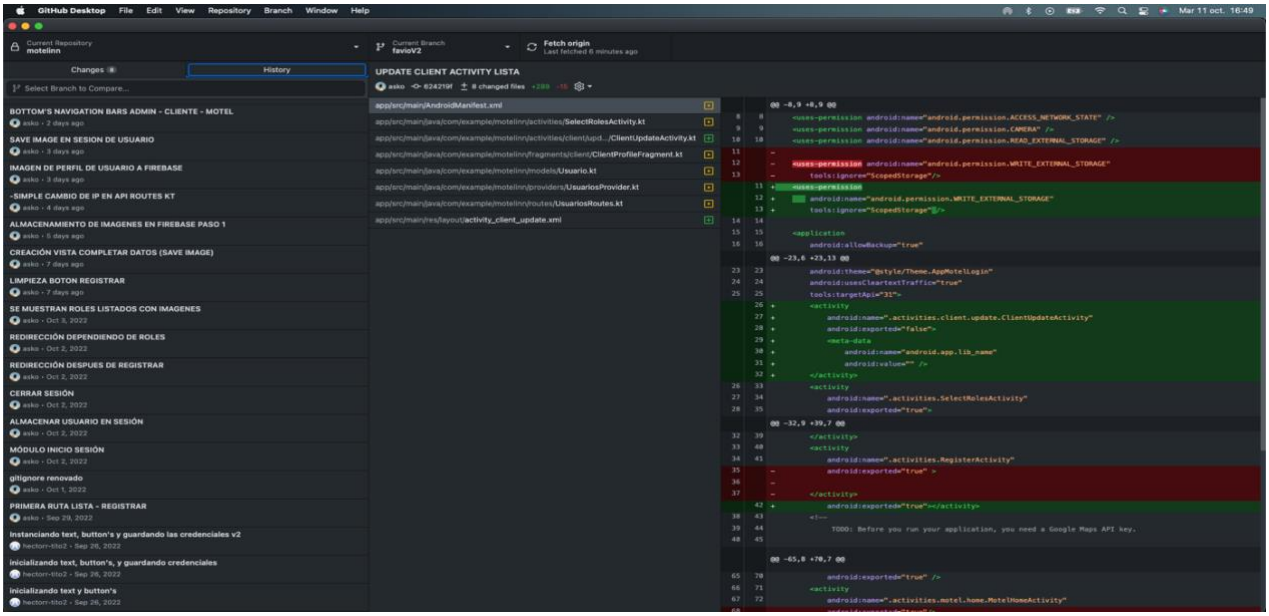
Por lo tanto cada desarrollador abarcaba un módulo en su respectiva rama, finalizaba su desarrollo, subía los cambios a la rama de testing, en donde eran integrados junto con los cambios del otro desarrollador, posteriormente testeados, y una vez aprobaban la fase de testing, estos cambios eran subidos a producción como una nueva versión.

Una de las muchas cosas que permite Git, es por ejemplo si se tiene un programa en su versión 1.0 estable, y se lanza la versión 2.0 pero esta viene con problemas, gracias a git y su historial de cambios, se puede volver a la versión anterior estable tan solo con un comando. De ahí su categorización como “control de versiones”.

Si se le da un buen uso a la herramienta, se puede llegar a modularizar el error de manera que si se separa por commit cada cambio efectuado, si un cambio presenta un error, al estar cada cambio en su respectivo commit, basta con eliminar dicho commit y el error desaparece sin tener que afectar o eliminar otro cambio ya efectuado en dicha versión.

En la siguiente imagen, específicamente en la 18, se puede observar una captura de pantalla de Github, en donde se visualizan los cambios o líneas añadidas de color verde. Los cambios o líneas eliminadas de color rojo. Además existe un historial de cambios y commits-push (cada commit-push es un cambio que fue lanzado por el desarrollador para posteriormente ser integrado), además de un sinfín de más herramientas que para estos efectos no es relevante mencionar.

Ilustración 18: Control de versiones - Github.



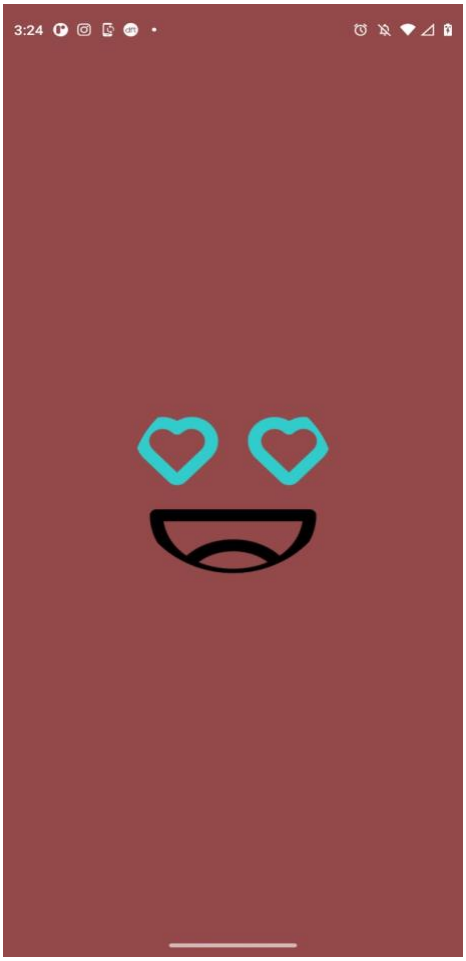
Fuente: elaboración propia

Iteración 4:

Luego de finalizar con las iteraciones anteriores se dio paso a realizar la iteración número cuatro, correspondiente a la confección del desarrollo del código aplicación, englobando todo lo que se definió en las etapas anteriores. En esta iteración también se desarrollaron las interfases visuales con las cuales constaría la aplicación, lo que en resumen traduce las funcionalidades del código hacía el usuario.

En la Ilustración 19 que se presenta a continuación, se puede observar la pantalla de lanzamiento de la aplicación, esta pantalla aparece una vez se accede a ella mediante el APK o launcher de esta misma.

Ilustración 19 - SplashScreen de la aplicación (Pantalla de lanzamiento)



Fuente: Elaboración propia

La Ilustración 20 presenta la pantalla de Login con la cual consta la aplicación, en esta se pueden observar los campos de “Nombre de usuario” y “Contraseña”, también se observan los botones correspondientes a “Iniciar sesión” y “Registrar aquí”. Una vez se completen las credenciales solicitadas y se confirmen, la aplicación iniciara sesión en cada respectiva cuenta.

Respecto al botón registrar aquí, este redirige a una nueva pantalla, la cual da la opción de registrarse en MotelApp.

Ilustración 20 - Pantalla de Login de la aplicación

Fuente: Elaboración propia

En cuanto a lo que respecta con la funcionalidad de registro en la aplicación, se puede evidenciar en la Ilustración 21, en esta se pueden observar los campos “nombre de usuario”,

“Correo electrónico”, “Confirmación clave”, “Clave”. Para poder registrarse en la aplicación es necesario rellenar todos estos campos correctamente, cada campo cuenta con sus respectivas restricciones, como por ejemplo que en el campo “Correo electrónico” es obligatorio insertar un “@” adjunto al dominio correspondiente.

También cuenta con la funcionalidad de registrarse como cliente o administrador, con la finalidad de asignar roles posteriormente.

Ilustración 21 - Pantalla de registro de la aplicación

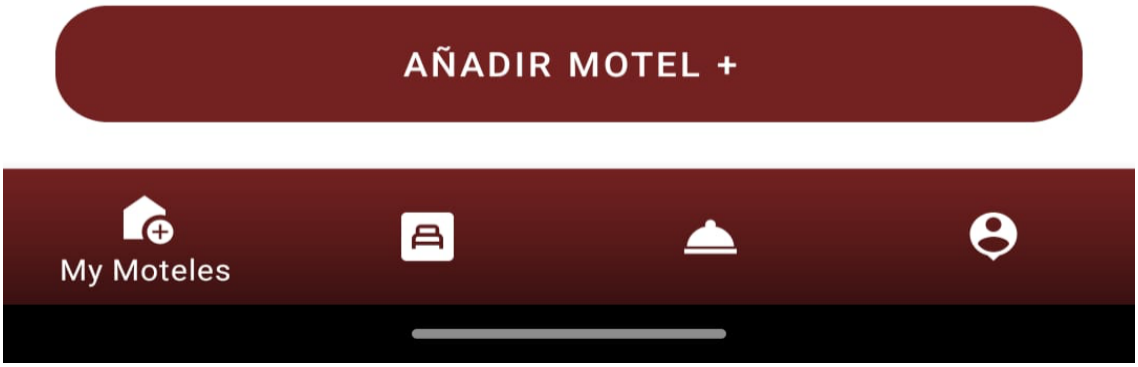
Fuente: Elaboración propia

Cabe destacar que una vez se haya registro e iniciado sesión el usuario, se arroja una pantalla la cual confirma asignación de rol del usuario, estos roles pueden ser “cliente” o “administrador”/”motel” como se mencionó anteriormente. Posterior a esta pantalla se arroja la

interfaz del menú de navegación principal, el cual consta con una barra con botones según el tipo de rol del usuario.

Si el usuario posee un rol de administrador, se muestra el menú de navegación que se puede observar el Ilustración 22. Este menú de navegación posee cuatro botones principales, los cuales se usan con la finalidad de seleccionar la vista deseada, el primer botón corresponde a añadir el motel en cuestión, el segundo botón sirve para añadir una nueva habitación y el tercer botón finalmente sirve para poder ver las habitaciones que se encuentran disponibles en el motel.

Ilustración 22 - Menú de botones rol "administrador" o "motel"



Fuente: Elaboración propia

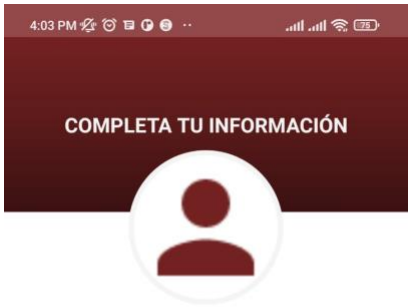
Respecto al rol cliente, el menú en cuestión se puede observar en la Ilustración 23, este menú de navegación posee diferentes botones en comparación con el rol de administrador, estos botones son los de “Home”, “Ubicación moteles”, “Realizar reserva” y el ultimo que corresponde a “Perfil del usuario”. En el primer botón se puede observar que moteles están disponibles, en el segundo se puede observar la ubicación del mismo, en el tercero se puede observar la reserva que se realizó y finalmente en el cuarto se observan los datos del perfil de usuario del rol cliente.

Ilustración 23 - Menú de botones rol cliente

Fuente: Elaboración propia

Previo a la visualización de la barra inferior de navegación, es decir, al momento en que el usuario presiona registrar y el sistema valida los campos de registro, se despliega una última ventana que le permite elegir si desea una imagen de perfil, como complemento al perfil de usuario.

Ilustración 24: Complementando perfil de usuario



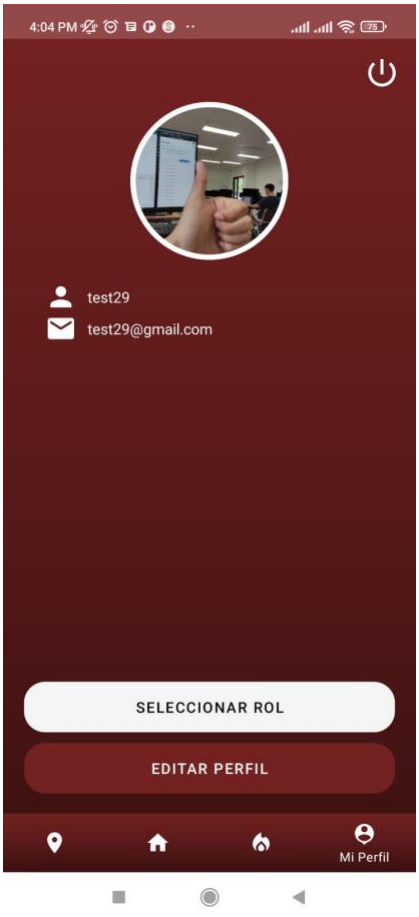
Fuente: Elaboración propia.

Como se ve en el anexo 2.1, en el punto referido al sistema funcional en fase de testing, es decir, previo al despliegue a producción, después de presionar la imagen para completar el perfil con alguna imagen.

Se despliega una ventana o modal que hace elegir al usuario entre elegir un foto ya existente, almacenada en el celular del usuario o en alguna nube de este, y por otro lado, se ofrece sacar una fotografía con la cámara del dispositivo.
Si el usuario decide sacar una fotografía con la cámara del dispositivo, como se ve en los anexos 2.3 y 2.4, la fotografía quedará almacenada en Firebase, que es en resumidas palabras, es una base de datos en la nube de Google.

Posterior a elegir la imagen, se da una previsualización de la imagen escogida o tomada por el usuario, y se da la opción de confirmar (anexo 2.5), una vez confirmado se redirecciona al usuario a su perfil, como se muestra en la siguiente imagen.

Ilustración 25: Perfil de usuario



Fuente: Elaboración propia.

La pantalla de perfil de usuario, almacena los datos del usuario y la imagen, y cuenta con tres botones.

En la parte superior derecha, se encuentra el botón de cerrar sesión, el cual lleva al usuario nuevamente a la pantalla de Login para reingresar al sistema (Ilustración 20).

Los dos botones restantes son “Seleccionar rol” y “Editar perfil”. El botón de selección de rol permite al usuario cambiar de roles, por ejemplo, le permite pasar de “modo cliente” a “modo administrador/motel” . Dicho módulo se presenta más adelante.

Por otro lado, el botón “Editar perfil”, despliega una pantalla en la que se permiten configurar los datos editables del usuario, como lo son el nombre y la imagen, para la actualización de imagen, se realiza el mismo proceso de selección de imagen o de toma de fotografía que se realiza en la Ilustración 26. A continuación una muestra de lo anteriormente explicado.

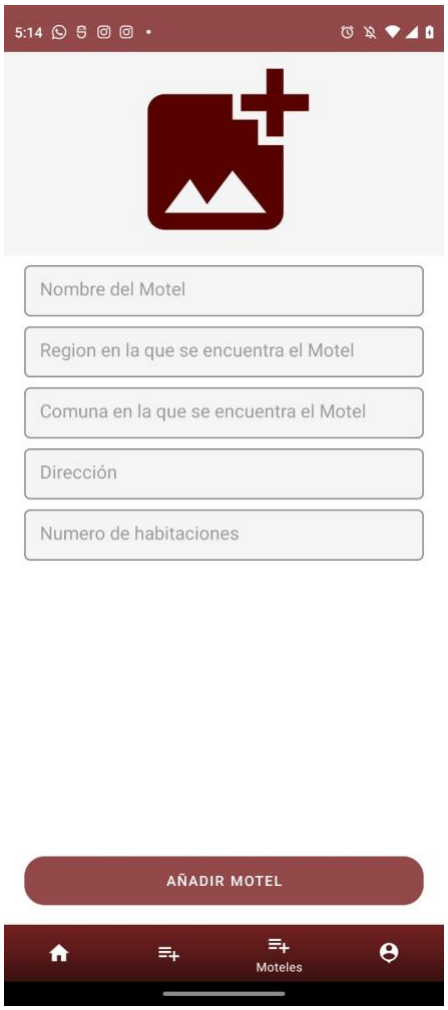
Ilustración 26: Editar perfil de usuario.

Fuente: Elaboración propia.

Después de presionar el botón actualizar, se realiza una función que actualiza al usuario con los nuevos campos y el usuario puede navegar nuevamente a su perfil a ver los cambios o bien navegar por la aplicación.

En el caso de que el usuario este registrado con el rol “Motel” la aplicación arroja el **menú** que se logra ver en la Figura 27, también se observa, que por defecto se arroja la vista “añadir Motel”, en la cual se puede añadir un Motel completando los datos solicitados. Cabe destacar que una vez creado el motel, también aparecerá en la vista del Cliente, al momento de escoger moteles, es por esto que para que al usuario tipo “Cliente” le aparezcan moteles que elegir, primero se debe registrar el motel como usuario tipo “Motel”.

Ilustración 27: Menú de rol Motel/ vista añadir Motel



Fuente: Elaboración propia

Posterior a la creación del Motel, se lanza la pantalla en la cual se guarda la dirección de dicho motel, acá se deben insertar los datos relacionados con la vista como se puede observar en la Ilustración 28. Una vez se selecciona el tercer campo rellenar se ejecuta la actividad la cual da paso a la siguiente pantalla en la que se asigna la dirección mediante la "API" integrada de Google Maps.

Ilustración 28 - Registro de ubicación del motel

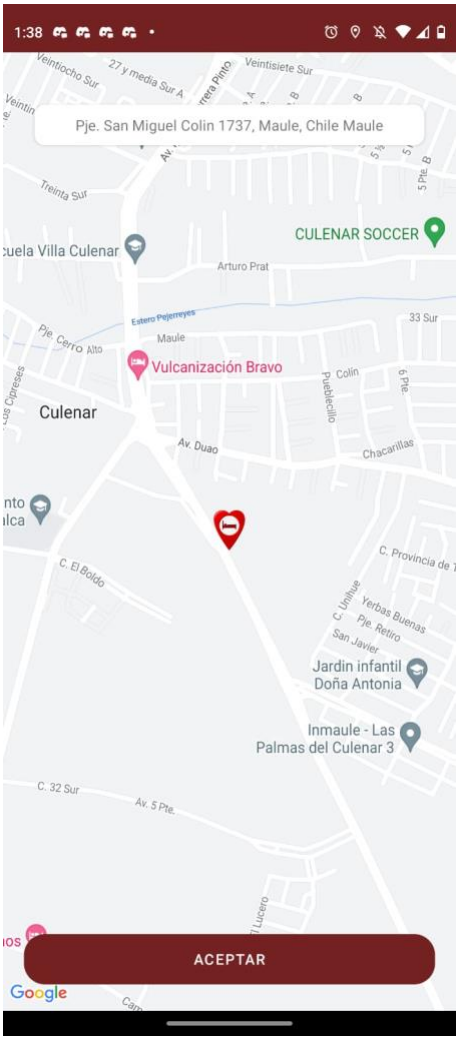
Fuente: Elaboración propia

Con fines posteriores de enrutar al cliente al motel, en el cual realizo la reserva, se agregó Google Maps en la aplicación, al consumir esta "API" se observa de manera precisa cual sería la ubicación de motel.

En cuanto a la pantalla en la cual se asigna la ubicación, se puede observar en la Ilustración 29, en la parte superior se observa la ubicación momentánea cada vez que el cursor se desplaza por el mapa, cabe destacar que estos datos se almacenan como latitud y longitud en la base de datos.

Una vez el administrador del Motel decidió la dirección y presione el botón "aceptar" los datos se almacena en la base de datos.

Ilustración 29 - Añadir la ubicación en Google Maps

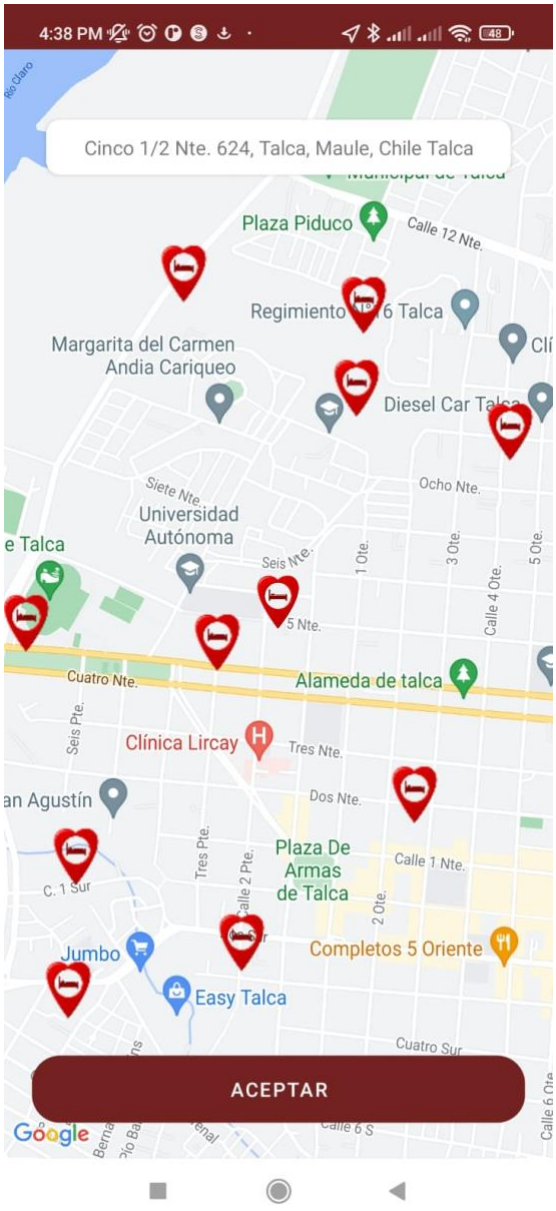


Fuente: Elaboración propia

Tomando como ejemplo una cuenta que tiene varios moteles agregado, se visualiza como en la Ilustración 30, destacando que cada marcador es un motel dentro de la aplicación.

En el anexo 2.6 se observa cómo es la vista del lado de una cuenta con el rol de Cliente.

Ilustración 30 - Ubicaciones de moteles



Fuente: Elaboración propia

Posterior a lo que se almacena la dirección de motel, el administrador se puede dirigir a la pantalla en la cual se asignan las habitaciones al motel, como se observa en la Ilustración 31, esta pantalla cuando esta sin completar cuenta con los campos “Nombre de habitación”, “Precio de la habitación”, “Descripción” y un campo correspondiente a las fotos que se le asignan a la habitación, siendo estas un mínimo de una foto y un máximo de cinco.

Cabe destacar que el campo correspondiente al “Precio de la habitación” se debe rellenar en función al valor de la hora de lo que cobrara el Motel.

Ilustración 31 - Añadir habitación al motel

Fuente: Elaboración propia

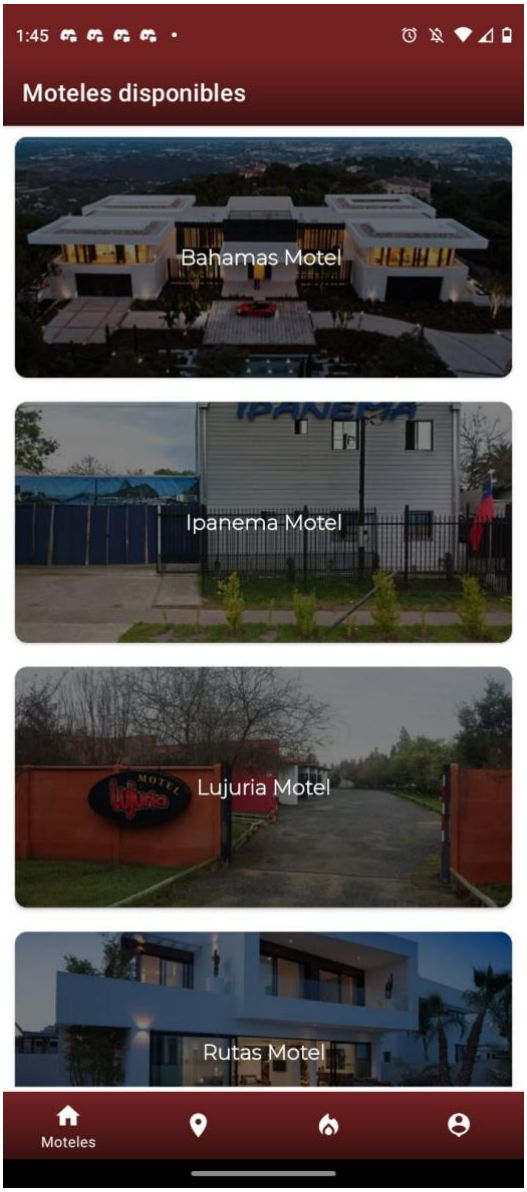
Ya con los campos completados, la pantalla se observa como en la Ilustración 32, es importante mencionar, que una vez completados estos campos se despliega un “spinner” en el cual se debe seleccionar el motel al cual se desea añadir la habitación. Dado que en este “spinner” solo se despliegan los moteles que el usuario en sesión tiene registrados, es decir, un usuario tipo “administrador” puede tener uno o más moteles registrados, si tiene un motel registrado solo aparecerá dicho motel, si tiene más de uno, aparecerá más de uno, pero no aparecerán todos los moteles que existen en la base de datos, solo los que ese usuario registró. Ya completados los pasos anteriormente mencionados se presiona el botón “añadir habitación” para realizar dicha acción.

Ilustración 32 - Seleccionar motel al cual se le añade la habitación

Fuente: Elaboración propia

En cuanto a lo que respecta la vista del usuario tipo “cliente”, el menú principal se lanza con la pantalla por defecto de “Moteles Disponibles”, como se observa en la Ilustración 32. En esta pantalla salen todos los moteles registrados que se agregaron correctamente a la base de datos por los usuarios tipo “administrador”. Esta vista cuenta con un scroller el cual se desplaza para ver la totalidad de moteles disponibles.

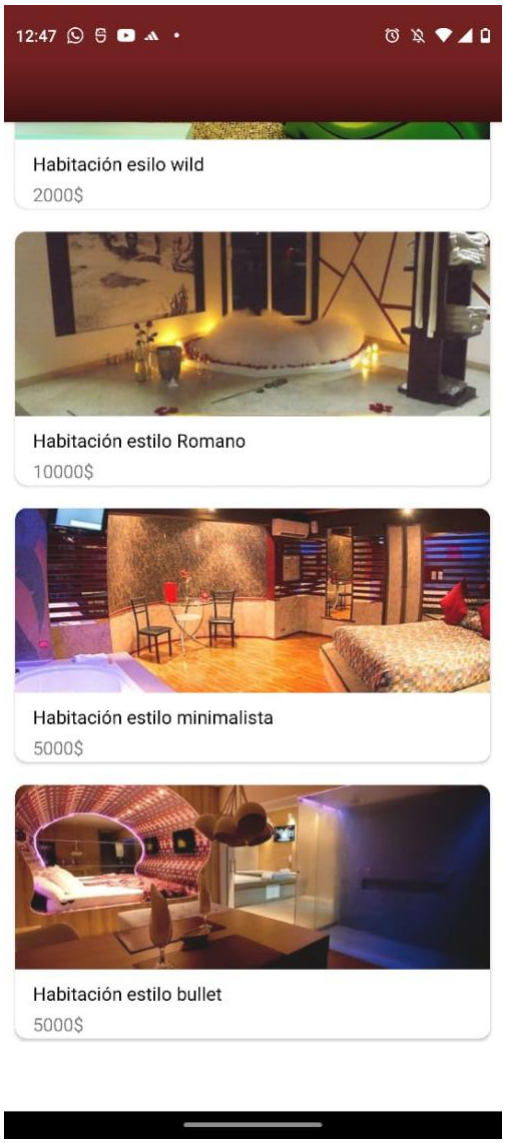
Ilustración 33 - Moteles disponibles



Fuente: Elaboración propia

Una vez el cliente selecciona el motel al cual desea ir, se ejecuta la vista que arroja las habitaciones que se encuentran disponibles en dicho motel, como se observa en la Ilustración 33. En esta pantalla también se muestra el nombre de la habitación y el valor por hora que se cobrara por la habitación. Cabe destacar que esta vista también cuenta con un scroller con el cual el cliente interactúa y observa la totalidad de habitaciones disponibles del motel correspondiente.

Ilustración 34 - Habitaciones disponibles



Fuente: Elaboración propia

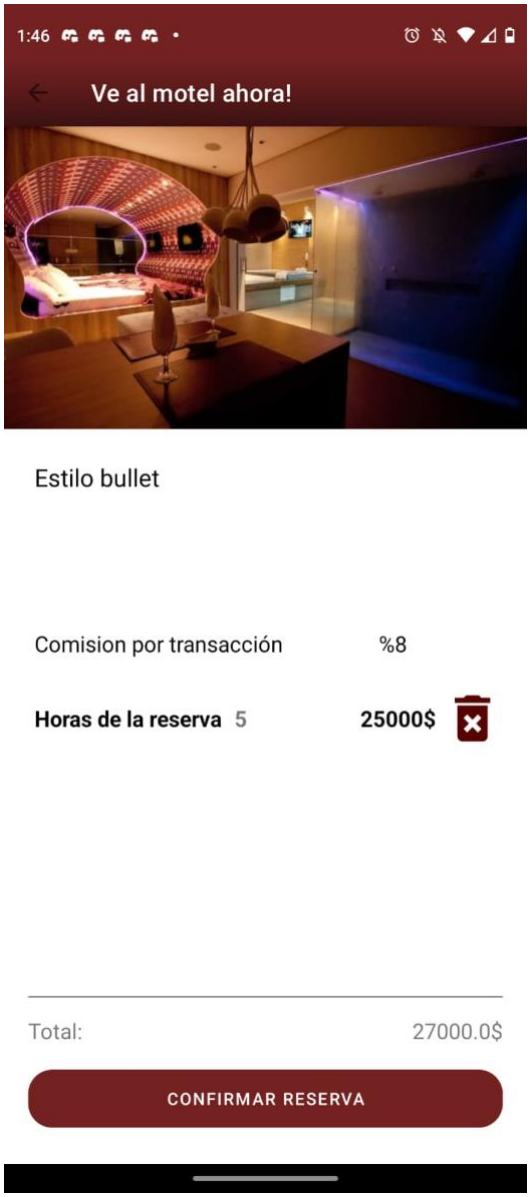
Posterior a que se selecciona la habitación, se arroja la vista con el detalle de la habitación, como se observa en la Ilustración 35. En esta pantalla existe un scroller que funciona de forma horizontal al momento de interactuar con las imágenes almacenadas de la habitación, siendo cinco el máximo de imágenes disponibles para ver por habitación. También se observa el nombre, descripción y la cantidad de horas por las cuales se desea reservar la habitación. Siendo claro que la habitación se reservara en ese mismo momento, por el tipo de servicio que ofrecen los moteles, que normalmente no se planifican, se requiere al mismo momento que se conoce el servicio.

Ilustración 35 - Detalle de la habitación y asignación de horas

Fuente: Elaboración propia

Una vez se presiona el botón “Reservar habitación” , se arroja la vista que se observa en la Ilustración 35. En esta pantalla se observa el nombre de la habitación y los cargos asociados a la reserva de la misma, que son: “comisión por la transacción” (que sería porcentaje de pago a la aplicación por su uso) y “valor de la habitación”, que se genera en función de la cantidad de horas de la reserva y el valor por hora, que el usuario “motel” le asigno a la habitación.

Ilustración 36 - Confirmar Reserva

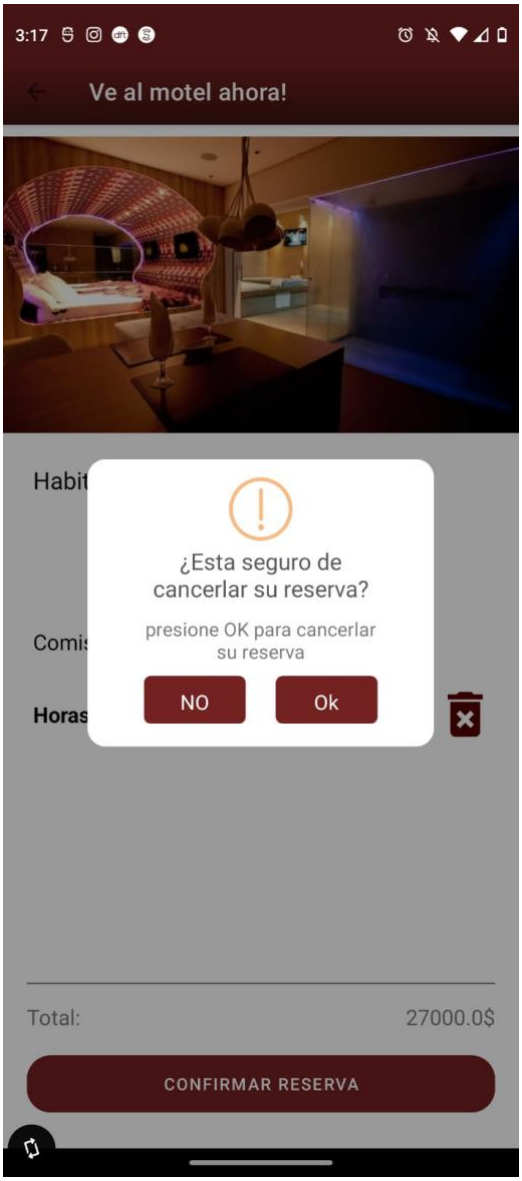


Fuente: Elaboración propia

Destacando que la vista anterior posee dos opciones al momento de interactuar, que son: “confirmar la reserva” y “cancelar la reserva”, esta última se ejecuta posterior al presionar el icono con forma de basurero que se puede observar en la ilustración 36.

Al momento de presionar este botón, se arroja una vista pequeña en la cual se confirma o se denega la cancelación de la reserva, como se puede observar en la Ilustración 37.

Ilustración 37 - Cancelar reserva



Fuente: Elaboración propia

En el caso contrario al mencionado anteriormente, y haber pulsado el botón “Confirmar reserva”, se registra la reserva en la base de datos, ocultando la habitación de “habitaciones disponibles”. En conjunto, la pantalla que procede a la vista anterior se observa en la Ilustración 38. Acá se observa el detalle de la reserva, como lo es el “nombre de la habitación”, la “cantidad de horas de la reserva” y el “precio a cancelar al llegar al motel”.

Con la finalidad de salvaguardar la confidencialidad de los datos personales, la reserva genera un código, el cual lo tiene el administrador del motel y el cliente. Finalmente con este código se obtiene el acceso al motel.

Ilustración 38 - Comprobante de reserva

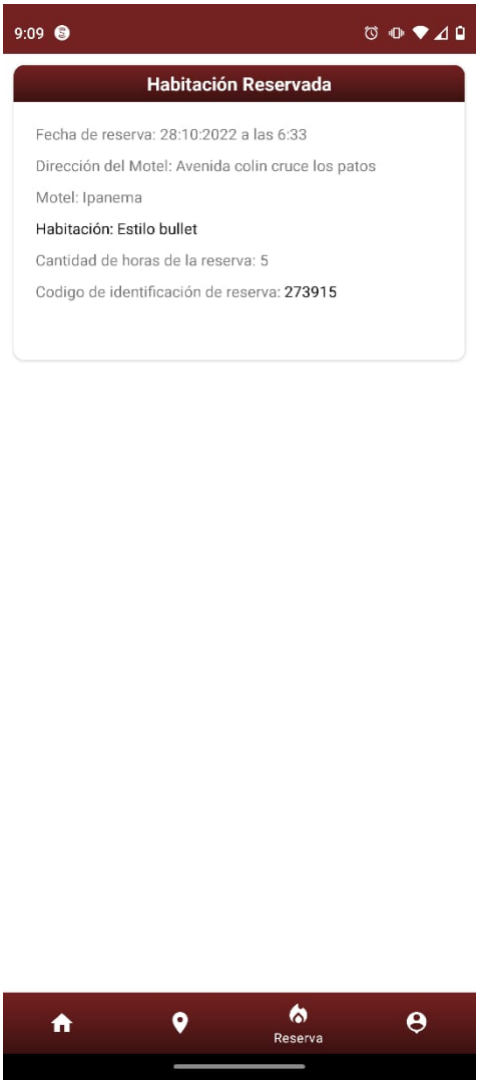
Fuente: Elaboración propia

A lo que respecta al “registro de la reserva” se visualiza en el botón “reserva” del menú de navegación, como se observa en la Ilustración 39, en esta vista se obtiene el horario en el que se realizó la reserva, este dato es de clave importancia, ya que desde esa hora se empieza a reducir el periodo de reserva, claro sumado a los 30 min de llegar al motel.

Se destaca de igual manera que también se obtiene la dirección del motel al cual se desea ir.

Como se mencionó anteriormente, la reserva genera un código el cual se obtiene en el destalle de la “habitación reservada”.

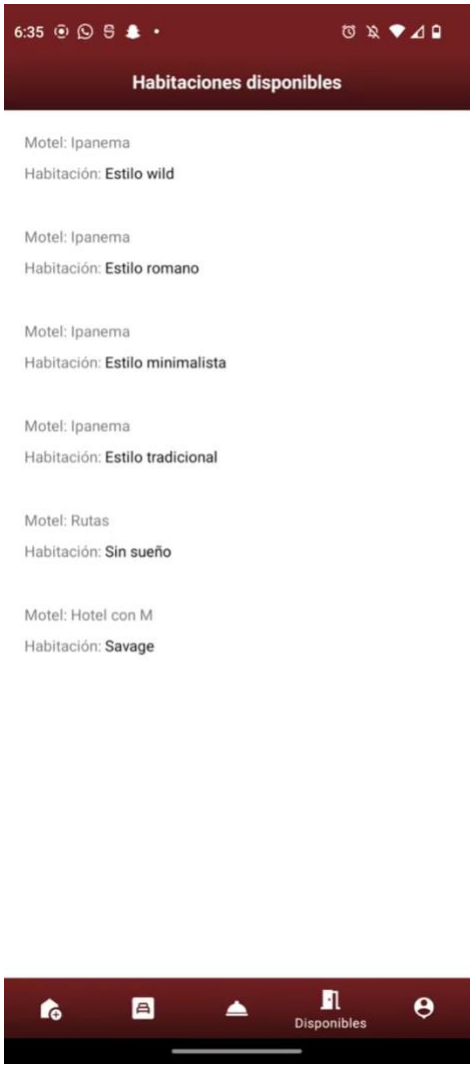
Ilustración 39 - Detalle de la habitación reservada



Fuente: Elaboración propia

Finalmente en la Ilustración 40 se puede observar como la habitación reservada ya no se encuentra en “habitaciones disponibles” del rol administrador o “motel”, esto quiere decir que cada vez que se reserve una habitación, esta no aparece en esta pantalla hasta que haya finalizado el periodo de reserva, teniendo un lapso intermedio entre los estados “Disponible” y “Reservada”, el cual se llama “En limpieza”. Este estado es el único en el cual la habitación no se visualiza en ninguna vista.

Ilustración 40 - Vista habitaciones disponibles, Rol "Motel"



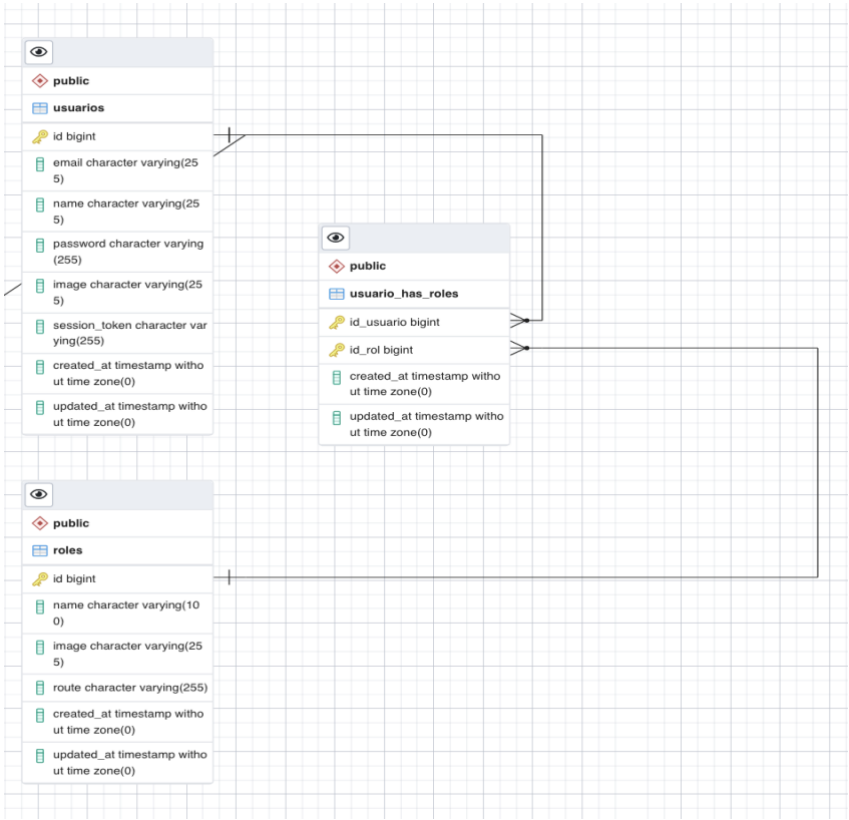
Fuente: Elaboración propia

Iteración 5:

Como se mencionó anteriormente existen roles en la aplicación, roles de administrador y usuarios respectivamente, que para contextualizar en este caso se hacen llamar rol de “Motel” y “Cliente”. Cabe destacar que para versiones futuras, está pensando agregar un rol de administrador que sería como una “rama” del rol “Motel” con diferentes permisos.

El algoritmo con el que se trabajan los roles se basa principalmente en la “extracción de información” de las consultas realizadas en SQL, esto atreves de la interacción de las tablas que se representan en la Ilustración 41. Acá se puede observar las relaciones que se desprenden entre Usuarios y Roles, en base a estas relaciones se crea la tabla Usuario tiene Roles.

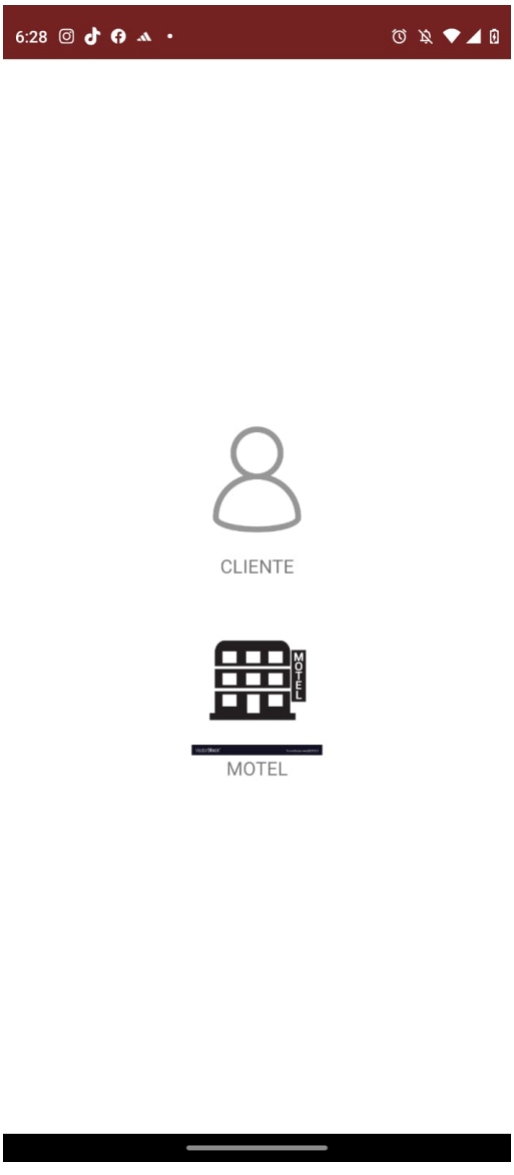
Ilustración 41 - Representación del algoritmo de interacción intermediaria, extracto de la estructura de bbdd



Fuente: Elaboración Propia

Respecto a cómo se ve visualmente en la aplicación al ejecutarse la opción de escoger el rol, se observa gráficamente en la Ilustración 42, cada icono intuitivamente representa un rol con el cual el usuario accede a deferentes pantallas y funcionalidades.

Ilustración 42 - Captura de pantalla de "seleccionar rol"

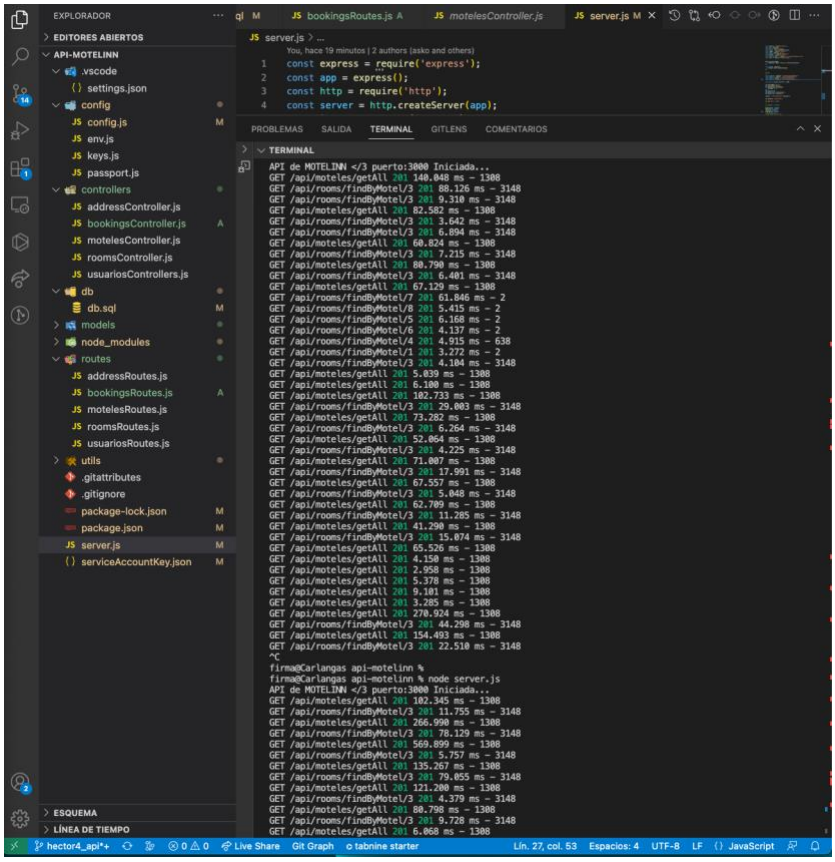


Fuente: Elaboración propia

Métrica de tiempos de respuesta:

Dado que el sistema posea una estructura de base de datos sólida y bien relacionada, y además la cantidad de datos que se manejaron para las pruebas de rendimiento no son exuberantes, como se puede evidenciar con el servidor en marcha en la Ilustración 40, esto da a entender que los resultados se consideran realmente positivos, ya que el intervalo de respuestas de las consultas a la base de datos va desde los 0,002 a los 0,56 segundos, es decir, los tiempos de respuesta según el fragmento de consultas que se muestra en la Ilustración 40, oscilan entre 0,002 y 0,56 segundos. En cuanto al tiempo promedio de respuesta que se registran en el fragmento es de 0,58 segundos, también cabe destacar que los tiempos de respuesta de la aplicación en general no superan los 2 segundos, siendo estos, unos tiempos de respuestas muy aceptables en el contexto de interfaces de usuarios.

Ilustración 43 - Fragmento de consultas con el servidor en marcha



Fuente: Elaboración propia

CAPÍTULO 4 - CONCLUSIONES

Se comprende que es de vital importancia innovar, agilizar y/o optimizar los procesos de un negocio, sin importar el rubro. En base a este simple pero objetiva conjetura la empresa "CREAPP SPA" implementará el desarrollo de esta aplicación en los moteles que se estime conveniente, levantando la app en plataformas de distribución de aplicaciones como lo son PlayStore y AppStore. La empresa se ha visto muy favorecida con el desarrollo de la aplicación, entregando un feedback muy positivo, principalmente gracias a la escalabilidad con la que cuenta la aplicación respecto a contextos de desarrollo. Recalcando también la importancia de que el sistema funcione como aplicación tipo "intermediaria" es decir que establece la comunicación entre el motel y el cliente en cuestión. Con lo cual se hace alusión a uno de los objetivos específicos de este proyecto, el cual era "Implementar un algoritmo que funcione como intermediario, entre el cliente y el motel" esto se logró en base a la estructura de roles con la cual se compone la aplicación, dando la posibilidad de interactuar a ambos usuarios al mismo tiempo mediante el sistema.

En cuanto al objetivo específico que hacía alusión a la toma de requerimientos, fue realizado correctamente en base a diferentes reuniones con la contraparte. Es importante recalcar lo fundamental del proceso de toma de requerimientos ya que fueron las bases que cimentaron el desarrollo de la aplicación.

Tomando en que cuenta al objetivo que hacía referencia a las métricas, se lleva a cabo concisa y apropiada, yendo al punto principal rápidamente respecto a "los tiempos de respuestas de la aplicación hacia los usuarios de la aplicación" , concluyendo que los tiempos eran apropiados y rápidos para el uso común de los usuarios.

Respecto a lo que nos compete personalmente, el proyecto fue muy difícil de llevar a cabo, dado el desconocimiento de las tecnologías que utilizamos para el desarrollo del mismo, ya que nunca habíamos desarrollado nada para móviles, teniendo un desconocimiento absoluto del área. Es por esto que en primera instancia nos tuvimos que instruir detalladamente de dichas tecnologías. Como resultado de esto, nos sentimos enormemente preparados para emplear estas tecnologías en diferentes proyectos y de diferentes magnitudes.

Finalmente, el desarrollo de este proyecto tuvo un impacto positivo en cuanto a los conocimientos que se adquirieron, lo que se tradujo en una aplicación con mucho potencial a futuro, dado el formato escalable del desarrollo del mismo, y por tipo de aplicación que es. Ya que la aplicación busca cubrir de forma rápida esta necesidad, si una persona desea el servicio de ir a un motel, lo tendrá rápido, en el mismo momento, no se necesitará agendar una hora o una fecha, podrá recurrir al motel en el instante que confirme la reserva. Funcionando similar a todos los servicios que entregan las aplicaciones de hoy en día. “Lo necesito ahora y lo obtengo ahora”.

Links de Repositorios de GitHub:

Backend: https://github.com/vskooooo/api-motelinn.git

Frontend: https://github.com/vskooooo/motelinn.git

REFERENCIAS

- Adami, C. (2016). What is information? . Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2063), 20150230. Recuperado de https://doi.org/10.1098/rsta.2015.0230

- Aguilar, Nora Isela Zamora et al. "Especificación de requerimientos con Áncora y el estándar 830." Res. Comput. Sci. 79 (2014): 109-119.

- A. Hernández. (2006). Los sistemas de información evolución y desarrollo. Recuperado de https://dialnet.unirioja.es/servlet/articulo?codigo=793097.

- C. Rondón (2019). 5 sistemas hoteleros para optimizar tu hotel o posada. Recuperado el 24 de mayo, de https://asksuite.com/es/blog/sistema-hotelero/.

- Correa Ospina, M. L., & Díaz Pinzón, B. H. (2018). Capacidad en tecnologías de la información y desempeño organizacional: un estudio en el contexto colombiano. Innovar. Recuperado de https://doi.org/10.15446/innovar.v28n69.71699

- Dori, D., Sillitto, H., Griego, R. M., Mc Kinney, D., Arnold, E. P., Godfrey, P., Martin, J., Jackson, S., & Krob, D. (2020, 1 juni). System Definition, System Worldviews, and Systemness Characteristics. IEEE Journals & Magazine | IEEE Xplore. Recuperado 26 de abril de 2022, de https://ieeexplore.ieee.org/abstract/document/8693820

- Enriquez, J. G., & Casas, S. I. (2014). Usabilidad en aplicaciones móviles. Informes Científicos Técnicos - UNPA, 5(2), 25–47. https://doi.org/10.22305/ict-unpa.v5i2.71

- Fang, Y. (2019). An app a day keeps a customer connected: Explicating loyalty to brands and branded applications through the lens of affordance and service-dominant logic. Information & Management

- Flaaut. M (2016) CASO DE USO. Recuperado el 24 de mayo de 2022, de http://maribelteamuml.blogspot.com/2015/10/caso-de-uso-maribel-flaaut.htm

- Getapp.com (2022). Sistemas de reservas. Recuperado el 24 de mayo, de https://www.getapp.cl/directory/83/reservation-online-booking/software.

- IBM (2022). ¿Qué es la gestión de requisitos?. Recuperado de https://www.ibm.com/cl-es/topics/what-is-requirements-management?mhsrc=ibmsearch_a&mhq=ingenieria%20de%20software

- Introducción a Adroid Studio. (2021) Introducción a Android Studio. https://developer.android.com. Recuperado el 24 de mayo de 2022, de https://developer.android.com/studio/intro?hl=es-419

- Kendall, K. E., & Kendall, J. E. (2011). Analisis Y Diseño De Sistemas (8.ª ed.). Recuperado de http://cotana.informatica.edu.bo/downloads/ld-Analisis%20y%20Diseno%20de%20Sistemas_Kendall-8va.pdf

- Maida, EG, Pacienzia, J. (2015). Metodologías de desarrollo de software. Tesis de Licenciatura en Sistemas y Computación. Facultad de Química e Ingeniería "Fray Rogelio Bacon". Universidad Católica Argentina. Recuperado de http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf

- Malagón, A., 2018. Revisión sistemática de teorías de integración de sistemas de gestión normalizados. 10th ed. Madrid: Signos.

- Oracle.com (2022). ¿Qué es el PMS hotelero?. Recuperado el 24 de mayo, de https://www.oracle.com/cl/industries/hospitality/what-is-hotel-pms/..

- ¿Qué es un sistema de gestión y para qué sirve? (2021, enero 31). ISBL. Recuperado de https://isbl.eu/2021/01/que-es-un-sistema-de-gestion-y-para-que-sirve/

- Rodríguez. H. (2021) ¿Qué es un framework y para qué sirve? https://www.crehana.com. Recuperado el 24 de mayo de 2022, de https://www.crehana.com/blog/desarrollo-web/que-es-un-framework/

- Salazar, O. A., Aguirre, F. A. M., & Osorio, J. A. C. (2011). Herramientas para el desarrollo rápido de aplicaciones web. *Scientia et technica*, *1*(47), 254-258.

- Sommerville, I. (2005). Ingeniería del Software. Madrid: Pearson Educación

- Sommerville, I. (2011). Ingeniería de software. sistemamid.com. Recuperado de https://sistemamid.com/panel/uploads/biblioteca/2018-06-11_03-37-12144643.pdf

- S.A.Sommerville, I. (2011). Ingenieria De Software (9.ª ed.). Recuperado de https://www.academia.edu/25063155/Ingenieria_de_Software_Somerville

- V. Fernández (2006) . Desarrollo de sistemas de información. Recuperado de https://d1wqtxts1xzle7.cloudfront.net/63922351/Desarrollo_de_Sistemas_de_Informacion_una_Metodologia_Basada_en_el_Modelado_de_Vicenc_Fernandez20200714-4043-1dzzais-with-cover-page-v2.pdf?Expires=1651012714&Signature=RoRjkqX91oKmXwZO0MO5P-qSZ-Hx-t9J-zG9FAPN4~U1UeYWbHZTwKRmJuQ-8kyyl~09mnJzH6KZjqn5DTTagNYYULL~~QoaCGSQPq6h8b2GIOxFN3nEkcmT7ETS1QRQoRHSeOKWuWTNUsR2WRLkl0RJPKWzK48VbAbEk6Lh8Z5FNImzRkgRTLv-BF9QCC87Zqm3WsD9p5p1nyPFl9nEQvPv9BQrZ8sFkqaMBhHdRp8QfrzSWaBzRWkMr1AGO8iypN3A4OtFCDmF0YN6ls~yuBs6xQgmcUiYd6eTN5nGfe7S2BiRRPDD8sOROYbWQhf2w0YjV6S9qCT1J6T7oxbiQg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

- Wang, Y. (2007). Software engineering foundations: A software science perspective. Auerbach Publications.

- Watson, C., McCarthy, J., & Rowley, J. (2013). Consumer attitudes towards mobile marketing in the smart phone era. International Journal of Information Management

- Zydney, J. M., & Warner, Z. (2016). Mobile apps for science learning: Review of research. Computers & Education, 94, 1-17.

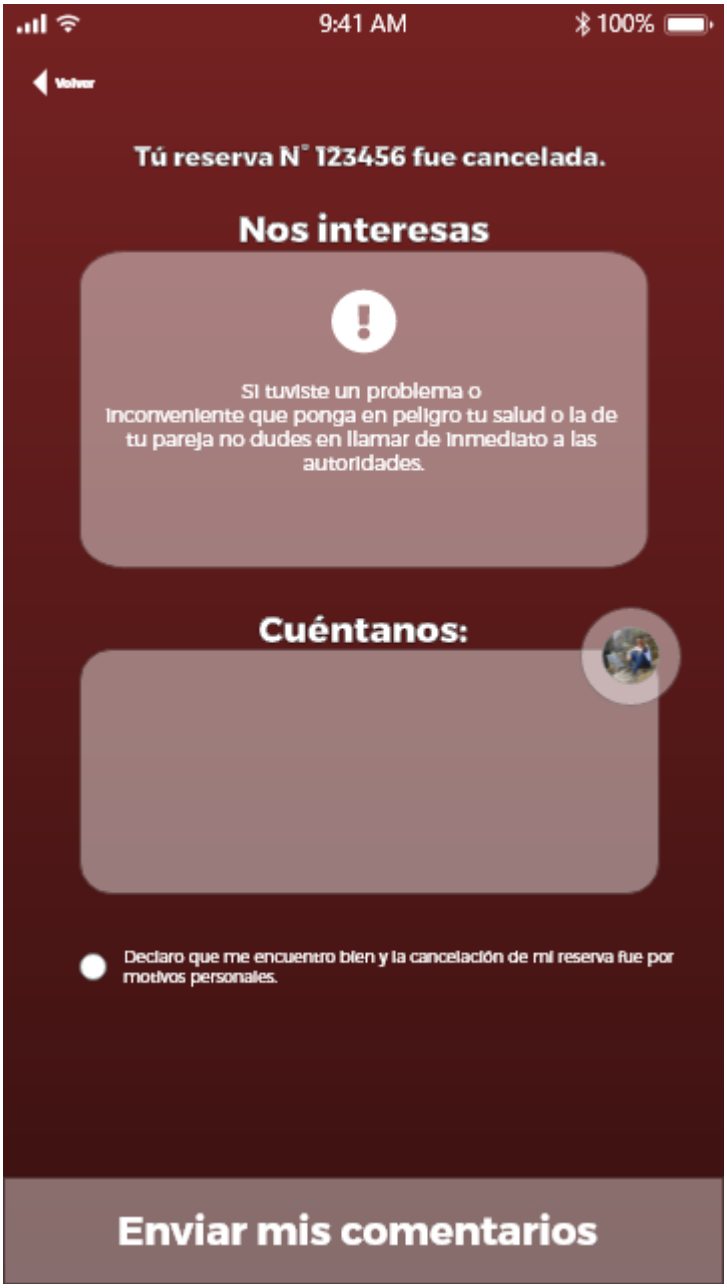
ANEXOS

ANEXO 1. MOCK UPS DE LA APLICACIÓN

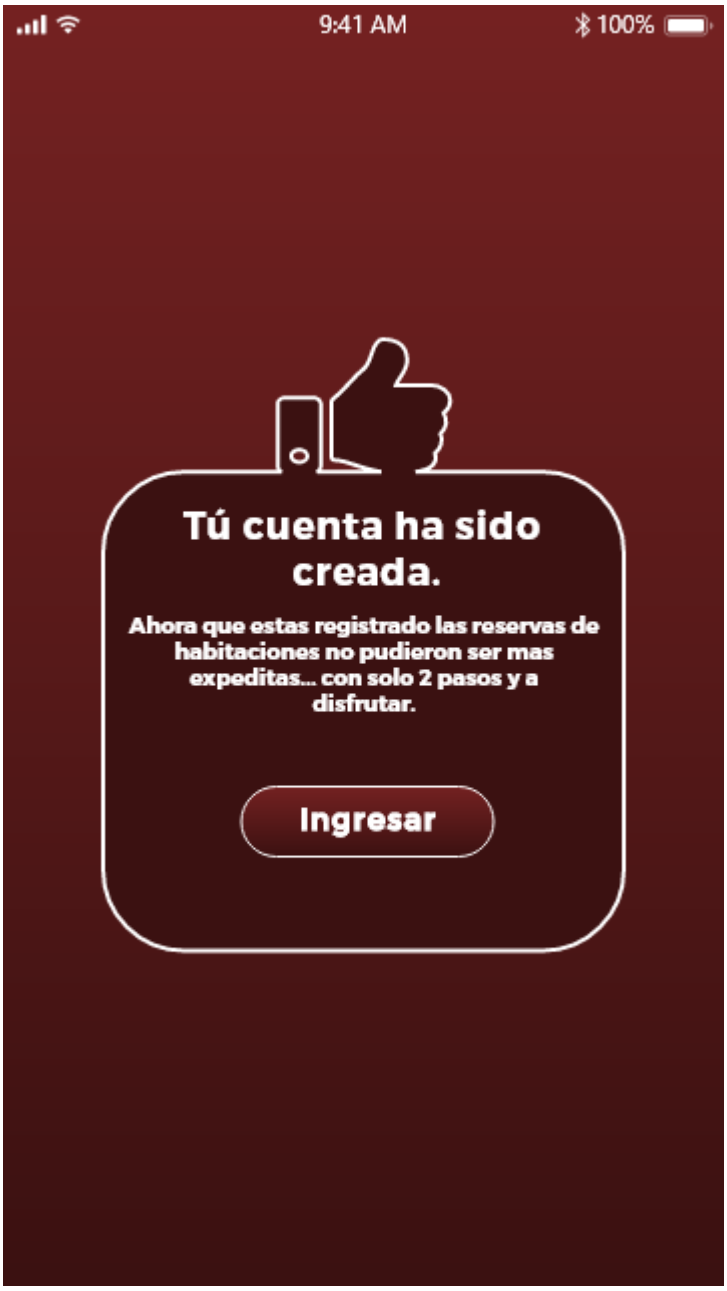
CANCELAR RESERVA

PREGUNTAS FRECUENTES

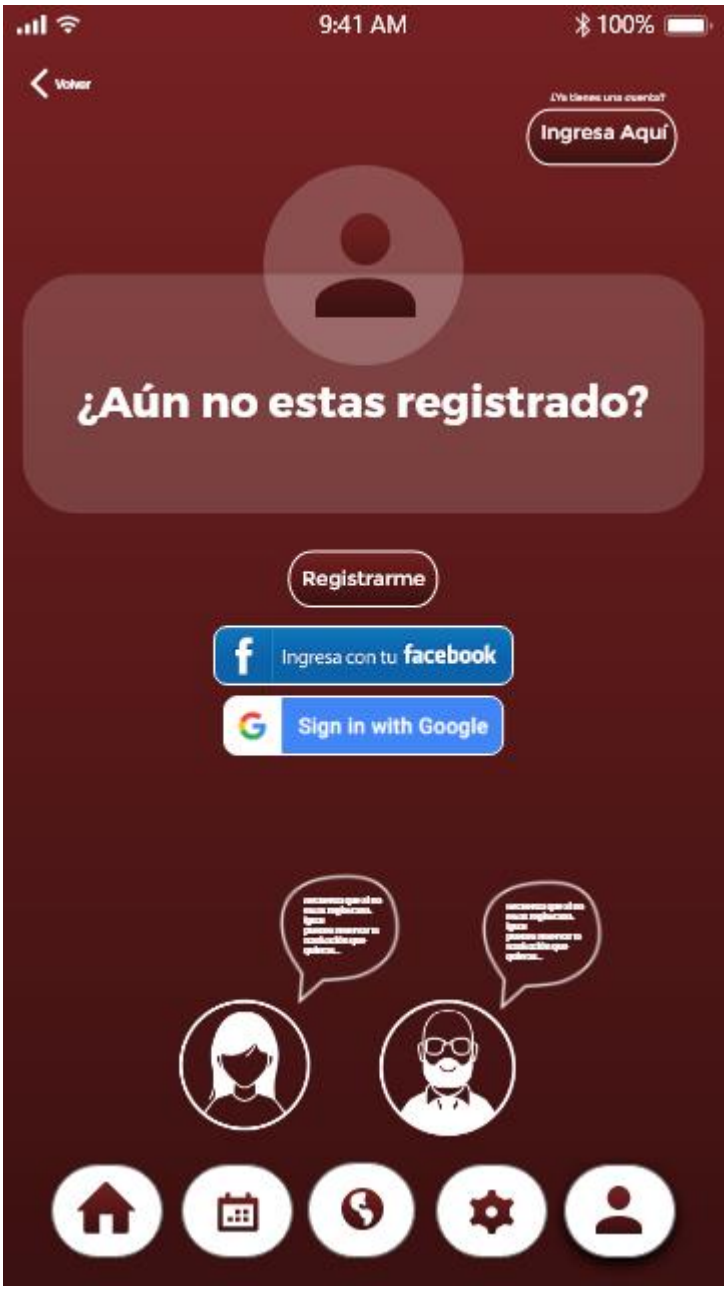
FEEDBACK CLIENTE



RESPUESTA FORMULARIO DE REGISTRO INTERNO

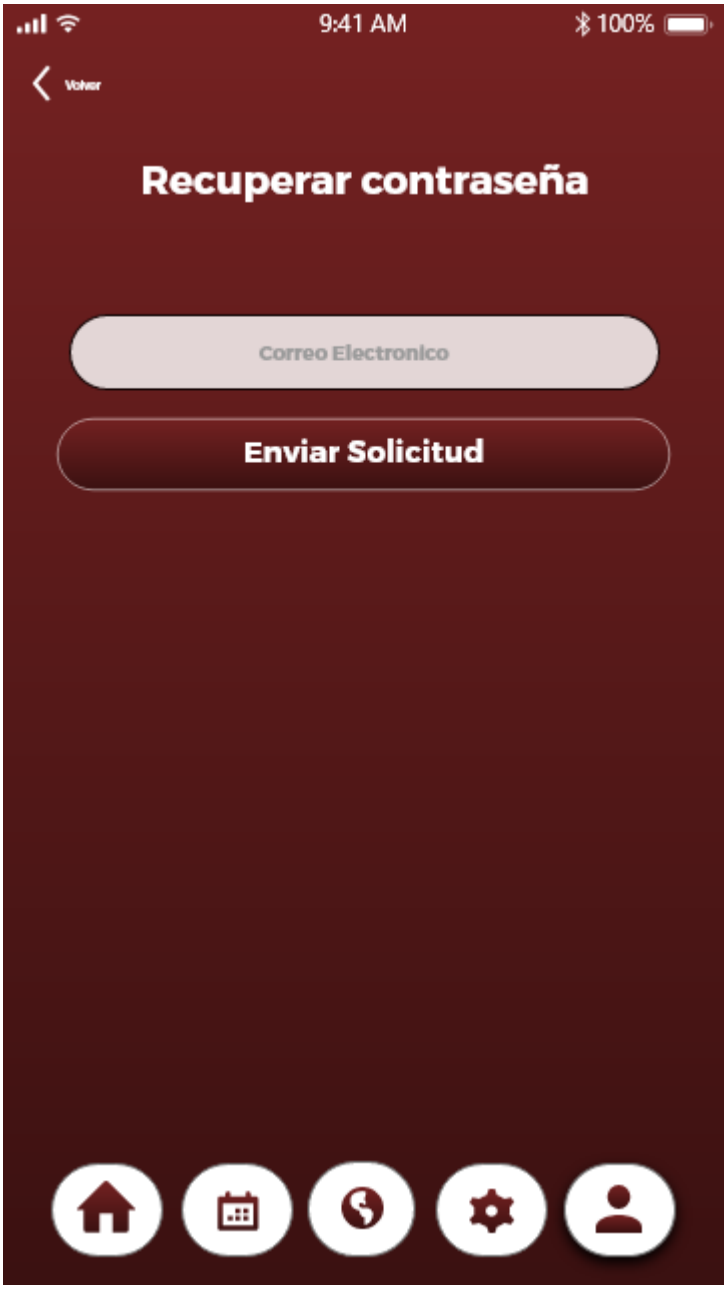


PERFIL NO LOGEADO



NOVEDADES – ELECCIÓN DE MOTEL

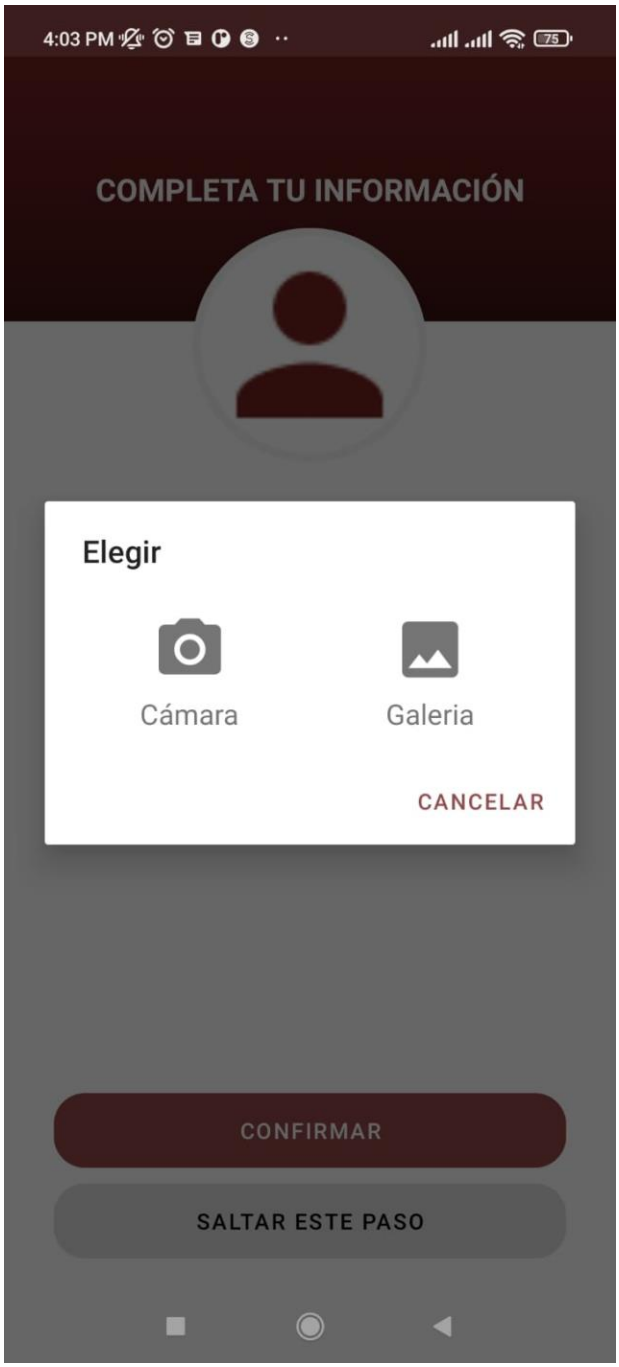
RECUPERAR CONTRASEÑA



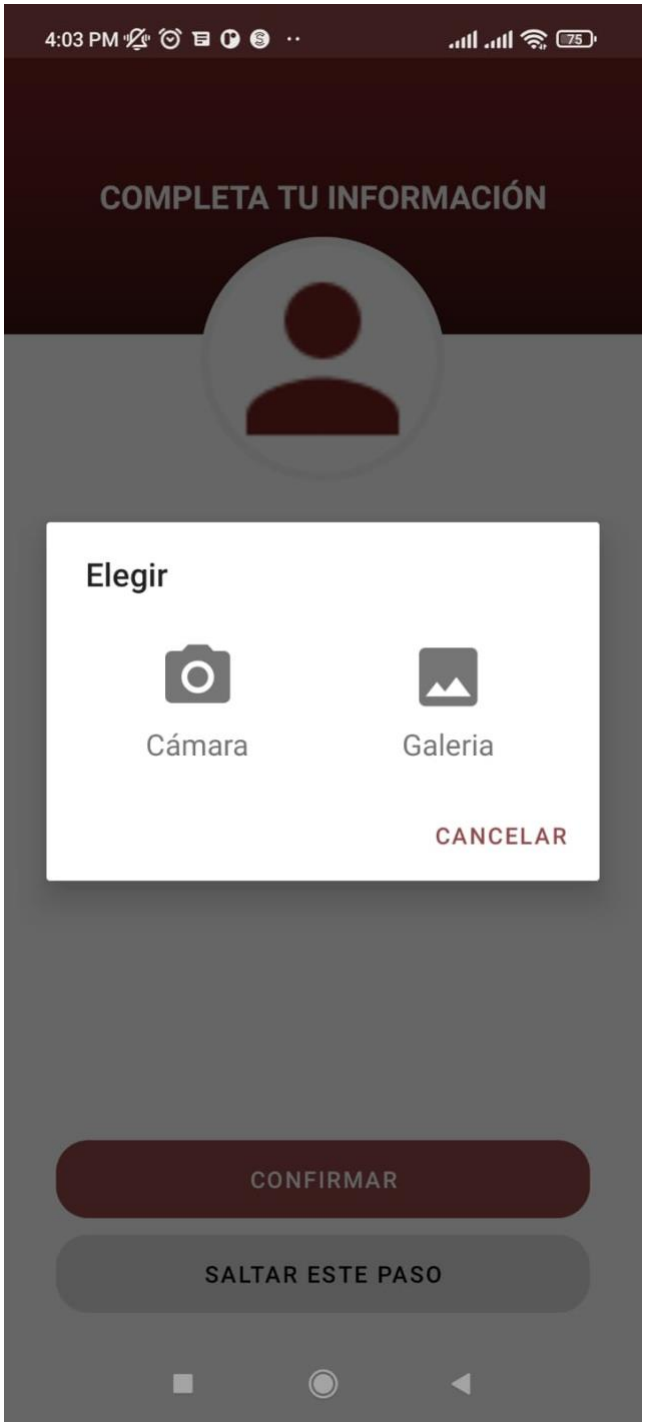
GENERAR RESERVA

ANEXOS 2. SISTEMA FUNCIONAL PREVIO A PRODUCCIÓN

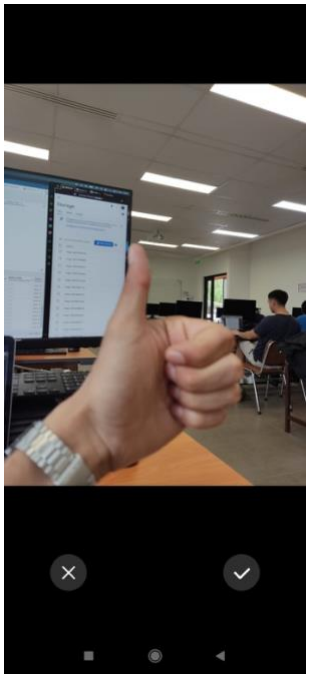
2.1



2.2


4:03 PM
COMPLETA TU INFORMACIÓN
Elegir
Cámara
Galeria
CANCELAR
CONFIRMAR
SALTAR ESTE PASO

2.3



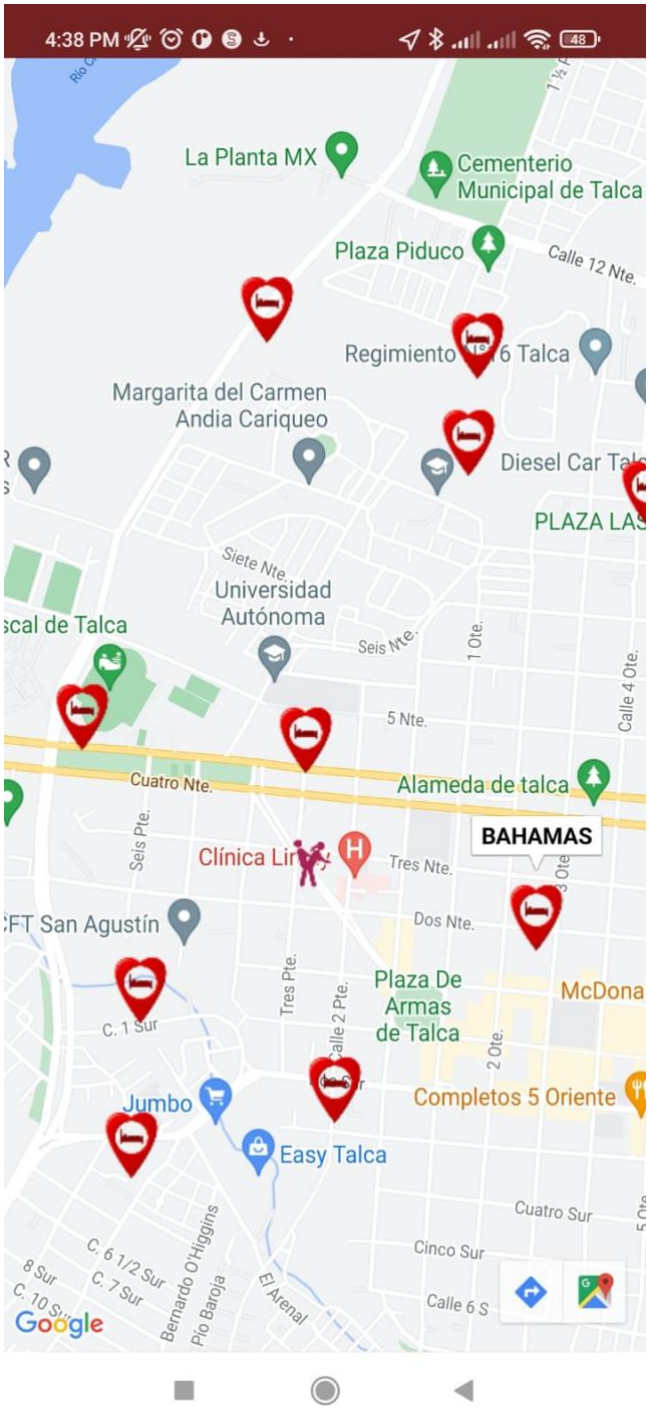
2.4

2.5

CONFIRMAR

SALTAR ESTE PASO

2.6


4:38 PM
La Planta MX
Cementerio Municipal de Talca
Plaza Piduco
Calle 12 Nte.
Regimiento 6 Talca
Margarita del Carmen Andia Cariqueo
Diesel Car Tal
PLAZA LAS
Siete Nte.
Universidad Autónoma
scal de Talca
Seis Nte.
1 Ote.
Calle 4 Ote.
5 Nte.
Cuatro Nte.
Alameda de talca
BAHAMAS
Seis Pte.
Clínica Lir
Tres Nte.
Dos Nte.
FT San Agustín
Tres Pte.
Calle 2 Pte.
Plaza De Armas de Talca
McDona
C. 1 Sur
2 Ote.
Jumbo
Completos 5 Oriente
Easy Talca
Cuatro Sur
Cinco Sur
C. 6 1/2 Sur
C. 7 Sur
8 Sur
C. 10 Sur
Bernardo O'Higgins
Pío Baroja
El Arenal
Calle 6 S
Google