

Universidad Distrital Francisco José De Caldas

Prototipado De Una Aplicación Móvil Para La Comunicación Entre Usuarios y Servicios Inherentes A Obras Civiles

Autores:
Harold Andres Salcedo
Everth David Cubides

Director:
Giovanny Tarazona PhD
Revisor:
John Jairo Londoño

Universidad Distrital Francisco José De Caldas
Facultad de Ingeniería
Especialización en Ingeniería de Software
Noviembre 29, 2020
Bogotá, Colombia



Universidad Distrital Francisco José De Caldas

Prototipado De Una Aplicación Móvil Para La Comunicación Entre Usuarios y Servicios Inherentes A Obras Civiles

Autores:
Harold Andres Salcedo
Everth David Cubides

Director:
Giovanny Tarazona PhD
Revisor:
John Jairo Londoño

Proyecto de grado para optar al título de especialista en ingeniería de software

Universidad Distrital Francisco José De Caldas
Facultad de Ingeniería
Especialización en Ingeniería de Software
Noviembre 29, 2020
Bogotá, Colombia

TABLA DE CONTENIDO

I DESCRIPCIÓN DE LA INVESTIGACIÓN 7

1. DESCRIPCIÓN DE LA INVESTIGACIÓN 9

1.1. Introducción . . . 10
1.2. Estudio del problema de investigación . . . 11
1.2.1. Planteamiento del problema. . . . 11
1.2.2. Formulación del problema. . . . 11
1.2.3. Sistematización del problema. . . . 12
1.3. Objetivos . . . 13
1.3.1. Objetivo general . . . 13
1.3.2. Objetivos especificos . . . 13
1.4. Justificación . . . 14
1.5. Hipótesis . . . 15
1.6. Marco referencial . . . 16
1.6.1. Marco teórico . . . 16
1.6.2. Aplicaciones Referente . . . 16
1.7. Marco Conceptual . . . 18
1.7.1. Aplicación hibrida . . . 18
1.7.2. IONIC . . . 18
1.7.3. .Net Core . . . 18
1.7.4. Geolocalización . . . 19
1.7.5. Profesionales del área civil . . . 19
1.7.6. Android . . . 19
1.7.7. iOS . . . 19
1.7.8. Scrum . . . 19
1.7.9. Metodología XP . . . 19
1.7.10. Bussines Intelligence . . . 20
1.7.11. Bases de datos relacionales . . . 20

1.8. Metodología de la investigación . . . 21
1.8.1. Tipo de estudio . . . 21
1.8.2. Método de investigación . . . 21
1.9. Alcances, limitaciones y resultados esperados . . . 21
1.9.1. Alcances . . . 21
1.9.2. Limitaciones . . . 21
1.9.3. Resultados esperados . . . 22

II DESARROLLO DE LA INVESTIGACIÓN 23

2. REQUERIMEINTOS DEL PROTOTIPO 25
2.1. Modelo del proceso . . . 26
2.1.1. Registro e inicio de sesión usuaros y proveedores . . . 27
2.1.2. Solicitud y atención de servicios . . . 28
2.2. Interfaz grafica . . . 29
2.2.1. Prototipo de pantalla para inicio de sesión . . . 30
2.2.2. Prototipo de pantalla de solicitud de servicio . . . 30
2.2.3. Prototipo de pantalla de confirmación de servicio . . . 31

3. DISEÑO DE LA BASE DE DATOS 33
3.1. Diseño de base de datos . . . 34
3.1.1. Modelo del contexto . . . 34
3.1.2. Modelo notacional de conjuntos . . . 35
3.1.3. Modelo conceptual asegurado . . . 36
3.1.4. Modelo relacional . . . 37

4. ARQUITECTURA EMPRESARIAL 39
4.1. Arquitectura empresarial . . . 40
4.1.1. Introducción . . . 40
4.1.2. MISIÓN . . . 40
4.1.3. VISIÓN . . . 40
4.1.4. OBJETIVOS . . . 40
4.1.4.1. Objetivo general . . . 40
4.1.4.2. Objetivos específicos . . . 40
4.1.5. Capa de motivación . . . 41
4.1.5.1. Punto de vista: Interesados o StakeHolder . . . 41

4.1.5.2. Punto de vista: Realización de objetivos . . . 42
4.1.5.3. Punto de vista: Principios . . . 42
4.1.5.4. Punto de vista: Motivación . . . 43
4.1.6. Capa de negocio . . . 45
4.1.6.1. Punto de vista: Organización . . . 45
4.1.6.2. Punto de vista: Cooperación de actor . . . 45
4.1.6.3. Punto de vista: Función de negocio . . . 46
4.1.6.4. Punto de vista: Proceso de negocio . . . 47
4.1.6.5. Punto de vista: Cooperación en proceso de negocio . . . 48
4.1.6.6. Punto de vista: Producto . . . 49
4.1.7. Capa de aplicación . . . 50
4.1.7.1. Punto de vista: Comportamiento de aplicación . . . 50
4.1.7.2. Punto de vista: Cooperación de aplicación . . . 51
4.1.7.3. Punto de vista: Estructura de aplicación . . . 52
4.1.7.4. Punto de vista: Uso de Aplicación . . . 53
4.1.8. Capa de tecnologia . . . 55
4.1.8.1. Punto de vista: tecnología . . . 55
4.1.8.2. Punto de vista: Uso de tecnología . . . 56
4.1.8.3. Punto de vista: Despliegue e implementación . . . 56
4.1.8.4. Punto de vista: Estructura de información . . . 57
4.1.8.5. Punto de vista: Realización del servicio . . . 57

5. BACK-END Y FRONT-END 59

5.1. Back - End . . . 60
5.1.0.1. Net Core . . . 60
5.1.0.2. Arquitectura de micro servicios . . . 62
5.1.0.3. Persistencia de datos . . . 66
5.1.0.4. Despliegue de aplicación . . . 67
5.1.0.5. Docker . . . 68
5.2. Front - End . . . 69
5.2.0.1. Arquitectura de aplicaciones PWA . . . 69
5.2.0.2. HTTPS . . . 70
5.2.0.3. Ventajas . . . 70
5.2.0.4. IONIC Framework . . . 71
5.2.0.5. Prototipo de aplicación . . . 72
5.2.0.6. Geolocalización API Google . . . 76

6. NORMATIVIDAD LEGAL **79**
6.1. Normatividad legal 79
6.2. Normativa de protección de datos personales 80
6.2.1. Consideraciones básicas del tratamiento de datos personales 80
6.3. Normatividad de comercio electrónico 80
6.3.1. Consideraciones a tener en cuenta en la comunicación de mensajes de datos 80

III CIERRE Y CONCLUSIONES **83**

7. CIERRE Y CONCLUSIONES **85**
7.1. Conclusiones 86
7.2. Trabajos futuros 88

A. Anexos **93**
A.1. EDT del proyecto 93
A.2. Cronograma 94
A.3. Mapa mental del proyecto 96
A.4. Matriz de contrastacion 97
A.5. Presupuesto del proyecto 98

LISTA DE FIGURAS

2.1. Modelo de registro usuarios, Fuente: Autores 27
2.2. Modelo de registro provedores, Fuente: Autores 28
2.3. Modelo de solicitud y atención de servicio, Fuente: Autores 29
2.4. Prototipo inicio de sesión, Fuente: Autores 30
2.5. Prototipo pantalla de solicitud, Fuente: Autores 31
2.6. Prototipo pantalla de confirmación, Fuente: Autores 32

3.1. Modelo del contexto, Fuente: Autores 35
3.2. Modelo notacional de conjuntos, Fuente: Autores 36
3.3. Modelo conseptual asegurado, Fuente: Autores 37
3.4. Modelo relacional, Fuente: Autores 38

4.1. Punto de vista: StakeHolder, Fuente: Autores 41
4.2. Punto de vista: Realizacón de objetivos, Fuente: Autores 42
4.3. Punto de vista: Principios, Fuente: Autores 43
4.4. Punto de vista: Motivación , Fuente: Autores 44
4.5. Punto de vista: Organización , Fuente: Autores 45
4.6. Punto de vista: Cooperación de actor , Fuente: Autores 46
4.7. Punto de vista: Función de negocio , Fuente: Autores 47
4.8. Punto de vista: Proceso de negocio, Fuente: Autores 48
4.9. Punto de vista: Cooperación en proceso de negocio , Fuente: Autores 48
4.10. Punto de vista: Producto, Fuente: Autores 49
4.11. Punto de vista: Comportamiento de la aplicación. Fuente: Los autores 51
4.12. Punto de vista: Cooperación de aplicación. Fuente: Los autores 52
4.13. Punto de vista: Estructura de aplicación. Fuente: Los autores 53
4.14. Punto de vista: Uso de Aplicación. Fuente: Los autores 54
4.15. Punto de vista: tecnología: Los autores 55
4.16. Punto de vista: Uso de tecnología: Los autores 56

4.17. Punto de vista: Despliegue e implementación: Los autores 57
4.18. Punto de vista: Estructura de información: Los autores 57
4.19. Punto de vista: Realización del servicio: Los autores 58

5.1. Pilar de tecnologias .Net Framework 4.x [1] 60
5.2. .Net Framework vs .Net Core [1] 61
5.3. Enfoque monolitico. Fuente: Los autores. 62
5.4. Diagrama microservicios. Fuente: Los autores. 63
5.5. Arquitectura del Back-end. Fuente: Los autores. 64
5.6. Contenerización de aplicaciones [2] 68
5.7. Modelo mental del funcionamiento de IONIC framework. [3] 71
5.8. Comparación de arquitecturas de aplicaciones móviles. [3] 72
5.9. Inicio de sesión Felix app profesionales. Fuente: Los autores. 73
5.10. Inicio de sesión Felix app usuarios. Fuente: Los autores. 74
5.11. Nueva orden. Fuente: Los autores. 74
5.12. Solicitud servicio. Fuente: Los autores. 75
5.13. Finalizar orden. Fuente: Los autores. 75
5.14. Implementación del API de geolocalización de google. Fuente: Los autores. . 76
5.15. Implementación del API de geolocalización de google. Fuente: Los autores. . 77
5.16. Implementación del API de geolocalización de google. Fuente: Los autores. . 77
5.17. Modos geolocalización [4] 78

A.1. Edt del proyecto. Fuente: Los autores. 93
A.2. Diagrama de gantt. Fuente: Los autores. 94
A.4. Actividades del proyecto. Fuente: Los autores. 95
A.3. Fases del proyecto. Fuente: Los autores. 95
A.5. Mapa mental del proyecto. Fuente: Los autores. 96
A.6. Matriz contrastacion del proyecto. Fuente: Los autores. 97
A.7. Tabla de gastos generales del proyecto. Fuente: Los autores. 98
A.8. Grafico de gastos generales del proyecto. Fuente: Los autores. 98
A.9. Tabla de costos por servicios personales del proyecto. Fuente: Los autores. . . 99
A.10. Grafico de costos por servicios personales del proyecto. Fuente: Los autores. . 100
A.11. Tabla de costos totales del proyecto. Fuente: Los autores. 101
A.12. Grafico de costos totales del proyecto. Fuente: Los autores. 101

Parte I

DESCRIPCIÓN DE LA INVESTIGACIÓN

Capítulo 1

DESCRIPCIÓN DE LA INVESTIGACIÓN

1.1 Introducción

La presente investigación tiene como fin presentar una solución útil para poder unificar los servicios prestados por los diferentes profesionales del área civil, entiéndase por trabajadores del área civil a las personas que, por su conocimiento y experticia adquirida de manera formal o empírica en diferentes áreas como plomería, electricidad locativa, construcción y otras clases de arreglos locativos, este sector tiene el problema de ser en su mayor parte descentralizado, debido a esto es muchas ocasiones no es posible contactar al profesional adecuado para las necesidades de los usuarios.

Para analizar esta problemática tenemos que empezar a explorar las causas del sector, indefectiblemente el mayor problema es la descentralización y la informalidad del sector, esto es fácil de notar debido a que en su gran mayoría son personas que no están asociados a ninguna empresa por ende cuando se requiere un profesional del sector, se acude al conocimiento previo si este existe (buscar a una persona que haya arreglado algo antes), o el voz a voz entre amigos y familiares, o ir a los depósitos ferroeléctricos y preguntar por personas conocidas por los ferreteros. A partir de esto encontramos un segundo problema, la necesidad de un profesional en cualquier momento, esto debido a que los daños locativos en muchas ocasiones no son previsibles.

Buscar una solución ingenieril a este problema de comunicación entre prestadores de servicios y usuarios finales nos causa un interés académico para solventar con una aplicación móvil que permita un canal de comunicación efectiva, aumento de cobertura por los profesionales y una rápida atención al usuario final. Para cumplir con este propósito se desea utilizar las mejores prácticas, modelos, metodologías, arquitecturas, etc; que permitan llevar a buen puerto una aplicación estable y escalable.

1.2 Estudio del problema de investigación

1.2.1 Planteamiento del problema.

Un daño locativo en cualquier hogar o establecimiento es un problema que no es previsible y cuando sucede genera pérdidas de tiempo, disponibilidad y uso de las instalaciones, esto implica la necesidad de contactar un profesional que realice las respectivas reparaciones y adecuaciones para poder volver a tener las locaciones en óptimas condiciones. Un problema que derivado es la fiabilidad del profesional seleccionado para realizar los distintos trabajos pues si bien es cierto en algunas ocasiones se cuenta con un profesional quien ha trabajado anteriormente para el usuario esto no siempre se cumple o no se mantiene el contacto con el mismo especialista. [5]

Cuando un usuario no dispone del contacto con un especialista previo suele buscar información de contactos para solventar su problemática entre su círculo social con el fin de buscar contactos fiables para sus necesidades, si este método no funciona, aun cuenta con la posibilidad de buscar profesionales en otras fuentes como internet o las ferreterías locales en donde por lo general se tiene el contacto de personas que pueden solventar problemas de arreglos locativos.

El tiempo de respuesta para solventar estos problemas normalmente depende de la ubicación del especialista, el resultado del diagnóstico realizado por el mismo es por esto que disponer de especialistas en el sector del cliente se vuelve una necesidad para solventar la problemática. [6]

Por otra parte, la descentralización del gremio de los profesionales de obras civiles no les permite ampliar su espectro laboral ya que normalmente cuentan con una clientela definida y dependen del voz a voz entre los clientes para ser recomendados entre la ciudadanía, esto mantiene su nicho laboral bastante cercado. [4]

1.2.2 Formulación del problema.

¿Cómo obtener una estructura cognitiva que evidencie y soporte una comunicación efectiva entre usuarios y personal capacitado en obras civiles que están dispersos?

1.2.3 Sistematización del problema.

- ¿Los profesionales de las obras civiles estarían dispuestos a utilizar una aplicación móvil?
- ¿Es posible separar por especialidades a los profesionales de las obras civiles?
- ¿Los usuarios estarían dispuestos a solicitar sus reparaciones locativas por medio de su celular?
- ¿Qué partes de la ciudad serían las más beneficiadas con esta clase de aplicación?
- ¿Es posible ubicar una persona por medio de la plataforma de servicios de geolocalización de Google?

1.3 Objetivos

1.3.1 Objetivo general

- Prototipar una aplicación web progresiva disponible en IOS y Android para la comunicación efectiva entre profesionales de la obra civil y los posibles usuarios en la ciudad de Bogotá.

1.3.2 Objetivos especificos

- Documentar las características de las herramientas a utilizar en el desarrollo de la aplicación.

- Construir una interfaz gráfica con la herramienta IONIC para la interacción de usuario.

- Montar una estructura de servicios con .net core para soportar las peticiones de la interfaz gráfica.

1.4 Justificación

La revolución tecnológica, día a día hace que los dispositivos y wearables sean cada vez más potentes y a accesibles al público en general, producto de esto se ven más frecuentemente diferentes aplicaciones que permiten la comunicación entre prestadores de servicio y usuarios finales, casos como uber, rappi, indriver etc son un claro ejemplo de la masificación de plataformas tecnológicas al servicio del cliente. Mas sin embargo se sigue generando necesidades que aún no son cubiertas por estos servicios, como el presente caso para los prestadores de servicios de obra civil.

Una de las principales problemáticas para abordar dentro de esta investigación es la descentralización que por la naturaleza de su profesión genera este gremio. Ya que este tipo de labores se realiza a petición y estos trabajadores generalmente no tienen una ubicación única, más por el contrario son personas de constante movimiento dentro de la ciudad y sus alrededores, por parte del usuario este problema no mejora, ya que en la mayoría de los casos para contactar un profesional que solvente sus problemáticas de arreglos locativos se depende de un voz a voz o búsquedas en diferentes fuentes no unificadas (internet, ferreterías, etc) y en el mejor de los casos el contacto de un especialista con el cual ya había tenido una relación laboral previa.

Acorde a lo anteriormente expuesto se evidencia la necesidad de solventar este problema prototipando una aplicación móvil que permita generar un canal habitual de comunicación entre los profesionales de obra civil y sus usuarios finales inicialmente para la ciudad de Bogotá, esto con el fin de reducir la descentralización del gremio y permitiendo a los usuarios finales tener siempre un especialista disponible para sus necesidades de obras civiles.[4]

Principalmente este proyecto busca automatizar el proceso de comunicación reduciendo el tiempo de búsqueda de un usuario cuando se genera la necesidad de solventar daños locativos o se pretende realizar mejoras en los inmuebles, esta solución beneficia a profesionales y a usuarios ya que para los profesionales significa un aumento en la prestación de sus servicios y para los clientes representará un ahorro en tiempo.

1.5 Hipótesis

Al construir una aplicación hibrida que permita la comunicación entre usuarios finales y los profesionales de obra civil se conseguirá una reducción de tiempo y costos al momento de solventar las necesidades de arreglos locativos además de aumentar el espectro laboral para los trabajadores de obra civil.

1.6 Marco referencial

1.6.1 Marco teórico

El presente trabajo tiene como misión presentar la investigación, desarrollo y demás trabajos realizados con la finalidad de construir un prototipo de aplicación hibrida la cual tiene por finalidad construir un canal de comunicación entre el usuario final y los profesionales de obras civiles. Para este fin se plantea el uso de metodologías agiles de desarrollo tales como SCRUM, XP, RAD, etc, además de los conocimientos adquiridos sobre arquitectura de software, es con esto en mente que por medio de este marco profundizaremos en los conceptos tecnológicos y arquitectónicos que permiten dar vida al prototipo propuesto, de igual forma estableceremos las pautas sobre conceptos como geolocalización y concurrencia de usuarios que permitirán dar un valor agregado a la solución propuesta.

Para dar un paso inicial empezaremos analizando los antecedentes a nuestra solución propuesta.

1.6.2 Aplicaciones Referente

Timbrit es una aplicación móvil que nace en Buenos Aires argentina para el año 2017 con la finalidadde eliminar las empresas intermediarias entre profesionales del rubro hogar y los clientes, la promesade valor de la aplicación es generar el contacto sin intermediarios. Cuenta con aplicacionesindependientes para Android y IOS. Actualmente a fecha de 13 de abril de 2020 cuenta con alrededor de 1080 reviews en Google play donde su calificación promedio es de 3.4/5, al revisar los feedback de los usuarios se evidencian problemas con sus métodos de comunicación con los profesionales, problemas con su sistema de registro, problemas con la geolocalización del usuario, entre otras. En app store cuenta con alrededor de 312 descargas con una calificación promedio de 4.5 / 5 y en los feedback de la aplicación no se ven mayores inconvenientes en la aplicación salvo problemas de registro. Este ejemplo nos da una primera pincelada de las necesidades del usuario final pues la interfaz de la aplicación se considera intuitiva y la geolocalización se vuelve primordial. El contar con aplicaciones separadas para desarrollo en Android y IOS hace que las sensaciones del usuario final sean diferentes para las dos plataformas mostrándose en este caso mejor la experiencia en IOS.

Chepe Pepe es un emprendimiento colombiano que nace en Medellín Colombia para el año 2018, realizando su primer reléase para julio de 2018. Su objetivo es ser una aplicación

móvil gratuita que cruza la oferta y la demanda de servicios de reparaciones para la casa y la oficina. Concretamente, todo lo relacionado con instalaciones, reparaciones, mantenimientos y atención de emergencias. Todo en una plataforma muy sencilla e intuitiva, que convierte la experiencia del usuario en una gestión ágil y rápida.

No cuenta dentro de sus funcionalidades con geolocalización para usuarios y clientes dando un catálogo de usuarios sin tener en cuenta la localización del cliente, esto aunque aumenta la visibilidad de los trabajadores hace que los clientes tengan un proceso de búsqueda mas extenso para sus necesidades.

Actualmente cuenta con una calificación promedio en Google Play de 3.9/5 con 116 reviews donde las principales quejas de los usuarios son referentes a los altos costos en los servicios ofrecidos por medio de la aplicación.

En app store cuenta con 21 reviews y una calificación promedio de 4.2, en los reviews no se evidencian muchos problemas, aunque no es una muestra considerable de usuarios para tener en cuenta.

Podemos afirmar que es un emprendimiento en crecimiento ya que aumento sus zonas de cobertura de la ciudad de Medellín a tener disponibilidad en las ciudades de Cali y Bogotá.

Rappi Favor Rappi favor nace como un servicio prestado por la plataffforma rappi para la solucion de necesidades generales (pago recibos publicos, encomiendas, etc). cuentan con una plataforma tecnologica bastante robusta ya que es parte de la suite que ofrece rappi a sus clientes mas no es enfocado en clientes del area civil

1.7 Marco Conceptual

El desarrollo de nuestro prototipo se encuentra ubicado en el estudio y posterior aplicación de tecnologías de última generación aplicadas al desarrollo de aplicaciones híbridas. Con el fin de manejar un marco común conceptual especificaremos conceptos relevantes para nuestro proyecto.

1.7.1 Aplicación hibrida

Las aplicaciones híbridas son aplicaciones móviles diseñadas en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un framework que permite adaptar la vista web a cualquier vista de un dispositivo móvil.[7]

Ventajas:

- Es de creación más sencilla y sus costos son significativamente más bajos.
- El código base es reutilizable en distintas plataformas.
- Unificación del desarrollo.

1.7.2 IONIC

Ionic es una estructura tecnológica (Framework) de código abierto que se utiliza en el desarrollo de aplicaciones móviles híbridas,[8] es decir, se combinan el HTML5, CSS y JavaScript dando como resultado aplicaciones con una interfaz amigable e intuitiva para el usuario que luego se comercializan o descargan en plataformas como Android o IOs. [9]

1.7.3 .Net Core

.NET Core es un framework informático administrado, gratuito y de código abierto para los sistemas operativos Windows, Linux y macOS. Es un sucesor multiplataforma de .NET Framework. El proyecto es desarrollado principalmente por Microsoft bajo la Licencia MIT.1. [10]

1.7.4 Geolocalización

La geolocalización ayuda a identificar la ubicación geográfica real de los objetos, como dispositivos móviles o cualquier terminal conectada a Internet. El término ”geolocalización”se refiere tanto al proceso de localización geográfica de los objetos como a la ubicación geográfica real identificada. [11]

1.7.5 Profesionales del área civil

Personal que por sus estudios o experiencia empírica tienen conocimientos en áreas como, plomería, construcción, ebanistería, electricidad, arquitectura, diseño de interiores, y demás especialidades de arreglos locativos.

1.7.6 Android

Android es un sistema operativo que se emplea en dispositivos móviles, por lo general con pantalla táctil. De este modo, es posible encontrar tabletas (tablets), teléfonos móviles (celulares) y relojes equipados con Android, aunque el software también se usa en automóviles, televisores y otras máquinas. [3]

1.7.7 iOS

El iOS es el sistema operativo diseñado por Apple para sus productos, iPhone, iPad, iPod Touch, y Apple TV [12].

1.7.8 Scrum

Scrum es una de las Metodologías Ágiles más populares y conocidas en la actualidad. Se trata de una herramienta muy útil en espacios donde los grupos de trabajo tienen dificultades para hacer las acciones u operaciones que les lleven a objetivos en común. Dicho de otro modo, Scrum sirve para que equipos multidisciplinares trabajan en entornos complejos, donde los requisitos son muy cambiantes, y los resultados se tienen que obtener en un plazo corto de tiempo. [13]

1.7.9 Metodología XP

La Metodología XP “Extreme Programming” o “Programación Extrema” es una de las llamadas metodologías ágiles de desarrollo de software más exitosas. Es habitual relacionarla

con scrum, y la combinación de ambas asegura un mayor control sobre el proyecto, y una implementación más efectiva y eficiente.[14]

1.7.10 Bussines Intelligence

Hace referencia a un conjunto de métodos y técnicas por medio de las cuales se transforman datos básicos nativos en información entendible para la organización[15], esto implica que los datos recopilados en el sistema se transforman en información valiosa para la subsecuente toma de decisiones y así mejorar el flujo del proceso en la organización. [16]

1.7.11 Bases de datos relacionales

El concepto de bases de datos relacionados fue postulado por Edgar F. Codd, científico británico, y hace referencia a que una relación representa un conjunto de entidades con las mismas propiedades. Esta serie de relaciones se componen de una serie de filas o registros cuyos valores dependen de los atributos o columnas las cuales componen las tablas las cuales son sistemas de clasificación construidas por filas horizontales y columnas verticales que permite que permiten agrupar datos, los atributos de los que se componen vienen definidos del sistema relacional con el que se haya diseñado. [17]

1.8 Metodología de la investigación

1.8.1 Tipo de estudio

Estudio explicativo

1.8.2 Método de investigación

Método de observación: seleccionamos el método de observación como método de investigación debido a que con este tenemos la capacidad de describir y explicar el comportamiento al haber obtenido los datos correspondientes al desarrollo de la aplicación y a su eventual uso por parte de usuarios finales y especialistas, esta observación se realizará en el ámbito práctico del proyecto y la misma se realizará de la manera más objetiva posible. Esta observación no modifica nuestro objeto de análisis y nos ayudará a confirmar nuestra hipótesis de trabajo.

1.9 Alcances, limitaciones y resultados esperados

1.9.1 Alcances

Prototipado y desarrollo de una aplicación móvil que permita la comunicación asertiva entre usuarios y profesionales de obras civiles, la aplicación se centra en buscar la facilidad de que los usuarios encuentren diferentes tipos de servicios locativos a través de la aplicación.

1.9.2 Limitaciones

- Al ser un prototipado no se realiza un despliegue en ambiente productivo.
- Los datos personales de las personas encuestadas están disponibles bajo la autorización de cada uno de los encuestados.
- Al ser una aplicación PWA se debe desarrollaran sobre framework IONIC y .Net Core
- Al final del proyecto se prototipará un MVP (Mínimo Producto Viable).
- La aplicación solo abarca información de servicios locativos disponibles para clientes.

- El rendimiento de la aplicación dependerá del acceso a internet y de las características propias del equipo en el que se instale.
- El acceso a la información solo se podrá realizar con una conexión a internet.

1.9.3 Resultados esperados

- Construir un mínimo producto viable que permita una comunicación asertiva y eficaz entre usuarios y profesionales de obra civil.
- Obtener una base de conocimiento sobre los usuarios potenciales de la aplicación.
- Profundizar, aplicar y conceptualizar metodologías de programación aplicadas en los diferentes frameworks utilizados para este prototipo.

Parte II

DESARROLLO DE LA INVESTIGACIÓN

Capítulo 2

REQUERIMEINTOS DEL PROTOTIPO

2.1 Modelo del proceso

Dentro del diseño del prototipo al cual apunta este proyecto de investigación, se desea elaborar dos aplicaciones móviles, una para usuarios y otra apara proveedores de servicios respectivamente, de tal forma que podamos cumplir con el objetivo de este proyecto el cual es su base permitir la comunicación entre usuarios y proveedores.

Inicialmente se procede a realizar un diseño de un modelo objetivo del prototipo, el cual busca de manera puntual ser funcional y resistente al cambio en todo el proceso de desarrollo, teniendo en cuenta esto, a continuación, se describen los pasos necesarios para el proceso de solicitud de servicios locativos por parte de los usuarios a través de la herramienta:

- Registro e inicio de sesión app usuarios.
- Registrar la ubicación actual del usuario.
- Solicitar servicio.
- Aceptar proveedor del servicio.
- Realizar finalización del servicio.

De igual manera, para el proveedor especificamos el proceso que se debe seguir para la aceptación de las solicitudes y posterior a esto la culminación del servicio prestado:

- Registro e inicio de sesión app proveedores.
- Aprobación de EMBER S.A.S para poder ejercer funciones.
- Registrar la ubicación actual del proveedor.
- Aceptar servicio de acuerdo a la profesión.
- Realizar servicio.
- Realizar finalización del servicio.

De esta manera y con los procesos anteriormente mencionados indicamos a través de distintos modelos estructurales y gráficos el comportamiento de esta solución.

2.1.1 Registro e inicio de sesión usuaros y proveedores

Este escenario ocurre cuando un usuario o proveedor de servicios requiere hacer parte de la herramienta con el fin de solicitar o realizar los diferentes servicios locativos que se pueden prestar a través de la aplicación, de esta manera se inicia alimentando la base de datos con los usuarios que harán parte de ella, de esta manera realizan todo el proceso de registro y validación de datos por parte de los actores presentes para este proceso.

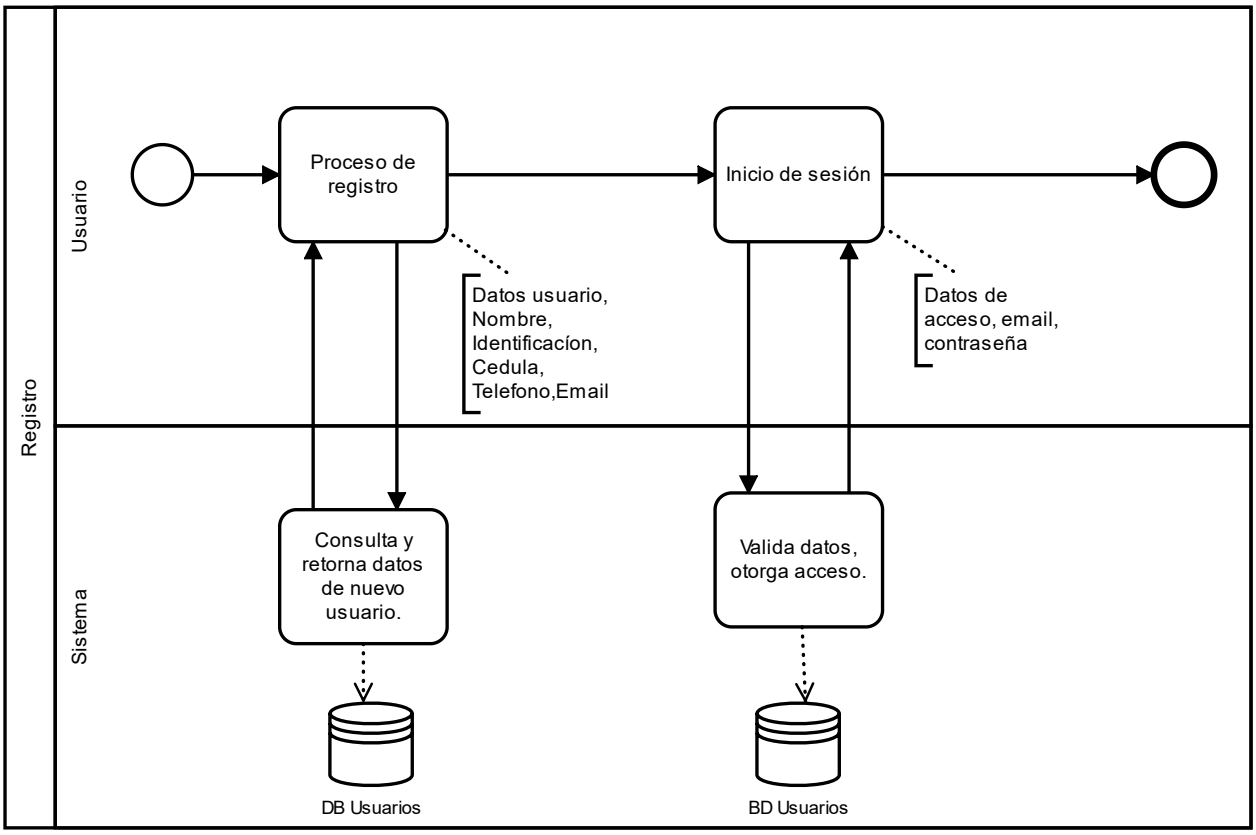


Fig. 2.1: Modelo de registro usuarios, Fuente: Autores

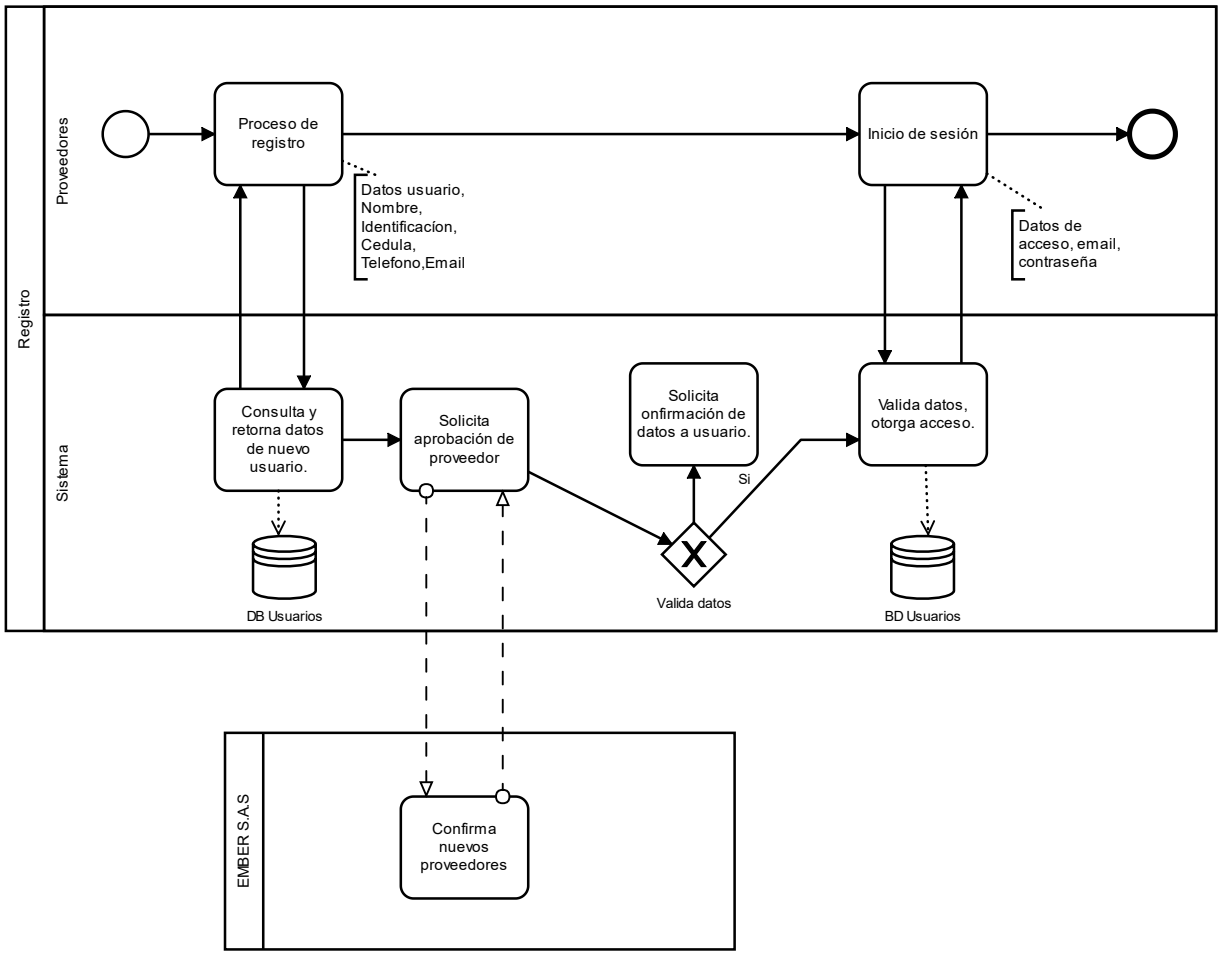


Fig. 2.2: Modelo de registro provedores, Fuente: Autores

2.1.2 Solicitud y atención de servicios

En este escenario participan los dos actores del proceso de solicitud, tanto el cliente como el proveedor, donde inicia con una solicitud de servicio específico (eléctrico, plomería, pintura, etc.), este puede ser una solicitud programada o de atención inmediata, la solicitud llegara de manera contigua a los proveedores de servicios que se encuentren específicamente en el tipo solicitado, posterior a esto el proveedor acepta el servicio y se traslada al lugar del evento para realizar el respectivo trabajo, seguidamente se finaliza el evento desde la aplicación al igual que el cliente, de modo que ambas partes se encuentren satisfechas con la atención de la solicitud.

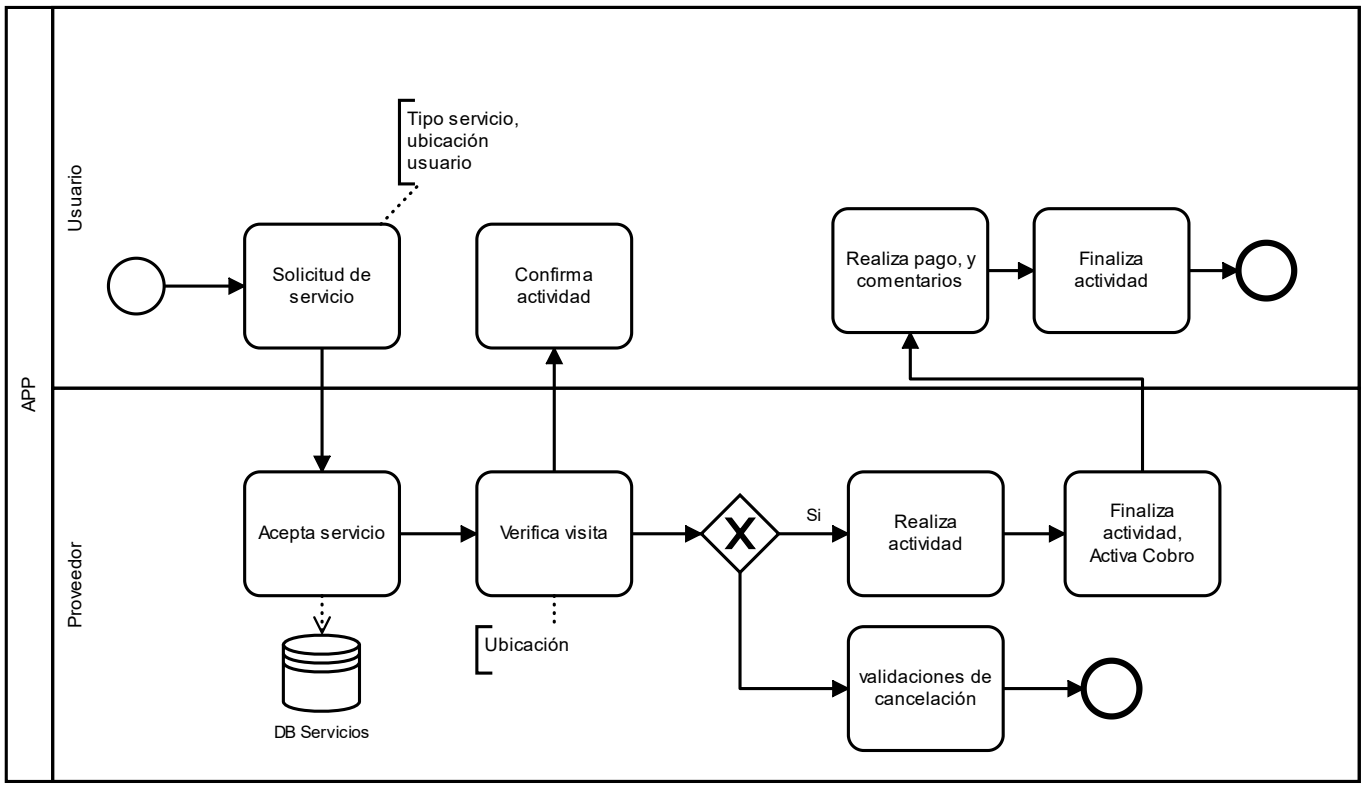


Fig. 2.3: Modelo de solicitud y atención de servicio, Fuente: Autores

2.2 Interfaz grafica

El proceso de desarrollo de la interfaz grafica de usuario hace parte fundamental de la ingeniería de software, es la capa del desarrollo que interactúa directamente con el usuario y es fundamental hacer un análisis del comportamiento humano en el manejo de este tipo de herramientas. Se tiene en cuenta que para este punto es viable parametrizar puntos fundamentales como la ergonomía, la facilidad de uso, la amigabilidad, la simpleza, entre otras, todo esto con el fin de reducir en la mayor cantidad el esfuerzo de un usuario por ejecutar las diferentes acciones que permita realizar la aplicación, otras características a tener en cuenta son la flexibilidad y naturalidad, estas dos últimas características se refieren al modelo de la

realidad que pueda ver reflejado un usuario en la pantalla del dispositivo.[18]
Teniendo en cuenta las características anteriormente mencionadas se muestra a continuación el prototipo de diseño para la aplicación móvil FELIXAPP.

2.2.1 Prototipo de pantalla para inicio de sesión

Los usuarios ingresaran a la aplicación a través de un inicio de sesión el cual solicitara los datos de email y contraseña para poder acceder a los servicios, si aun este no se ha registrado la misma opción mostrara el link respectivo para realizar el registro satisfactorio en la aplicación.

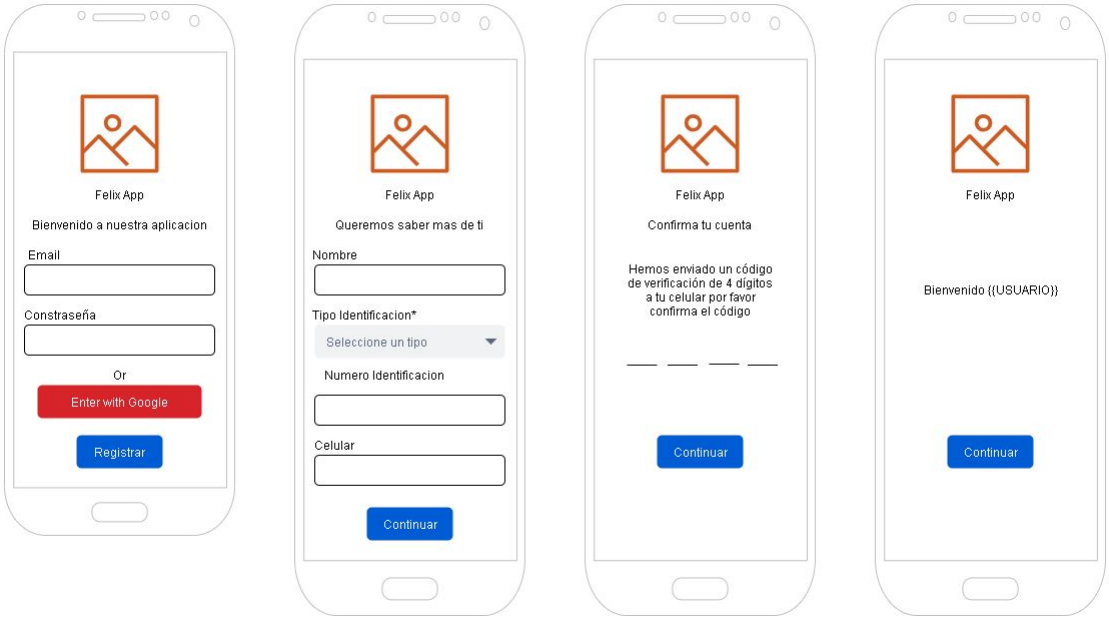


Fig. 2.4: Prototipo inicio de sesión, Fuente: Autores

2.2.2 Prototipo de pantalla de solicitud de servicio

Al ingresar a la aplicación se habilitará la funcionalidad de mapa en la cual se marcará la ubicación actual del usuario, siguiente a eso permitirá desde la vista inicial, realizar la solicitud de un servicio indicando el tipo (eléctrico, plomería, pintura etc.) la ubicación del servicio y una descripción explicativa del servicio solicitado.

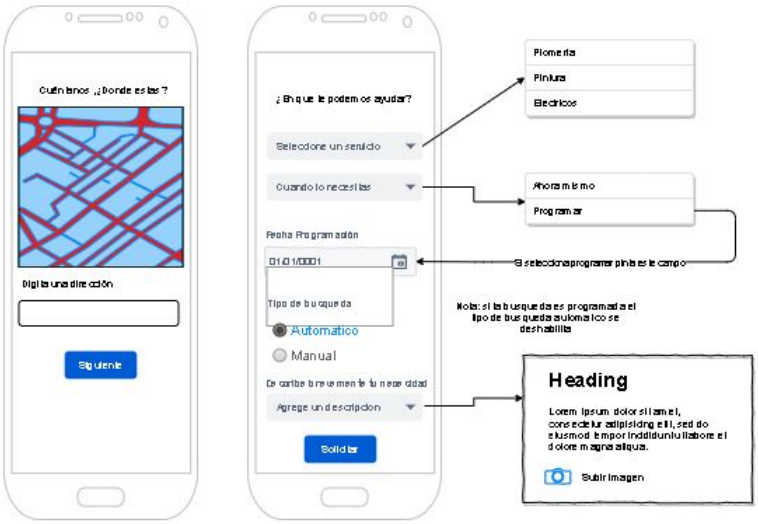


Fig. 2.5: Prototipo pantalla de solicitud, Fuente: Autores

2.2.3 Prototipo de pantalla de confirmación de servicio

Para este escenario ya sea una solicitud de servicio inmediata o programada, tendrá una confirmación en la cual se indique si fue tomada por un proveedor de servicios o no, de ser positiva la confirmación indicara los datos del proveedor y el tiempo estimado de llegada, adicionando datos pertinentes que informen al usuario, de esta manera se sostiene una un prototipo funcional del proceso de solicitud de servicios a través de la herramienta no sin antes mencionar que estas vistas pueden aumentar en el trascurso del desarrollo de la aplicación, con el fin de dar el mejor nivel calidad al proceso de diseño y arquitectura de software de este proyecto.

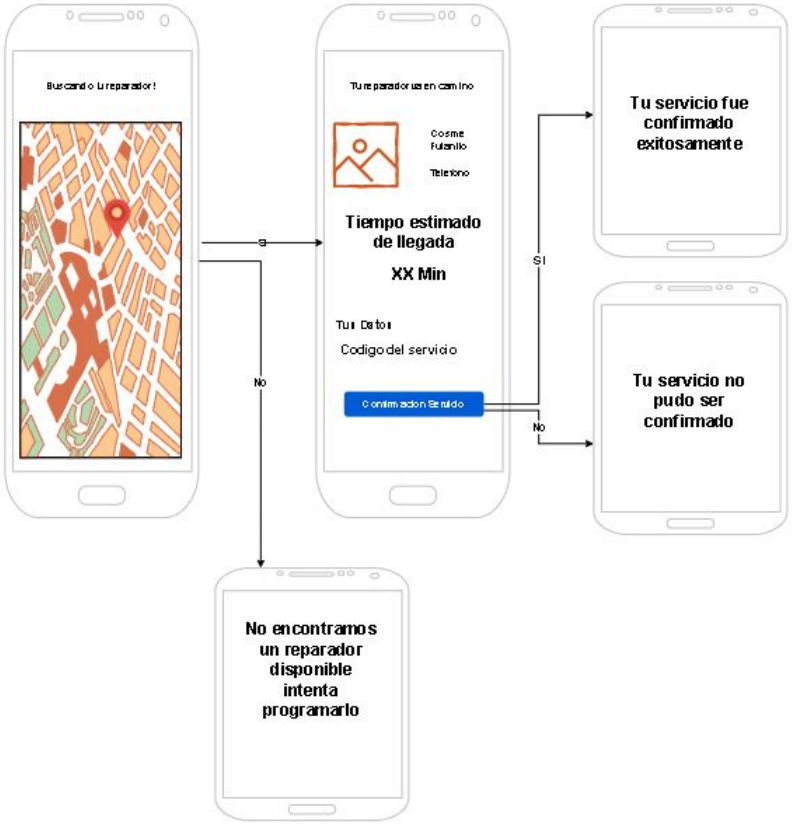


Fig. 2.6: Prototipo pantalla de confirmación, Fuente: Autores

Capítulo 3

DISEÑO DE LA BASE DE DATOS

3.1 Diseño de base de datos

Es preciso que antes de realizar el diseño del software y de la base de datos se debe conocer de forma clara y precisa la lógica del contexto del negocio [30], con base en esta logica, se podra proponer la construcción segura, funcional y pertinente de la base de datos, para lo cúal como se menciono anteriormente es necesario definir en esencia el contexto de la logica de negocio sobre el cual se va a desarrollar este diseño [19].

En el momento de realizar el diseño de un base de datos es necesario tener en cuenta diferentes ítems esenciales que se asocian a la construcción, estos escenarios son: la logica de negocio, la metodoligia del diseño, el motor de base de datos y la futura inteligencia de negocio.

De los escenarios anteriormente mencionados una de los mas importantes es la logica de negocio, pues de este no solo se desprende el diseño de la base de datos, sino también el diseño del software, este debe ser pensado con base en la misión inherente a la logica de negocio permitiendo así derivar los sistemas y subsistemas que a través de los cuales permitirá cumplir el propósito del objetivo del proyecto. La forma inicial de realizar la abstracción de la lógica de negocio y sobre la cual se dará inicio al diseño de la base de datos muestran las clases sobre las que se ejercerá control; para realizar esto es común que se realice una narrativa guiada por parte de los creadores del proyecto, que permita de forma clara y puntual separar la lógica de negocio, de la lógica de la inteligencia de negocios para tener un acierto acorde con el presente diseño.

3.1.1 Modelo del contexto

Para realizar la identificación de la lógica del negocio y en consecuencia cada una de las Çlases"que hacen parte está de esta logica, se hace necesario realizar las dos siguientes preguntas:

¿Que se va controlar? y ¿Para quien se va a controlar?

El siguiente diagrama representa el modelo del contexto en el cual visualizamos la lógica del negocio y las clases principales que interactúan en el modelo.

Después de contextualizar la misión y el propósito del negocio se procede a asignar las

diferentes clases al contexto y sus respectivos atributos, a este proceso se le conoce como Autodeterminación, teniendo en cuenta el modelo del contexto, a continuación se identificaran las dependencias funcionales entre cada una de las clases del modelo de la logica de negocio [19].

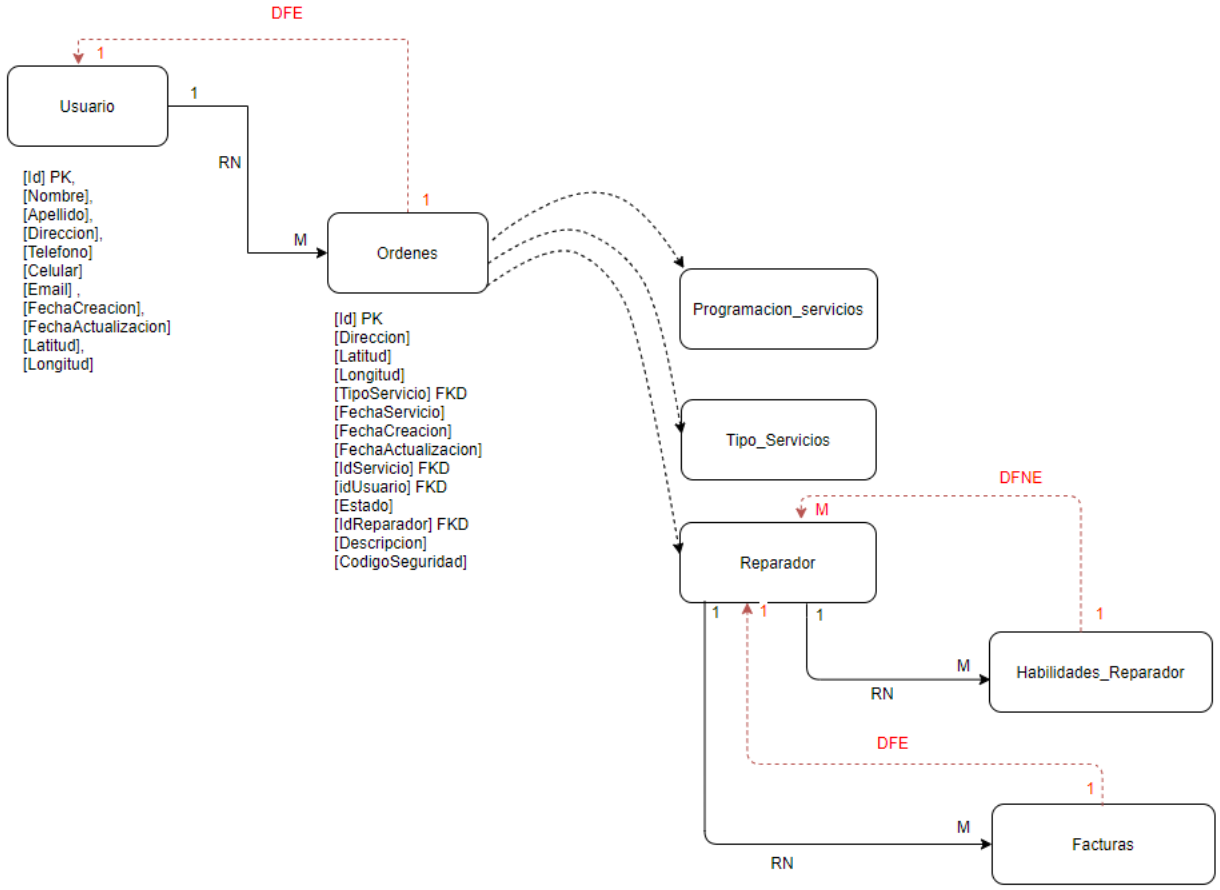


Fig. 3.1: Modelo del contexto, Fuente: Autores

3.1.2 Modelo notacional de conjuntos

Modelo notacional de conjuntos nos permite dar una primera impresión de las necesidades de modelamiento y manejo para cumplir con la lógica de negocio de la aplicación permitiendo empezar así el desarrollo de un posterior modelo conceptual asegurado donde divisamos las diferentes relaciones y cómo se interconecta la base de datos entre sí.

el modelo de negocio para félix app nos indica que es una aplicación encargada de controlar las ordenes de servicio de los usuarios y controlar la gestión de pagos para los reparadores de

las órdenes de trabajo, con esto en cuenta nuestra entidad de mayor valor es la orden desde donde desprendemos la narrativa guiada para este ejercicio.

la compañía ember sas con el fin de realizar la implementación de la aplicación felix app necesita realizar el diseño de la base de datos la cual permite gestionar las órdenes de servicios locativos para los usuarios y gestionar el pago de dichas órdenes a los reparadores.

```
Usuario
{PKP idUsuario, nombre, apellido, dirección, teléfono, celular,
 email, fechaCreacion, fechaActualizacion, Latitud, Longitud}

Órdenes
{PK Id, direccion latitud longitud tiposervicio FKD, fechaServicio,
 fechaCreacion,FechaActualizacion idServicio FKD,
 idUsuario FKD, estado FKD, idReparador FKD, Descripcion, codigoSeguridad}

TipoServicio
{PKP tipoServicio, nombre}

Servicio
{PKP IdServicio, nombre, activo}

Estado
{PKP idEstado, nombre}

Reparador
{PKP idUsuario, nombre, apellido, dirección, teléfono, celular, email,
 fechaCreacion, fechaActualizacion, Latitud, Longitud, estado}

HabilidadesReparador
{(PKC)(idReparador,idServicio)}

OrdenPago
{(PKC)(idReparador,idOrden) valorTotal,ValorReparador,ValorServicio,ValorFelix,Pagada,
DescripcionTrabajo,FechaCreacion,calificacionServicio}

```

Fig. 3.2: Modelo notacional de conjuntos, Fuente: Autores

3.1.3 Modelo conceptual asegurado

El siguiente modelo Inicia localizando a la izquierda del diseño, el conjunto en modo "entidad", que soluciona la "última clase" de la lógica de negocio. asignando las relaciones de derecha a izquierda de uno a muchos. El diseño deberá ser leído en adelante de derecha a izquierda e irá al motor también de derecha a izquierda. De esta manera y teniendo en cuenta los anteriores modelos mostrados, identificamos el modelo conceptual asegurado para la base de datos de FelixApp. Demostrando así un correcto funcionamiento en la lectura del diseño para esta aplicación.

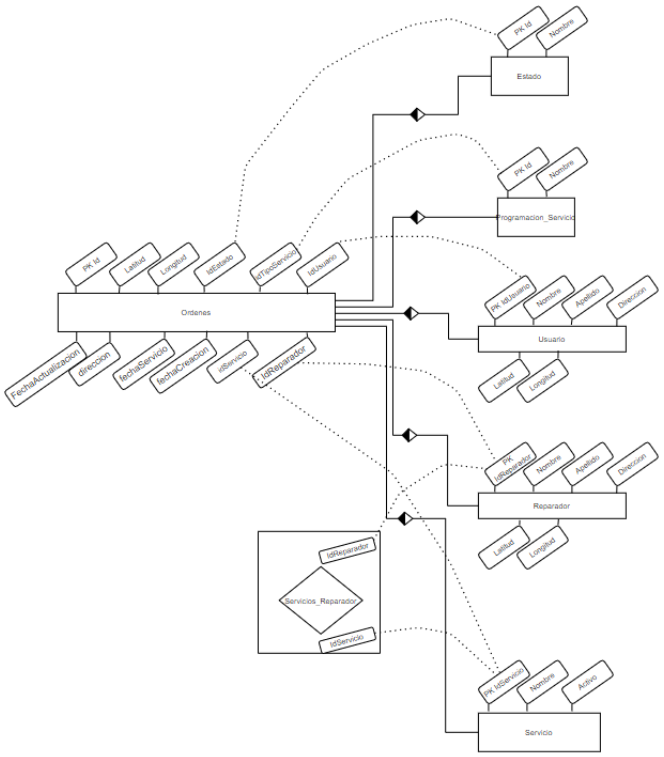


Fig. 3.3: Modelo conseptual asegurado, Fuente: Autores

3.1.4 Modelo relacional

Luego de contextualizar el proceso de negocio que se desea modelar es pertinente realizar la asignación de los atributos abstractos a las clases correspondientes adicionando a esto la asignación de las dependencias funcionales de cada una de las clases según corresponda, luego de tener resueltas las diferentes dependencias funcional podremos obtener el modelo de datos relacional tal como se observa a [17].

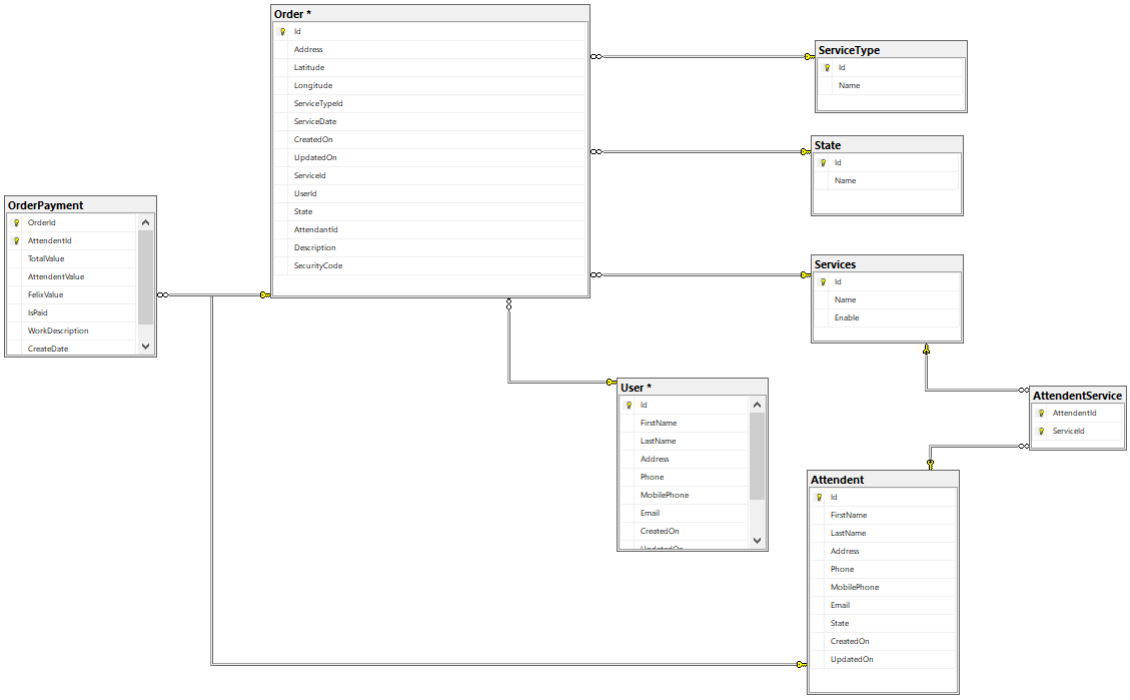


Fig. 3.4: Modelo relacional, Fuente: Autores

El presente modelo debe construirse de izquierda a derecha y debe ser leído de derecha a izquierda siempre teniendo en cuenta la orientación de uno a muchos [19].

Capítulo 4

ARQUITECTURA EMPRESARIAL

4.1 Arquitectura empresarial

4.1.1 Introducción

Para realizar la respectiva arquitectura empresarial, colocamos el caso hipotético de la empresa cliente, que para tal fin será EMBER S.A.S, la cual será una compañía de transformación digital y desarrollo de software a la medida, una organización que trabajara utilizando las mejores practicas de desarrollo y últimas tecnologías para así ofrecer a los cliente herramientas y soluciones de alta calidad, buscando hacer la diferencia entre las empresas competidoras en el mercado de software y desarrollo de apps.

4.1.2 MISIÓN

"Proveer a nuestros clientes y/o socios servicios tecnológicos innovadores que les permitan ampliar el liderazgo y generación de nuevas oportunidades de negocio que brinden ventaja competitiva haciendo el debido uso de los recursos económicos para el logro de sus propósitos, con el apoyo de funcionarios capacitados que contribuirán al cumplimiento de los planes organizacionales."

4.1.3 VISIÓN

"Ser a nivel nacional una empresa de servicios tecnológicos innovadores con posicionamiento a nivel nacional a través del uso de sus productos y servicios.."

4.1.4 OBJETIVOS

4.1.4.1. Objetivo general

"Ser una marca líder con reconocimiento en el país por brindar soluciones tecnológicas de calidad."

4.1.4.2. Objetivos específicos

- *"Construir herramientas de software de alta calidad que permitan solucionar las necesidades objetivas de los clientes."*
- *"Implementar arquitecturas de software y buenas prácticas de desarrollo para aumentar la calidad de los productos desarrollados."*

4.1.5 Capa de motivación

La capa de motivación hace referencia a los conceptos iniciales del proyecto, en los cuales se identifican a partir de los objetivos de la empresa, para visualizar la conceptualización de esta capa se identifican seis diferentes puntos de vista que son: Interesados o Stakeholder, realización de objetivos, contribución, principios realización de requerimientos y motivación.[20]

4.1.5.1. Punto de vista: Interesados o StakeHolder

A continuacion se detalla los elementos que intervien en el punto de vista de interesados o Stakeholders para la empresa EMBER S.A.S.

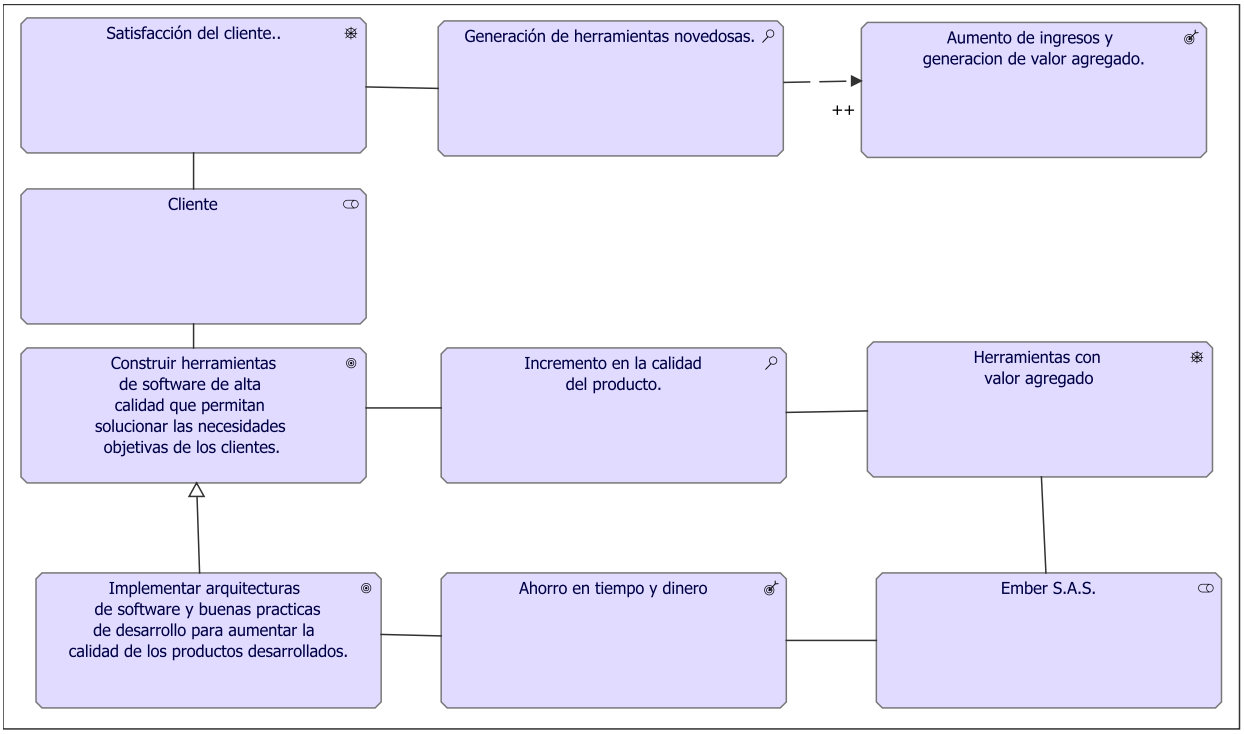


Fig. 4.1: Punto de vista: StakeHolder, Fuente: Autores

En este punto de vista se detallan como a partir de un objetivo específico de la empresa se muestran los implicados del proyecto, para este caso los clientes y EMBER S.A.S, donde a partir de diferentes manejadores se espera llegar a los resultados definidos uno de ellos el aumento de ingresos y la generación de valor agregado.

4.1.5.2. Punto de vista: Realización de objetivos

A continuacion se detalla los elementos que intervien en el punto de vista realización de objetivos para la empresa EMBER S.A.S.

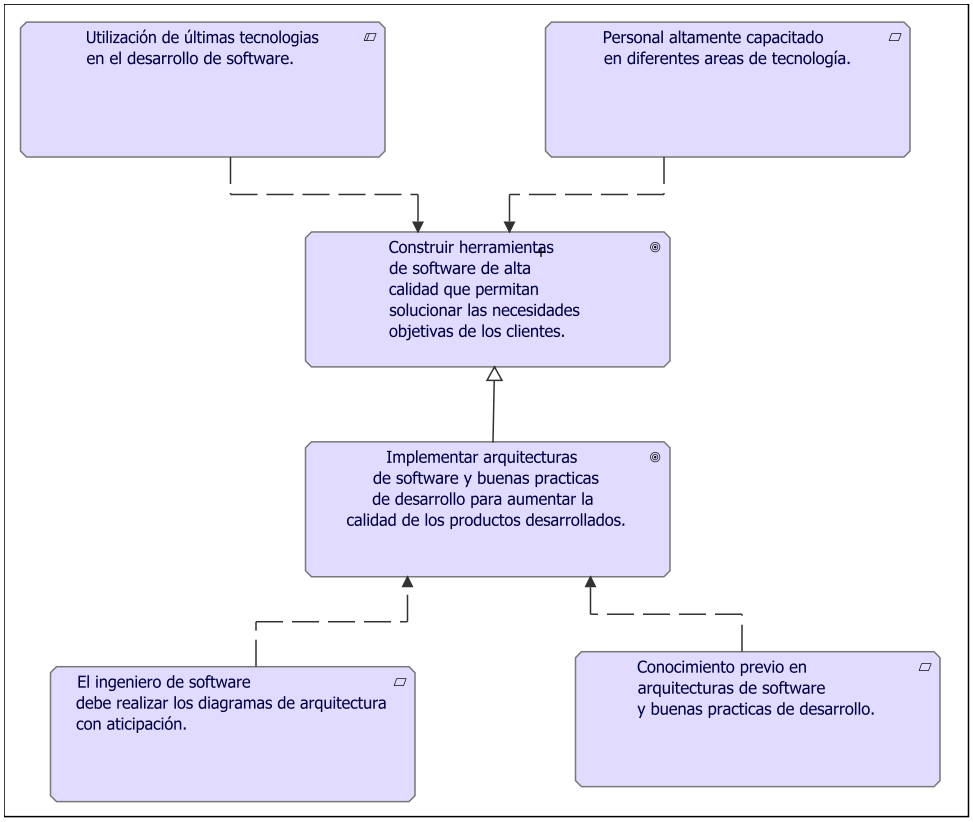


Fig. 4.2: Punto de vista: Realizacón de objetivos, Fuente: Autores

En este caso se observan dos de nuestros más importantes objetivos específicos, donde se destacan para cada uno de ellos las restricciones y los requerimientos que necesitan cada uno de ellos para poderse cumplir en el entorno del proceso de desarrollo de la empresa.

4.1.5.3. Punto de vista: Principios

A continuacion se detalla los elementos que intervien en el punto de vista de principios para la empresa EMBER S.A.S.

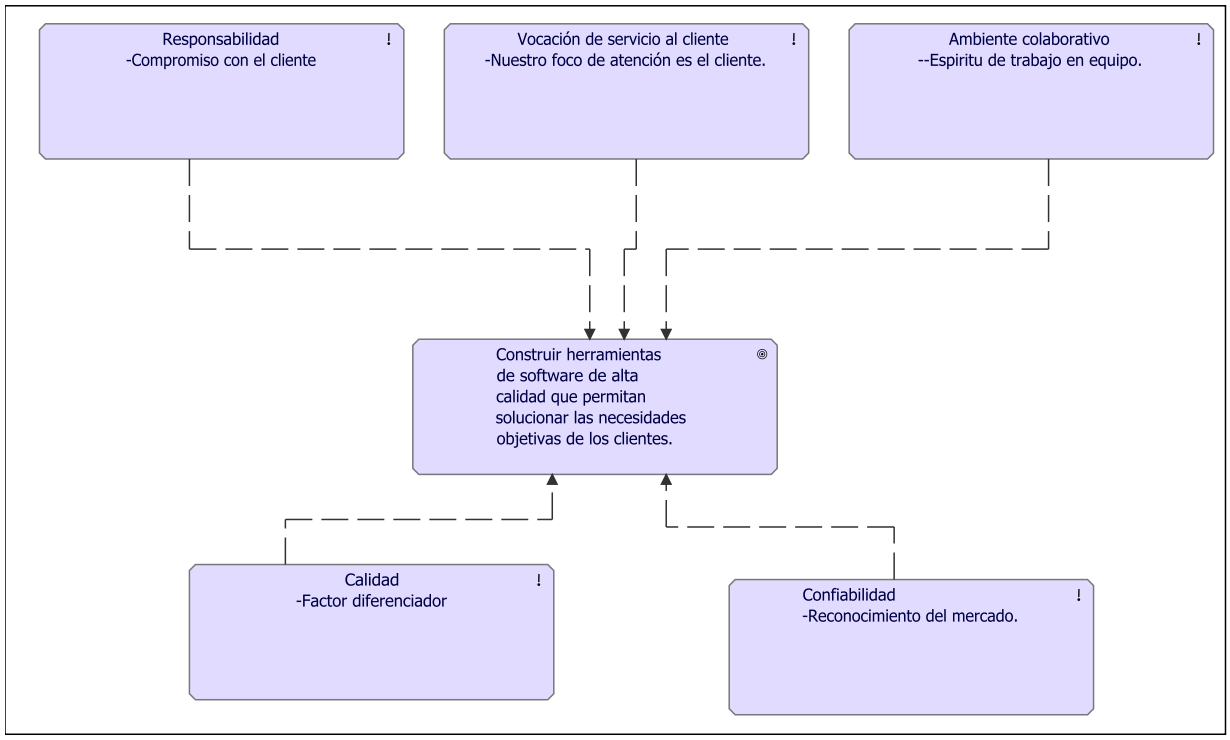


Fig. 4.3: Punto de vista: Principios, Fuente: Autores

En este punto de vista destacamos los principios que desarrolla la compañía como valores agregados, estos se desprenden de un objetivo específico y se evidencia como la responsabilidad, la vocación al servicio al cliente, el ambiente colaborativo, la calidad y la confiabilidad se convierten en piezas que ancladas al cumplimiento de los objetivos suponen un factor diferenciados para el desarrollo de este objetivo.

4.1.5.4. Punto de vista: Motivación

A continuacion se detalla los elementos que intervien en el punto de vista de motivacón para la empresa EMBER S.A.S.

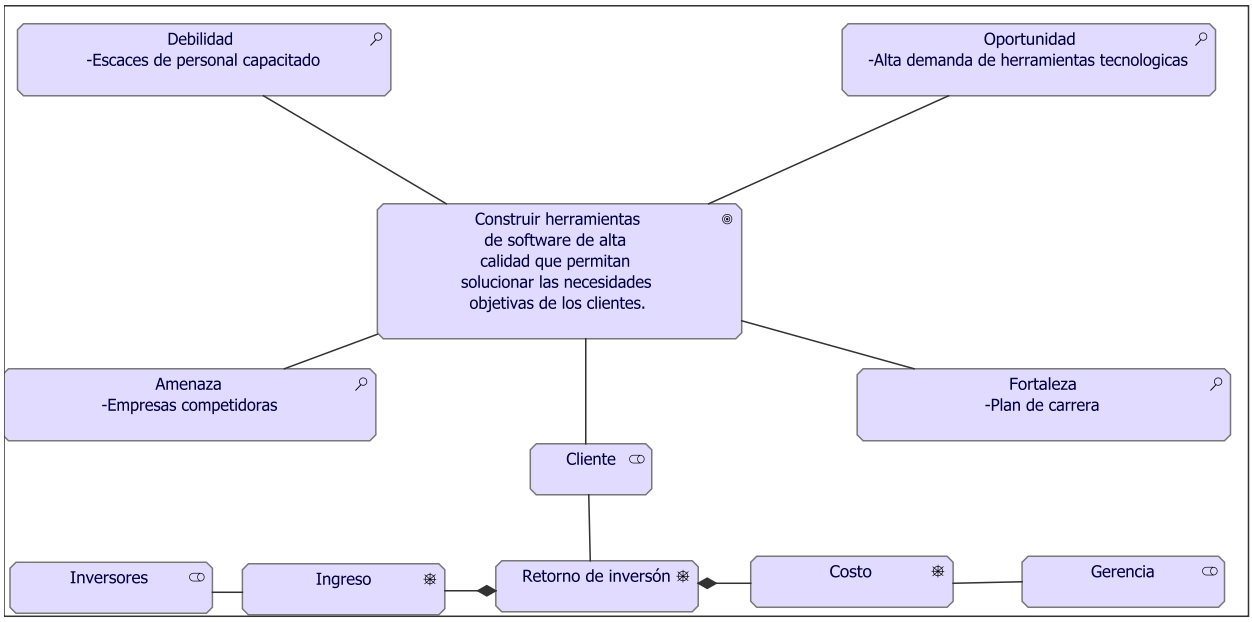


Fig. 4.4: Punto de vista: Motivación , Fuente: Autores

Visualizamos en este punto de vista como a través de un DOFA encontramos las oportunidades, debilidades amenazas y fortalezas, y como convergen entre ellas a la realización del objetivo principal, de igual manera se observa el proceso en el que interactúan los implicados que con un resultado buscan generar y cumplir el objetivo hacia el cliente.

4.1.6 Capa de negocio

La capa de negocio nos permite observar la empresa de un punto de vista operativo e independiente de los recursos tecnológicos, en esta capa se busca representar las entidades que componen la organización y las relaciones que tienen entre sí.
Se busca con esta hacer la relación entre la empresa y los objetivos del negocio, establecemos las relaciones entre los procesos, las funciones, servicios, roles del negocio, así como los principios del funcionamiento del producto.

4.1.6.1. Punto de vista: Organización

A continuacion se detalla los elementos que intervien en el punto de vista de organización para la empresa EMBER S.A.S.

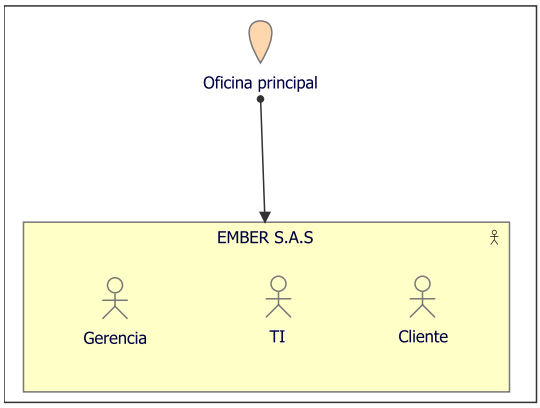


Fig. 4.5: Punto de vista: Organización , Fuente: Autores

EMBER S.A.S estará constituido principalmente por la gerencia y por un departamento de TI (Tecnologías de la información), de este departamento sobre sale el área de desarrollo e infraestructura donde radica la construcción de software y soluciones, sobre este modelo se representarán los modelos de negocio que se quieren implementar.

4.1.6.2. Punto de vista: Cooperación de actor

A continuacion se detalla los elementos que intervien en el punto de vista Cooperación de actor para la empresa EMBER S.A.S.

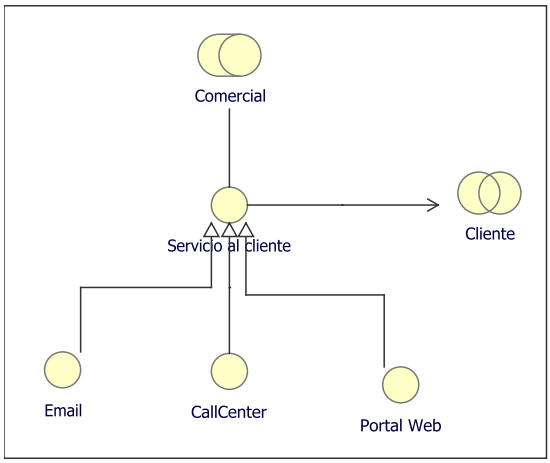


Fig. 4.6: Punto de vista: Cooperación de actor , Fuente: Autores

Para la organización es indispensable el servicio al cliente, en este intervienen diferentes actores como los son el cliente y el comercial, el cual ejecutan el proceso de servicio al cliente imprentando diferentes medios para este proceso como lo son el envio de correos electrónicos, comunicación a través de CAllCenter y por medio del portal web de la compañía, esto con el fin de prestar un correcto servicio pos y pre venta.

4.1.6.3. Punto de vista: Función de negocio

A continuacion se detalla los elementos que intervien en el punto de vista Función de negocio para la empresa EMBER S.A.S.

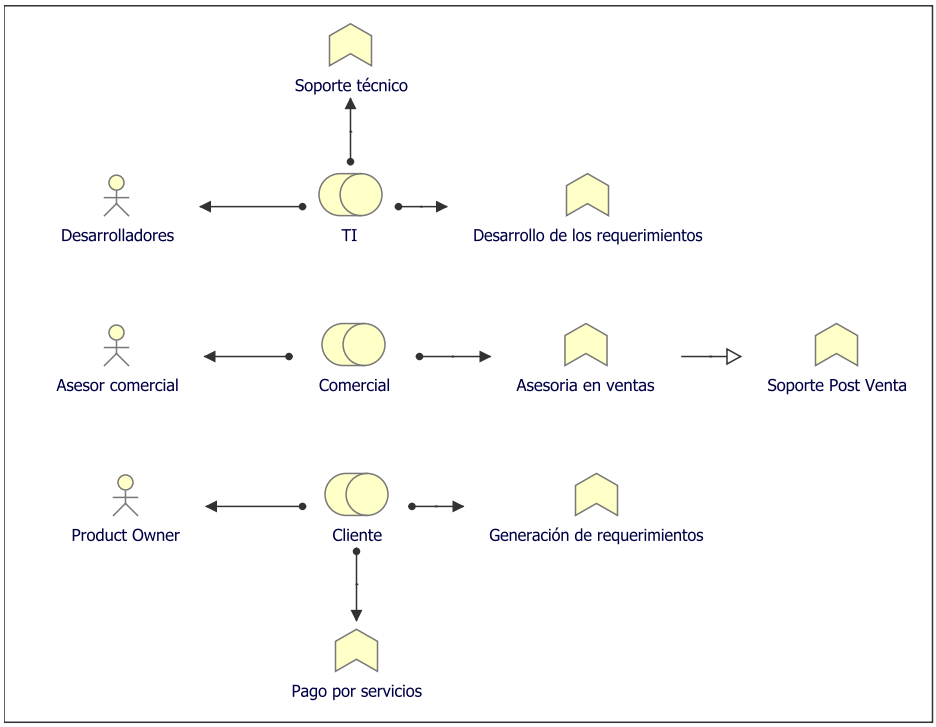


Fig. 4.7: Punto de vista: Función de negocio , Fuente: Autores

A través del diagrama representamos las funciones que ejecutan los actores del proceso, los desarrolladores, los cuales hacen parte del área de TI, deberán realizar tareas como lo son el desarrollo de las soluciones y el debido soporte técnico que es implícito dentro del proceso de venta de las soluciones, Los asesores comerciales, los cuales pertenecen al área comercial prestaran el servicio de asesoría en ventas y el servicio postventa siendo nodo fundamental entre la comunicación con el cliente y el soporte técnico pre y post venta que se da, el product Owner el cual hace referencia al cliente serán los encargados de generar los requerimientos y el pago de los mimos, el conjunto de esta funciones generar el debido proceso para la generación de productos de software.

4.1.6.4. Punto de vista: Proceso de negocio

A continuacion se detalla los elementos que intervien en el punto de vista Proceso de negocio para la empresa EMBER S.A.S.

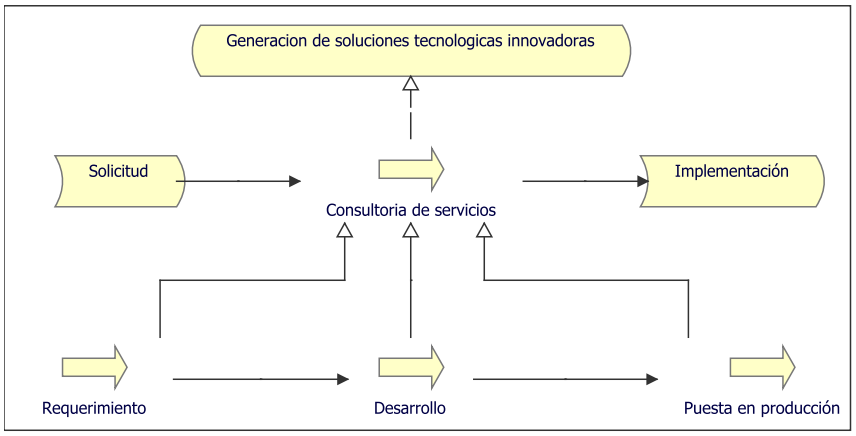


Fig. 4.8: Punto de vista: Proceso de negocio, Fuente: Autores

El proceso de negocio se deprende del servicio, el cual para este caso es la generación de soluciones tecnológicas innovadoras, el proceso inicia con un evento de solicitud y finaliza con un evento de implementación, para realizar estos eventos se debe ejecutar la función de consultoría de servicios, para que esta función se cumpla debe pasar por un proceso de requerimiento en su inicio, pasando por el desarrollo de la solución o herramienta y terminando con las puesta en producción, de esta manera se representa el proceso de negocio principal de EMBER S.A.S.

4.1.6.5. Punto de vista: Cooperación en proceso de negocio

A continuacion se detalla los elementos que intervien en el punto de vista Cooperación en proceso de negocio para la empresa EMBER S.A.S.

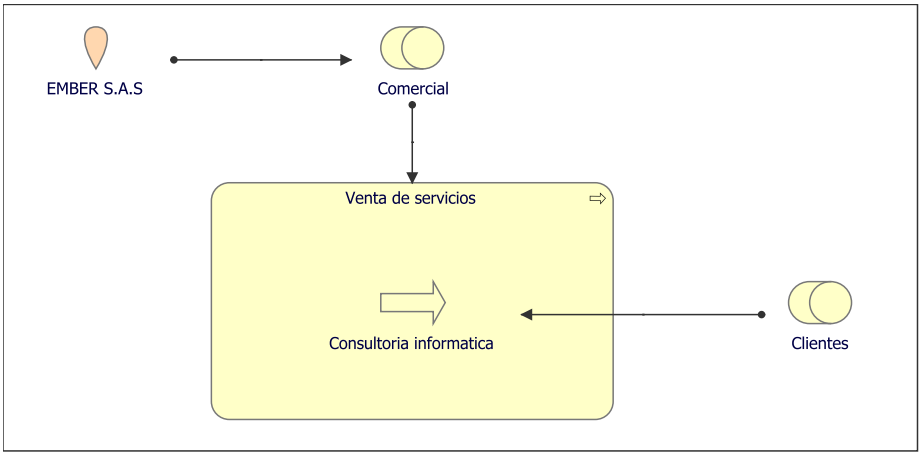


Fig. 4.9: Punto de vista: Cooperación en proceso de negocio , Fuente: Autores

Con la finalidad de cumplir con el proceso de caso de estudio para este caso, que corresponde a la venta de servicios tecnológicos y construcción de software, identificamos los actores que intervienen, encontramos que el proceso principal es la consultoría informática donde participa el actor comercial como parte de EMBER S.A.S y el actor cliente, de esta manera se evidencia la cooperación de los actores que intervienen en el proceso.

4.1.6.6. Punto de vista: Producto

A continuacion se detalla los elementos que intervien en el punto de vista Producto para la empresa EMBER S.A.S.

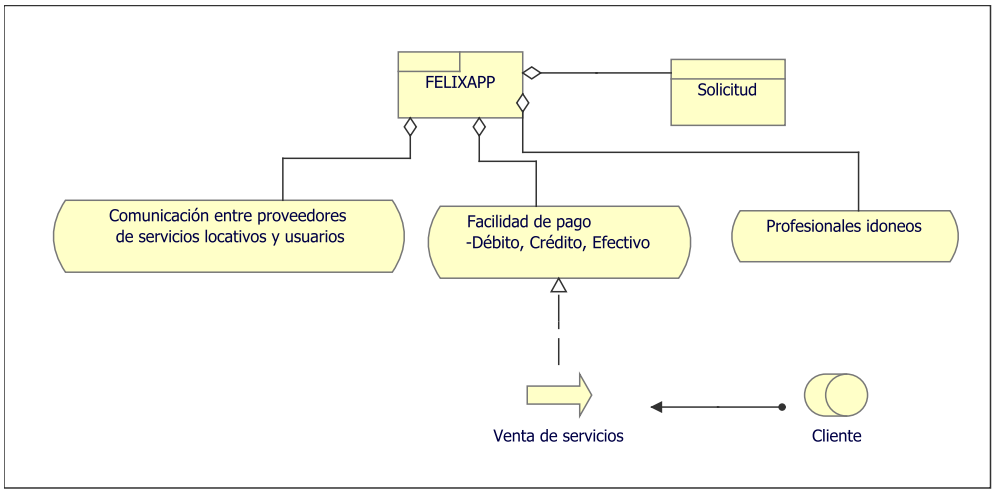


Fig. 4.10: Punto de vista: Producto, Fuente: Autores

Identificamos el primer servicio, para este caso una aplicación móvil denominada FELIXAPP, una aplicación la cual permite la comunicación entre usuarios y proveedores de servicios locativos, este hace parte del objeto de negocio y va dirigido al rol cliente, para la venta de este producto especificamos tres servicios que hacen parte del producto, los cuales definimos como la comunicación entre proveedores y usuarios, la facilidad de pago de los servicios a través de la herramienta, y la prestación de servicios locativos por parte de profesionales idóneos todo esto apuntando a la venta de servicios hacia el cliente.

4.1.7 Capa de aplicación

Para tratar esta capa, partimos de la información adquirida en la capa de negocio, ya que de esta obtenemos los principales conceptos que se utilizaran en esta capa de aplicación, en la cual pretendemos demostrar las interacciones que se presentan entre la capa de aplicación y la capa de negocio, para una adecuada apropiación de estos términos de se implementaran cuatro diferentes puntos de vista los cuales son: comportamiento de la aplicación, cooperación de la aplicación, estructura de la aplicación y uso de la aplicación [20] .
Un aspecto fundamental de la capa de aplicación es demostrar la forma en la que interactúan los componentes que realizan la comunicación entre ellos, donde aplicamos el concepto de colaboración. Adicional a esto se evidencia las operaciones y eventos que exponen los componentes que interactúan a través del canal lógico mediante el cual se accede a los servicios expuestos.

4.1.7.1. Punto de vista: Comportamiento de aplicación

El punto de vista comportamiento interno de la aplicación expone en su diagrama el comportamiento interno de la aplicación para este caso el producto principal de EMBER S.A.S el cual será FelixApp como aplicación móvil. Se hace útil para diseñar el comportamiento principal de la aplicación y para identificar la funcionalidad principal de las aplicaciones [20] .

Dentro de este punto de vista identificamos los principales servicios que desprenden de la aplicación móvil FelixApp y como sus componentes se relacionan entre sí.

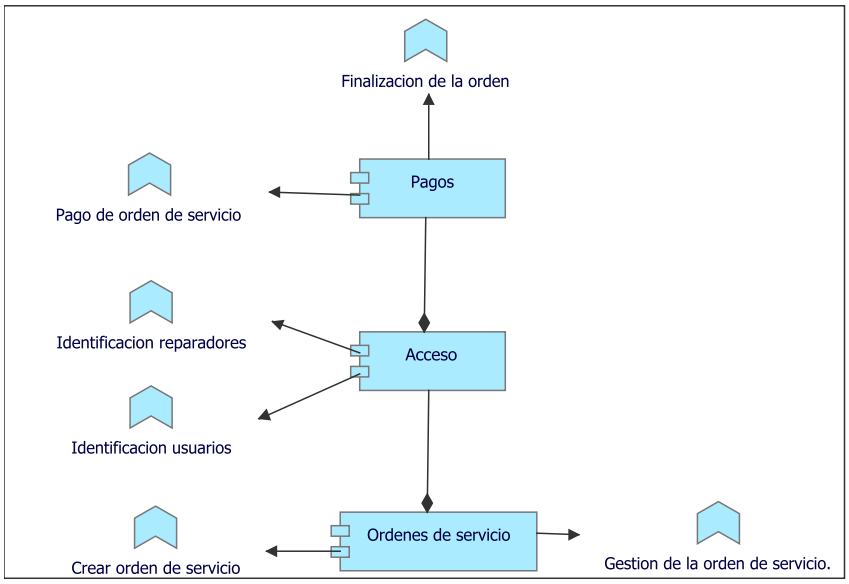


Fig. 4.11: Punto de vista: Comportamiento de la aplicación. Fuente: Los autores

4.1.7.2. Punto de vista: Cooperación de aplicación

EL punto de vista cooperación de la aplicación describe directamente la relación que tienen los componentes de la aplicación en términos de los flujos de la información que radican entre ellos adicional a esto en términos de los flujos de servicios que se ofrecen entre ellos [20]
.
Adicional a esto se evidencia desde este punto de vista el comportamiento de los micro servicios que desprenden de la aplicación y la orquestación entre ellos mostrando la cooperación que existen entre los diferentes componentes de la aplicación en función del proceso de negocio [20] .
Aplicando este concepto para la aplicación móvil FelixApp se definen las relaciones entre los componentes para el proceso de solicitud y atención de servicios locativos a través de esta.

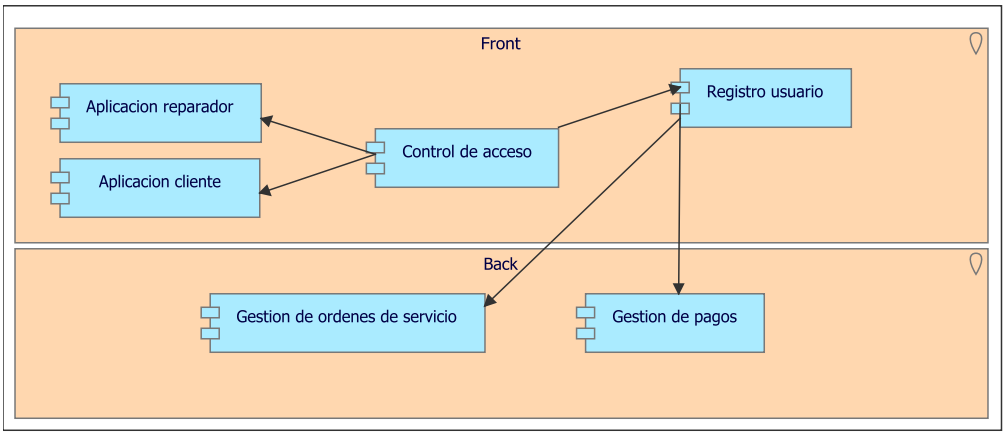


Fig. 4.12: Punto de vista: Cooperación de aplicación. Fuente: Los autores

4.1.7.3. Punto de vista: Estructura de aplicación

Para el desarrollo de la aplicación de servicios FelixApp de la empresa EMBER S.A.S. se plante al caso para el punto de vista estructura de la aplicación con los siguientes elementos que interactúan en dicho punto de vista.

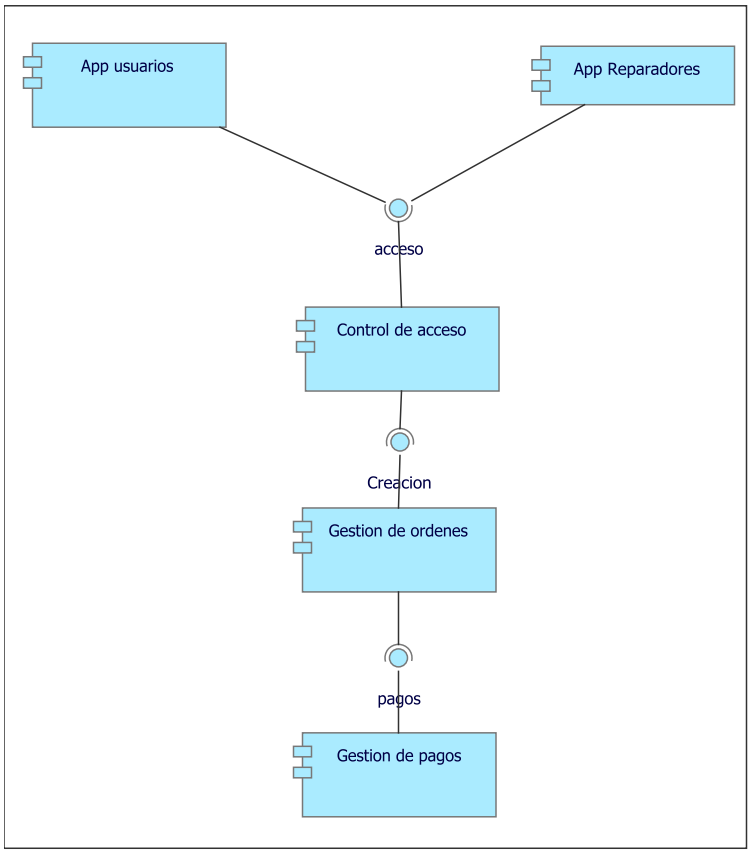


Fig. 4.13: Punto de vista: Estructura de aplicación. Fuente: Los autores

4.1.7.4. Punto de vista: Uso de Aplicación

A continuación, se describen los elementos que interactúan para el punto de vista uso d la aplicación para la empresa EMBER S.A.S., en el cual se identifica un proceso base para el uso de la aplicación principal FelixApp en la solicitud y atención de órdenes de servicio en reparaciones locativas tal como lo trata el presente proyecto de investigación.

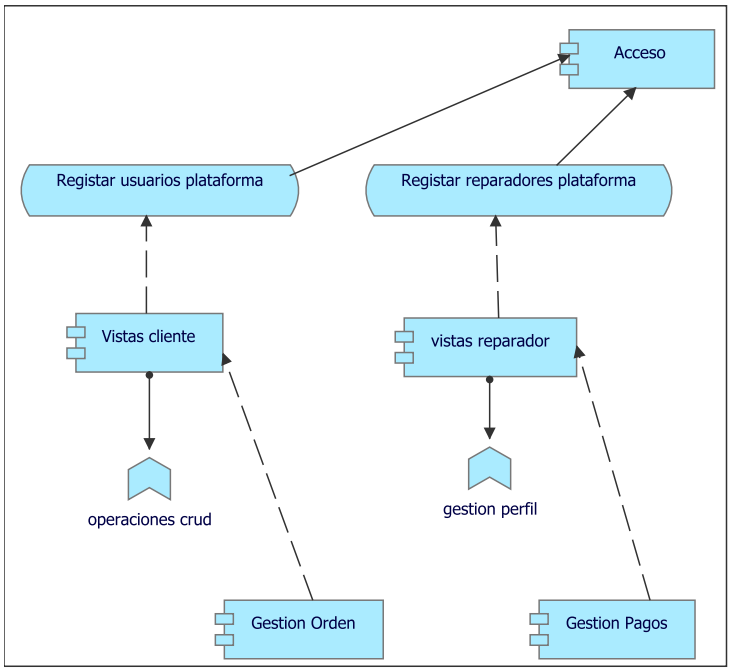


Fig. 4.14: Punto de vista: Uso de Aplicación. Fuente: Los autores

4.1.8 Capa de tecnologia

Teniendo en cuenta la arquitectura de la aplicación y sabiendo que esta presenta un entorno estable para su despliegue, presentamos la capa de tecnología de la organización donde implementamos los conceptos de Nodo y Dispositivo, con los cuales identificamos las diferentes entidades físicas o recursos de computación y de infraestructura que requiere el producto principal de la empresa EMBER S.A.S para este caso la aplicación móvil FelixApp.

Adicional a esto pretendemos evidenciar en esta capa los diferentes elementos de software y de hardware que harán parte de la implementación de este proyecto, para esto es indispensable tener en cuenta los conceptos mostrados en la capa de aplicación y en la capa de negocio, con esto se nos facilitara demostrar en la capara de tecnología como interactúan los diferentes actores y elementos dentro del proceso. Para conceptualizar aún más esta capa trabajaremos en 6 puntos de vista los cuales son tecnología, uso de tecnología, despliegue e implementación, estructura de información, realización del servicio, fisco y capas.

4.1.8.1. Punto de vista: tecnología

A continuación, se describen los elementos que participan en el punto de vista de tecnología para la aplicación principal FelixApp en la empresa EMBER S.A.S.

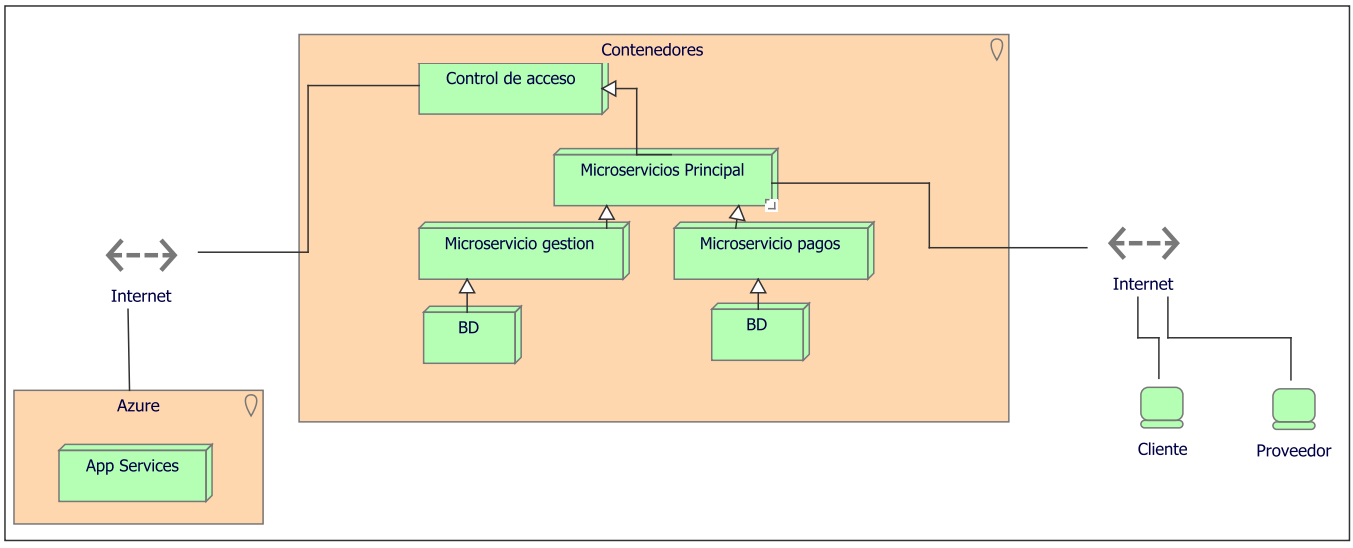


Fig. 4.15: Punto de vista: tecnología: Los autores

- *Servidor de BD: Este componente hace referencia a la persistencia de la aplicación*

donde se almacenará todo aquello correspondiente al proceso de solicitud y atención de servicios locativos a través de la app.

- *Rest API: Este componente hace referencia al desarrollo de servidor o back-end el cual publica los servicios asociados a los procesos que se ejecutan en la aplicación móvil el mismo da la disponibilidad de los servicios y los expone a través de internet.*
- *Internet: componente que se distingue por su comportamiento como canal de transmisión de la información y los servicios entre el Rest Api y las apps móviles.*
- *Cliente: hace referencia al componente de App móvil el cual utilizaran los diferentes actores dentro del proceso de solicitud de servicios.*

4.1.8.2. Punto de vista: Uso de tecnología

A continuación, se describen los diferentes componentes que hacen parte del punto de vista uso de tecnología de la presente capa para la empresa EMBER S.A.S en la implementación de la aplicación FelixApp.

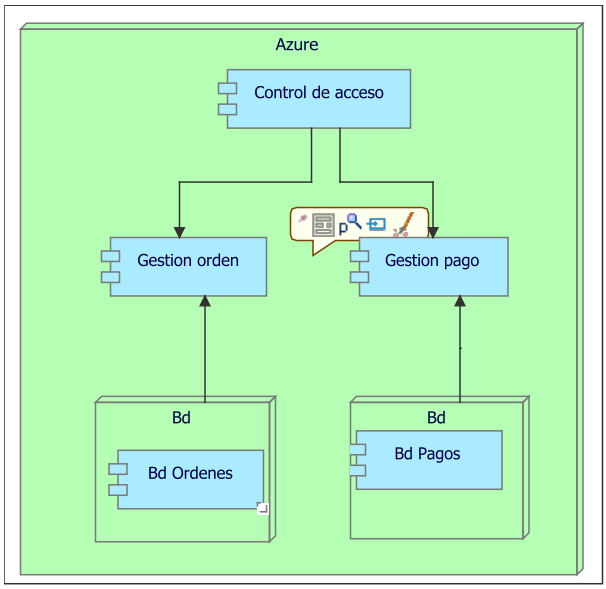


Fig. 4.16: Punto de vista: Uso de tecnología: Los autores

4.1.8.3. Punto de vista: Despliegue e implementación

A continuación, evidenciamos los componentes que hacen parte en el proceso de despliegue de la aplicación FelixApp en la organización EMBER S.A.S. en él se describen claramente el comportamiento de los elementos que intervienen en este proceso de implementación.

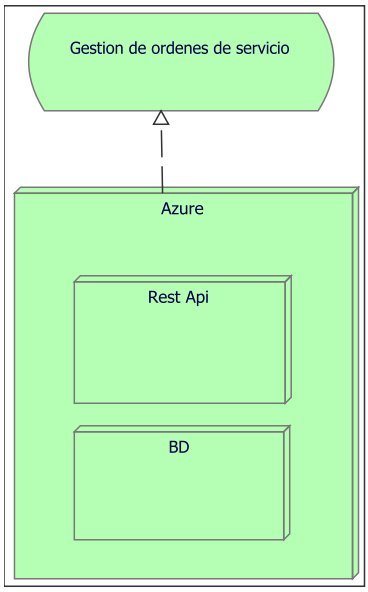


Fig. 4.17: Punto de vista: Despliegue e implementación: Los autores

4.1.8.4. Punto de vista: Estructura de información

A continuación, se evidencian los componentes que hacen parte del punto de vista estructura de la información para la empresa EMBER S.A.S.

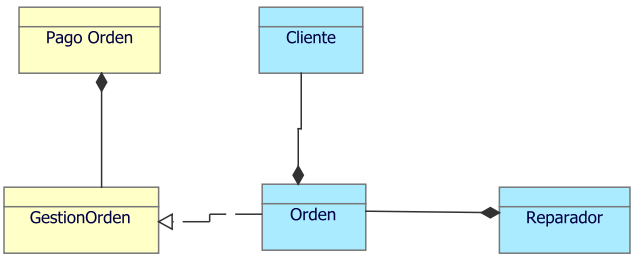


Fig. 4.18: Punto de vista: Estructura de información: Los autores

4.1.8.5. Punto de vista: Realización del servicio

A continuación, se evidencian los componentes que hacen parte del punto de vista realización del servicio para la empresa EMBER S.A.S.

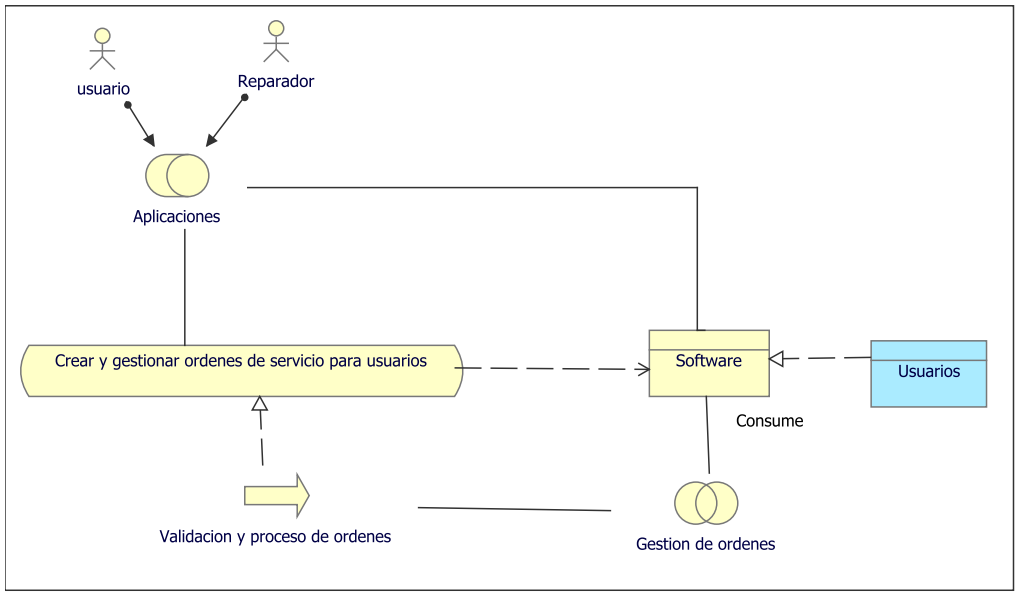


Fig. 4.19: Punto de vista: Realización del servicio: Los autores

Capítulo 5

BACK-END Y FRONT-END

5.1 Back - End

5.1.0.1. Net Core

.Net core es la evolución del framework anterior de microsoft .Net Framework. trae consigo nuevas ventajas ser un framework de código abierto y multiplataforma permitiendo crear aplicaciones pensadas en infraestructura en la nube y on-premises.

Este framework nace como solución a muchos problemas de su antecesor, el principal de ellos era que .Net framework estaba pensado como un framework monolítico obligando a cualquier aplicación que se pensara desarrollar sobre este framework a cargar todo completamente así no se usará causando aplicaciones excesivamente sobrecargadas [21].

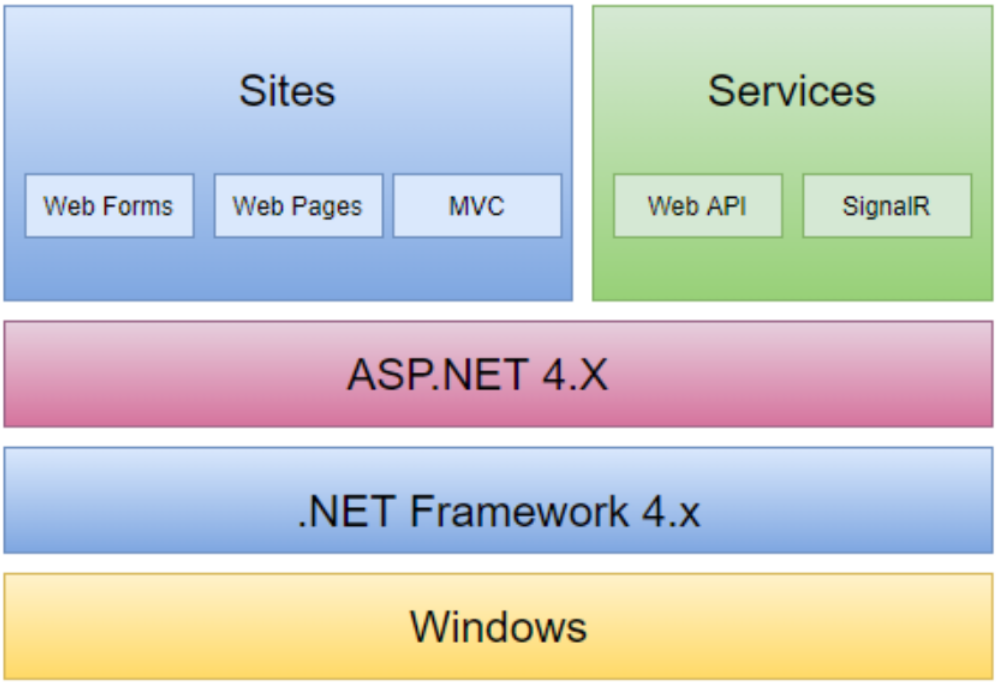


Fig. 5.1: Pilar de tecnologias .Net Framework 4.x [1]

Con la construcción de .Net Core se pensó en un framework enfocado a lo necesario, removiendo librerías como la System.Web y creando paquetes para encapsular funcionalidad y ser utilizados cuando estos sean necesarios [22].

En la siguiente gráfica podemos ver la comparativa de .Net Core y .Net Framework.

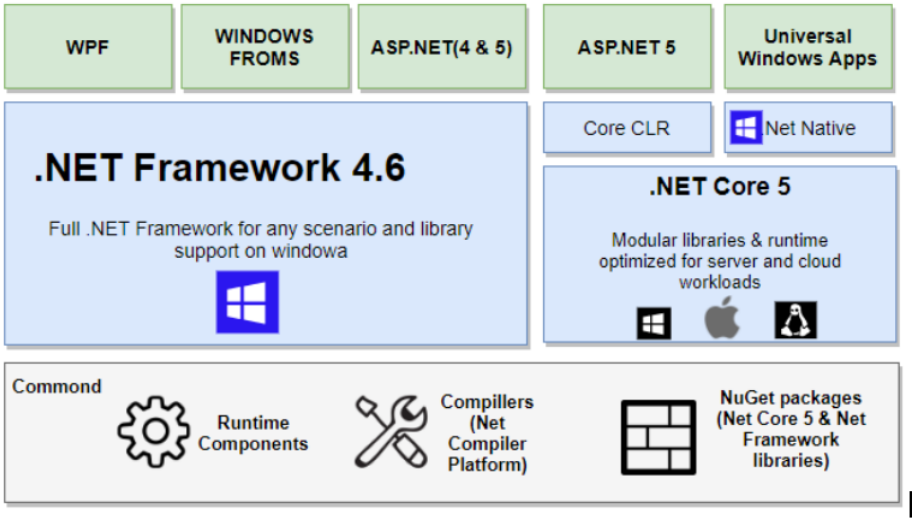


Fig. 5.2: .Net Framework vs .Net Core [1]

Como podemos observar para aplicaciones de tipo PWA .Net core es una opción mucho mejor debido a las ventajas presentadas . Dentro del desarrollo de félix app .net core fue seleccionado como framework para la construcción de apis debido a las facilidades presentadas anteriormente, es importante resaltar las facilidades brindadas por el framework para aplicar buenas prácticas de desarrollo de software (principios solid, tdd, manejo y control de métricas de desarrollo etc).

Otra gran ventaja presentada por este framework es una facil integracion con la plataforma azure para lograr interoperabilidad en los diferentes componentes de infraestructura, permitiendo de esta manera realizar despliegues a ambientes productivos más fácilmente y con un costo operacional más bajo ya que se integra totalmente con la suite de productos microsoft.

.Net Core es uno de los framework con mayor soporte por parte de microsoft como se puede evidenciar en su cantidad de releases en cuanto al tiempo de vida que tiene, desde su versión inicial lanzada en 2016 (.net core 1.0) a su versión actual se han realizado 7 actualizacion mayores al framework siendo su versión actual .net core 5.0 a la cual se le brinda soporte continuo por parte de microsoft y se realizan despliegues menores solucionando fallos detectados a cada versión.

5.1.0.2. Arquitectura de micro servicios

Las arquitecturas de microservicios tienen como fin desarrollar una aplicación pensándola como pequeños servicios funcionando de manera autónoma y comunicándose entre sí, cambiando el enfoque de aplicaciones monolíticas, para entender esto tomaremos como ejemplo una aplicación de ecommerce con funcionalidad de registro, compra y envío, en la siguiente gráfica veremos la distribución de esta aplicación con un enfoque monolítico.

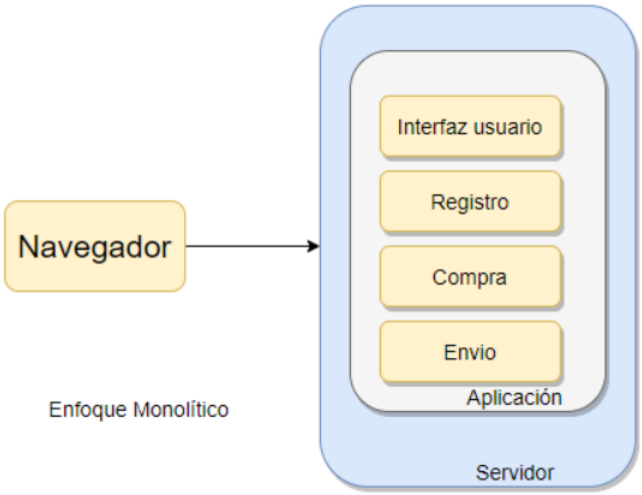


Fig. 5.3: Enfoque monolitico. Fuente: Los autores.

La gráfica nos muestra los módulos utilizados por la aplicación donde podemos ver que todos están encapsulados sobre una misma aplicación. esto trae inconvenientes pues si algún modulo falla la aplicacion entera fallara, la misma aplicación en una arquitectura de microservicios tendría desplegado cada una de las funcionalidades de la aplicación como servicios independientes y aislados.

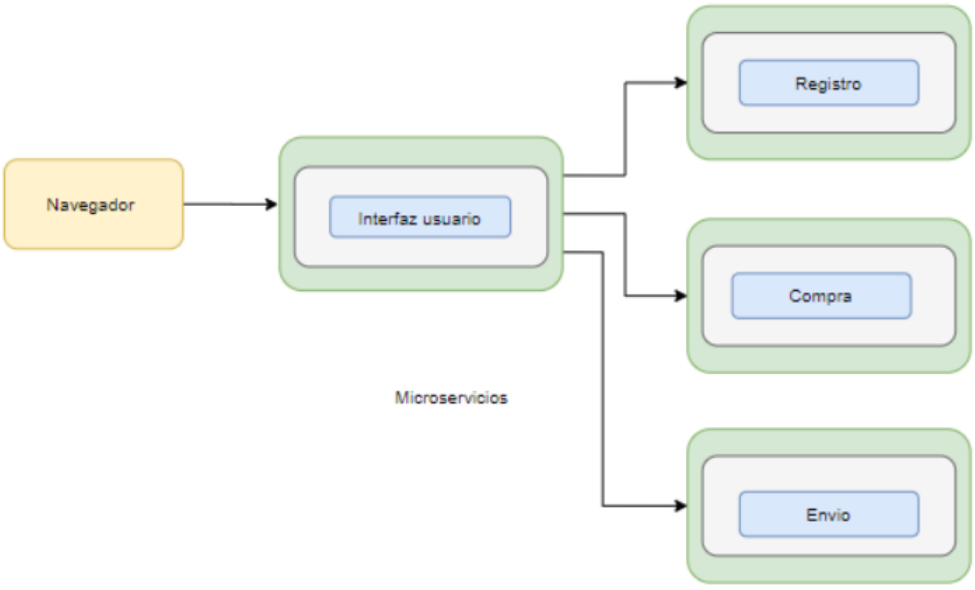


Fig. 5.4: Diagrama microservicios. Fuente: Los autores.

Como podemos observar las funcionalidades se encuentran aisladas cada una permitiendo generar despliegues individuales y parcializados acorde a cada una de las funcionalidades.

Para solucionar la arquitectura tecnológica que se propone: consta de un enfoque totalmente aplicado a micro servicios, donde se plantea como proveedor de infraestructura azure, en el diagrama a continuación se evidencia la arquitectura propuesta y cómo interactúa cada uno de los objetos del pod entre sí.

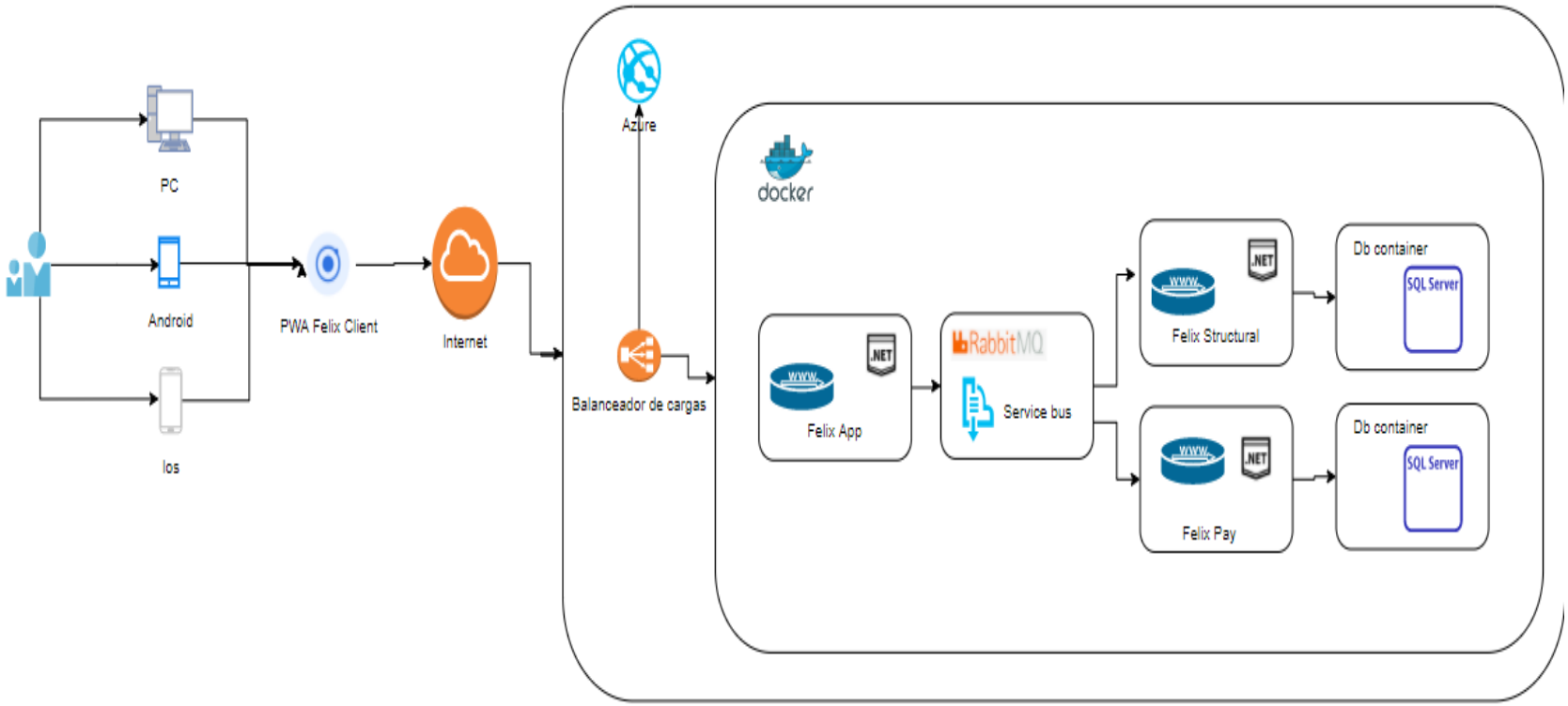


Fig. 5.5: Arquitectura del Back-end. Fuente: Los autores.

Como se observa se propone una aplicación contenida en azure donde se maneja un balanceador de cargas que permita hacer un escalamiento horizontal y vertical de recursos permitiendo así sostener el flujo continuo de llamados a la aplicación.

Se plantea un microservicio maestro quien es el encargado de servir como pasarela de coneccion para los diferentes front que se conectaran (web, android o IOS), este se encargará de redirigir cada petición a el microservicio encargado (órdenes o pagos) quienes a su vez mantienen insolación entre ellos manteniendo cada uno bases de datos independientes. en cuanto a respuesta entre microservicios se propone una solución basada en tipificación de peticiones (comandos y consultas) donde cada uno de los posibles tipos se manejan por sincronía o asincronía según sea el caso (comandos sincrónicos y consultas sincronicas).

Sincronia y asincroniza:

Para solventar las solicitudes a cada uno de los microservicios expuestos la tipificación de las peticiones indica que las consultas deben resolverse sincrónicamente por medio de respuestas http entre peticiones permitiendo tener respuestas sincrónicas a cada petición y respuesta inmediatas a cada solicitud. Un ejemplo de una consulta sincrónica puede evidenciarse en servicios como obtener servicios, obtener órdenes, obtener perfil, entre otras donde no es necesario realizar ninguna persistencia de datos u operación compleja que se salga del ámbito de consulta. Este tipo de comportamiento implica que las respuestas a servicios de este tipo necesitan respuestas en tiempo real.

Por otra para todos los comandos deben realizarse por medio de peticiones asíncronas ya que no necesitan una respuesta inmediata para continuar con el funcionamiento de la aplicación, es aquí donde el uso de Rabbit MQ como service bus solventa la comunicación entre microservicios eliminado la necesidad de llamados HTTP y siendo reemplazados estos por el uso de colas de llamadas, las respuestas de este tipo de servicios deben ser emitidas a los servicios interesados por medio envío de notificadores (uso e implementación de patrón suscriptor) permitiendo así a los servicios interesados estar a la escucha de los eventos de respuesta y notificar a los clientes conectados cada una de las respuesta de los servicios asincrónicos, un ejemplo de este tipo de servicios se puede evidenciar en la creación de órdenes, creación de usuarios, creación de servicios entre otras.

Inyección de dependencias:

.net core brinda la posibilidad de implementar la inyección de dependencias de forma nativa y sin necesidad de extensiones de terceros, la inyección de dependencias se establece como un patrón de diseño de software que permite crear componentes desacoplables para obtener componentes de fácil gestión de cambios en futuros cercanos, fácil implementación de pruebas unitarias, migración de clases y modificación de comportamiento sin afectar los componentes externos.

La inyección de dependencias también permite una mecanismo para la construcción de gráficos de dependencias los cuales permiten crear clases que dependen de las interfaces permitiendo isolar los comportamientos y desarrollar los externos sin depender de implementaciones, esto hace fuerte referencia a los principios SOLID especialmente a la inyección como principio y a el principio de responsabilidad única, además de cumplir con buenas prácticas de desarrollo de software como el principio de jerarquía perfecta.

La Inyección de Dependencias es un patrón de diseño de software que nos permite desarrollar componentes acoplados libremente o desacoplados para obtener como resultado la fácil gestión de cambios a futuro, implementación fácil de pruebas unitarias, factoría para emitir instancias de clases, prevención de fugas de memoria, entre otros.

DI (inyección de dependencias) proporciona un mecanismo para la construcción de gráficos de dependencia independientes de las definiciones de clase, aplicando esta técnica a nuestras clases van a depender de interfaces y no de implementaciones, esto se relaciona directamente con el primero de los 5 principios para desarrollar software de calidad, haciendo referencia a SOLID y el principio de responsabilidad única [21].

5.1.0.3. Persistencia de datos

La persistencia de datos se torna en un factor primordial para Felix app como aplicación. Para implementar almacenamiento de datos se propone un motor de base de datos SQL Server como servicio desplegado dentro de azure para cada uno de los micro servicios existentes.

Entity Framework Core: Entity framework core es un ORM de fácil uso y open source que permite la manipulación de los datos y la creación de modelos de base de datos a partir de entidades de negocio, su implementación presenta 2 paradigmas (code first or database first) cada uno de estos se especializa en diferentes necesidades de negocio, para la implementación de felix app se propone un modelo intermedio donde no se realizará totalmente una implementación code first ya que según lo expuesto en el capítulo 3 el modelo de base de datos se define inicialmente y guiará la lógica del negocio pero si se utilizaran las bondades de este paradigma para el manejo de entidades y conexiones a bases de datos.

SQL Server: Para la creación y gestión de la base de datos se realiza un proyecto especia-

lizado de base de datos desde donde se realizan los despliegues y se hace la implementación del modelo definido en el capítulo 3.

Este tipo de proyectos brindan la facilidad de generar archivos bakpac los cuales por medio de pipelines de despliegue permiten automatizar los releases de modificaciones y actualización a la estructura de la base de datos [10].

5.1.0.4. Despliegue de aplicación

El despliegue continuo en este tipo de aplicaciones se torna en un factor diferenciador puesto que la salida de nuevas funcionalidades se vuelve un factor diferenciador contra competidores.

Para solventar esta necesidad se propone el uso de Azure Devops como repositorio de código fuente y gestor para integración y despliegues. El despliegue de la aplicación se propone hacer dentro de un app service de azure en el cual se desplegarán contenedores de docker que mantengan las imágenes de la aplicación por medio de azure container registry.

La integración de cambios al aplicativo se realiza por medio de integraciones parciales a partir de pull request donde el equipo de desarrollo se encarga de hacer revisiones pares a los cambios que se desean integrar, cuando un cambio es considerado correcto se aprueba el pull request y se integra a la rama principal desde donde posterior al merge del nuevo cambio corren las pruebas unitarias para validar el nuevo código, y se genera una nueva compilación que si es satisfactoria se publica en azure container registry.

Posterior a que la nueva imagen se encuentra en azure container registry se genera un nuevo despliegue por medio de el lanzamiento de un pipeline que toma el último artefacto actualizado (generado en el paso anterior) y se publica dentro del app service para que finalmente quede disponible al uso productivo.

Este tipo de estrategias permite que los releases de nuevas funcionalidades se generen de forma automatizada e inyectando la mejor cantidad de errores posibles para los usuarios garantizando un código de calidad y limpio [23].

5.1.0.5. Docker

El desarrollo de aplicaciones hoy en día requiere muchos más que escribir código, se requieren diferentes tipos de marcos, múltiples lenguajes de programación, diferentes tipos de arquitecturas e interfaces discontinuas entre herramientas para cada ciclo de vida de una aplicación y esto genera una enorme complejidad a la hora de controlar la magnitud de las diferentes herramientas implementadas. Es aquí donde entra Docker a trabajar como herramienta que simplifica y agiliza este tipo de trabajos, cumpliendo su función de generar contenedores de trabajo que nos permite aislar la aplicación y su entorno de desarrollo permitiéndonos ejecutar la aplicación desde cualquier máquina en su proceso de desarrollo y mantenimiento [2].

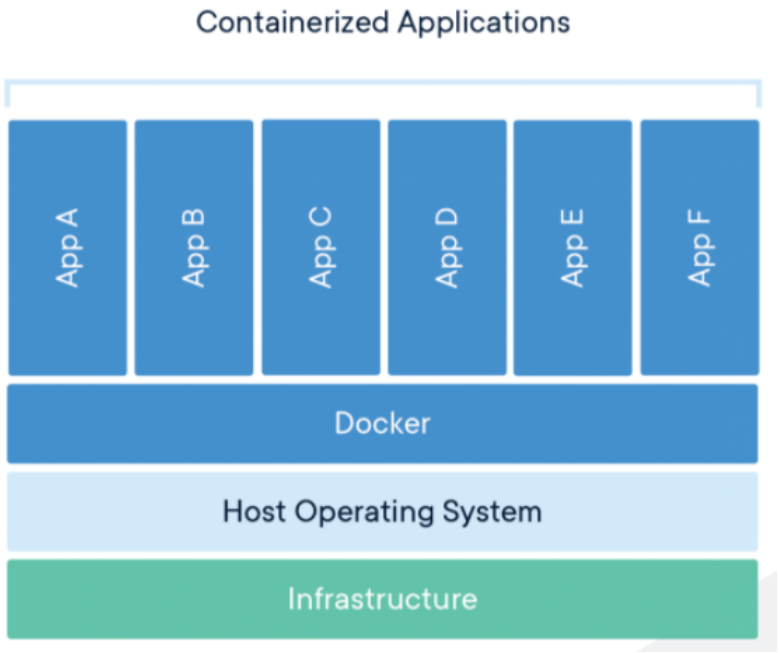


Fig. 5.6: Contenerización de aplicaciones [2]

5.2 Front - End

La unión internacional de telecomunicaciones emitió en 2015 que aproximadamente 3.200 millones de personas o casi la mitad de la población mundial estarán en línea para fines de este año, esto se presentará en diferentes tipos de dispositivos, a diferente velocidades de conexión y condiciones siempre cambiantes, como desarrolladores se hace difícil realizar aplicaciones que logren este alcance y que satisfagan las diferentes necesidades que se presentan en el mercado cambiante de hoy, y es aquí donde entran las aplicaciones web progresivas (PWA), las cuales le brindan a los desarrolladores la capacidad de crear sitios web más dinámicas, resistentes y adaptables a cualquier tipo acceso desde cualquier parte del mundo [24].

Podemos definir que una PWA es la que tiene capacidad de funcionar con características como ejecución fuera de línea, con la capacidad de tener tiempos de carga casi instantáneos y al mismo tiempo ser seguro a redes poco confiables, la mayoría de los navegadores web cuenta con estas características y solo hace falta aprovecharlas en la realización de estos desarrollos. otras características que contienen las PWA son ser sensibles, tener conectividad independiente, estar siempre actualizadas, ser instalables, enlazables y reconectables y sobretodo que están desarrolladas con tecnologías que se conocen comúnmente como son HTML. CSS y JavaScript.

Más que un conjunto de características las PWA es una evolución de cómo crear más y mejores sitios web adicionando a esto que es una colección de mejores prácticas para el diseño y desarrollo de estas, aplicando las características antes mencionadas y utilizando técnicas de diseño adaptable las PWA funcionan tanto en escritorio como en móviles utilizando una base única de código funcional en diferentes plataformas [25].

5.2.0.1. Arquitectura de aplicaciones PWA

Dadas las características de las PWA estas manejan una arquitectura o patrones:

- App Shell que básicamente es HTML, CSS y Javascript, que básicamente es el esqueleto que necesita para cargar la interfaz de usuario, esto permite que este tipo de aplicaciones carguen de forma casi instantánea y fiable casi igual a las aplicaciones nativas [24].
- PRPL, el cual se está implementa hoy en día diferentes frameworks como Angularjs, Ionic, View y su objetivo principal está en la optimización de recursos en cualquier dispositivo.

- Push: Empaquetador de recursos que sean indispensables.
- Render o el renderizado de las primeras vistas lo antes posible.
- Pre-Cache de aquellos recursos que se suponen son los más utilizados al inicial la aplicación.
- Lazy-load o carga de recursos bajo demanda y según el usuario vaya necesitando.

5.2.0.2. HTTPS

Otra característica fundamental de las PWA como Felix app es la protección de los datos y es acá donde entra la comunicación con las API's actuales que para nuestro caso propuesto se encuentra construido con .net core como framework base, aquí se realiza una implementación que no permite una conexión con redes inseguras y sin protocolo de autenticación para que la información del usuario no se vea en ningún momento comprometida.

por otra parte el uso de certificados ssl que permitan verificar el no repudio de las solicitude y respuestas entre backend y front-end son de vital importancia en este tipo de aplicaciones ya que si no se garantiza la validez de los emisores ataques del tipo hombre en el medio podría permitir a ciber-atacantes obtener información altamente sensible de los usuarios.

El concepto de un ataque de hombre en el medio es muy sencillo. Además, no se limita únicamente al ámbito de la seguridad informática o el mundo online. Este método sólo necesita que el atacante se sitúe entre las dos partes que intentan comunicarse; interceptando los mensajes enviados e imitando al menos a una de ellas. Por ejemplo, en el mundo offline, se crearían facturas falsas, enviándolas al correo de la víctima en interceptando los cheques de pago de dichos recibos. En el mundo online, un ataque MiTM es mucho más complejo, pero la idea es la misma. El atacante se sitúa entre el objetivo y la fuente; pasando totalmente desapercibido para poder alcanzar con éxito la meta [26].

5.2.0.3. Ventajas

- Interacciones similares a las apps nativas: Al adaptar las diferentes arquitecturas se pueden implementar experiencias con navegadores y experiencias similares a las apps nativas, complementadas con soporte sin conexión.
- Uso económico de datos: Está especialmente diseñadas para hacer un consumo mínimo de datos.

- Rendimiento confiable y rápido: Los recursos estáticos como los son el CSS el HTML y Javascript se almacenan en caché en la primera visita y estas se mantienen precargadas en caché [27]

5.2.0.4. IONIC Framework

Ionic es una herramienta poderosa para el desarrollo de aplicaciones web híbridas, es de las más reconocidas en el área de desarrollo y de las más estables y utilizadas, aplica con este el concepto fundamental de las PWA, está basado en componentes y se pueden desarrollar nuevos usando su generador de código abierto, permitiendo la integración con Angular, React y Vue, su integración con Cordova ayuda a la integración con las herramientas de diferentes dispositivos para hacerla adaptable a los mismos [3]

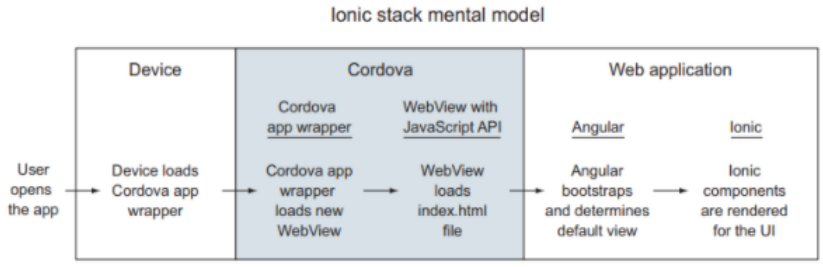


Fig. 5.7: Modelo mental del funcionamiento de IONIC framework. [3]

Sin embargo es importante aclarar que existen diferentes maneras de desarrollar aplicaciones móviles las cuales son aplicaciones nativas, sitios web móviles y aplicaciones híbridas, cada una de esta técnicas tiene sus respectivas ventajas y desventajas como se describen en la figura 7 donde podemos evidenciar la arquitectura del funcionamiento de cada una de las opciones descritas sin embargo en este artículo nos enfocamos en las virtudes de las PWA y las aplicaciones híbridas como ventaja para el desarrollo de aplicaciones móviles.

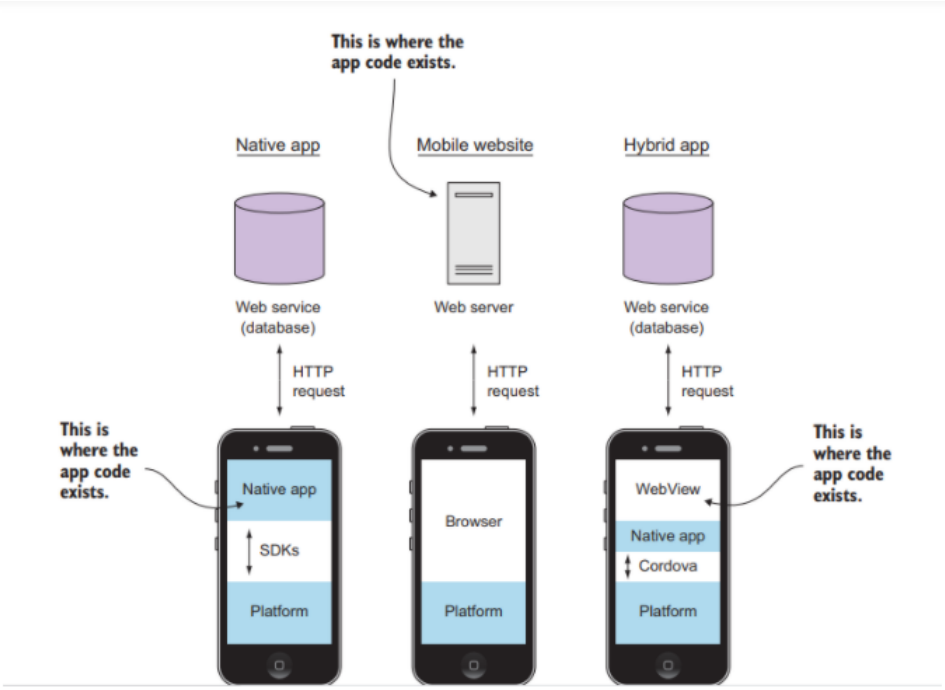


Fig. 5.8: Comparación de arquitecturas de aplicaciones móviles. [3]

Dentro de Felix app ionic se transforma en el core de las aplicaciones front end permitiendo el desarrollo para todas las plataformas necesarias sin necesidad de reescribir código entre ellas, se despliega en arquitectura orientada a componentes donde cada una de las páginas se visualiza como un componente independiente en el cual se pueden también agregar componentes externos y [3].

Como lenguaje intermedio se eligió angular 10 la cual permite la creación de componente reutilizables dentro de la aplicación, esta funcionalidad se utilizó principalmente en el desarrollo componente para geolocalización y menús.

5.2.0.5. Prototipo de aplicación

Teniendo en cuenta los conceptos de aplicaciones PWA e implementando el framework IONIC se desarrollan por parte del Front - end dos aplicaciones que se permiten compilar en apk para su funcionamiento en android y su respectiva compilación para su funcionamiento en dispositivos IOS, FElixApp usuarios y FelixApp reparadores, estas dos aplicaciones permitirán realizar la solicitud y atención de servicios locativos solicitados por los usuarios y atendidos por los proveedores de servicios.

Teniendo en cuenta esto, se implementan en ambas aplicaciones una arquitectura dada por el framework el cual trabaja con tecnología angular, dentro de estas buscamos una organización de carpetas la cual nos permite diferenciar las pages de los componentes implementados en cada vista que utilizan los usuarios, el proceso de solicitud y atención de servicios realizado en IONIC se puede ver en funcionamiento en las siguientes imágenes del prototipo implementado para este proyecto [8].

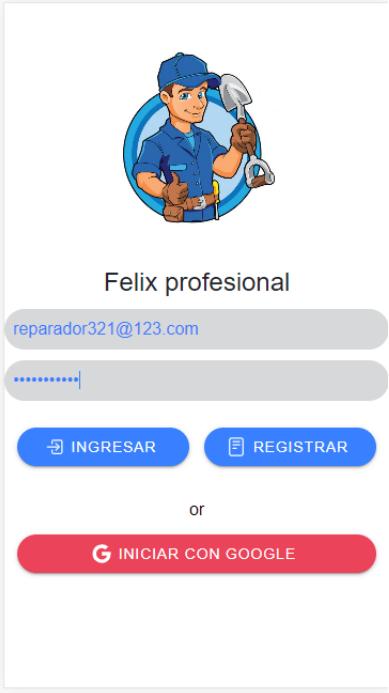


Fig. 5.9: Inicio de sesión Felix app profesionales. Fuente: Los autores.

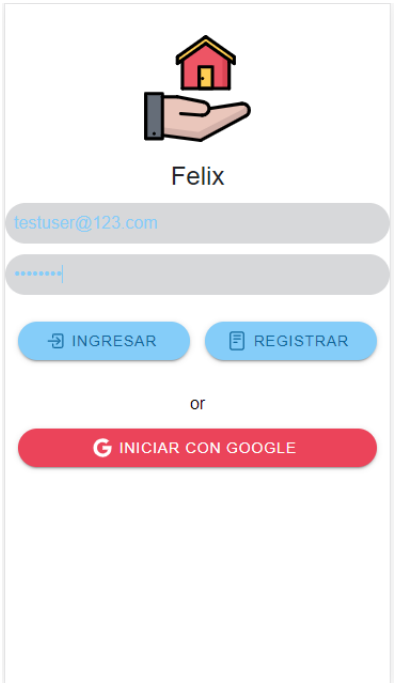


Fig. 5.10: Inicio de sesión Felix app usuarios. Fuente: Los autores.

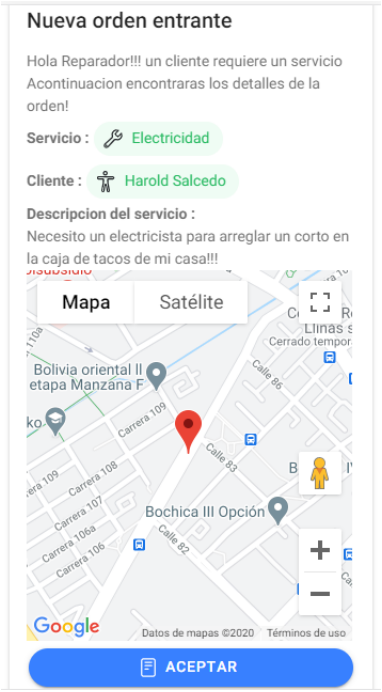


Fig. 5.11: Nueva orden. Fuente: Los autores.

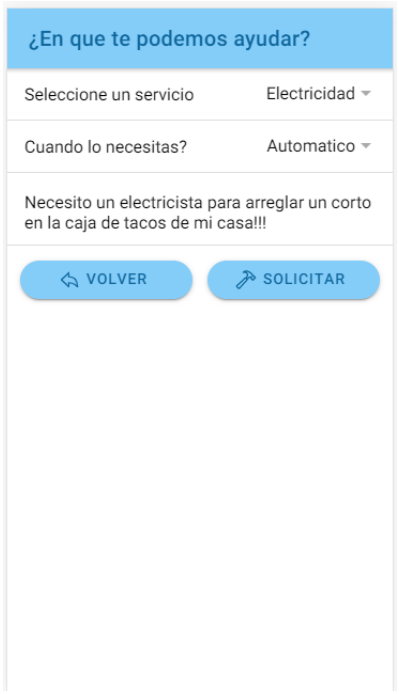


Fig. 5.12: Solicitud servicio. Fuente: Los autores.

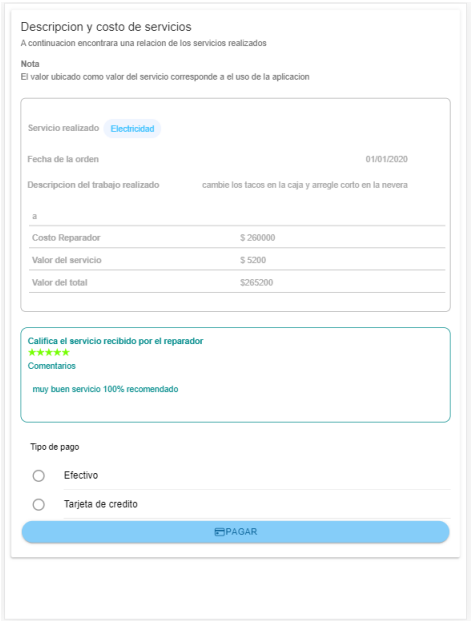


Fig. 5.13: Finalizar orden. Fuente: Los autores.

Las anteriores vistas son sólo algunas dentro de todo el proceso de solicitud y atención de órdenes de servicio solicitadas y atendidas a través de las dos aplicaciones desarrolladas para

este proyecto, el funcionamiento en concreto se puede observar en el demo de demostración del prototipo desarrollado para tal fin.

5.2.0.6. Geolocalización API Google

Georeferenciación o posicionamiento: Permitir la ubicación física de un objeto o individuo por medio de un sistema de coordenadas para en base a esto obtener información específica de terreno, espacio, etc. ejemplos claros que se pueden dar en la vida real son la ubicación de prisioneros por medio de dispositivos GPS, Sistemas de conducción dirigida, sistemas de telemetría.

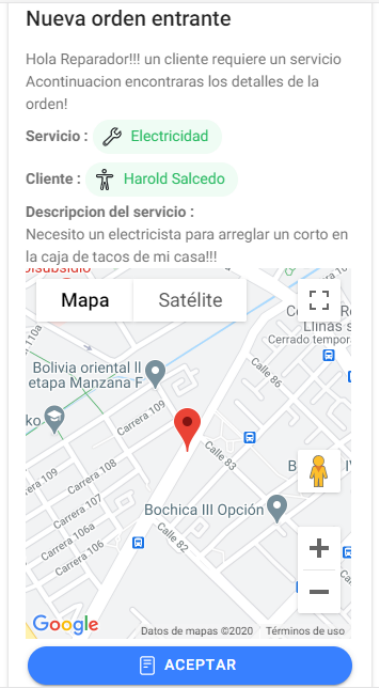


Fig. 5.14: Implementación del API de geolocalización de google. Fuente: Los autores.

Geocodificación: Permite realizar búsquedas entorno a objetos codificados para dar un ejemplo del término se aplica a búsqueda de lugares en un entorno, buscar una gasolinera en mi zona de radio, obtener hoteles en una ciudad, obtener cámaras de control de velocidad en una ruta de tráfico.

Geoetiquetado: Permite agregar información de posicionamiento a objetos del común como por ejemplo fotografías, imágenes, documentos entre otros.

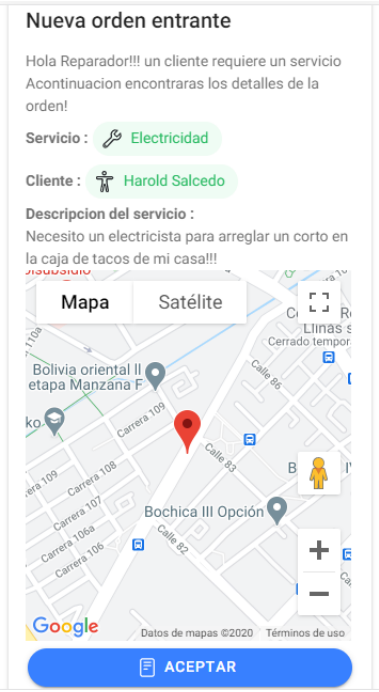


Fig. 5.15: Implementación del API de geolocalización de google. Fuente: Los autores.

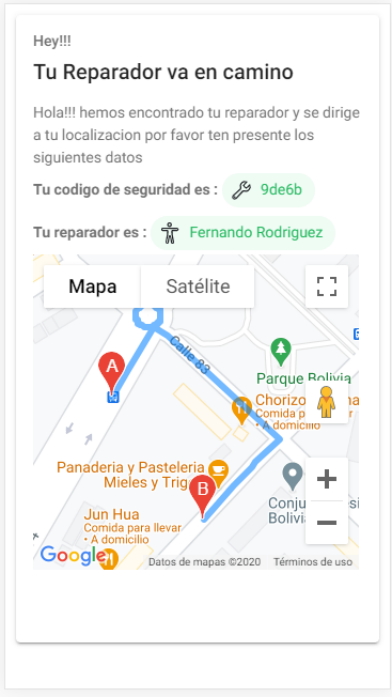


Fig. 5.16: Implementación del API de geolocalización de google. Fuente: Los autores.

Seguridad de la geolocalización: La masificación de este tipo de tecnología y su aplicación

dentro de aplicaciones de acceso común genera problemas asociados a la seguridad de la información (privacidad e información sensible). debido a esto es importante tener conocimiento acerca de los problemas relativos al aseguramiento y privacidad de la información para poder utilizar esta tecnología de manera segura y responsable.

Los datos de geolocalización son generados y recopilados de cualquiera de las dos formas siguientes: en un modo activo conocido como geolocalización basada en el dispositivo del usuario o un modo pasivo conocido como búsqueda en tabla o geolocalización basada en servidores por correlación de datos.

Modo	Método de recolección	Tecnologías involucradas
Activo: Basado en el dispositivo del usuario	• Utiliza el firmware y el software de la computadora o dispositivo inalámbrico del usuario • La ubicación se determina a través del chip del GPS y/o triangulación mediante la información de las torres celulares • Modelo de petición-respuesta	• GPS • GPS asistido (A-GPS) • Wi-Fi - Posicionamiento inalámbrico • 3G/4G • Aplicaciones móviles: iPhone, dispositivos Android, BlackBerry®
Pasivo: Búsqueda de datos basada en servidores	• Involucra el uso de proveedores de servicios de geolocalización de terceros, por ej., Quova®, NatGeo, Bering Media • Basado en la dirección IP que no dependen de una ubicación específica adquirida del dispositivo del usuario o de los identificadores del conjunto de servicios (SSID) para redes inalámbricas • La correlación con las bases de datos de IP o SSID obtenidas de los registros de compras, la información suministrada por los usuarios, los análisis de red de las rutas de rastreo y los nombres almacenados del sistema de nombres de dominio (DNS)	• Ubicación IP: búsqueda Whois, DNS LOC, nombres geográficos en el usuario del nombre del dominio o información de aplicaciones, datos de horarios mediante la inferencia de ping basada en los datos de enrutamiento, por ej., monitorización de Traceroute de las redes del proveedor del servicio de Internet (ISP) • 3G/4G • Wi-Fi - Posicionamiento inalámbrico

Fig. 5.17: Modos geolocalización [4]

Capítulo 6

NORMATIVIDAD LEGAL

6.1 Normatividad legal

En el presente capitulo mostramos un panorama general de como en Colombia se regulan de manera legal el desarrollo de aplicaciones móviles (aquellas que específicamente se instalan nativamente en un dispositivo móvil). Para ello buscamos evidenciar cuales son los aspectos legales fundamentales y de mayor relevancia que deben ser considerados en el presente proyecto.

Aun cuando el tema de las aplicaciones móviles es un concepto a nuevo a nivel nacional, es importante aclarar que no se tienen normas o leyes específicas en este ámbito sin embargo dado el auge, impacto e importancia que este tipo de desarrollos en la sociedad diferentes entidades se han encargado de regular a un nuevo concepto las diferentes aplicaciones que impactan de una u otra manera a los usuarios colombianos.

Dado que el gobierno ha evidenciado que el área de las aplicaciones móviles es un campo muy aprovechable para diferentes emprendimientos se evidencia en diferentes documentos que brindan un conglomerado de reglas claras para su implementación es esta área y lo que buscamos es mostrar las principales de tal forma que sea de conocimiento mutuo y aplicables este proyecto a los derechos y deberes de los diferentes actores que intervienen en este proceso [28].

6.2 Normativa de protección de datos personales

6.2.1 Consideraciones básicas del tratamiento de datos personales

Para Colombia rige la Ley Estatutaria 1581 de 2012, para los temas y disposiciones generales de protección de datos personales.

La Ley de Protección de Datos abarca el derecho que tienen los ciudadanos a conocer, actualizar y rectificar información recopilada sobre ellos en bases de datos o archivos registrados por entidades públicas o privadas. Es necesario recalcar que su cobertura se limita al estado colombiano o territorios extranjeros siempre y cuando se posean acuerdos o tratados de carácter internacional que dictaminen su validez; sin embargo, no aplicará principalmente en: bases de datos o archivos usados en un ámbito personal o doméstico, bases de datos con información periodística y bases de datos para asuntos de seguridad y defensa [29].

6.3 Normatividad de comercio electrónico

En este marco de referencia actúa la Ley 527/99 la cual interactúa en el ámbito del comercio electrónico, sin embargo, para entender correctamente esta Ley hay que hacer referencia a dos conceptos claves los cuales son mensaje de datos y comercio electrónico, los cuales se definen a continuación:

Mensaje de datos, es toda información tratada por medios electrónicos, contempla labores como el envío, recepción, transmisión, almacenamiento, etc. Al ser por medios electrónicos abarca las comunicaciones del tipo correo electrónico, ópticos, telefax y todo tipo de comunicaciones que puedan relacionarse por su naturaleza electrónica [30].

Comercio electrónico, es toda relación comercial, contractual o no, donde se evidencie algún tipo de suministro o intercambio de bienes y servicios a partir del empleo de mensajes de datos[30].

6.3.1 Consideraciones a tener en cuenta en la comunicación de mensajes de datos

En las comunicaciones que se empleen mensajes de datos es necesario conocer que:

- Salvo que previamente se acuerde entre las partes, un contrato puede ser celebrado empleando un mensaje de datos [30].
- Un mensaje de datos tendrá reconocimiento jurídico, por tanto, la misma validez y fuerza obligatoria que documentos físicos [30].
- Se entenderá que el mensaje de datos proviene del iniciador si ha sido enviado por el iniciador, alguna persona facultada para actuar en su nombre o inclusive si se genera mediante un sistema automático que opere en su nombre [30].
- El destinatario del mensaje puede actuar consecuentemente, según el acuerdo pactado siempre y cuando el mensaje recibido provenga del iniciador. Habiendo concordancia entre el mensaje enviado y el recibido [30].
- Acuse de recibido, si entre iniciador y destinatario se convino emplear acuses de recibo al enviar o antes de enviar el mensaje, pero no se definió por medio de cual método hacerlo, es posible hacerlo por medio de comunicación automática o no del destinatario o cualquier acto que el destinatario pueda emplear para hacer conocer que ha recibido el mensaje. En consecuencia, si no se recibe acuse, ese mensaje se considerará como no enviado [30].

De esta manera evidenciamos los ítems legales que corresponden al prototipo de software que desprende de esta investigación para la comunicación asertiva y prestación de servicios a través de la misma entre usuarios y proveedores de servicios de arreglos locativos.

Parte III

CIERRE Y CONCLUSIONES

Capítulo 7

CIERRE Y CONCLUSIONES

7.1 Conclusiones

A continuación, describimos las deducciones obtenidas a partir de los objetivos planteados en el proyecto de investigación:

- Durante el desarrollo del proyecto de investigación se logro dar alcance al objetivo principal del proyecto, dado que el diseño y construcción del prototipo de aplicación móvil propuesto para la comunicación entre usuarios y proveedores de servicios inherentes a obras civiles, permitió hacer las pruebas necesarias de funcionalidad al momento de realizar el procesos de comunicación entre los actores, avanzando así a realizar un panorama de solicitud y atención de servicios inherentes a obras civiles a través de las aplicaciones generadas para tal fin.

- Concerniente en el desarrollo del prototipo, en los objetivos específicos de este proyecto de investigación, fue posible cumplir con los mismos, dado que la herramienta en prototipo cuenta con dos estructuras denominadas comúnmente como back-end y front-end, utilizando las mejores practicas de desarrollo y arquitectura de software, se logra implementar los diferentes servicios a través de un API RestFull completamente funcional y enfocada a una estructura de microservicios , la cual da un performance de calidad a la estructura del back-end propuesta en tecnología C y construyéndolo con el framework .Net Core

- Por otra parte, las características y funcionalidad del framework IONIC para la creación de aplicaciones PWA, nos permite implementar dos IU, una para usuarios denominada en este proyecto como Felix Usuarios y otra denominada como Felix Proveedores, la cual guarda las características y funcionalidades básicas para la implementación de las aplicaciones móviles, funcionales y adaptativas a diferentes sistemas operativos como lo son Android y IOS.

- Teniendo en cuenta el diseño de bases de datos, encontramos que al tener claro y preciso la lógica del contexto del negocio, realizamos la construcción segura, funcional y pertinente de la base de datos, dado que los modelos tales como conceptual asegurado, notacional de conjuntos y modelo relaciona nos permiten generar las respectivas clases que intervendrán en este sistema, se evidencian como entidades o clases dentro del proyecto del back-end en su capa de persistencia o también denominado para este caso capa de base de datos.

- Mediante el análisis realizado a la estructura de la empresa Ember S.A.S se determino el contexto y el modelo de la cadena lógica del negocio del sistema, esto fue posible

gracias a la caracterización de las dependencias funcionales, permitiendo alienar el diseño y construcción tanto de la base de datos como del prototipo del producto de investigación desarrollado para este proyecto.

- Con el desarrollo de este proyecto de investigación fue posible comprobar que la comunicación entre usuarios y proveedores de servicios locativos se puede realizar de manera mas asertiva, controlada y eficaz a través del prototipo construido, sin embargo, es necesario realizar segundas validaciones en un ambiente productivo que permita evidenciar a gran escala el objetivo de este proyecto, permitiendo garantizar una mayor confiabilidad al momento de realizar la comunicación entre los actores de este proceso.

7.2 Trabajos futuros

Con el desarrollo de este proyecto de investigación es posible generar en una escala mayor un proyecto de emprendimiento sostenible, que permita la generación de empleos a través de los servicios inherentes a obras civiles prestados u ofrecidos a través de este medio hacia los usuarios o clientes solicitantes, permitiendo así una comunicación rápida, efectiva y eficaz a entre los actores que proponemos interactúen en este canal de comunicación.

Con el apoyo de la inteligencia de negocio y el análisis de datos que se permita realizar de este proyecto a gran escala, poder identificar y crear nuevos módulos de servicios que se permitan materializar a través de este canal de comunicación, creando así propuestas de valor que a partir del análisis de datos obtenidos de un proyecto implementado a gran escala.

Bibliografía

[1] Academy. Pila de tecnologias net framework 4.x. https://utech.academy/blog/net-framework-vs-net-core/, September 2019. Accessed: 2020-10-24.

[2] docker.com. Docker y contenedores, que es docker? https://www.docker.com/resources/what-container, September 2020. Accessed: 2020-10-24.

[3] Jeremy Wilken. *Ionic in Action, HYBRID MOBILE APPS WITH IONIC AND ANGULARJS*. United States of America: Manning Publications., 2017.

[4] unai zorrilla. Modos de geolocalizacion. https://www.evaluandosoftware.com/la-geolocalizacion-funciona/, September 2020. Accessed: 2020-10-21.

[5] Camara de comercio de bogota. Mercado laboral. https://www.ccb.org.co/observatorio/Economia/Economia-dinamica-incluyente-e-innovadora/Mercado-laboral/. Accessed: Nov 19, 2020,.

[6] Universidad Nacional de colombia. ¿cuáles son los problemas de las construcciones en colombia? http://unradio.unal.edu.co/nc/detalle/cat/observatorio-de-gobierno-urbano/article/cuales-son-los-problemas-de-las-construcciones-en-colombia.html/. Accessed: oct 19, 2020,.

[7] W. Dev. Progressive web apps. https://web.dev/progressive-web-apps/. Accessed: Jun 20, 2020,.

[8] N.D. Hosting an angular progressive web application (pwa) with azure — premier developer. https://devblogs.microsoft.com/premier-developer/hosting-an-angular-progressive-web-application-pwa-with-azure/. Accessed: May 24, 2020,.

[9] Fu Cheng. *Build Mobile Apps with Ionic 4 and Firebase: Hybrid Mobile App Development*. Auckland, New Zealand: Apress., 2018.

[10] J. M. Aguilar. ¿por qué asp.net core? https://www.campusmvp.es/recursos/post/Por-que-ASPNET-Core.aspx. Accessed: Apr 19, 2020,.

[11] Bhatia J.S. Verma, Pankaj. Design and development of gps-gsm based tracking system with google map based monitoring. (0706.2788), June 2013. Accessed: 2018-11-04. Vol.3, No.3, June 2013,pp-33 -40.

[12] Apple inc. ios 14. https://www.apple.com/co/ios/ios-14/. Accessed: Nov 19, 2020,.

[13] Joel Francia. Que es scrum? https://www.scrum.org/resources/blog/que-es-scrum. Accessed: Nov 19, 2020,.

[14] Vila Grau Juan Luis. La metodología xp: la metodología de desarollo de software más existosa. https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-metodologia-xp/. Accessed: Nov 19, 2020,.

[15] Barrios Arce Juan Ignacio. Sistemas de información. el business intelligence. https://www.juanbarrios.com/sistemas-de-informacion-el-business-intelligence/. Accessed: Nov 19, 2020,.

[16] N.D. ¿qué es business intelligence (bi) y qué herramientas existen? https://blog.signaturit.com/es/que-es-business-intelligence-bi-y-que-herramientas-existen. Accessed: May 14, 2020,.

[17] N.D. Bases de datos relacionales: qué son y por qué se utilizan - ionos. https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/bases-de-datos-relacionales. Accessed: May 14, 2020,.

[18] Ivana Harari Francisco Javier D´ıaz and Ana Paola Amadeo. *Guía de recomendaciones para diseño de software centrado en el usuario.* Editorial de la Universidad Nacional de La Plata (EDULP)„ 2013.

[19] John Jairo Londoño Perez. *Modelo Seudomatemático para el diseño de las bases de datos relacionales.* 2016.

[20] The Open Standard. Open group group. ArchiMate 2.0 Specification.

[21] Mark J. Price. *C 8.0 and .NET Core 3.0 – Modern Cross-Platform Development.* BIRMINGHAM - MUMBAI: Packt., 2019.

[22] Hammer Jacob Vibe Reid Jon D. Perkins, Benjamin. *Beginning C 7 Programming with Visual Studio® 2017 —— ASP.NET and ASP.NET Core.* United States of America: John Wiley Sons., 2017.

[23] N.D. Introducción a la creación de servicios web con .net. http://www.joaquin.medina.name/web2008/documentos/informatica/documentacion/logica/serviciosWeb/2007_08_31_CreacionDeServiciosWeb.html#Refh2_Ventajas_Y_Desventajas_De_Los_Servicios_Web. Accessed: May 14, 2020,.

[24] n.d. Hosting an angular progressive web application (pwa) with azure. https://devblogs.microsoft.com/premier-developer/hosting-an-angular-progressive-web-application-pwa-with-azure/, September 2019. Accessed: 2020-10-24.

[25] n.d. Building progressive web apps with the amplify framework and aws appsync. https://aws.amazon.com/es/blogs/mobile/building-progressive-web-apps-with-the-amplify-framework-and-aws-appsync/, September 2019. Accessed: 2020-10-24.

[26] Adam. Freeman. *Essential Angular for ASP.NET Core MVC —— Structuring an Angular Application.* United States of America: Apress., 2017.

[27] A. Osmani. El modelo de "shell de app." retrieved april 19. https://developers.google.com/web/fundamentals/architecture/app-shell?hl=es#app-shell-benefits, September 2019. Accessed: 2020-10-24.

[28] J. D. Rodriguez Fuentes. *PROPUESTA NORMATIVA PARA APLICACIONES MÓVILES EN COLOMBIA: DERECHOS Y DEBERES DE ACTORES INVOLUCRADOS EN LA CREACIÓN Y GESTIÓN DE APLICACIONES NATIVAS.* 2017.

[29] Congreso de la república. *Ley Estatutaria 1581 de 2012.* 2012.

[30] Congreso de la república. *Por medio de la cual se define y reglamenta el acceso y uso de los mensajes de datos, del comercio electrónico y de las firmas digitales, y s e establecen las entidades de certificación y se dictan otras disposiciones.* 1999.

Apéndice A

Anexos

A.1 EDT del proyecto

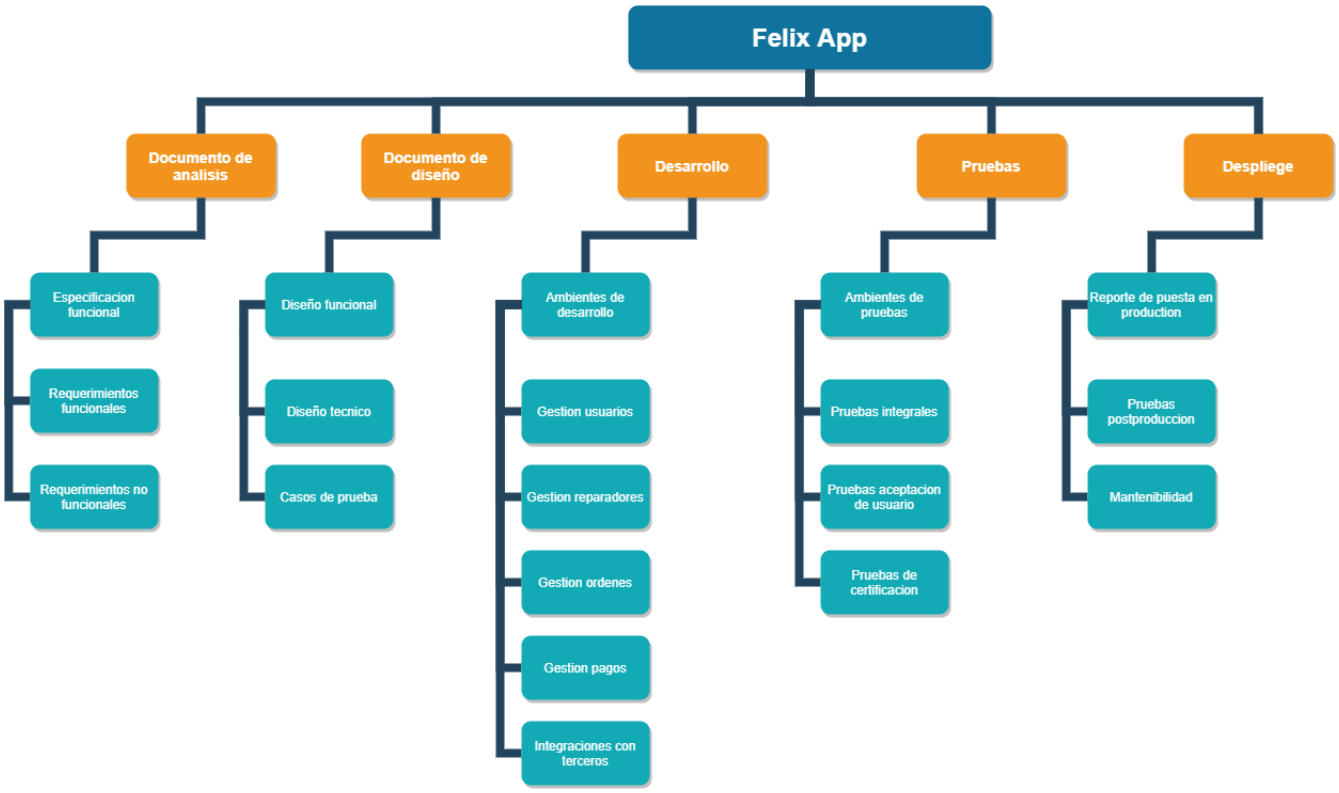


Fig. A.1: Edt del proyecto. Fuente: Los autores.

A.2 Cronograma

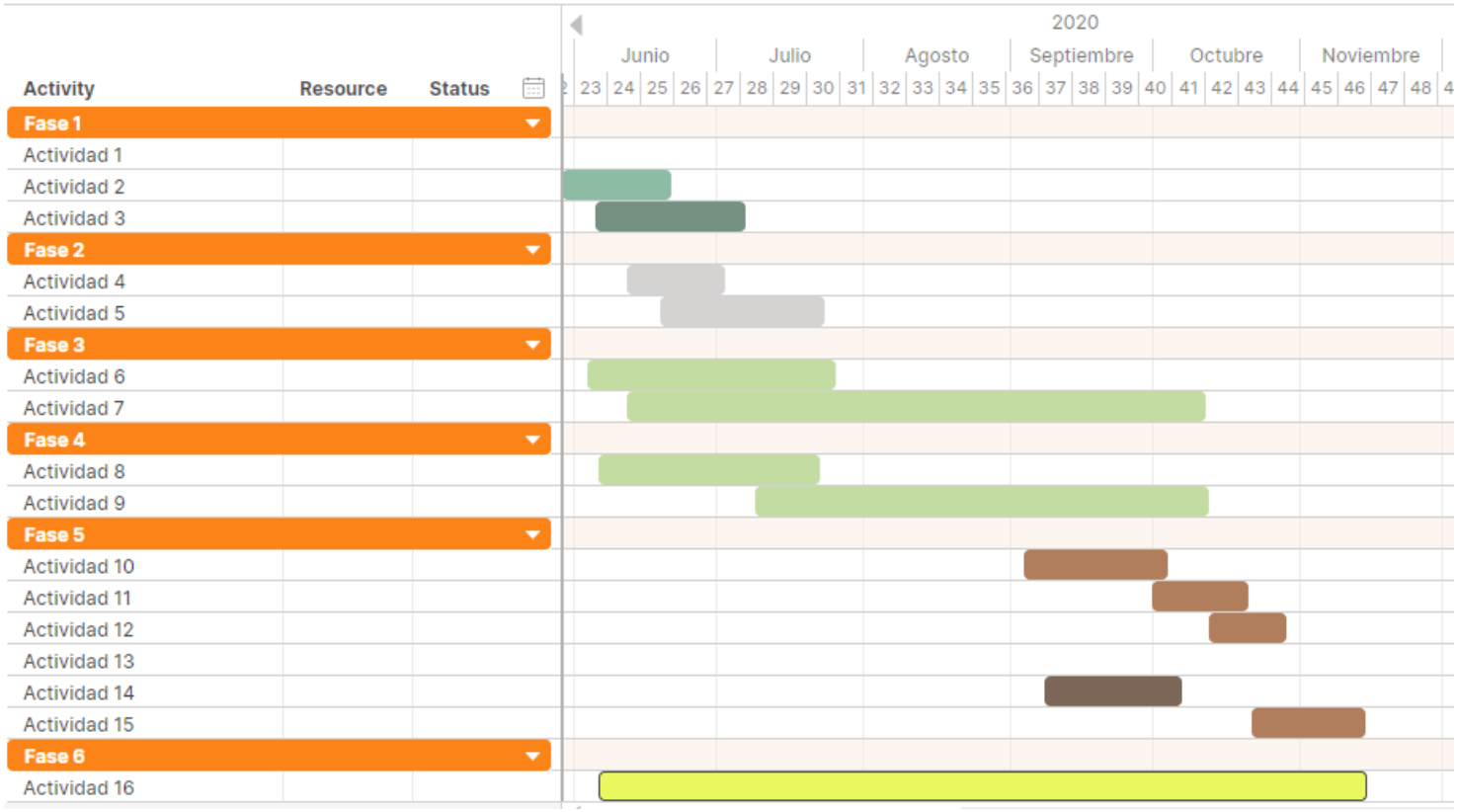


Fig. A.2: Diagrama de gantt. Fuente: Los autores.

Actividad	Descripción
1	Recopilación de la información en área de desarrollo de apps.
2	Recopilación de información en arquitectura de desarrollo de software.
3	Recopilación de información de aplicaciones de comunicación asertiva entre proveedores y usuarios finales de diferentes servicios.
4	recopilación de diferentes investigaciones y desarrollos de aplicaciones que hayan surgido de problemáticas distintivas con el proyecto.
5	Modelar las diferentes interfaces que tendrá la solución de este proyecto.
6	Diseño de la interfaz grafica que permita la interacción de los usuarios de la aplicación.
7	Realizar algoritmo de desarrollo implementando el framework definido para este fin.
8	Diseño de la solucion que permita la comunicación entre la interfaz grafica y la persistencia de los datos
9	Realizar un algoritmo implementando el framework definido y utilizando la arquitectura de desarrollo seleccionada.
10	Pruebas de la simulación de la aplicación.
11	Obtener trazas del comportamiento de la aplicación desarrollada.
12	Análisis de la información obtenida del comportamiento de la aplicación desarrollada.
13	Elaboración del cuestionario a realizar a usuarios.
14	Realizar Encuestas.
15	Análisis de las encuestas y el funcionamiento de la aplicación.
16	Elaboración informe final sobre el prototipo de la aplicación desarrollada, análisis de desarrollo y usabilidad por parte de los usuarios.

Fig. A.4: Actividades del proyecto. Fuente: Los autores.

Fases
Fase 1: Revisión bibliografica
Fase 2: Modelo de desarrollode la aplicación
Fase 3: Plantemainto de la interfaz grafica o Front-End.
Fase 4: Planteamiento del Back-End
Fase 5: Analisis de resultados
Fase 6: Informe

Fig. A.3: Fases del proyecto. Fuente: Los autores.

A.3 Mapa mental del proyecto

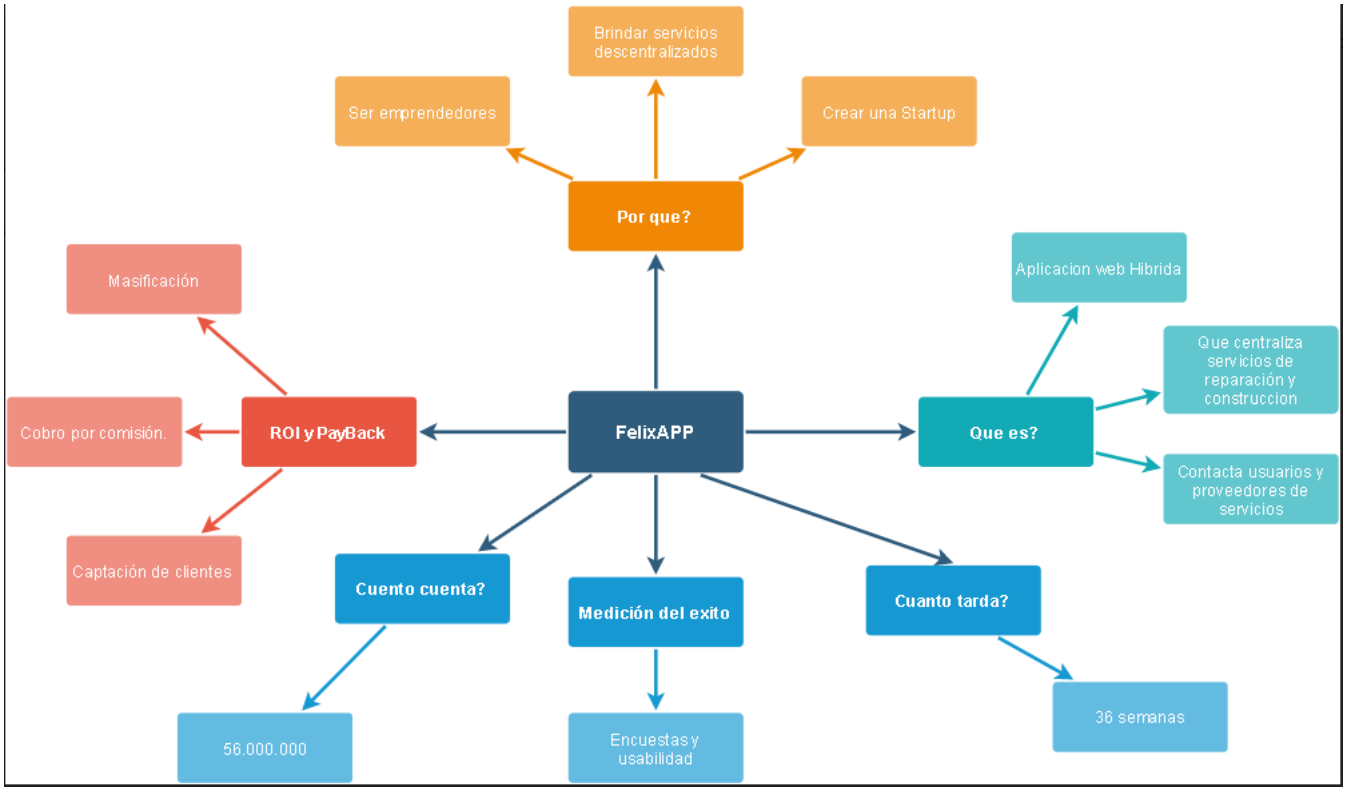


Fig. A.5: Mapa mental del proyecto. Fuente: Los autores.

A.4 Matriz de contrastacion

Objetivo	Descripcion	Cumplimiento	Evidencia
1	Prototipar una aplicación web progresiva disponible en IOS y Android para la comunicación efectiva entre profesionales de la obra civil y los posibles usuarios en la ciudad de Bogotá.	100%	Capitulo 2 Requerimientos del prototipo, Capitulo 7 Cierre y conclusiones
2	Documentar las características de las herramientas a utilizar en el desarrollo de la aplicación.	100%	Capitulo 3 Diseño de base de datos, Capitulo 1 numeral 1.7 Marco conceptual
3	Construir una interfaz gráfica con la herramienta IONIC para la interacción de usuario.	100%	Capitulo 5 numeral 5.2
4	Montar una estructura de servicios con .net core para soportar las peticiones de la interfaz gráfica.	100%	Capitulo 5 numeral 5.1

Fig. A.6: Matriz contrastacion del proyecto. Fuente: Los autores.

A.5 Presupuesto del proyecto

Recurso	Descripción	Cantidad	No Horas	Valor Mensual	Valor Hora	Valor Unitario	Total
Internet	Internet 50 Mb	1	560	$ 300,000.00	$ 1,250.00	N/A	$ 700,000.00
Luz	Servicio luz locación	1	560	$ 200,000.00	$ 833.33	N/A	$ 466,666.67
Agua	Servicio agua locación	1	560	$ 190,000.00	$ 791.67	N/A	$ 443,333.33
Locación	Locación para el desarrollo del proyecto	1	560	###########	$41,666.67	N/A	$ 23,333,333.33
Dispositivos Móviles	Celulares para pruebas	3	N/A	N/A	N/A	##########	$ 3,000,000.00
Dispositivos Móviles	Iphone 12 pro	2	N/A	N/A	N/A	##########	$ 11,999,998.00
Computadores	Macbook air 13	1	N/A	N/A	N/A	##########	$ 3,800,000.00
Computadores	Macbook pro	2	N/A	N/A	N/A	##########	$ 14,400,000.00
Computadores	Computador para el analista legal del proyecto	1	N/A	N/A	N/A	##########	$ 3,000,000.00
Computadores	Computadores para el testeo del proyecto	2	N/A	N/A	N/A	##########	$ 8,000,000.00
Computadores	Computadores para el desarrollo del proyecto	2	N/A	N/A	N/A	##########	$ 12,000,000.00
Licencias							$ 18,755,333.33
	Windows	4	N/A	N/A	N/A	##########	$ 6,000,000.00
	McAfee	4	N/A	N/A	N/A	$ 160,000.00	$ 640,000.00
	Oficce 365	7	N/A	N/A	N/A	$ 300,000.00	$ 2,100,000.00
	Suite Adobe	4	N/A	N/A	N/A	##########	$ 9,600,000.00
	Visual studio profesional	4	560	$ 178,000.00	$ 741.67	N/A	$ 415,333.33
							$ 99,898,664.67

Fig. A.7: Tabla de gastos generales del proyecto. Fuente: Los autores.

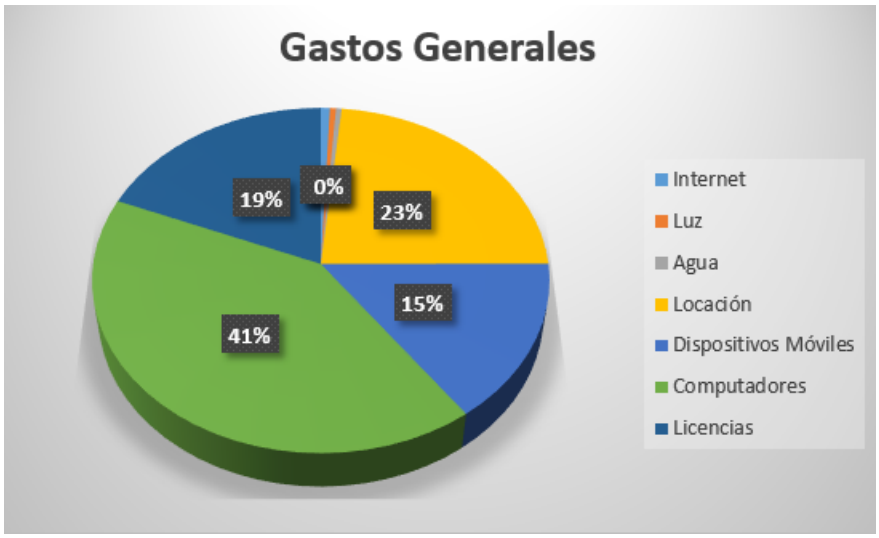


Fig. A.8: Grafico de gastos generales del proyecto. Fuente: Los autores.

Recurso	Descripción	Cantidad	No Horas por recurso	Valor Hora	Total
Director	Director del proyecto de grado encargado de direccionar el proyecto revisando mitologias y propuesta de valor del proyecto	1	60	$120,000.00	$7,200,000.00
Revisor	Revisor del proyecto de grado encargado de verificar procesos, mejorar las deudas técnicas y metodológicas encontradas alrededor del proyecto	1	60	$120,000.00	$7,200,000.00
Analista de desarrollo	Desarrollo de la aplicación	2	280	$31,000.00	$8,680,000.00
Analista de requerimiento	Levantar los requerimientos necesarios para el desarrollo de la aplicación	1	160	$25,000.00	$4,000,000.00
Analista de calidad	Revisor encargado de la calidad de la aplicación, identificación de bugs, reportes de calidad	1	320	$25,000.00	$8,000,000.00
Analista de infraestructura	Encargado de realizar el diseño y la implementación de la infraestructura para soportar la aplicación	1	120	$25,000.00	$3,000,000.00
Diseñador grafico	Encargado de realizar el diseño de la UI del producto	1	60	$20,000.00	$1,200,000.00
Analista legal	Encargado de realizar los términos y condiciones de uso de la aplicación, conformación legal de la compañia	1	70	$20,000.00	$1,400,000.00
			1130		$40,680,000.00

Fig. A.9: Tabla de costos por servicios personales del proyecto. Fuente: Los autores.

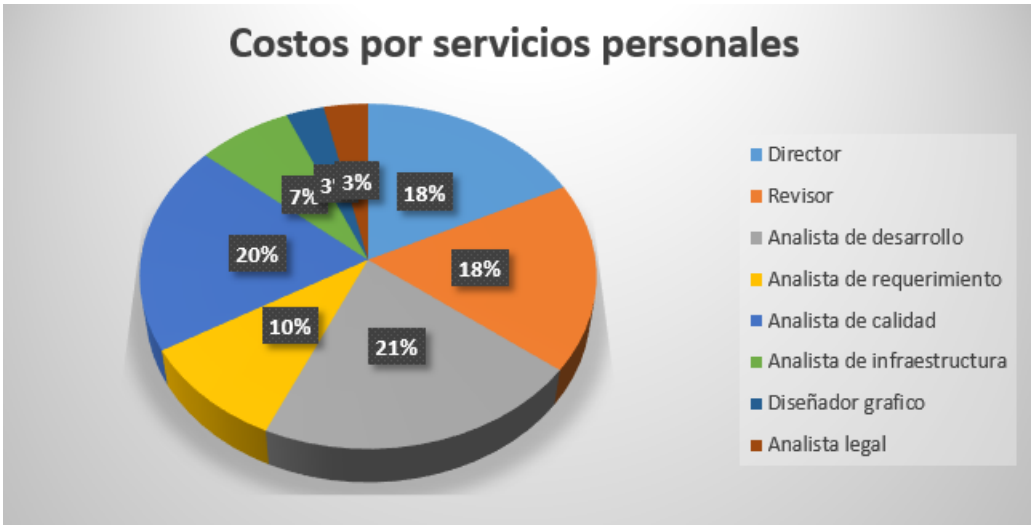


Fig. A.10: Grafico de costos por servicios personales del proyecto. Fuente: Los autores.

	Totales	
Gastos Generales	$	99,898,664.67
Costos por servicios personales	$	40,680,000.00
	$	**140,578,664.67**

Fig. A.11: Tabla de costos totales del proyecto. Fuente: Los autores.

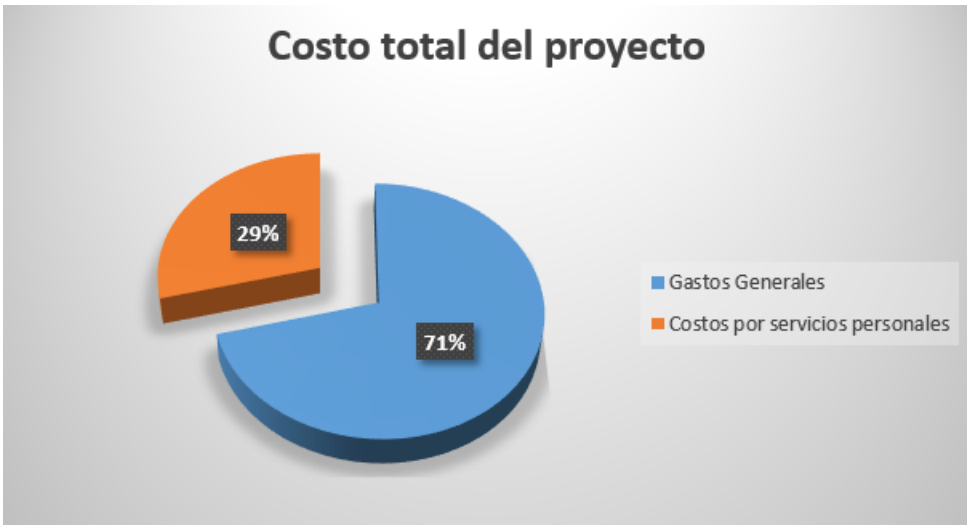


Fig. A.12: Grafico de costos totales del proyecto. Fuente: Los autores.