

**APLICACIÓN DE DESIGN THINKING CÓMO METODOLOGÍA PARA EL
DESARROLLO DE UNA APLICACIÓN DE OFERTA Y DEMANDA DE SERVICIOS
CÓMO PROYECTO DE INNOVACIÓN**

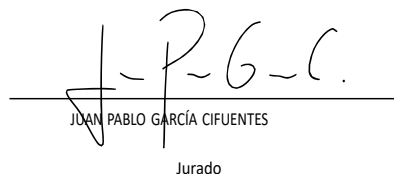
ANDRES FELIPE LOPEZ BEDOYA

Nota de Aceptación

Certificamos que el presente Trabajo de Grado Satisface, en alcances y calidad, todos los requisitos que demanda un Trabajo de Grado de Maestría.



JUAN CARLOS MARTÍNEZ ARIAS
Director Trabajo de Grado



JUAN PABLO GARCÍA CIFUENTES
Jurado



JULIAN FELIPE SIERRA CANO
Jurado

Aprobado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana Cali, para optar el título de Magister en INGENIERIA DE SOFTWARE.



HERNÁN CAMILO ROCHA NIÑO Ph. D.

Decano Facultad de Ingeniería y Ciencias



JUAN CARLOS MARTÍNEZ ARIAS
Director Posgrados de Ingeniería y Ciencias

Cali, 09 de febrero de 2023



Pontificia Universidad
JAVERIANA
Cali

Facultad de Ingeniería y Ciencias

Acta de Correcciones al Documento de Trabajo de Grado

Santiago de Cali, 09 de febrero 2023

Autor: Andres Felipe Lopez Bedoya

Título del Trabajo de Grado: *“APLICACIÓN DE DESIGN THINKING CÓMO METODOLOGÍA PARA EL DESARROLLO DE UNA APLICACIÓN DE OFERTA Y DEMANDA DE SERVICIOS CÓMO PROYECTO DE INNOVACIÓN”*

Director:

Como indica el artículo 2.13 de las Directrices para Trabajo de Grado de Maestría, he verificado que el estudiante indicado arriba ha implementado todas las correcciones que los Jurados del Proyecto de Trabajo de Grado definieron que se efectuaran, como consta en el Acta de Evaluación correspondiente.

Director del Trabajo de Grado

Juan Carlos Martínez Arias

DATOS DEL ESTUDIANTE
TRABAJO DE GRADO

1. **ESTUDIANTE:** Andres Felipe López Bedoya
2. **CORREO ELECTRÓNICO:** ingenierlopez@javerianacali.edu.co
3. **TELÉFONO DE CONTACTO:** 3225689050
4. **DIRECCIÓN:** Carrera 19A # 8-93 Jamundí
5. **PROFESIÓN:** Ingeniero de Sistemas – Unidad Central del Valle del Cauca (UCEVA)
6. **EMPRESA:** COEXITO SAS
7. **CARGO:** Coordinador de Arquitectura TI

RESUMEN
TRABAJO DE GRADO

Los proyectos de desarrollo de software se ejecutan bajo metodologías tradicionales, ágiles o relacionadas a procesos de desarrollo de software. Se abordó este trabajo por medio de la metodología Design thinking aplicado al desarrollo de software de una aplicación de oferta y demanda de servicios para la aplicabilidad de ejecutar proyectos de desarrollo de software basado en las necesidades de los usuarios. De igual manera se valida por medio de la metodología la aplicación cómo proyecto de innovación que apoye en el impulso de la economía de la región del Valle del Cauca.

ABSTRACT
TRABAJO DE GRADO

Software development projects are executed under traditional methodologies, agile or related to software development processes. This work was approached using the Design Thinking methodology applied to the software development of a supply and demand service application for the applicability of executing software development projects based on user needs Likewise, the methodology validates the application of an innovation project that supports the promotion of the economy of the Valle del Cauca region.



Pontificia Universidad
JAVERIANA
Cali

**APLICACIÓN DE DESIGN THINKING COMO METODOLOGÍA PARA EL DESARROLLO
DE UNA APLICACIÓN DE OFERTA Y DEMANDA DE SERVICIOS COMO PROYECTO
DE INNOVACIÓN**

Andres Felipe López Bedoya

*Proyecto de trabajo de grado para optar al título de
Magister en Ingeniería de Software*

Director
Juan Carlos Martínez Arias

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN INGENIERÍA DE SOFTWARE
SANTIAGO DE CALI, NOVIEMBRE 21 DE 2022

TABLA DE CONTENIDO

Contenido

1. INTRODUCCIÓN.....	7
2. GLOSARIO.....	8
3. DEFINICIÓN DEL PROBLEMA	11
3.1 PLANTEAMIENTO DEL PROBLEMA	12
3.2 FORMULACIÓN DEL PROBLEMA.....	14
4. OBJETIVOS DEL PROYECTO.....	15
4.1 OBJETIVO GENERAL	15
4.2 OBJETIVOS ESPECÍFICOS	15
4.3 RESULTADOS ESPERADOS	15
5. MARCO TEÓRICO DE REFERENCIA Y ANTECEDENTES	16
5.1 METODOLOGÍA DESIGN THINKING	16
5.2 TÉCNICAS DE DESIGN THINKING	16
5.2.1 Mapa de actores.....	17
5.2.2 Inmersión cognitiva	17
5.2.3 Mooboard	17
5.2.4 ¿Qué?, ¿cómo?, ¿por qué?.....	17
5.2.5 Cinco ¿por qué?	17
5.2.6 Toolkit	17
5.2.7 Lienzo modelo negocio.....	17
5.2.8 Curvas de valor	18
5.2.9 Mapas del contexto	18
5.2.10 Dibujo del problema.....	18
5.2.11 Estudio de tendencias	18
5.2.12 Entrevistas a expertos	18
5.3 ETAPAS DE DESIGN THINKING	19
5.4 PROYECTO DE DESARROLLO DE SOFTWARE APLICANDO DESIGN THINKING	20
5.5 TRABAJOS RELACIONADOS.....	21

6. DESIGN THINKING Y PROCESOS DE DESARROLLO DE SOFTWARE.....	22
6.1 Adaptabilidad	24
6.2 Procesos de Desarrollo de software y Design Thinking	24
6.2.1 Análisis.....	24
6.2.2 Prototipo	25
6.2.3 Implementación	25
6.3 Retroalimentación y sinergia de los procesos.....	25
7. CONTEXTO DEL PROYECTO A DESARROLLAR.....	26
8. OBTENCIÓN Y ANÁLISIS DE REQUISITOS APLICANDO DESIGN THINKING.....	29
8.1 Artefactos para obtención de requerimientos.....	30
8.2 Resultados obtenidos del trabajo de campo.....	33
8.2.1 Técnicas de ideación aplicadas.....	34
8.3 Análisis de requerimientos.....	36
8.3.1 Diagrama de Caso de Uso general del sistema	37
8.3.2 Definición de requerimientos.....	38
8.3.3 Diseño del sistema.....	42
9. VALIDACIÓN DE PROTOTIPACIÓN ENFOCADO EN LA ADAPTACIÓN DEL PRODUCTO.....	46
9.1 Prototipado	46
9.2 Validación prototipación	47
10. DESARROLLO Y PRUEBAS DEL PRODUCTO	50
10.1 Implementación técnica	50
10.1.1 Desarrollo Back.....	50
10.1.2 Despliegue Back con Elastic Beanstalk	52
10.1.3 Desarrollo Front.....	54
10.1.4 Despliegue de aplicativo en Android	55
10.1.5 Despliegue de aplicativo en Android	56
11. RESULTADOS EN LA ADAPTABILIDAD DE PROCESOS DE DESIGN THINKING Y DE SOFTWARE CON EL MODELO PROPUESTO.....	57
11.1 Adaptación entre la metodología Design Thinking y procesos de desarrollo de software....	57
11.2 Aplicabilidad del modelo definido.....	58
11.3 Resultados obtenidos a partir del modelo	60
11.4 Principios de la adaptación de procesos de software con Design Thiking.....	61

11.5 Guía de aplicabilidad del modelo	61
12. RESULTADO DEL DESARROLLO PROTOTIPO FINAL.....	64
13. PRUEBAS REALIZADAS AL PROTOTIPO.....	67
14. CONCLUSIONES	69
14.1 Trabajos futuros:.....	69
15. REFERENCIAS BIBLIOGRÁFICAS	70

LISTA DE FIGURAS

Ilustración 1 Ciclo de vida de metodologías [2]	11
Ilustración 2 Standish Group fracasos de proyectos TI [9]	12
Ilustración 3 Combinar el Design Thinking [10].....	14
Ilustración 4 Modelo Design Thinking [13].....	19
Ilustración 5 Modelo definido de procesos de software y Design Thinking parte 1	22
Ilustración 6 Modelo definido de procesos de software y Design Thinking parte 2	23
Ilustración 7 Que va primero Idea o Usuario	26
Ilustración 8 Indicadores de desarrollo de actividades por cámara y comercio Cali [18]	27
Ilustración 9 Indicadores comportamiento por Cámara y comercio Cali [18]	27
Ilustración 10 Conexión emocional [2].....	29
Ilustración 11 Obtención y análisis de requisitos.....	30
Ilustración 12 Resultado flujo de Demanda de clientes	31
Ilustración 13 Resultado plataformas de comunicación utilizadas	31
Ilustración 14 Resultado de adquisición de clientes	31
Ilustración 15 Resultado de adaptación al cambio de conectar oferta y demanda.....	32
Ilustración 16 Resultado recomendaciones	32
Ilustración 17 Canvas Hallazgos.....	35
Ilustración 18 Canvas de Negocio.....	35
Ilustración 19 Contexto Dominio Sistema	37
Ilustración 20 Caso de Uso general del Sistema.....	38
Ilustración 21 Diagrama de procesos de aplicativo	43
Ilustración 22 C4-Contexto	44
Ilustración 23 C4 – Contenedor	45
Ilustración 24 Proceso prototipación modelo definido.....	46
Ilustración 25 Validación prototipo Inicio sesión	47
Ilustración 26 Validación información prototipo servicio	47
Ilustración 27 Validación colores y letras plataforma	48
Ilustración 28 Validación opiniones abiertas usuarios	48
Ilustración 29 Etapa de desarrollo y pruebas validación.....	50
Ilustración 30 Spring Boot inicial	51
Ilustración 31 Estructura del proyecto Spring Boot	51
Ilustración 32 Implementación de clases	52
Ilustración 33 Crear aplicación Web.....	52
Ilustración 34 Configuraciones del servicio en Beanstalk	53
Ilustración 35 Despliegue del servicio con Beanstalk.....	53
Ilustración 36 Estructura del proyecto Front.....	54
Ilustración 37 Dependencias de Flutter	54
Ilustración 38 Generar clave de aplicativo	55
Ilustración 39 Comando Flutter generar APK Android	56
Ilustración 40 Creación aplicación en Google Developer.....	56

Ilustración 41 Resultado de Artefactos por Usuario	60
Ilustración 42 Login	64
Ilustración 43 Buscar servicio	64
Ilustración 44 Información del servicio	65
Ilustración 45 Reservar servicio.....	66
Ilustración 46 Generación código de reserva	66
Ilustración 47 Pruebas código back.....	67
Ilustración 48 Consumo de APIS.....	67
Ilustración 49 Seguridad JWT APIS	68

LISTA DE TABLAS

Tabla 1 Información relevante entrevista experto	33
Tabla 2 Resultados trabajo campo	34
Tabla 3 Benchmark comparativo atributos y aplicaciones existentes.....	36
Tabla 4 Definición de requerimientos funcionales.....	40
Tabla 5 Definición requerimientos restrictivos	42
Tabla 6 Herramientas utilizadas por etapa del modelo.....	59

1. INTRODUCCIÓN

La situación de informalidad antes de pandemia, entre los años 2017 y 2019, tuvo un comportamiento de 47 % a 48,2 % para el año 2017 y entre un 46,5 % a 47,6 % al cierre del 2019, donde la tasa porcentual se ha mantenido en un rango elevado en este ítem.

El mercado laboral de la ciudad de Cali ha presentado en los últimos años tasas de desempleo superiores a la tasa de desempleo nacional, entre dos y tres puntos porcentuales[1]. Situación similar se observa para el departamento del Valle. A esta complicada situación se le suman los efectos de medidas como la cuarentena, implementadas a finales del mes de marzo de 2020, con el fin de reducir los impactos del COVID-19 sobre las vidas de los colombianos; los cierres de empresas en la ciudad y demás factores no antes vistos en la región.

Además, las estadísticas del DANE reafirman que el trabajo informal se distribuye en hombres con un 47,4 % representando un aumento de 2,2 puntos a comparación del 2019, mientras que las mujeres representan el 49,1 % con un aumento de casi un punto de acuerdo con estadísticas publicadas por el DANE en los periodos de diciembre 2020 a febrero del 2021. De acuerdo con dicha información, la informalidad en la región ha aumentado viéndose reflejada la dificultad de conseguir un empleo estable.

Los servicios de oferta y demanda son trabajos que han tratado de disminuir el índice de desempleo, siendo promovidos desde diferentes entes gubernamentales, locales, empresas privadas o personas naturales, sin embargo, esta estrategia no ha sido efectiva, por el contrario, esta se ha visto retrasada los últimos dos años debido a las diversas complicaciones causadas por la crisis sanitaria, generando así, dilatación en los tiempos para alcanzar los niveles de estabilización de la región.

De acuerdo con lo anterior, se han introducido en el mercado nuevas herramientas tecnológicas y aplicaciones que permitan comunicar de forma más eficiente y ágil estas dos poblaciones las empresas que requieren capital humano y las personas que requieren un empleo por medio de diferentes aplicaciones con uso de metodologías tradicionales.

Este proyecto concluye que la metodología Design Thinking, aplicada en este campo de desarrollo, específicamente en este tipo de proyectos e implementación de software es efectiva y logra reducir tiempos de entrega, prototipos y satisfacción hacia el cliente, determinando que la metodología adoptada en proyectos de innovación pueda ser usada en otros campos.

2. GLOSARIO

Co-creación: Involucramiento de los usuarios como forma de generar valor para las personas y éxito para las organizaciones [2].

Contexto dominio: Forma gráfica de representar un sistema con todos sus componentes como usuarios, documentación, funciones, entes externos.

COVID-19: Es una enfermedad causada por el nuevo coronavirus conocido como SARS-CoV-2 de acuerdo con la OMS (Organización Mundial de la Salud) [3].

DANE: Departamento Administrativo Nacional de Estadística de Colombia.

Design Thinking: Es una metodología ágil de innovación centrada en las personas [2].

Diagrama caso de uso: Forma gráfica y estática de representar funcionalmente un sistema con sus respectivas dependencias.

DynamoDB: Base de datos no relacional del proveedor en nube pública Amazon Web Service que ofrece una concurrencia de carga y multimedia en tiempo real.

Empatizar: Es la etapa de la metodología Design Thinking donde se explora el problema y el contexto aplicado a alguno de los mapas del contexto como técnica.

Entrevista de profundidad: Se denominan a este tipo de entrevistas donde se busca encontrar un significado más profundo y oportunidades de información no clara [2].

Gartner: Grupo de expertos para investigar tendencias aplicándolas en clientes reales.

Insights: Son una síntesis, una expresión concisa de lo que se ha aprendido a través de la investigación [2].

Metodologías ágiles: Estas metodologías ayudan en el desarrollo de proyectos que necesitan mayor enfoque para adecuarse a las necesidades del cliente [4].

Nicho de mercado: Es una porción de un segmento de mercado que está compuesto por un grupo reducido de personas o empresas [5].

Oficios varios: Profesión que se conoce en Colombia hacia una ocupación o actividad económica.

Patrones de información: Es aquella información que coincide con el resultado de las entrevistas

aplicadas y se puedan agrupar de manera concisa.

Procesos de desarrollo de software: Son aquellos que se componen de diferentes actividades dividido por lo regular en las fases de análisis, diseño e implementación para crear un producto de software.

Programación extrema (XP): Es una metodología de desarrollo de la ingeniería de software.

Prototipar: Es la etapa de la metodología Design Thinking donde indica fallar de forma económica y veloz, es una premisa, un mecanismo más exploratorio.

Requerimientos funcionales: Expresa las necesidades y restricciones definidas sobre un producto de software que contribuye a la solución de algún problema del mundo real [6].

Requerimientos no funcionales: Son aquellos requerimientos que están asociados en función de cumplir los atributos de calidad de un producto de software como la disponibilidad, seguridad, despliegue, entre otros importantes.

RoadMap: Es la hoja de ruta de un proyecto o la guía para llevarlo a cabo y alcanzar los objetivos.

Scrum: Es una de las principales metodologías de agilidad, aplicando los conceptos de entregas tempranas y satisfacción del cliente.

Sesgo de confirmación: Es aquel que define que la información sea compatible con el resultado de las entrevistas aplicadas [2].

Sinergia: Es aquella acción que hace que dos o más cosas puedan actuar conjuntamente.

Stakeholders: Son los involucrados de un proyecto o partes interesadas.

Standish Group: Ofrece una amplia gama de herramientas y servicios, brindando asesoramiento e investigación a través de los programas de membresía ofrecidos [7].

Testear: Es la última fase de la etapa de la metodología donde se refleja el resultado del lanzamiento de los prototipos y ya se tiene una percepción del usuario.

Usuario extremo: Es aquel usuario que puede aportar información necesaria para la exploración del contexto y búsqueda de la información.

Usuario experto: Es aquel usuario conocedor del tema principal de investigación que ayuda a ahondar y detallar aún más en la investigación.

Usuario objetivo: Es aquel usuario identificado como el usuario que va a utilizar el producto a desarrollar.

Workshops: Son aquellas actividades que hacen parte de los talleres prácticos entre grupo de personas involucradas en un proyecto.

Zona gris: Puntos no claros en un contexto del dominio a definir.

3. DEFINICIÓN DEL PROBLEMA

Tradicionalmente, los proyectos de desarrollo de software se han basado sobre metodologías en procesos de desarrollo de software, tanto de manera tradicional, como de modo ágil, tales como cascada, programación extrema (XP), Scrum, entre otros, donde cada una de estas metodologías abarcan una gran cantidad de las siguientes fases.



Ilustración 1 Ciclo de vida de metodologías [2]

Las metodologías enfocadas en desarrollo de software tienen en particular las fases de análisis, diseño, implementación y pruebas, su objetivo es llegar a obtener un producto de software de manera procedimental, desde una etapa temprana de acercamiento del producto, hasta su etapa más trascendental, llegando así a un producto más maduro y aceptado por el cliente[8]. De acuerdo con lo anterior, los proyectos de software tienen el gran reto de superar barreras de factores de fracaso, donde la gran mayoría se ven reflejados en aspectos técnicos o administrativos, ya que son productos orientados a clientes, donde se afectan directamente en tiempos y presupuesto.

A continuación, se muestran los factores de fracasos de proyectos de TI:

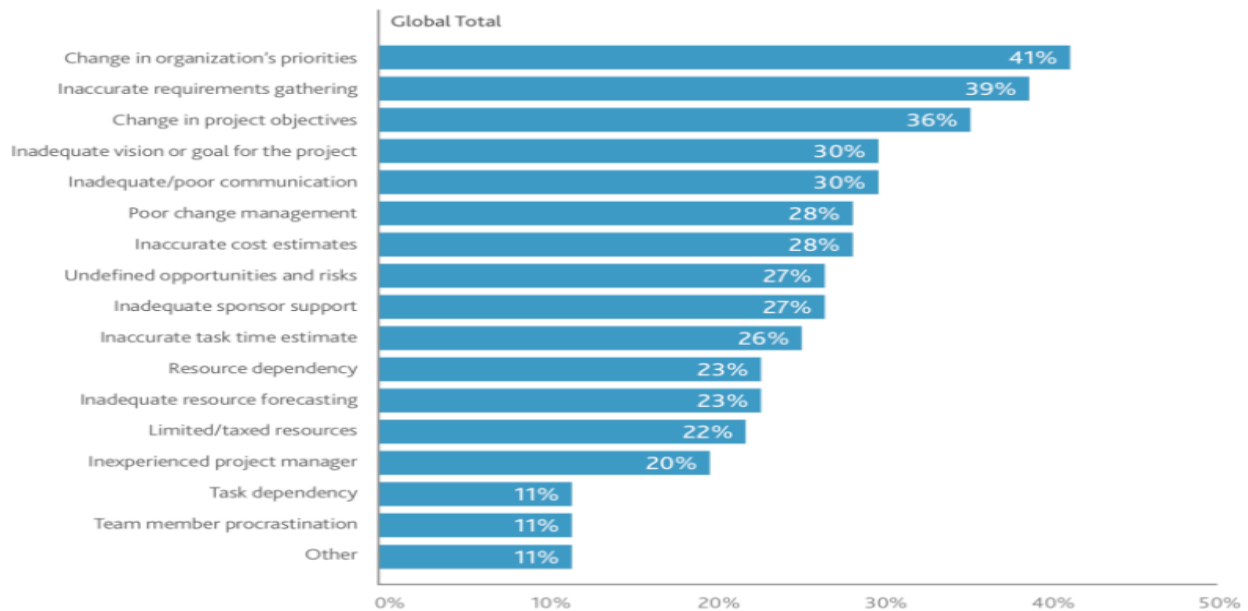


Ilustración 2 Standish Group fracasos de proyectos TI [9]

De acuerdo con Standish Group, la gran parte de las razones de fracasos de proyectos de TI salen por cambios en prioridades, inapropiados requerimientos, cambios de objetivos en el proyecto, entre otros importantes. De igual manera, independientemente de la metodología de desarrollo a aplicar, esta brecha sigue apareciendo entre lo que requiere en realidad el cliente y lo que se entrega como producto de software.

3.1 PLANTEAMIENTO DEL PROBLEMA

En la región del Valle del Cauca, la tasa de trabajo informal tanto para hombres como para mujeres alcanza más de un 47 %, donde los gobiernos locales y nacionales no han podido encontrar la forma de comunicar los potenciales ejecutados con empleadores interesados para impulsar la economía de una manera más rápida. Es de resaltar que la curva respecto a la informalidad de empleos va en tendencia creciente y aún no se tiene un claro horizonte. La búsqueda empírica de empleabilidad se hace compleja debido a que no fluye la efectividad de esta, no solo debido a la falta de oportunidades de empleo, sino también, por el poco relacionamiento entre los ofertantes y los demandantes de los servicios de una manera más eficaz, desembocando el mayor desempleo y desaprovechamiento del recurso humano.

De acuerdo con lo mencionado anteriormente, es importante tener en cuenta la problemática de los servicios de oferta y demanda que tienen como reto la informalidad, teniendo como factor

importante la ejecución de un proyecto de innovación que esté orientado a las necesidades de los usuarios mediante el desarrollo de un software.

Por otro lado, como se indicó anteriormente, los proyectos de desarrollo de software de acuerdo con la Ilustración uno (1) de Standish Group, tienden a fracasar por varios ítems principales como lo son los cambios en prioridades, inapropiados requerimientos obtenidos, cambios de objetivos en el proyecto, entre otros. Esto sucede, donde la dinámica no ha sido la adecuada en el momento de la entrega o la satisfacción con el usuario final, encontrando inconformidades con el producto, ocurrencias de nuevas funcionalidades después de la entrega, desfase en los tiempos, comunicaciones no fluidas, entre otros. Así mismo, las metodologías de desarrollo de software tienen en particular el manejo de las fases de análisis, diseño, implementación, despliegue pensadas o enfocadas en dar como fin un producto de software, siendo estas un poco complejas referente a demoras en tiempos de entregas.

Teniendo en cuenta lo mencionado anteriormente, los proyectos de desarrollo de software que involucran la aplicabilidad de la metodología de *Design thinking* requerirá implementar otro tipo de fase como lo son: empatizar, definir, idear, prototipar y testear, las cuales están orientadas al usuario final. De acuerdo con la problemática y las necesidades que se requiere abordar en el proyecto, es necesario aplicar una metodología que se enfoque en dos frentes importantes como las necesidades de los usuarios y la tecnología aportando un valor al usuario y una oportunidad de mercado.

De acuerdo con lo anterior, *Design Thinking* es una metodología que se puede complementar con otras metodologías o puede cumplir con el ciclo completo de un prototipo lanzado en un mercado específico. A continuación, se mostrarán unas Ilustraciones que validan el proceso de la metodología.

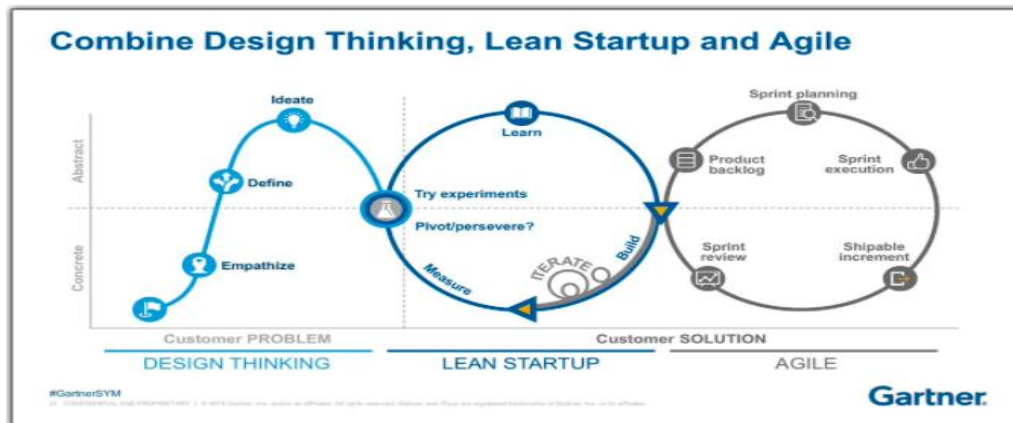


Ilustración 3 Combinar el Design Thinking [10]

De acuerdo con la anterior Ilustración de Gartner, el *Design Thinking* es una metodología que se enfoca en los problemas a partir del pensamiento de las personas en cuanto a «dolores del usuario», redefinición de problemas, creación de ideas nuevas e innovadoras testeables en el nicho de mercado, mediante prototipos validados. Es una metodología corta y a bajo costo para que los productos puedan ser probados y validados.

Teniendo en cuenta las necesidades de la región del Valle del Cauca, logramos evidenciar una oportunidad que nos permitirá evaluar una metodología que se pueda adaptar a un proyecto de desarrollo de software, con el cual se ejecute el desarrollo de un prototipo de oferta y demanda en un nicho de mercado de oficios varios, empleando la exploración de una metodología en el área de innovación como *Design Thinking* que se enfocará en usuarios y aplicabilidad de la tecnología en un entorno particular de la economía de la región en la ciudad de Cali.

3.2 FORMULACIÓN DEL PROBLEMA

¿Cómo evaluar la efectividad de la metodología de *Design Thinking* en el desarrollo de software, para el prototipo de la aplicación de oferta y demanda de servicios de empleo para oficios varios en la ciudad de Cali?

4. OBJETIVOS DEL PROYECTO

4.1 OBJETIVO GENERAL

Valorar la efectividad de la metodología de Design Thinking en el desarrollo de software para el prototipo de la aplicación de oferta y demanda de servicios de empleo para oficios varios en la ciudad de Cali.

4.2 OBJETIVOS ESPECÍFICOS

- Identificar cómo la metodología Design Thinking podría ajustarse a los proyectos de desarrollo de software.
- Definir los requerimientos funcionales y no funcionales para la aplicación de oferta y demanda de servicios, utilizando la metodología Design Thinking.
- Aplicar las fases de Design Thinking en el desarrollo del prototipo de la aplicación de oferta y demanda de servicios.
- Desarrollar el prototipo de una aplicación de oferta y demanda de servicios de acuerdo con la obtención de requerimientos, aplicando la metodología Design Thinking.
- Concluir la efectividad del uso de la metodología dentro del desarrollo del prototipo de la aplicación de oferta y demanda de servicios.

4.3 RESULTADOS ESPERADOS

- Prototipo de la aplicación que contenga las principales necesidades del usuario final aplicando la metodología Design Thinking.
- Aportes al desarrollo de software basados en la metodología Design Thinking.
- Resultados obtenidos de la evaluación de la viabilidad y usabilidad por parte de los usuarios finales.

5. MARCO TEÓRICO DE REFERENCIA Y ANTECEDENTES

La metodología Design Thinking es una metodología que ha sido utilizada por lo regular en proyectos de innovación donde va de la mano del pensamiento del diseño y la tecnología.

5.1 METODOLOGÍA DESIGN THINKING

A continuación, se define la metodología. «Design Thinking se trata de una disciplina que usa la sensibilidad y métodos de los diseñadores para hacer coincidir las necesidades de las personas con lo que es tecnológicamente factible y con lo que una estrategia viable de negocios puede convertir en valor para el cliente y en una oportunidad para el mercado» [11]. De acuerdo con Tim Brown, CEO de la firma IDEO, esto es una disciplina que nos permite aterrizar por medio de la tecnología las necesidades de las personas, enfatizando en que los diseñadores deben permitir ese vínculo de necesidad, para convertirlas en valores agregados y generar grandes oportunidades de negocios al final del ejercicio.

En este sentido, una definición del Designer «es pensando de manera deductiva que el designer constantemente desafía sus estándares, haciendo y deshaciendo conjeturas, y transformándose en oportunidades para la innovación. Es esa habilidad, de deshacerse del pensamiento lógico cartesiano, que hace que el designer se mantenga fuera de la caja» [12]. Lo anterior nos indica que no se pueden solucionar problemas con un mismo tipo de pensamiento y cabe resaltar que esto se adecúa perfectamente a los desarrollos de software donde no existe pieza perfecta para una necesidad en general y existen muchas oportunidades de innovación de acuerdo con las necesidades en particular encontradas.

El Design Thinking viene a complementar las visiones encontradas de mercado, enfatizando en que para innovar es necesario enfocarse en el desarrollo de nuevas tecnologías. Para comprender mejor el concepto el «Design» debemos tener en cuenta que esto es una disciplina que lidia con los significados, pero ya involucrando todo el significado de la frase «Design Thinking», son soluciones que generan nuevos significados como aspectos cognitivos, emocionales y sensoriales, los cuales se trasladan a la experiencia humana con el fin de innovar para introducir nuevos significados a productos y servicios.

5.2 TÉCNICAS DE DESIGN THINKING

Esta metodología la componen un conjunto de herramientas o técnicas que se pueden aplicar o no, que van de la mano para sacar su mejor provecho o rendimiento a un producto en específico. A continuación, se nombrarán algunas de las técnicas que se adaptan mejor en lo que se pretende desarrollar en este proyecto de software.

5.2.1 Mapa de actores

Esta técnica consiste en identificar a los usuarios. Aquí se define el mapa de usuarios de acuerdo a sus roles [13].

5.2.2 Inmersión cognitiva

Esta técnica describe la posibilidad de entender la realidad existente colocándose en la posición del usuario. Se trata de poder vivir las experiencias del usuario y entender la realidad que lo rodea [13].

5.2.3 Mooboard

Consiste en obtener información visual cómo la selección de imágenes, videos y materiales donde se lleva a la aproximación de la solución mediante esta técnica, algo similar se trabaja en el mundo del desarrollo del software con los mockups, que es basado en los prototipos de interfaces de usuario[13].

5.2.4 ¿Qué?, ¿cómo?, ¿por qué?

Esta técnica consiste en aumentar la capacidad de descubrimiento de supuestos e ir relacionando cada tema a medida que lo indagamos, debemos tener en cuenta que si se puede hacer este, el por qué puede ser clave para el usuario [13].

5.2.5 Cinco ¿por qué?

Esta técnica consiste en llevarlo hacia las entrevistas preguntando hasta cinco veces «¿por qué?», para llegar a una respuesta más concreta, va orientado a los comportamientos o pensamientos de las personas entrevistadas [13].

5.2.6 Toolkit

Esta técnica se encarga de recolectar todo el material de apoyo, siendo transversal en todas las fases de la metodología y sirviendo como herramienta de co-creación de un producto[13].

5.2.7 Lienzo modelo negocio

Mediante esta técnica se puede tener un acercamiento de los posibles clientes, propuestas de valor y canales de comunicación. Para este punto, el modelo de negocio es fundamental diseñarlo bajo herramientas como los Toolkit, las entrevistas con expertos, la identificación de los posibles clientes y el valor diferenciador, con el fin de encontrar la mejor estrategia [13].

5.2.8 Curvas de valor

El objetivo de dicha estrategia es identificar cómo la competencia genera valor agregado y cómo se pueden identificar parámetros diferenciadores con base en la estrategia de valor de la competencia [13].

5.2.9 Mapas del contexto

Se define esta técnica desde los diferentes puntos de vista donde se desea trabajar, para definir el contexto como por ejemplo tendencias, factores económicos, factores sociales, entre otros [13].

5.2.10 Dibujo del problema

Es relevante tener claro qué reto se necesita solucionar, del mismo modo, es importante que en este punto los posibles clientes evidencien los problemas que ellos presentan para dar una pronta solución, siendo fundamental que se pueda plasmar de manera visual, con el fin de obtener un lienzo o un dibujo del problema requerido y llegar a una conclusión general [13].

5.2.11 Estudio de tendencias

El estudio de tendencias nos permite conocer a profundidad el material especializado acerca de las diferentes soluciones que estén asociadas, con el fin de identificar una posible solución con innovación [13].

5.2.12 Entrevistas a expertos

Es de gran aporte tener la experiencia técnica o especialista acerca del tema a solucionar, donde ayuda a generar ideas y crear un mayor alineamiento a la solución [13].

Todas estas técnicas y muchas más que aporta la metodología Design Thinking son de gran ayuda y generan un lineamiento de las mejores prácticas para un mejor provecho de la metodología con el fin de generar proyectos de innovación con bases sólidas.

De acuerdo con lo anterior, es importante explorar y validar cómo esta metodología tiene aplicabilidad y adaptabilidad por medio de un producto de software enfocado al usuario final. Es importante tener en cuenta que las anteriores técnicas descritas sacan mejor provecho a un proyecto de software ya que va de la mano con la innovación aplicada al ámbito tecnológico, cuyo fin es construir un producto intangible no físico, donde algunos de los proyectos de software tienden a fracasar por no tener en cuenta la validación con los usuarios, la evaluación de las necesidades y los componentes diferenciadores e innovadores.

Es importante tener en cuenta el fundamento teórico de cada una de las etapas de la metodología, puesto que esto nos permitirá seguir de forma adecuada el proceso propuesto.

5.3 ETAPAS DE DESIGN THINKING

Es importante dentro de la metodología Design Thinking seguir las respectivas etapas, aplicando las técnicas que se adecuen mejor al proceso, teniendo en cuenta el contexto en que se vaya a desarrollar el proyecto de innovación. A continuación, se ilustra el ciclo de etapas de la metodología Design Thinking.

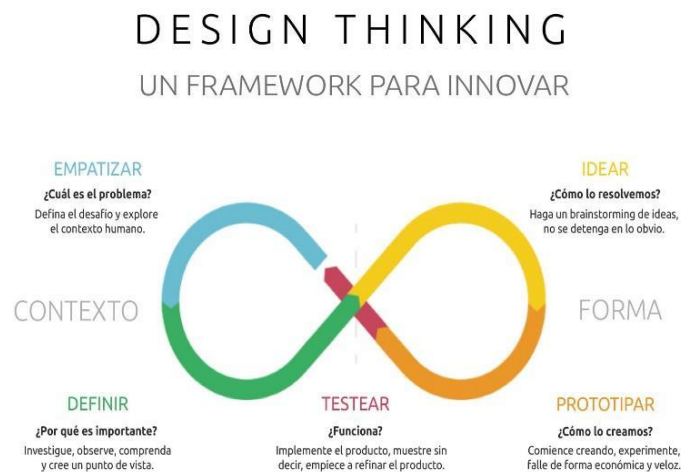


Ilustración 4 Modelo Design Thinking [13]

Empatizar es la etapa donde se explora el problema y el contexto, aplicando algunas herramientas como lo son los mapas del contexto como técnica de seguimiento, con el fin de hacer un lineamiento del problema a solucionar y el usuario que lo requiere. En este punto, llevándolo al mundo del desarrollo del software, no es viable empezar a levantar requerimientos o información sin tener claro la necesidad del usuario. Para este punto es fundamental empezar a aplicar las técnicas de la metodología Design Thinking como el mapa de actores, Toolkit y demás herramientas que permitan abordar este proyecto de una forma más intuitiva y cercana para el usuario final.

Definir: Es la etapa donde se debe preguntar acerca del porqué es importante el proyecto de innovación, comprendiendo desde un punto de vista la posición del usuario, mediante una inmersión tanto cognitiva como técnica, la cual se puede aplicar para entender la realidad de su entorno, permitiendo así «colocarse en los zapatos del usuario» y vivir la problemática de esta perspectiva.

Idear: En esta etapa se puede resolver la problemática planteada, apoyándose de nuevo en las técnicas como lo son el estudio de tendencias y las entrevistas a expertos. En este punto, desde la perspectiva del desarrollo de software, se tiene una similitud por medio de técnica de entrevistas con un ítem diferenciador, puesto que las entrevistas son un mejor contenido en cuanto a herramientas de exploración de la información con aplicación de toolkit, utilizando el material de apoyo como imágenes, videos entre otros. En este ciclo de la metodología ya se le da forma al proyecto, visualizando una posible solución.

Prototipar: El fallar de forma económica y veloz es una premisa en este punto de la metodología como un mecanismo más exploratorio, generando un acercamiento al usuario final con fragmentos de productos reducidos.

Testear: Es la conclusión del resultado del lanzamiento de los prototipos, donde ya se tiene una percepción por parte de los usuarios finales, la clave está en poder iterar el producto para empatizar aún más con el usuario final.

5.4 PROYECTO DE DESARROLLO DE SOFTWARE APLICANDO DESIGN THINKING

A nivel mundial la metodología Design Thinking ha sido utilizada en el ámbito del desarrollo de software por empresas reconocidas y dando gran aporte a las mismas. Por ejemplo, la compañía Apple tiene dentro de su ADN el diseño de sus productos enfocado en las necesidades y gustos de los usuarios, y BBVA aplicó tecnología pensando en el acercamiento con los usuarios y haciendo que sus cajeros fueran más interactivos y útiles en su uso.

A continuación, se nombran cada una de ellas.

- **Apple:** Fue una de las empresas que apostó por la metodología Design Thinking, donde gracias a ellos lograron unificar el diseño, la tecnología y la simplicidad, estos valores se identifican en cada uno de los respectivos productos de la marca.
- **Airbnb:** Una compañía que pasó de ganar 200 euros semanales a ser la millonaria compañía donde aplicaron el concepto de ponerse en los zapatos del cliente para entender sus necesidades reales.
- **BBVA:** En el sector bancario encontramos al banco español que se enfocó en crear cajeros más humanizados que se adaptan mejor a las necesidades de los clientes [14].

5.5 TRABAJOS RELACIONADOS

En los trabajos relacionados con el tema del documento sobre la aplicación de la metodología Design Thinking en proyectos de desarrollo de software, evidenciamos diversas metodologías tradicionales, las cuales no comparten amplias fuentes que nos permitan indagar a profundidad, a continuación, citamos algunas referencias:

<https://misaliados.com.co/>

La Alianza creada entre Bancolombia y Sura para conectar las necesidades cotidianas donde el enfoque se encuentra en proteger al aliado, su trabajo, sus clientes y todo el servicio en general. La idea de *misaliados* va orientada para ofrecer un seguro de riesgos laborales, mientras que, a través de este proyecto se busca ofertar las necesidades cotidianas de una manera más directa hacia el vínculo oferta/demanda, el cual está enfocado en la región del Valle del Cauca, en el nicho de mercado de la ciudad de Cali .

«Mis aliados» busca conectar la oferta y la demanda de diversos servicios, actualmente, esta estrategia está posicionada en la Ciudad de Medellín y se encuentra disponible en ambiente Web, donde el público específico no llega muy fácil por dicho medio.[15].

<https://www.jelpit.com/>

Plataforma web para ofrecer los servicios de oferta y demanda sobre precios establecidos y por turnos de disponibilidad.

«Diseño e implementación de aplicativo móvil para solicitud de clases al instante». Es una plataforma móvil para solicitudes de clases al instante desarrollado en la Universidad Javeriana Cali en el pregrado de Ingeniería de Sistemas y Computación, donde se logró desarrollar la aplicación en cuatro iteraciones validadas con el usuario final logrando así una satisfacción con el usuario final de las necesidades reales y aprendizajes por cada una de las iteraciones[16].

«El Design Thinking aplicado en el desarrollo de un Sistema de Información, permite incrementar la satisfacción de los operarios al reducir los tiempos de atención de Capital Humano». Este trabajo de grado está enfocado en atender las necesidades reales del usuario, aplicando la metodología Design Thinking en el diseño de un sistema de información para incrementar la satisfacción y reducir los tiempos de atención del capital humano[17].

6. DESIGN THINKING Y PROCESOS DE DESARROLLO DE SOFTWARE

Por medio de este capítulo se pretende explicar cómo se adaptan los procesos de Desarrollo de software con la metodología Design Thinking, siendo esta una metodología de innovación. A continuación, se definen los procesos de desarrollo de software y los de Design Thinking.

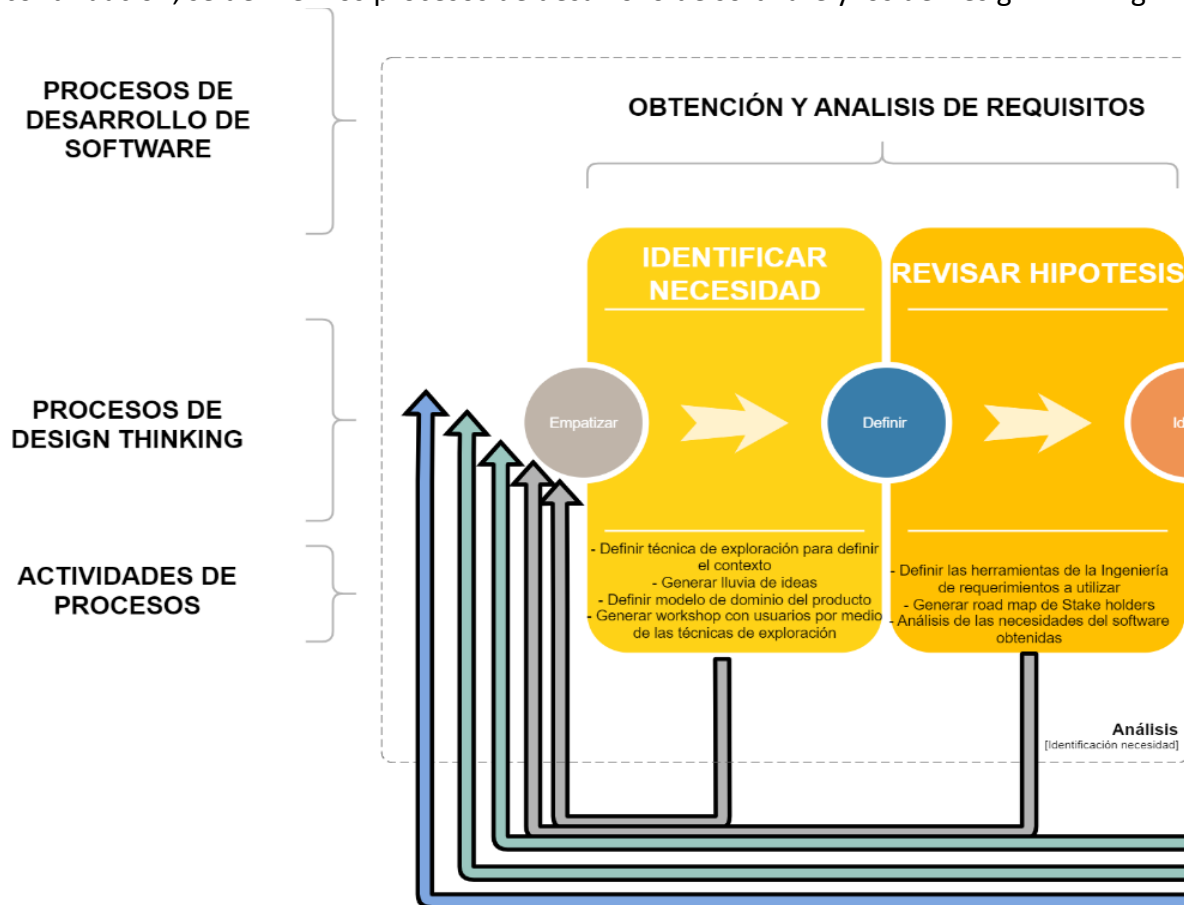


Ilustración 5 Modelo definido de procesos de software y Design Thinking parte 1

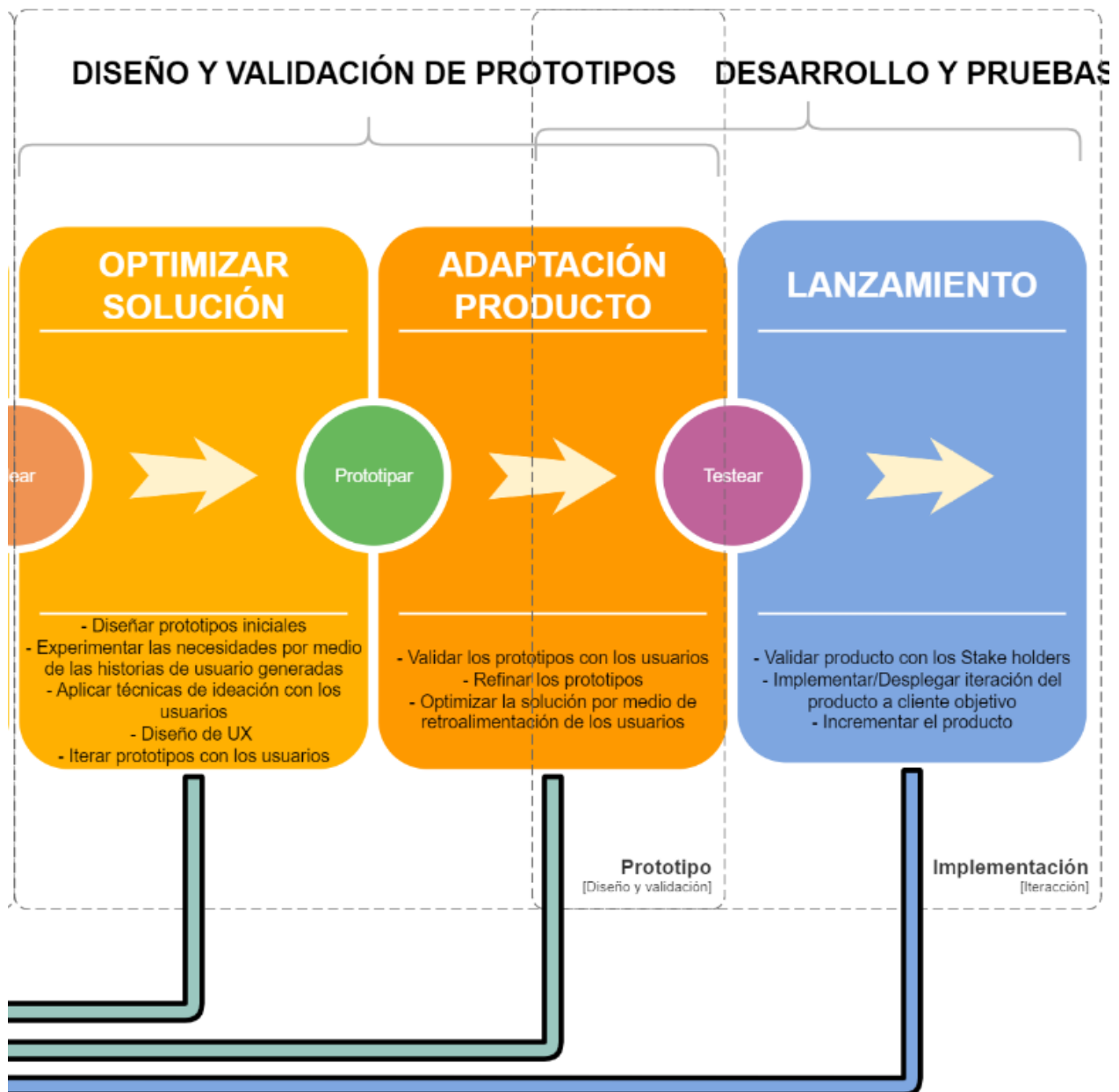


Ilustración 6 Modelo definido de procesos de software y Design Thinking parte 2

De acuerdo con el modelo anterior, se complementan las actividades de los dos mundos de los procesos de desarrollo de software y de la metodología Design Thinking, uno enfocado en el producto tecnológico con lanzamientos cortos y a bajo costo y el otro centrado en el usuario y las técnicas aplicadas por la metodología Design Thinking, todo esto, sin descuidar la sinergia que tienen ambos procesos para operar de manera natural con el modelo definido.

6.1 Adaptabilidad

El enfoque dado por la metodología *Design Thinking* está en construir un producto a bajo costo, aceptado y que se pueda testear por el usuario final, un producto estructurado e interactivo en cada una de sus fases. Los procesos de desarrollo de software y los procesos de la metodología *Design Thinking* tienen una sinergia en cada una de sus etapas, donde comparten conceptos para cumplir un mismo objetivo que es lanzar un producto para satisfacer las necesidades del usuario final. La primera está orientada a productos de software intangibles y la segunda etapa se enfoca en productos tangibles con diseño y estética que aporten a satisfacer las necesidades del cliente, pero independientemente de su enfoque tangible o intangible, las dos etapas se orientan en dos frentes: usuario final y producto con características.

De acuerdo con lo anterior, la empatía entre ambos procesos se da debido a la naturaleza en las fases que tienen. Dentro de sus similitudes, una etapa preliminar de preparar el ambiente por medio de técnicas para el **análisis** que están asociadas a obtención de información, diseño de contextos, generar lluvias de ideas, identificación de los usuarios; otra adaptación entre ambos procesos es buscar el **Prototipado** donde existen dos focos fundamentales que son validar por medio de prototipos, interacción con el usuario final, refinamiento de las necesidades del usuario donde en ambos procesos tanto en desarrollo de software como en *Design Thinking* el foco está en el usuario final y validación del producto a corta escala para lograr optimizar de mejor manera las características principales de un producto, por último y no menos importante es la **Implementación** del producto, si el producto desarrollado está cumpliendo o no con las necesidades requeridas. Esto se logra debido a incremento del producto temprano y a bajo costo, validación del producto dirigido a cliente objetivo y despliegue interactivo.

6.2 Procesos de Desarrollo de software y Design Thinking

Cada uno de los procesos tienen como base la identificación de necesidades, diseño y validación del producto y desarrollo y pruebas de validación. A continuación, se mencionan los puntos más importantes en el aporte del diseño metodológico de cada uno de los procesos.

6.2.1 Análisis

En esta etapa el proceso de *Design Thinking* aporta dos etapas que son la definición y la ideación a través de herramientas como la empatía, definición «ToolBox» con generación de lluvia de ideas, generación de Workshops, generación de RoadMap de Stakeholders, entre otros importantes. Por su parte, los procesos de desarrollo aportan en cuanto a definición y exploración de contexto, generación de modelo del dominio y definición de herramientas de requerimientos. Toma relevancia esta etapa ya que es el punto de partida en la obtención de información donde se transformaron los requerimientos funcionales y no funcionales en oportunidades de innovación, llevando la secuencia adecuada hacia las demás etapas existiendo un involucramiento directo del

usuario para las validaciones a corto plazo.

6.2.2 Prototipo

Durante esta etapa, las metodologías tienen su fundamento principal con los resultados obtenidos basados en un buen diseño y validación del producto con los usuarios finales, con algunas actividades de los procesos como diseño y validación de los prototipos con los usuarios finales, experimentación de las necesidades por medio de las historias de usuario, iteración continua con los usuarios finales.

6.2.3 Implementación

En esta etapa se busca que ambos procesos trasladen la validación del producto hacia el desarrollo de este como algo ya funcional, permitiendo así, que este sea aprobado por el cliente objetivo, generando como punto importante la iteración con desarrollo a bajo costo.

6.3 Retroalimentación y sinergia de los procesos

Para que los procesos no se comporten de manera estática, se busca una retroalimentación del ciclo de vida, donde escalan el producto de manera corta y a bajo costo; para el caso de *Design Thinking*, se generan interacciones cortas por cada subproceso y como pieza clave involucramiento del usuario final en cada una de ellas. Por consiguiente, los procesos de desarrollo de software trabajan con sinergia en cada una de sus etapas enfocadas en la validación del producto de software y aceptación de las necesidades con el usuario final, este punto es factor fundamental en el modelo para la disminución de errores que estos generan al final del proyecto ahorro en costos.

Como se observa en este capítulo, se propuso un modelo que integra de forma adecuada el *Design Thinking* y los procesos de desarrollo de software de manera estructurada con artefactos que entregan suficiente información, optimizable por prioridad de requerimientos, validaciones con el usuario final por cada fase, iteraciones cortas y baratas de fallar generando un valor agregado al usuario.

7. CONTEXTO DEL PROYECTO A DESARROLLAR

Por medio de la metodología Design Thinking se pretende validar cómo con proyectos de desarrollo de software se adapta en la aplicabilidad de la metodología, donde en el contexto del proyecto se pretende aplicar en el sector económico de oferta y demanda de servicios de empleo, para oficios varios en la ciudad de Cali, en donde de acuerdo con el DANE, existe un índice de informalidad del 49,1 % al año 2021.

Por otro lado, nos encontramos en una época de cambio exponencial basado en la definición del siglo VUCA [2], el cual analiza la volatilidad, incertidumbre, complejidad y ambigüedad, es por ello la importante de introducir herramientas que ayuden con el pensamiento del diseño y la solución tecnológica con procesos de desarrollo de software que se adapten de manera natural.

Partiendo de la siguiente premisa de la Metodología *Design Thinking*:

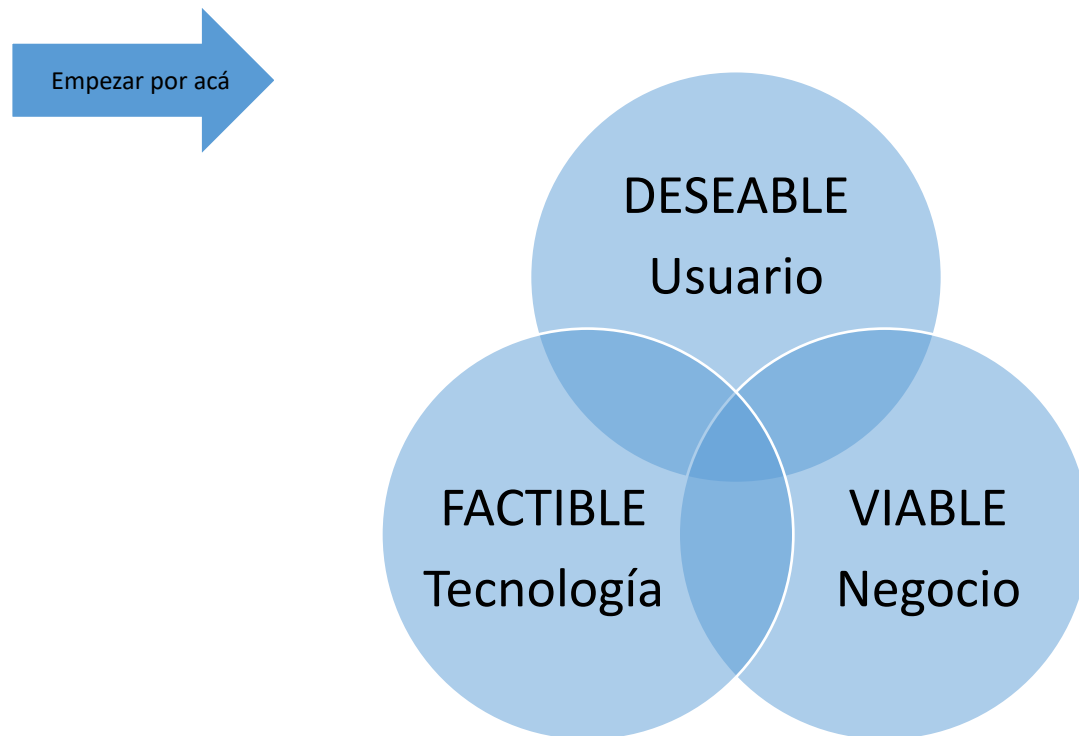


Ilustración 7 Que va primero Idea o Usuario

De acuerdo con la Ilustración anterior, es importante indagar por dónde empezar y cómo se va a llevar a cabo la validación de la metodología para analizar cómo se adapta a los proyectos de software. Lo primero es comenzar por lo deseable, es decir, lo que requiere el usuario para encontrar el problema y pensar en la posible solución.

El proyecto inició buscando generar empatía con el usuario final, por medio de entrevistas de profundidad, con el objetivo de obtener la información correspondiente de lo que requiere el usuario, después de ello, se continuó con el desarrollo de un prototipo de software enfocado en ofertas de servicio y demanda para oficios varios, del mismo modo, la validación del prototipo con respecto al cómo la metodología se adapta a los procesos de desarrollo de software.

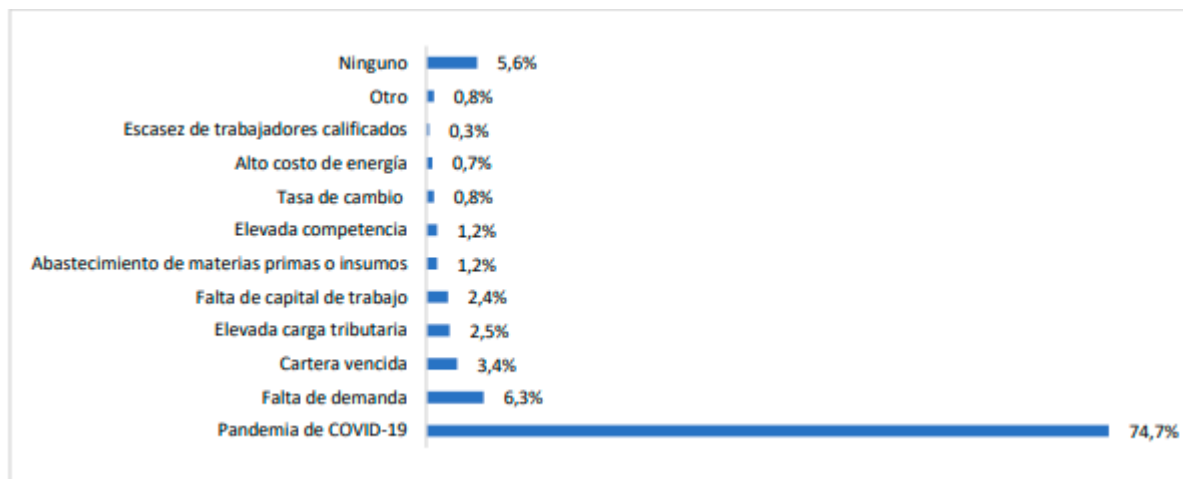


Ilustración 8 Indicadores de desarrollo de actividades por cámara y comercio Cali [18]

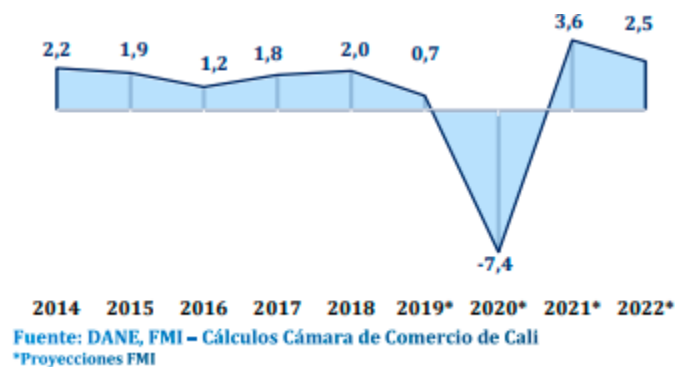


Ilustración 9 Indicadores comportamiento por Cámara y comercio Cali [18]

De acuerdo con los indicadores de comportamiento, existió falta de demanda para el desarrollo de actividades.

Por otro lado, es un gran reto desarrollar el trabajo de grado aplicando la metodología *Design Thinking* con los procesos de desarrollo de software para validar la adaptabilidad entre ambos.

Para ello, se pensó en una solución innovadora como prototipo basado en el contexto del problema, aplicando el modelo definido con procesos de la metodología *Design Thinking* y de desarrollo de software, después de ello se aplicaron cada uno de los artefactos que pertenecen a la fase de análisis, prototipo e implementación; con esto se llevó la información a herramientas de comparación, análisis de resultados, priorización de requerimientos y diseño del sistema con un alcance definido. Por consiguiente, se implementó y se lanzó el producto para que pueda ser testeado por el usuario objetivo, donde se obtuvo retroalimentación y fallo del producto a bajo costo, teniendo en cuenta lecciones aprendidas y ajustes en el proceso.

Se proyecta introducir el prototipo desarrollado en el mercado con un enfoque «marginal», mejorando la propuesta de valor actual teniendo en cuenta el costo y el funcionamiento como valor diferencial centrado en el usuario final, donde los usuarios se puedan sentir identificados con la solución desarrollada.

8. OBTENCIÓN Y ANÁLISIS DE REQUISITOS APLICANDO DESIGN THINKING

Este capítulo está enfocado en la obtención y análisis de requisitos aplicando la metodología *Design Thinking*. Se describen las diversas estrategias por medio de las herramientas que implementa la metodología, el análisis correspondiente para la obtención de la información de los usuarios, la definición del contexto general, entre otros factores importantes.

Es importante tener en cuenta al momento de obtener la información empatizar con el usuario, diseñar las correspondientes entrevistas de profundidad hacia los mismos, identificar los usuarios extremos, indagar con expertos donde estas historias contadas deberán transformarse en oportunidades que permitan organizar la información para encontrar el problema. A continuación, se ilustra cómo conectar con el usuario por medio del diseño de las entrevistas.

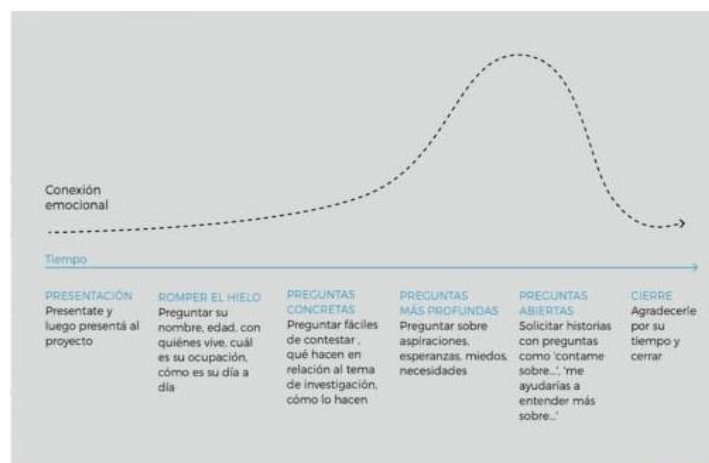


Ilustración 10 Conexión emocional [2]

La conexión emocional con el usuario es una de las fases más importantes para lograr resolver sus problemas por medio de las emociones de manera divergente, utilizando preguntas concretas, profundas y abiertas, para llegar a un **sesgo de confirmación** o generar que la información sea compatible con el resultado de los entrevistados.

Los procesos de desarrollo de Software planteados para la obtención y análisis de requisitos, donde se debe realizar técnicas de exploración para definir contexto como las entrevistas, la definición de framework para organización de información y la identificación de los usuarios.

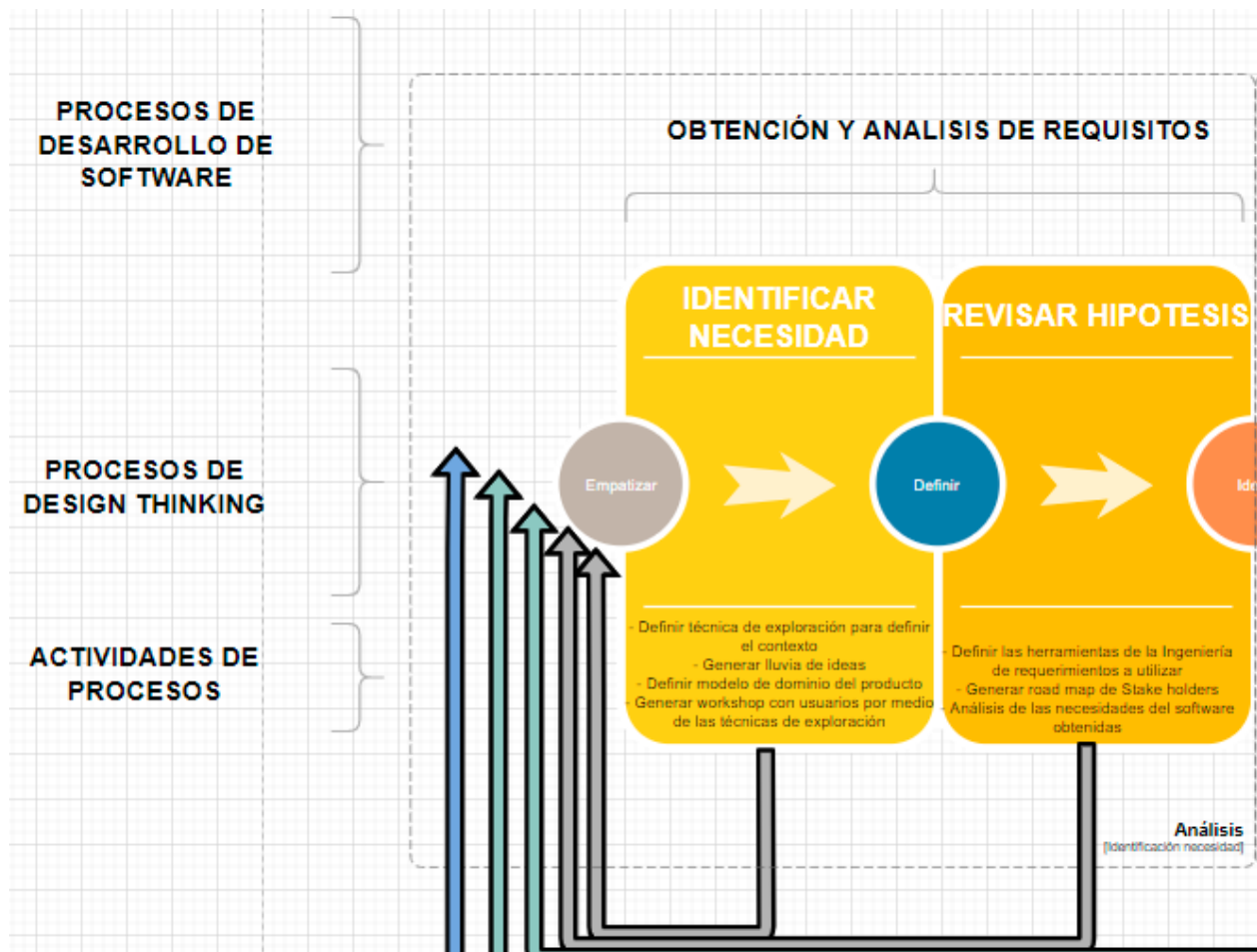


Ilustración 11 Obtención y análisis de requisitos

La primera etapa de análisis consolida conceptos como la exploración de información, la empatía con el usuario, la definición del modelo de Dominio del sistema y el análisis de necesidad por medio de lenguaje natural. Esta es una de las etapas más importantes definidas en el modelo propuesto, puesto que marca la pauta en términos de *Insights* baratos, priorizaciones, bajo costo, información de primera mano del usuario, entre otras importantes que se van a ir desarrollando en el capítulo.

8.1 Artefactos para obtención de requerimientos

Se diseñaron entrevistas para la obtención y el análisis de requerimientos enfocadas en diferentes usuarios como objetivo, extremos y expertos. Gracias a esta metodología, se permite explorar la conexión entre la oferta y la demanda centrado en el usuario, el agrupamiento y análisis de la información y la priorización de requerimientos.

Para los usuarios objetivos, los cuales ofrecen servicios informales, se aplicó una encuesta, arrojando los siguientes resultados relevantes. Ver el Anexo 1, «Entrevista de profundidad».

6. La ocupación en la que te desempeñas ¿Cómo es el flujo de demanda por parte de los clientes?

[Más detalles](#)



Ilustración 12 Resultado flujo de Demanda de clientes

8. ¿Cuál es la plataforma de comunicación por el que te contactan?

[Más detalles](#)

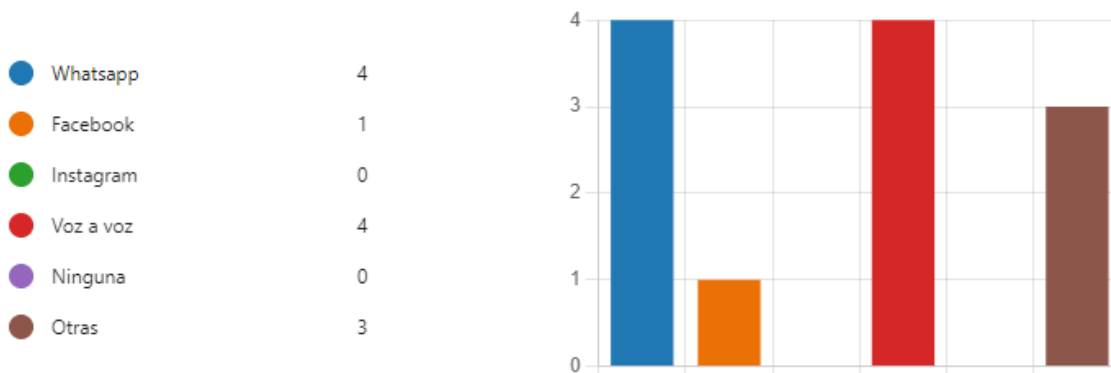


Ilustración 13 Resultado plataformas de comunicación utilizadas

9. ¿Cuál de las siguientes alternativas consideras lo difícil de adquirir clientes?

[Más detalles](#)

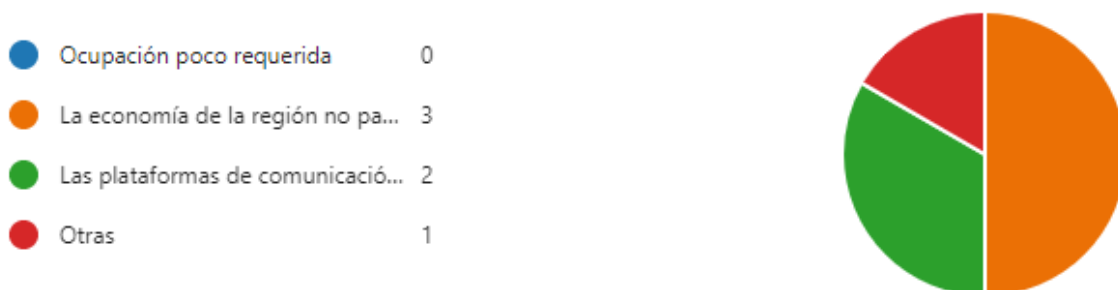


Ilustración 14 Resultado de adquisición de clientes

11. En este mundo de adaptación al cambio, ¿Consideras que tu ocupación debe dar un siguiente paso para aumentar la demanda de los clientes?

[Más detalles](#)



Ilustración 15 Resultado de adaptación al cambio de conectar oferta y demanda

12. ¿Ayúdanos a mencionar soluciones similares a la planteada para los servicios de oferta y demanda?

[Más detalles](#)

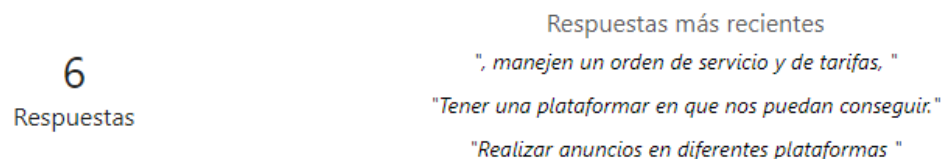


Ilustración 16 Resultado recomendaciones

Con un promedio de edad de 53 años sobre los encuestados, estos se dedican a profesiones de oficios varios con un flujo de demanda medio. En cuanto a la ocupación de su trabajo, se indica que existe un nicho de mercado con ocupación media para que pueda ser potenciado con un valor diferencial que aporte en el sector; por otro lado, los usuarios objetivos manifiestan necesidades como dar a conocer el trabajo realizado por medio de una herramienta efectiva, con adaptación al cambio para aumentar la demanda de los clientes, una plataforma donde se logren reconocer de manera fácil costos de los servicios ofrecidos y recomendaciones como necesidades importantes a considerar. Ver el Anexo 1 para más detalle.

De acuerdo con los anteriores resultados más significativos, los usuarios objetivo manifiestan una tendencia a la necesidad de conectar la oferta con la demanda por medio de una plataforma sencilla y efectiva que pueda vincularse con un valor diferencial, supliendo las necesidades de los usuarios por medio del lanzamiento de prototipos baratos e interactivos, probando por medio de validaciones de los usuarios hasta lograr madurar el producto.

Por otro lado, se logró diseñar y aplicar una entrevista con un experto en el área del *Design Thinking*, quien hace parte de la incubadora en la Universidad Javeriana Cali, Diana Isabel Riveros Ospina como Directora de Innovación y emprendimiento. De acuerdo a la entrevista se extrae la siguiente información de manera importante que da aporte al tema de investigación.

Indagación	Respuesta
¿Cómo observa la economía en el Valle del Cauca?	De acuerdo con un estudio realizado por el GEM la economía cayó generando más desempleo, donde la misión como Universidad es luchar contra todos estos factores adversos.
¿Cómo ves la informalidad en la región del Valle del Cauca?	Es gigante en la región, donde allí existe un potencial de mercado en oferta y demanda.
¿Cómo crees que se puede ir involucrando una plataforma en una primera fase?	No pensarlo mucho desde el punto de vista de monetizar desde el inicio, sino en conectar oferta con demanda y encontrar ese factor diferencial que no ofrecen las muchas plataformas existentes. Es sobre pensar cómo atraer muchos usuarios y ver un modelo de negocio con background y no una plataforma como negocio.
¿Teniendo en cuenta algunos procedimientos o Frameworks para llevar a cabo un proyecto que involucre al usuario, necesidad, solución tecnológica que recomendarías?	Recomiendo validaciones baratas, cosas rápidas, bitácora de aprendizaje, implementar rápidamente esas validaciones hasta llegar a un punto de lanzar el producto. Mucha interacción con el usuario y centrarse en él es lo fundamental.

Tabla 1 Información relevante entrevista experto

De acuerdo con la anterior tabla resumen, con la información más relevante de la entrevista con el usuario experto, se logra obtener información valiosa desde el punto de vista técnico, metodológico, enfoque de necesidad y mercado, consejos de diseño de aplicaciones que los usuarios solicitan en el mercado.

Se logran captar varios puntos con el usuario experto donde se profundiza en conceptos del cómo iniciar con una plataforma de este estilo, donde la recomendación es captar la mayor cantidad de información y de usuarios con un modelo gratuito, siempre centrar los prototipos y validaciones con el usuario para que se puedan sentir identificados con el producto a lanzar, es interesante el problema a resolver donde existe una oportunidad de lograr conectar oferta con demanda de manera potencial. Ver Anexo 5 para el detalle de la entrevista.

8.2 Resultados obtenidos del trabajo de campo

Los resultados del trabajo de campo se realizan por medio de las entrevistas, para identificar los patrones de conducta y extraer los *Insights* que generen sentido por medio de los patrones, para orientar las decisiones. A continuación, se presenta la obtención de información captada de los artefactos de entrevistas.

Hechos: <i>Datos relevantes para entender el contexto del usuario</i>	Aspectos Negativos: <i>Puntos de dolor o situaciones no deseadas</i>	Aspectos positivos: <i>Puntos de felicidad o situaciones deseadas</i>	Atajos: <i>Soluciones informales a problemas actuales</i>	Ideas: <i>Soluciones que los usuarios proponen</i>	Casos: <i>Ejemplos o casos similares</i>
<ul style="list-style-type: none"> - Edad promedio 53 años. - Flujo de demanda media y alta. - Ocupación de encuestados de oficios varios, transportistas, maestros obra. - Encuestados con nucleo familiar conformado. 	<p>Poca información con respecto al día a día de los encuestados. Economía de la región no pasa por buen momento.</p>	<ul style="list-style-type: none"> - Los encuestados manejan sistemas de comunicación. Los encuestados consideran un cambio implementar una plataforma que ayude en la oferta de servicios. 	<ul style="list-style-type: none"> - Comunicación voz a voz. - Manejar una orden de servicio y tarifas. - Recomendación de clientes. - Los encuestados utilizan mayormente Android. 	<ul style="list-style-type: none"> - Realizar anuncios en diferentes plataformas. - Diversificar el trabajo en una plataforma. - Calificación de los servicios realizados. - Chat de conversación. 	<ul style="list-style-type: none"> - Los encuestados manifiestan dar a conocer el trabajo realizado. - Una plataforma que los ayude a encontrar clientes. - Gran porcentaje de los encuestados utilizan el Whatsapp y el voz a voz.

Tabla 2 Resultados trabajo campo

Con respecto a la anterior tabla, la agrupación de información de las encuestas de profundidad aplicadas se categorizan para que se puedan aplicar patrones de información y así lograr analizar dicha información como datos relevantes, tales como las ideas de los usuarios, los aspectos, los casos reales de aplicabilidad en el contexto y las soluciones informales donde dicha recopilación se convertirá en Canvas de patrones de información de manera consolidada y clarificada.

8.2.1 Técnicas de ideación aplicadas

De acuerdo con la información recolectada de las encuestas aplicadas, se pretende agrupar la información en patrones que ayuden a detectar oportunidades para desarrollar la mejor solución con respecto a las necesidades manifestadas por los encuestados, todo ello por medio del componente tecnológico como producto tangible de solución.

Canvas (Patrones de información): Técnica que ayuda a categorizar la información de acuerdo a las prioridades y conductas obtenidas[19].



Ilustración 17 Canvas Hallazgos

- **Lluvia de ideas:** Utilización de técnica de lluvia de ideas con grupo de personas colaboradoras, para generar nuevas perspectivas sin perder el foco de la información[19].
- **Canvas de negocio:** Es la etapa en la cual se define la empatía de lo que se pretende proponer al usuario bajo tres conceptos: necesidad, beneficio y valor diferencial en el mercado[19].

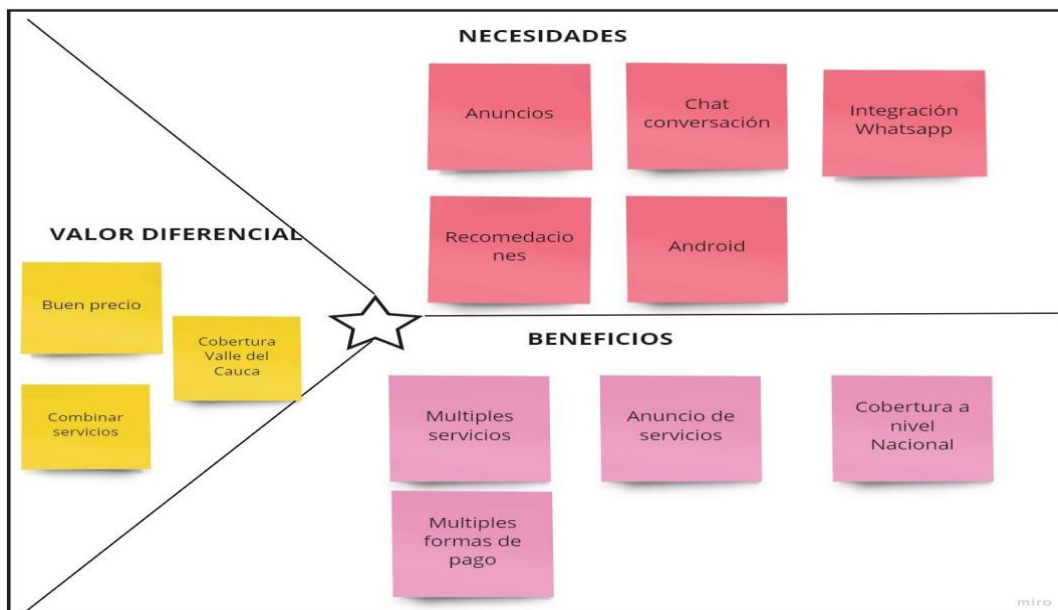


Ilustración 18 Canvas de Negocio

- **Benchmark:** Es la práctica por el cual se observan las tendencias para inspirar la propia idea a desarrollar, buscando la mayor cantidad de información posible y referenciando casos de éxito y fracasos de la idea a desarrollar.

Atributos	Hogaru	Jelpit	MisAliados
Buen precio	O	O	O
Página Web	X	X	X
Aplicación móvil	X	X	O
Cobertura Valle del Cauca	O	O	O
Cobertura Nivel Nacional	O	O	O
Todos los servicios (Jardinería, Transporte, Hogar, Instalaciones, etc)	O	O	O
Combinación de servicios	O	O	O
Múltiples forma de pago (En línea, efectivo, Transferencias)	X	X	X
Anuncios de servicios	O	O	O

Tabla 3 Benchmark comparativo atributos y aplicaciones existentes

O = No cuenta con el atributo

X = Cuenta con el atributo

De acuerdo con la tabla anterior, se identifican oportunidades para generar un factor diferencial en el mercado de las fortalezas y debilidades de las plataformas actuales, teniendo en cuenta la información obtenida de los usuarios, se logra evidenciar una preferencia por las plataformas con bajo costo, sencillas y con múltiples servicios.

8.3 Análisis de requerimientos

El análisis y obtención de los requerimientos es parte fundamental del proceso de la Ingeniería de Software, donde se logran identificar las zonas grises, los requisitos definidos y los procesos relacionados al proyecto que ayuden a su respectiva construcción. A continuación, se menciona el contexto del dominio donde se define la visión general del proyecto.

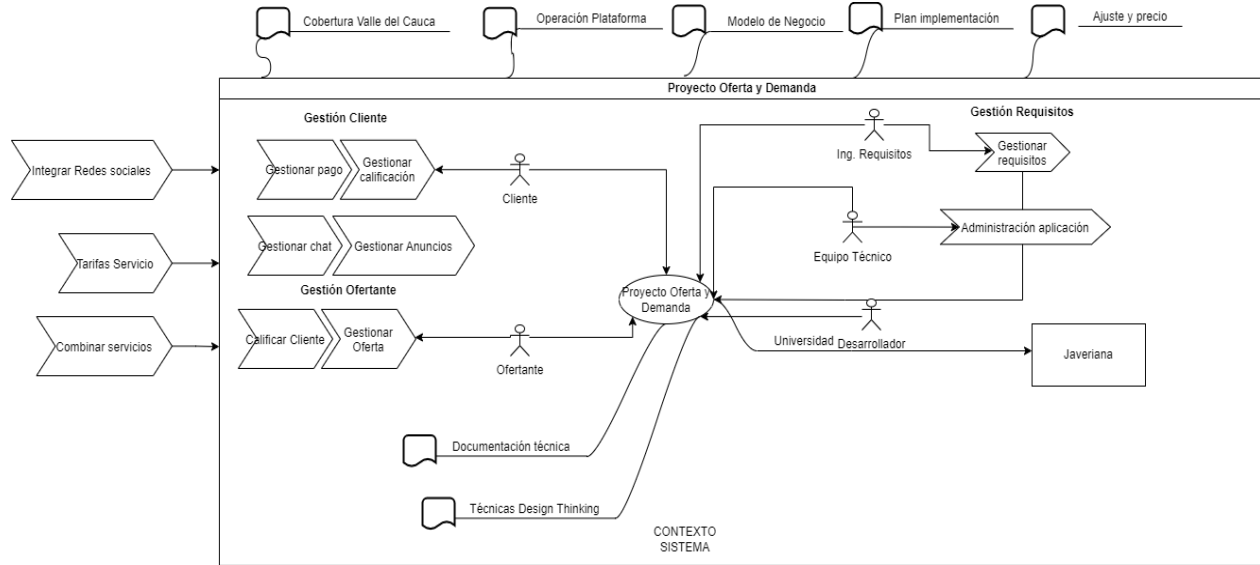


Ilustración 19 Contexto Dominio Sistema

De acuerdo con la Ilustración anterior, se define el contexto del dominio del sistema con respecto a la información obtenida de las entrevistas asociadas, para ello se tienen en cuenta los diferentes actores, entidades y documentación asociada. De igual manera, se identifican las zonas grises o zonas que no son tan claras en el primer acercamiento de interacción en el contexto, como por ejemplo: funcionalidades de tarifas de servicios, modelo de negocio e integraciones que se encuentran por fuera del contexto del sistema.

8.3.1 Diagrama de Caso de Uso general del sistema

A continuación, se describe el caso de uso general del sistema, de acuerdo con la información obtenida por las técnicas de creatividad y las respectivas encuestas y entrevistas.

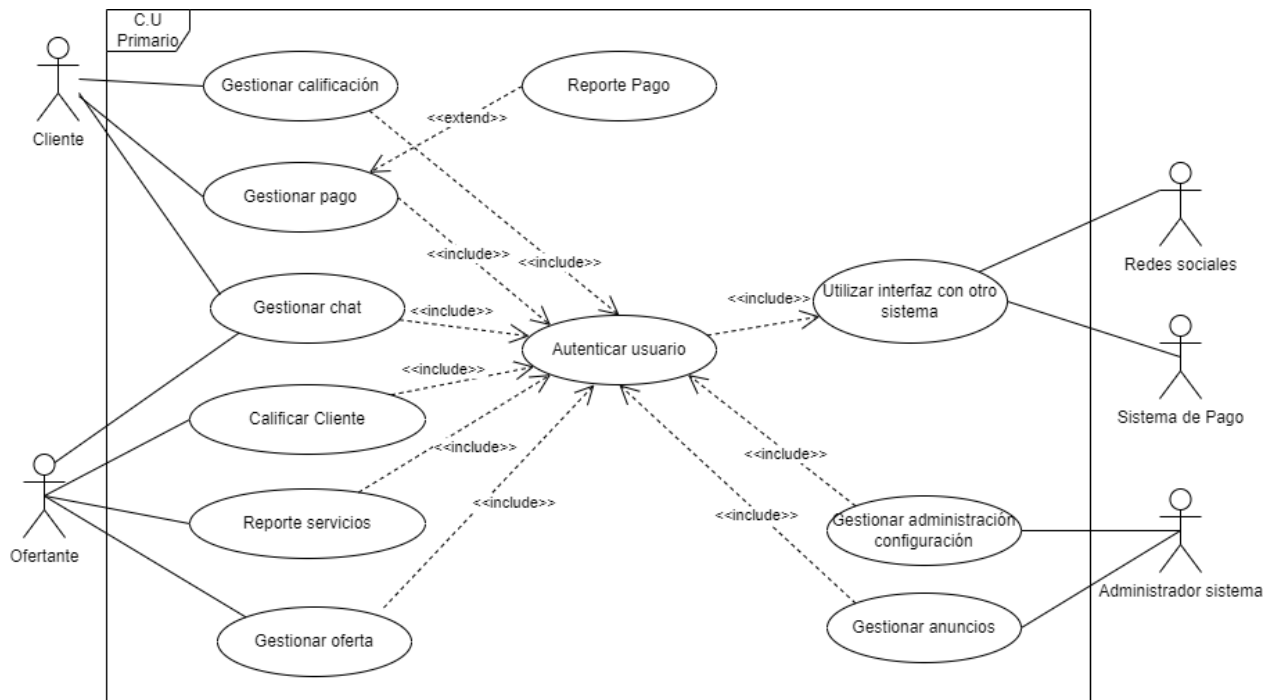


Ilustración 20 Caso de Uso general del Sistema

De acuerdo con la Ilustración anterior, se define el diagrama de Caso de Uso principal del sistema, identificando las principales funcionalidades del sistema obtenidas de las técnicas de *Design Thinking* como *Workbench*, entrevistas, encuestas, documentación existente de sistemas actuales que hacen parte del tema en investigación; con respecto al diseño propuesto de adaptar los procesos de ingeniería de software, con la metodología *Design Thinking* se saca mejor provecho en optimizar una mejor solución cubriendo las necesidades del usuario final que fueron plasmadas en las actividades descritas con anterioridad.

8.3.2 Definición de requerimientos

- **Definición de requerimientos funcionales:** En el Anexo 2, se identifican las diferentes funcionalidades y alcance del sistema en términos de funcionamiento. De acuerdo con lo anterior, los requerimientos se describen en lenguaje natural para que pueda ser interpretado por cualquier usuario sin un lenguaje técnico avanzado. A continuación, se describen los requerimientos funcionales del proceso principal del sistema desarrollado.

CATEGORIA	PROPUESTA REQUERIMIENTOS REFINADOS	1. Priorización in/out	2. Priorización tres niveles (A = Alta, M = Medio, B = Baja)
RF-COMUNICACION-001	El sistema deberá permitir chat o conversación sobre los servicios ofrecidos a los diferentes usuarios del sistema (Cliente, Ofertante)	IN SECCION A Y B	M
RF-SERVICIOS-003	El sistema deberá permitir anunciar los servicios por medio de diferentes filtros como mejor servicio calificado y precio más solicitado.	IN SECCION A Y B	M
RF-USABILIDAD-006	El sistema deberá desplegar la información o detalle de cada uno de los servicios ofertados como el precio, descuentos, características y ubicación geográfica.	IN SECCION A Y B	A
RF-SERVICIOS-004	El sistema deberá permitir al usuario cliente buscar la oferta que requiera utilizar de sus servicios y abrir contacto con el usuario ofertante.	IN SECCION A Y B	A
RF-SERVICIOS-005	El sistema deberá permitir al usuario cliente solicitar un servicio de acuerdo a los criterios de búsqueda.	IN SECCION A Y B	A
RF-SERVICIOS-006	El sistema deberá permitir al usuario ofertante reservar el servicio solicitado por el usuario cliente y no estar disponible para otro servicio hasta liberar el servicio.	IN SECCION A Y B	A
RF-SERVICIOS-007	El sistema deberá permitir al usuario cliente cuando solicite el servicio desplegar los diferentes medios de pago disponibles en la plataforma, sin este paso no es posible finalizar la transacción.	IN SECCION A Y B	A
RF-SERVICIOS-008	El sistema deberá permitir al usuario cliente declinar la transacción del servicio donde el servicio quedará disponible de nuevo para que puede ser tomado por otro usuario ofertante.	IN SECCION A Y B	A

RF-SERVICIOS-011	El sistema deberá permitirle al usuario recibir el código de la reserva cuando se confirme la transacción en el sistema, puesto que aquí finaliza el último proceso del medio de pago.	IN SECCION A Y B	A
------------------	--	------------------	---

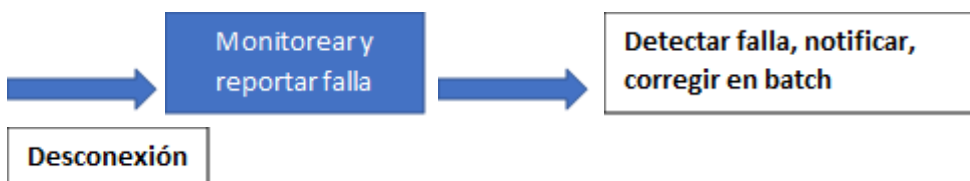
Tabla 4 Definición de requerimientos funcionales

Los requerimientos hacen referencia al flujo principal de la aplicación desarrollada, donde cuentan con la prioridad alta y media en desarrollo, cubriendo las necesidades principales del usuario con lo mínimo viable con la validación y testeo en el lanzamiento del producto. Los requerimientos fueron redactados en lenguaje natural, posteriormente, se realizó la depuración de la información repetida, hasta llegar a una idea concreta y estructurada.

- **Definición de requerimientos no funcionales:** En el Anexo 3, se definen los requerimientos no funcionales o atributos de calidad, con un lenguaje natural y de manera estructurada, teniendo en cuenta estímulos, respuesta, artefacto y medición de respuesta. A continuación, se mencionan las tácticas de arquitectura por las cuáles se resuelven cada uno de los requerimientos no funcionales.

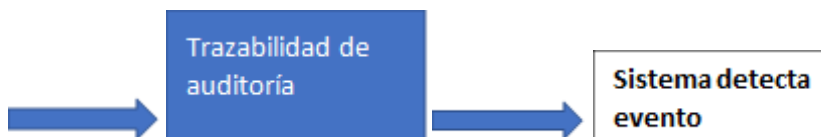
- **Disponibilidad**

- **Ping/Echo:** Táctica de arquitectura que permite mantener la medida de respuesta del sistema y detectar las fallas correspondientes, lo que garantiza que el sistema deberá auto mantenerse en los tiempos adecuados.



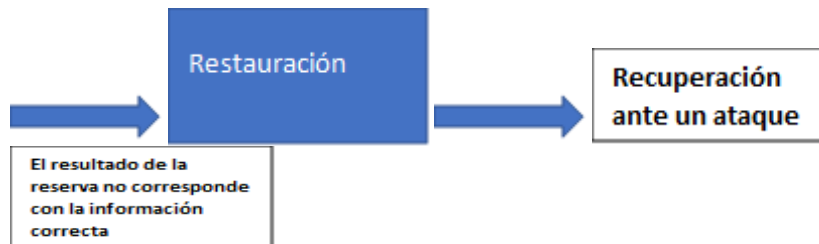
- **Seguridad**

- **Trazabilidad de auditoría:** En caso de un ataque o un evento inesperado se deberá tener una trazabilidad de auditoría, detectar y notificar el problema.



Un dispositivo produce un evento inesperado

- **Restauración:** Una de las tácticas principales es lograr que el sistema en tiempos de caída sea autónomo y tolerante a fallos.



- **Despliegue**

- **Canary:** Por medio de esta táctica se pretende realizar actualizaciones controladas de despliegue del sistema por grupos específicos de usuarios o terminales.

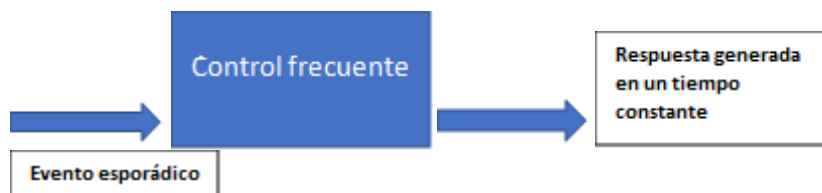


- **Continuos Deployment:** Esta táctica ayuda en la liberación continua de versiones en el ambiente productivo, con la integración adecuada del código y testeado de los módulos.



- **Rendimiento**

- **Control frecuente:** Por medio de esta táctica se pretende tener un control constante del sistema, donde no se vean afectadas las funcionalidades y puedan ser restauradas rápidamente.



- **Definición de requerimientos tipo restricción:** Se definen los requerimientos y tipo restricción a mapear las posibles limitantes referente al alcance, tiempo o costo del proyecto que puedan afectar el desarrollo del mismo.

CATEGORÍA	PROPUESTA DE RESTRICCIONES
REST-COSTO-001	El sistema será desarrollado con un presupuesto máximo de \$1'000.000 patrocinado por el estudiante del trabajo de grado y teniendo en cuenta cero costo de mano de obra en la ingeniería de requisitos y de desarrollo de software.
REST-TIEMPO-001	El sistema será desarrollado en dos etapas: 1. Enero 2022 a Junio 2022: La primera fase considera la definición de la arquitectura, ingeniería de requisitos, aplicación de Metodología Design Thinking. 2. Julio 2022 a Noviembre 2022: La segunda fase considera el desarrollo y retroalimentación del producto con los usuarios finales.
REST-ALCANCE-001	El sistema deberá enfocarse exclusivamente al proceso de servicios de oferta y demanda como un prototipo del sistema, demostrando la aplicabilidad de la metodología <i>Design Thinking</i> en los procesos de desarrollo de software.

Tabla 5 Definición requerimientos restrictivos

8.3.3 Diseño del sistema

En esta sesión del diseño del sistema, la intención es representar el detalle técnico con los flujos principales y arquitectura, para entender de manera gráfica las acciones, los usuarios involucrados, los proveedores requeridos y el contexto del sistema, logrando llevar el análisis obtenido a un lenguaje visual de manera clara, sencilla y precisa, permitiendo ejecutar las actividades de los procesos de software definidos en el modelo propuesto.

- **Diagrama de actividades**

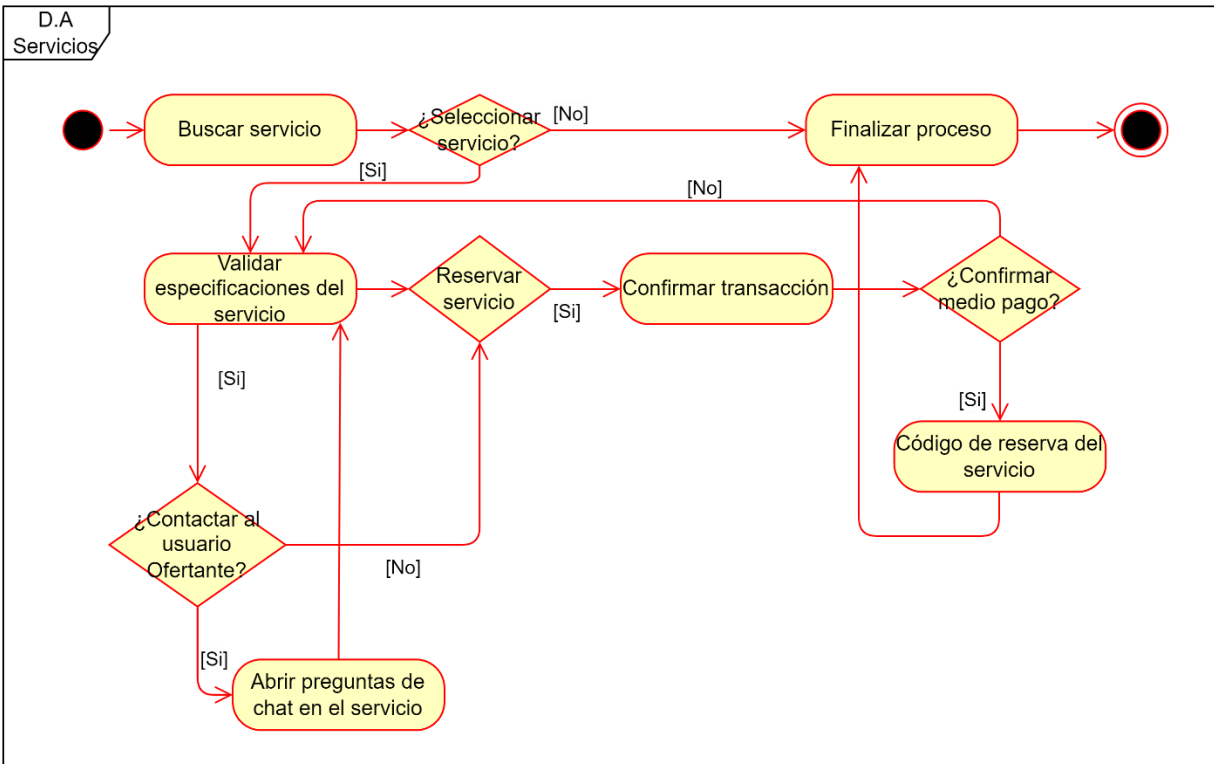


Ilustración 21 Diagrama de procesos de aplicativo

La Ilustración anterior, define el Diagrama de actividades principal del proceso núcleo del sistema a diseñar. Aquí se describen los procesos definidos como la búsqueda de un servicio, los controles de validación en el momento de separar un servicio, inclusive la declinación de este si no fuese posible confirmar la transacción. De manera gráfica y en forma estática se explica el flujo del proceso de cómo pasan los servicios ofertados y la interacción entre los usuarios.

- **Diagrama de Arquitectura**

El modelo de Arquitectura C4, permite plasmar de forma más intuitiva la información para usuarios técnicos como no técnicos.

Este diagrama se compone de 4 capas: Contexto, Contenedor, Componente, Código donde cada nivel tiene un propósito diferente y más detallado llevándolo cada vez a la minucia técnica.

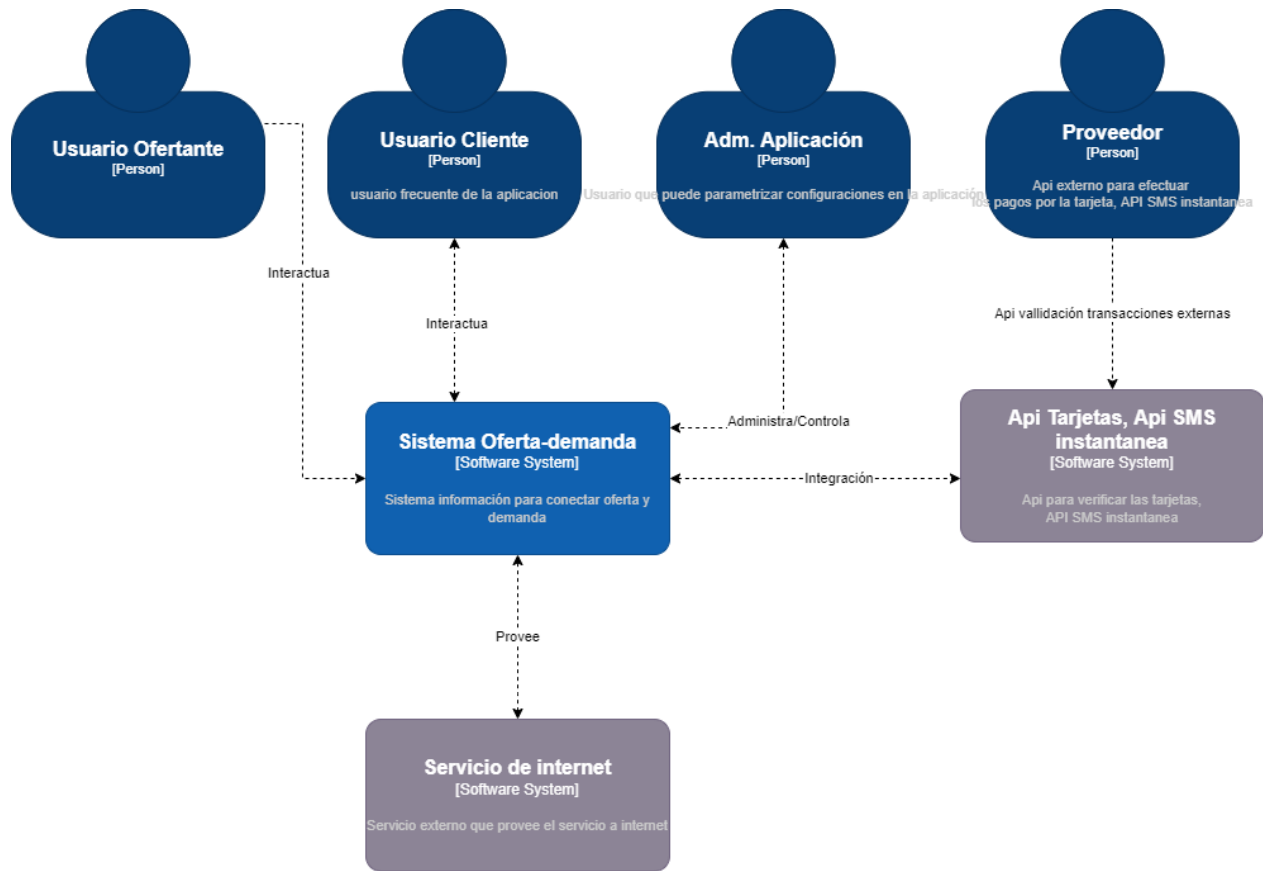


Ilustración 22 C4-Contexto

La Ilustración anterior nos indica que se identifican cuatro actores en el sistema, desde usuarios de aplicación hasta usuarios como proveedores externos, integración con el sistema de oferta y demanda para los pagos electrónicos y SMS instantánea. Por otro lado, y no menos importante, se requiere el servicio de internet para que el sistema pueda funcionar.

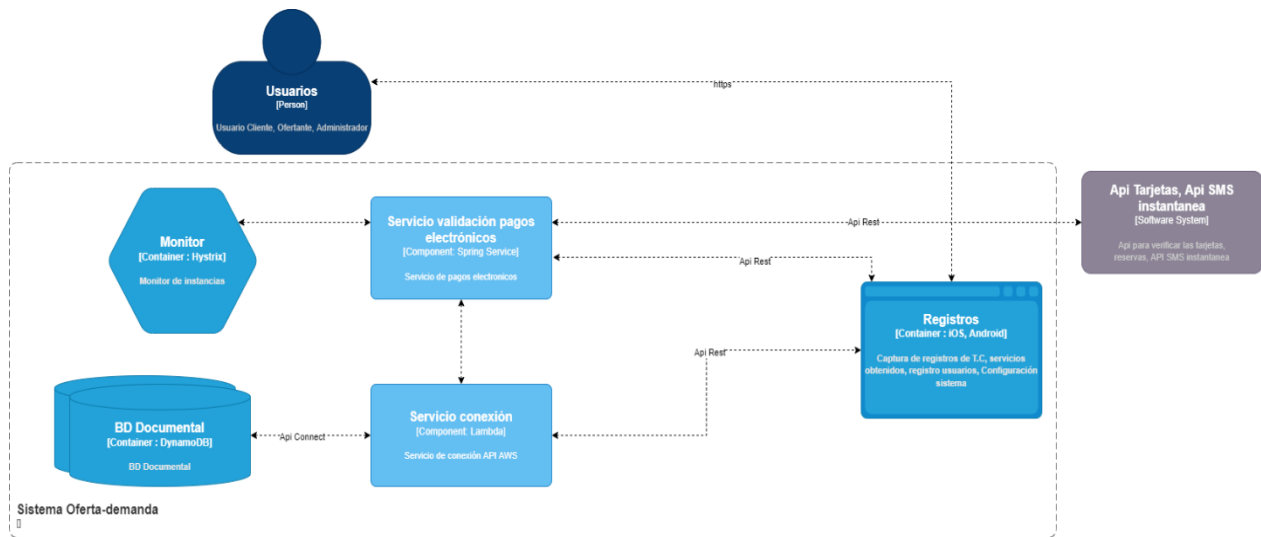


Ilustración 23 C4 – Contenedor

De acuerdo con la Ilustración anterior, llevando a un nivel más detallado, se identifica que se requiere un motor de base de datos documental, de acuerdo con los casos de uso implementados en la industria del software, lo ideal para este caso es DynamoDB, puesto que permite cargar información multimedia en tiempo real con alta concurrencia, del mismo modo, ejecuta de manera ágil y práctica el despliegue de la información a través de dispositivos móviles.

9. VALIDACIÓN DE PROTOTIPACIÓN ENFOCADO EN LA ADAPTACIÓN DEL PRODUCTO

En el modelo propuesto para los procesos de desarrollo de software y *Design Thinking* para comenzar a realizar diseño, validación de prototipos y producto basado en la experimentación aplicando técnicas de ideación, es consecuente tener las anteriores etapas validadas por medio de diferentes técnicas, tales como definición del contexto, entrevistas, definición y priorización de requerimientos, entre otros importantes mencionados en el capítulo anterior.

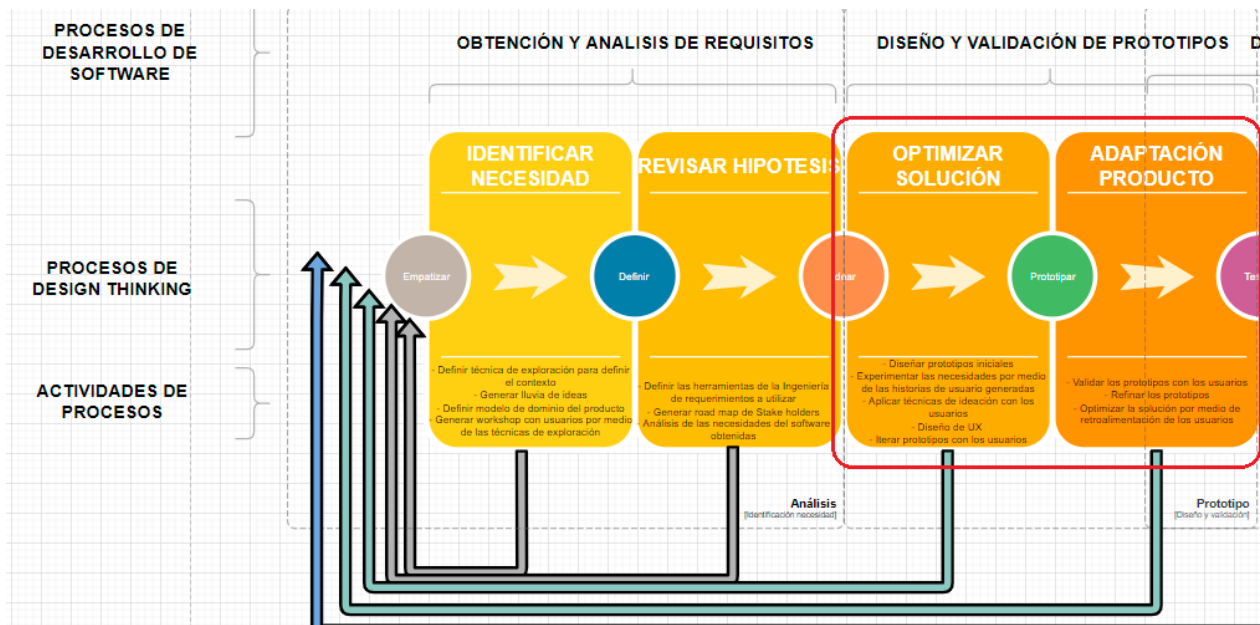


Ilustración 24 Proceso prototipación modelo definido

De acuerdo con la Ilustración anterior, se evidencia en el modelo propuesto el enfoque en el diseño y prototipado con iteraciones cortas, para validar rápidamente con el usuario objetivo, sin tener muy desarrollado interfaces de usuario o un sistema para mostrar al usuario, sino basado en el refinamiento de prototipos validados por los usuarios, para sacar un Insights cada vez más maduros.

9.1 Prototipado

El concepto de prototipado en el modelo definido tiene un punto importante que es diseñar barato y rápido para enfocarse en la validación con el usuario objetivo. Ver Anexo 4, donde se ilustra la prototipación de acuerdo con las técnicas de ideación aplicadas por medio de la metodología diseñada, acercando los dos mundos de los procesos del desarrollo del software y el Design Thinking de acuerdo con las necesidades de los usuarios que brindaron información suficiente para llegar a dicho acercamiento.

9.2 Validación prototipación

En esta parte del proceso se debe tener en cuenta lo recolectado en las etapas anteriores: análisis de información, Benchmark y Canvas, para que finalmente se pueda convertir dicha información en requerimientos en un lenguaje natural y se pueda adaptar a un producto de software.

De acuerdo con lo dicho anteriormente, se debe comenzar a realizar diseño, validación de prototipos y producto basado en la experimentación, aplicando diversas técnicas para lo cual se aplicó una encuesta a usuarios de nicho de oficios varios, como empleadas de servicios, transportadores, entre otros, obteniendo los siguientes resultados:

3. Validación de prototipo para el Inicio de sesión. ¿Qué opinas de la interfaz de la pantalla? (0 punto)

[Más detalles](#)

● Buena	6
● Regular	1
● Mala	0

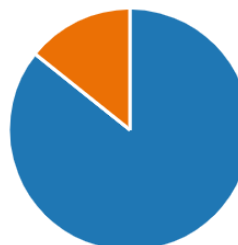


Ilustración 25 Validación prototipo Inicio sesión

6. Si usted fuera a ofrecer un servicio en una plataforma, ¿Cómo observas el contenido de la pantalla para crear un servicio y pueda ser visualizado por sus Clientes?

[Más detalles](#)

● Completo	6
● Regular	1
● Malo	0

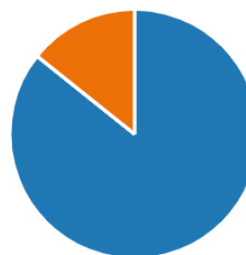


Ilustración 26 Validación información prototipo servicio

8. Considerando un mejor diseño de interfaz, ¿Que combinaciones tendrías en cuenta? (0 punto)

[Más detalles](#)

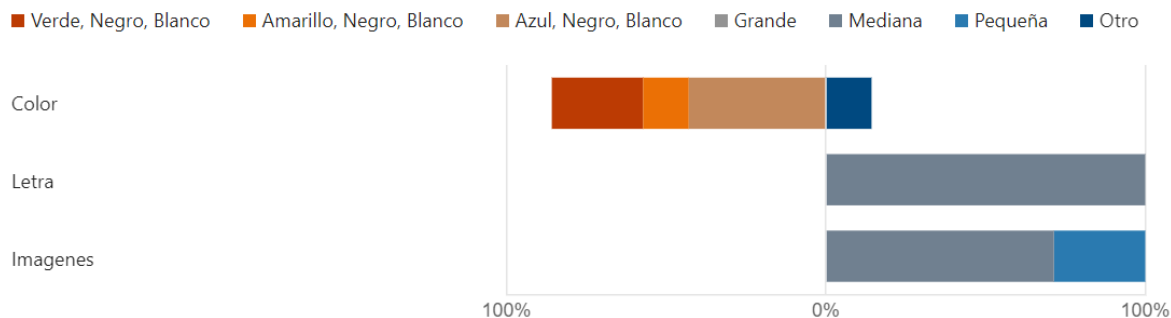


Ilustración 27 Validación colores y letras plataforma

9. Agradeciéndole por su valioso tiempo, ¿Qué comentarios o aportes adicionales agregarías en el diseño y las funcionalidades presentadas?

7 Respuestas

1	anonymous	A mi parecer me gusta más pantallas donde no tenga que hacer tanto scroll
2	anonymous	que sería importante retener el pago hasta que se preste el servicio y ellos puedan cargar las evidencias
3	anonymous	Que los botones no sean tan grandes, sean mas sutiles
4	anonymous	Cuando se piden los datos personales y datos de contacto se satura con muchos inputs y no da la mejor vista
5	anonymous	Muy clara la interacción interfaz - usuario
6	anonymous	Ninguna
7	anonymous	Ninguna

Ilustración 28 Validación opiniones abiertas usuarios

De acuerdo con las ilustraciones anteriores de los resultados más relevantes, los usuarios tienen buena percepción del diseño de los prototipos empleados en la plataforma. Los usuarios notan oportunidades de mejora como experiencia en el uso de aplicaciones que no presenten formularios extensos, letra grande, imágenes pequeñas, entre otras características importantes del producto a considerar. Para observar más detalle de la encuesta Ver Anexo 6.

Siguiendo con lo planteado en el modelo propuesto para adaptar los procesos de Desarrollo de

Software y el Design Thinking se logró que los artefactos aplicados llegaran de manera directa al usuario con el fin de explorar e identificar las necesidades con la agrupación de patrones de información, prototipado, implementación y validación del producto de software. Siguiendo con lo anterior, en el artículo titulado «Design Thinking in Practice: Understanding Manifestations of Design Thinking in Software Engineering», donde mencionan todas las prácticas de la metodología en los procesos de software, como los posibles modelos de proyectos, orientación en equipos, unificación de proyectos, modelos de herramientas [21], entre estas se encuentra el tipo de encuestas realizadas, la idea de elegir específicamente a nichos de mercado de oficios varios permitirá un mayor acercamiento a la realidad y así mismo, obtener características reales e importante junto a los artefactos que sirvieron en la obtención de estas, tales como las lluvias de ideas, co-creación y validación de prototipos desde el punto de vista inspiracional, ideación e implementación. Todo ello permite definir el alcance de un proyecto desde una etapa temprana, agilizando así el proceso de desarrollo.

10. DESARROLLO Y PRUEBAS DEL PRODUCTO

Este capítulo se encuentra enfocado en la manera como se implementa el prototipo y despliega el software con algunos stakeholders, con el fin de ir validando y corrigiendo el software junto con el cliente objetivo de la mano.

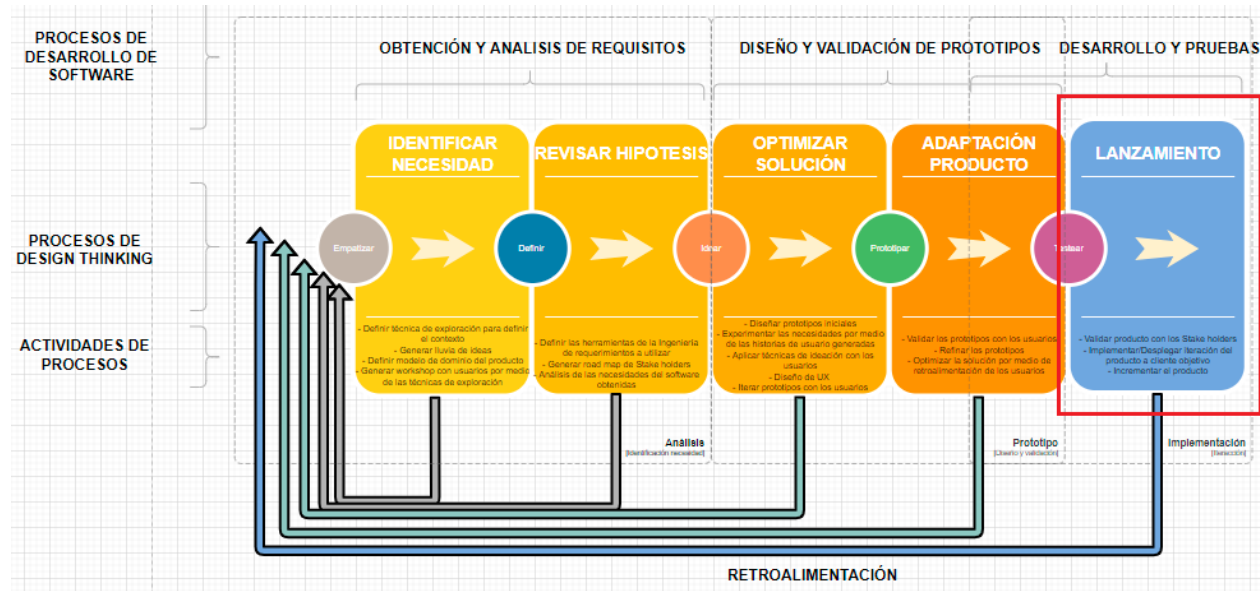


Ilustración 29 Etapa de desarrollo y pruebas validación

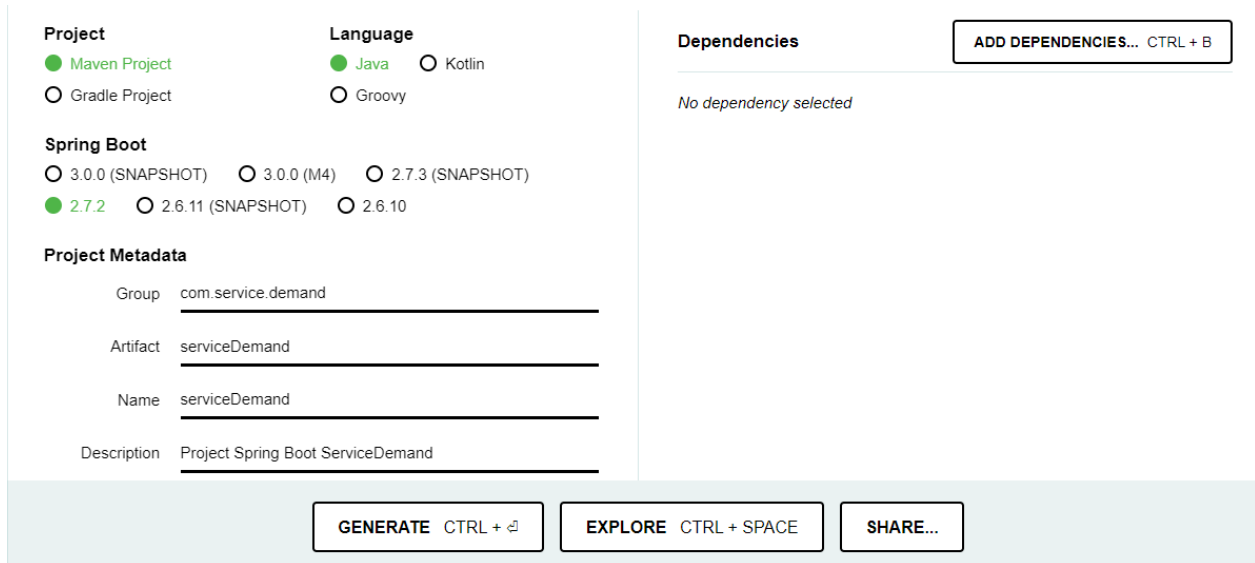
De acuerdo con la ilustración anterior, se describe el paso a paso de la ejecución del proceso de implementación, lanzamiento y validación del producto, teniendo en cuenta la información de las etapas anteriores como el alcance del dominio del sistema, requerimientos, prototipado, y validada.

10.1 Implementación técnica

El proyecto fue desarrollado bajo las condiciones técnicas de tecnologías comprobadas y con una alta compatibilidad sobre el proveedor de nube pública seleccionado, ya que la base de datos no relacional DynamoDB se basa en casos de éxito con proyectos similares al desarrollado.

10.1.1 Desarrollo Back

Se logra desarrollar el Back bajo tecnología Java con Spring Boot ya que ofrece robustez en la transaccionalidad que se pueda obtener en el aplicativo bajo altas cargas, escalabilidad y despliegues automáticos sobre la Nube de AWS (Amazon Web Service) con la funcionalidad de *Elastic Beanstalk*.



The screenshot shows the Spring Boot IDE configuration interface. It is divided into several sections:

- Project:** Includes radio buttons for "Maven Project" (selected), "Gradle Project", "Language" (with "Java" selected and "Kotlin" and "Groovy" as options), and "Spring Boot" (with "2.7.2" selected and other versions like "3.0.0 (SNAPSHOT)", "3.0.0 (M4)", "2.7.3 (SNAPSHOT)", "2.6.11 (SNAPSHOT)", and "2.6.10" as options).
- Project Metadata:** A form with fields for "Group" (com.service.demand), "Artifact" (serviceDemand), "Name" (serviceDemand), and "Description" (Project Spring Boot ServiceDemand).
- Dependencies:** A section with a button "ADD DEPENDENCIES... CTRL + B" and the text "No dependency selected".
- Bottom Bar:** Contains three buttons: "GENERATE CTRL + G", "EXPLORE CTRL + SPACE", and "SHARE...".

Ilustración 30 Spring Boot inicial

El lenguaje de programación Java ofrece despliegue multiplataforma y el rendimiento adecuado en el momento de la ejecución de las aplicaciones bajo un *Framework* optimizado como lo es el Spring. Se desarrolló el proyecto back llamado serviceDemand con su respectiva estructura de paquetes base.

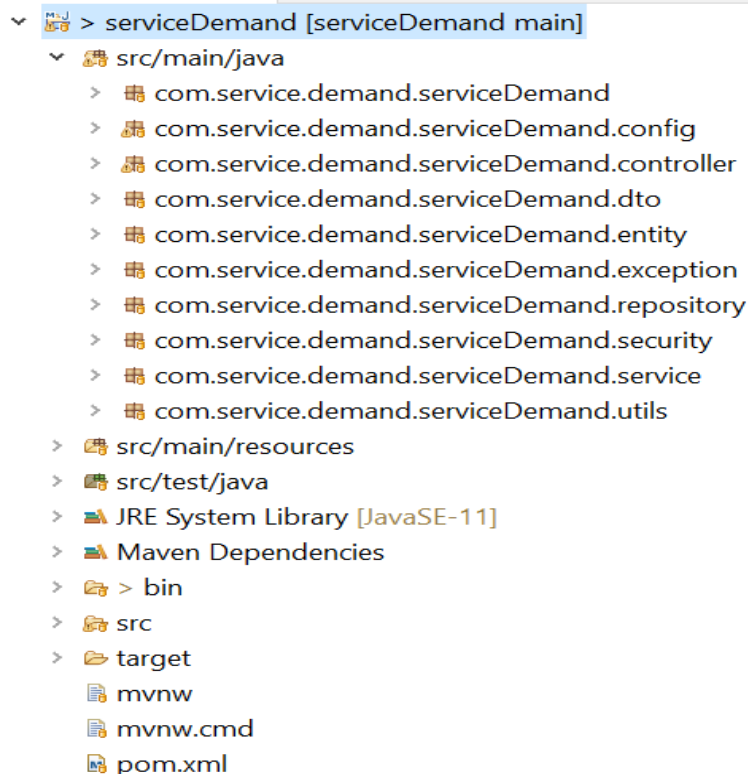


Ilustración 31 Estructura del proyecto Spring Boot

```
import java.util.List;
@Data
@AllArgsConstructor
@NoArgsConstructor
@DynamoDBTable(tableName = "service")
@Builder
@Component
public class ServiceDemand {

    @DynamoDBHashKey
    @DynamoDBAutoGeneratedKey
    private String id;

    @DynamoDBAttribute
    private String userId;

    @DynamoDBAttribute
    // @DynamoDBRangeKey
    private String name;

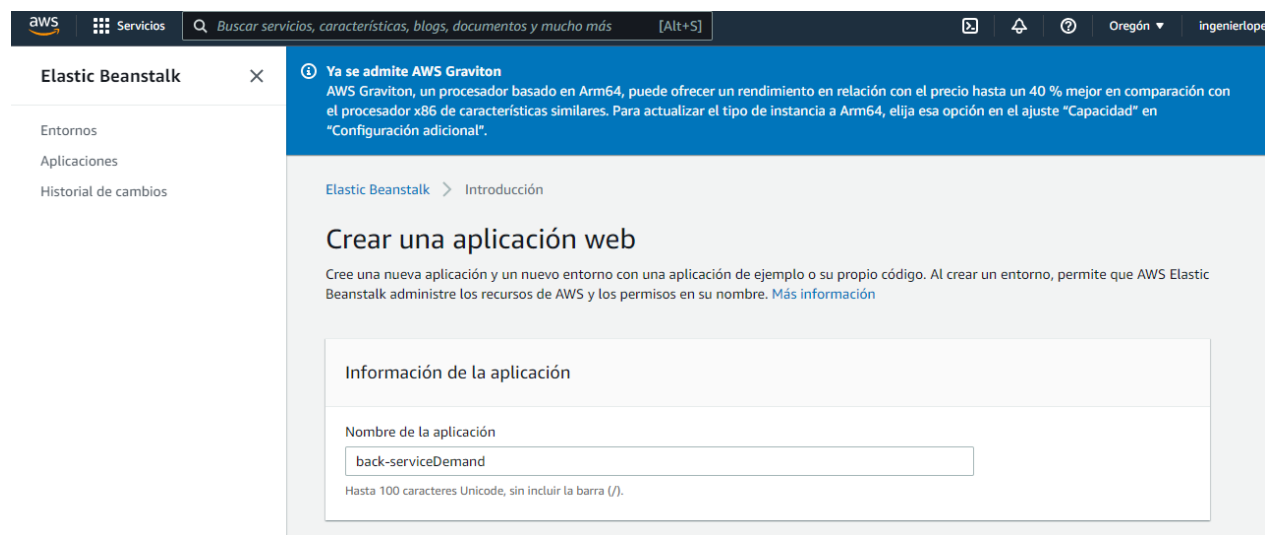
    @DynamoDBAttribute
    // @DynamoDBRangeKey
    private String description;
}
```

Ilustración 32 Implementación de clases

De acuerdo con la estructura de las capas definidas del proyecto y separando los repositorios, entidades, servicios y utilidades, se pueden generar buenas prácticas de mantenibilidad, escalamiento y seguridad en el código, por el modo que se definen las clases y variables de modo privado y sin declarativas de obtener y asignar valores por buenas prácticas en el desarrollo de código seguro.

10.1.2 Despliegue Back con Elastic Beanstalk

La nube pública de AWS (Amazon Web Service) ofrece funciones que permiten desplegar con facilidad proyectos con las características de Java, versión y generación de enlace público para que pueda ser consumido en el momento del despliegue de manera controlada.




The screenshot shows the AWS console interface for creating a web application on Elastic Beanstalk. At the top, there is a search bar with the text "Buscar servicios, características, blogs, documentos y mucho más" and a search icon. Below the search bar, there is a navigation menu with "Elastic Beanstalk" selected. A blue banner at the top right contains a message about AWS Graviton processors. The main content area is titled "Crear una aplicación web" and includes a sub-header "Información de la aplicación". A form field for "Nombre de la aplicación" (Application name) is visible, with the text "back-serviceDemand" entered. Below the form field, there is a note: "Hasta 100 caracteres Unicode, sin incluir la barra (/)." (Up to 100 Unicode characters, excluding the slash (/)).

Ilustración 33 Crear aplicación Web

Nombre	Valor
GRADLE_HOME	/usr/local/gradle
M2	/usr/local/apache-maven/bin
M2_HOME	/usr/local/apache-maven
SERVER_PORT	5000

Ilustración 34 Configuraciones del servicio en Beanstalk

 **Creando Backserviedemand-env**
 Este proceso tardará unos minutos.

AWSEBAutoScalingGroup-19OVILKT1GAMY:policyName/awseb-e-heyefi5sd-stack-AWSEBAutoScalingScaleDownPolicy-zEEjqaKNYyaG

9:37am Created Auto Scaling group policy named:
 arn:aws:autoscaling:us-west-2:684145662537:scalingPolicy:ccc30ec5-5097-4aff-bb0b-e70a270ab688:autoScalingGroupName/awseb-e-heyefi5sd-stack-AWSEBAutoScalingGroup-19OVILKT1GAMY:policyName/awseb-e-heyefi5sd-stack-AWSEBAutoScalingScaleUpPolicy-7ExFFQGne9Di

9:37am Waiting for EC2 instances to launch. This may take a few minutes.

9:37am Created Auto Scaling group named:
 awseb-e-heyefi5sd-stack-AWSEBAutoScalingGroup-19OVILKT1GAMY

9:36am Environment health has transitioned to Pending. Initialization in progress (running for 18 seconds). There are no instances.

9:35am Created security group named:
 awseb-e-heyefi5sd-stack-AWSEBSecurityGroup-DLM96U9B06MB

9:35am Created security group named:
 sg-03e6f8a8296724bae

9:35am Created target group named:
 arn:aws:elasticloadbalancing:us-west-2:684145662537:targetgroup/awseb--AWSEB-TORHFG0UCAQ/7982d6aa5b3e5ec3

9:35am Using elasticbeanstalk-us-west-2-684145662537 as Amazon S3 storage bucket for environment data.

9:35am createEnvironment is starting.

Ilustración 35 Despliegue del servicio con Beanstalk

El despliegue sobre la nube de AWS ayuda en la facilidad y las configuraciones del servicio, ofreciendo enlace público para que pueda ser consumido el servicio por internet desde la aplicación desarrollada. De igual manera, la función de *Elastic Beanstalk* tiene la característica que en el momento de actualizar una versión del despliegue puedan operar versiones sin

inconveniente y conservar la disponibilidad del servicio sin interrumpirse por los cambios afectados por el despliegue.

10.1.3 Desarrollo Front

Se logra desarrollar el prototipo del sistema bajo el *Framework Flutter* donde es soportado y desarrollado por la compañía Google bajo el lenguaje de programación Dart, ofreciendo la compatibilidad de ser multiplataforma cuando se requiera.

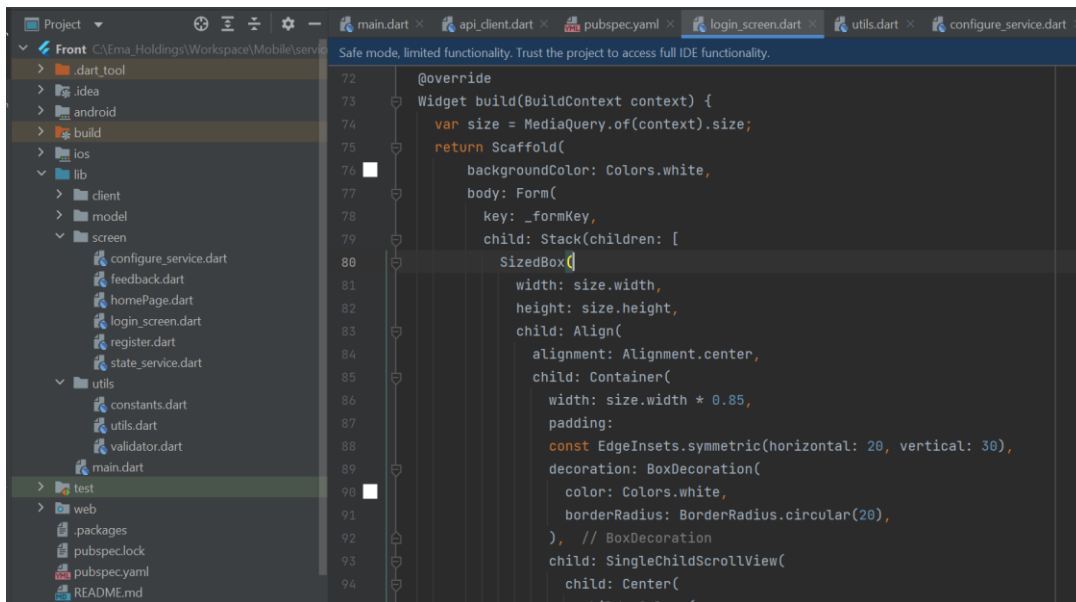


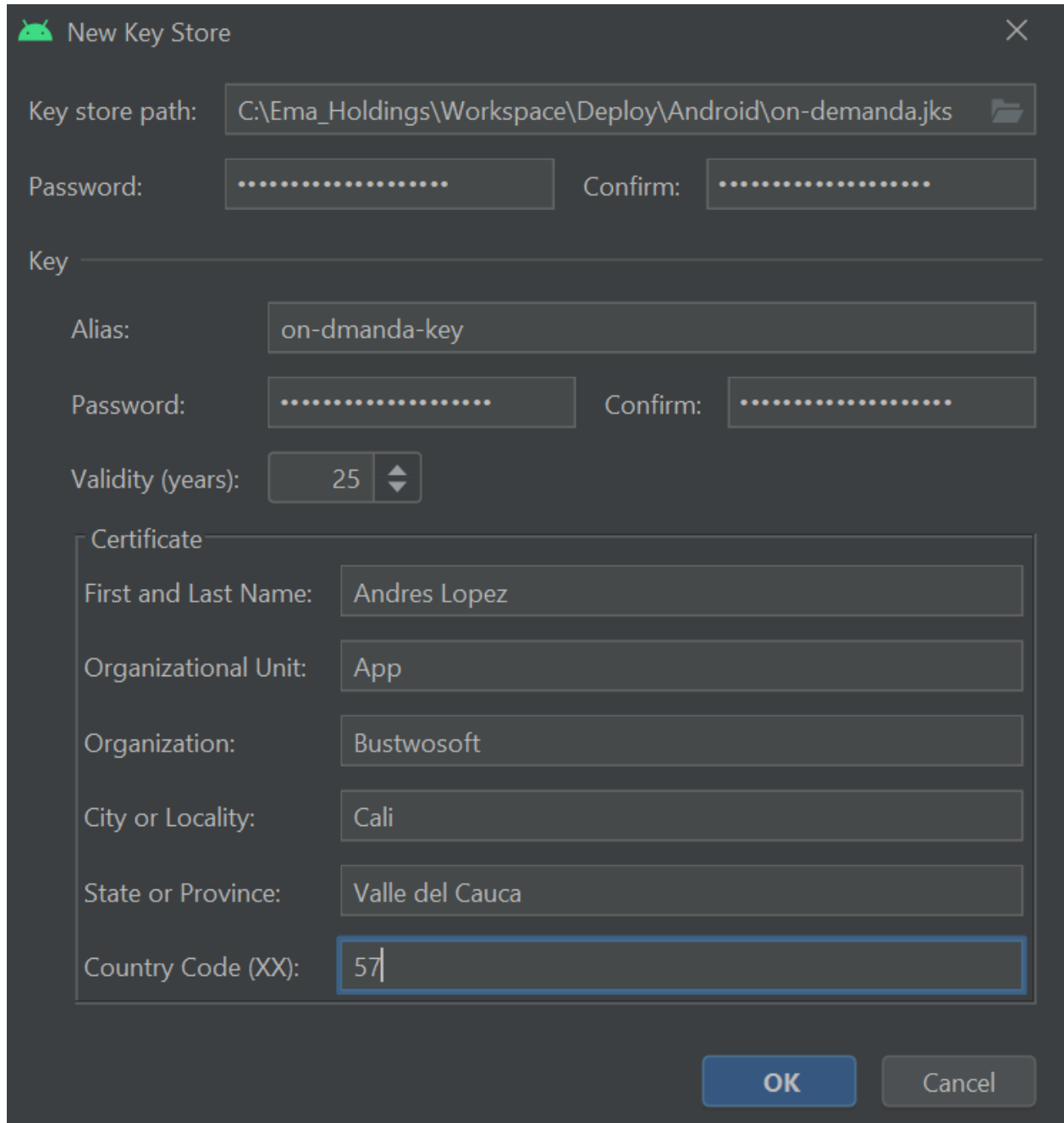
Ilustración 36 Estructura del proyecto Front

```
dependencies:
  flutter:
    sdk: flutter
  localstorage: ^3.0.6
  dio: ^4.0.6
```

Ilustración 37 Dependencias de Flutter

El proyecto Front desarrollado bajo el *Framework Flutter* es compatible con plataformas como Android, Web, iOS, ofreciendo versatilidad y facilidad en la configuración del entorno de desarrollo, agilizando el desarrollo por la forma de construir el código fuente ya que se acerca a los nativos de Android y iOS con una robustez en la ejecución.

10.1.4 Despliegue de aplicativo en Android



New Key Store

Key store path: C:\Ema_Holdings\Workspace\Deploy\Android\on-demanda.jks

Password: Confirm:

Key

Alias: on-dmanda-key

Password: Confirm:

Validity (years): 25

Certificate

First and Last Name: Andres Lopez

Organizational Unit: App

Organization: Bustwosoft

City or Locality: Cali

State or Province: Valle del Cauca

Country Code (XX): 57

OK Cancel

Ilustración 38 Generar clave de aplicativo

En la anterior Ilustración, se realiza la generación de la clave para la generación del compilado de la app que exige la tienda de Google Developer.

10.1.5 Despliegue de aplicativo en Android

```
PS C:\Ema_Holdings\Workspace\Mobile\serviceDemand\Front> flutter build apk --release
Downloading android-arm-profile/windows-x64 tools... 1.109ms
Downloading android-arm-release/windows-x64 tools... 735ms
Downloading android-arm64-profile/windows-x64 tools... 609ms
Downloading android-arm64-release/windows-x64 tools... 476ms
Downloading android-x64-profile/windows-x64 tools... 826ms
Downloading android-x64-release/windows-x64 tools... 585ms

Building with sound null safety

Running Gradle task 'assembleRelease'...
```

Ilustración 39 Comando Flutter generar APK Android

De acuerdo con la Ilustración anterior, este es el comando para generar el APK (ejecutable plataforma Android) para que se pueda montar y ejecutar sobre la tienda de Google Play.

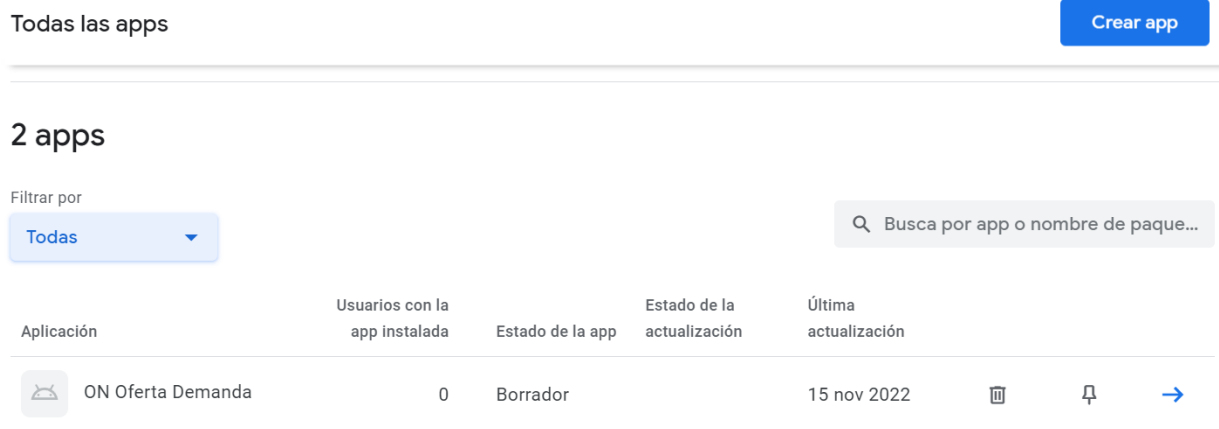


Ilustración 40 Creación aplicación en Google Developer

De acuerdo con la Ilustración anterior, se crea una aplicación en Google Developer para que pueda ser publicada la aplicación generada.

11. RESULTADOS EN LA ADAPTABILIDAD DE PROCESOS DE DESIGN THINKING Y DE SOFTWARE CON EL MODELO PROPUESTO

A continuación, se describe el resultado de la investigación y de cómo se evalúa la efectividad de la metodología de *Design Thinking* en el desarrollo de software, por medio de un prototipo de la aplicación de oferta y demanda de servicios de empleo para oficios varios en la ciudad de Cali. Partiendo desde la búsqueda del problema con una tasa de desempleo del 47 % en el Valle del Cauca, con un nicho de mercado potencial, donde las personas buscan conectar oferta con demanda. Partiendo de la información inicial recolectada por artículos, foros, investigaciones y entrevistas, se diseña un modelo en el cual se busca aterrizar las mejoras prácticas de ambos mundos de la metodología *Design Thinking* y los procesos de desarrollo de software, para que puedan ser coherentes en la aplicabilidad. A partir de esto, se propone un modelo que cuenta con la capacidad de retroalimentarse desde cualquiera de sus etapas, que está conectado entre sí, que es de ágil ejecución y económico al momento de su desarrollo, es importante resaltar que cada una de estas etapas están centradas en el usuario objetivo. Ver Ilustración 5 para mayor entendimiento de lo planteado.

11.1 Adaptación entre la metodología Design Thinking y procesos de desarrollo de software

Para llegar a este punto es importante pensar en lo siguiente: ¿es posible converger el pensamiento del diseño con desarrollo de software? Partiendo de lo anterior, donde el *Design Thinking* se enfoca en lo intangible (Descubrimiento, Empatía, Centrado en el Usuario) y el desarrollo de software en lo tangible (Producto, Artefactos, Lanzamientos, Validaciones).

Teniendo en cuenta estos dos horizontes: centralización del usuario y la tecnología por medio de los procesos de desarrollo de software para llegar a una solución, se indica que hay una alta cohesión. Se debe empezar primero por lo que desea el usuario y verificar la viabilidad técnica para llegar a un producto deseable; con todo esto se engloban una serie de técnicas o herramientas que poseen tanto el *Design Thinking* como los procesos de desarrollo de software en general y pueden desarrollarse por etapas, teniendo en cuenta de lado y lado que es lo mejor adaptable posible para que pueda ser ejecutado de manera natural bajo los siguientes criterios:

- **Tiempo:** Para este parámetro es considerable que se logre lanzar un producto con pocas características en brevedad de tiempo, para que pueda ser validado e iterado con agilidad.
- **Recursos:** El secreto de la metodología *Design Thinking* es desarrollar rápido y a bajo costo, es por esto por lo que se deben lanzar productos con prototipos desechables y validados.
- **Efectividad:** Siempre es importante por cada etapa centrar las actividades con el usuario objetivo por medio de técnicas de empatía como encuestas, validaciones, análisis de información, herramientas para consolidar información como los requerimientos,

modelamiento dominio, canvas y comparativos que no ocupen mucho tiempo, pero que aporten al desarrollo y entendimiento del sistema, con el objetivo de llegar a un valor diferencial en el mercado.

- **Cambio:** Para que la metodología pueda adaptarse, es considerable tener la variable de cambio ya que el proceso debe ser evolutivo, ágil, barato y deseable para el usuario.

11.2 Aplicabilidad del modelo definido

Después de definido el modelo y teniendo en cuenta lo mencionado con anterioridad, aquí es donde se involucra la parte práctica del modelo y en donde se debe hacer uso de las actividades y artefactos aplicados. A continuación, se muestra la tabla consolidada de lo aplicado en el modelo.

Actividad	Estático / Recolecta información	Etapas	Involucra Usuario	Artefacto
Contexto de Dominio	Estático	Análisis	NO	- Entrevista de profundidad
Generación lluvia de ideas	Recolecta información	Análisis	SI	- No Aplica
Requerimientos funcionales	Recolecta información	Análisis	SI	- Entrevista de profundidad
Requerimientos no funcionales	Recolecta información	Análisis	NO	- Entrevista de profundidad
Resultado de trabajo de campo	Recolecta información	Análisis	NO	- Entrevista de profundidad
Canvas	Recolecta información	Análisis	NO	- Entrevista de profundidad - Entrevista con experto
Benchmark	Recolecta información	Análisis	NO	- Entrevista con experto

Diagrama Caso Uso General	Estático	Análisis	NO	<ul style="list-style-type: none"> - Entrevista de profundidad - Canvas
Diagrama de Actividades	Estático	Análisis	NO	<ul style="list-style-type: none"> - Entrevista de profundidad - Requerimientos funcionales
Diagrama de Arquitectura	Estático	Análisis	NO	<ul style="list-style-type: none"> - Entrevista de profundidad - Canvas - Benchmark - Requerimientos funcionales y no funcionales - Modelo del dominio
Diseño y validación de prototipación	Recolecta información	Prototipo	SI	<ul style="list-style-type: none"> - Encuesta de validación
Implementación	Recolecta información	Implementación	SI	<ul style="list-style-type: none"> - Validación del producto - Diagrama de Arquitectura - Requerimientos funcionales y no funcionales - Diseño y validación de prototipación

Tabla 6 Herramientas utilizadas por etapa del modelo

De acuerdo con la tabla, se logra evidenciar que en todas las etapas del modelo se encuentra el involucramiento del usuario, esto quiere decir que es un modelo centrado en el usuario objetivo, logrando así validaciones cortas por actividades y etapas, llevando de la mano al usuario, por otro lado, existen actividades que por su característica de entregable se logran convertir fácilmente en artefactos, queriendo decir que esto es fuente de información y validación para otras actividades a desarrollar. Por último, ¿por qué se encuentra involucrado el usuario en la etapa de

implementación? Todo esto es porque existe un artefacto de validación del producto a medida que se va lanzando por pequeñas características, este juega un papel importante en la implementación dando un *feedback* muy corto para ajustes y cambios cortos.

11.3 Resultados obtenidos a partir del modelo

Con respecto a la aplicabilidad del modelo definido, se obtienen resultados en la adaptabilidad del modelo en los procesos de desarrollo de software, definición del modelo con las necesidades de los usuarios, validación del producto en menores costos e iteraciones cortas y sencillas en la ejecución del proceso.

Los resultados obtenidos son producto de la aplicabilidad del modelo en el prototipo de un desarrollo de software para oferta y demanda de servicios de oficios varios para la ciudad de Cali. Aquí se lograron utilizar artefactos y actividades plasmados en el modelo, teniendo en cuenta una variable dependiente en el proyecto que es el usuario, quien es el que define el alcance, funcionalidades futuras y las validaciones de un proyecto de software en cada una de las fases de este. Por consiguiente, se elabora un gráfico donde se consolidan artefactos por fases e involucramiento del usuario en cada una de estas de manera dependiente.

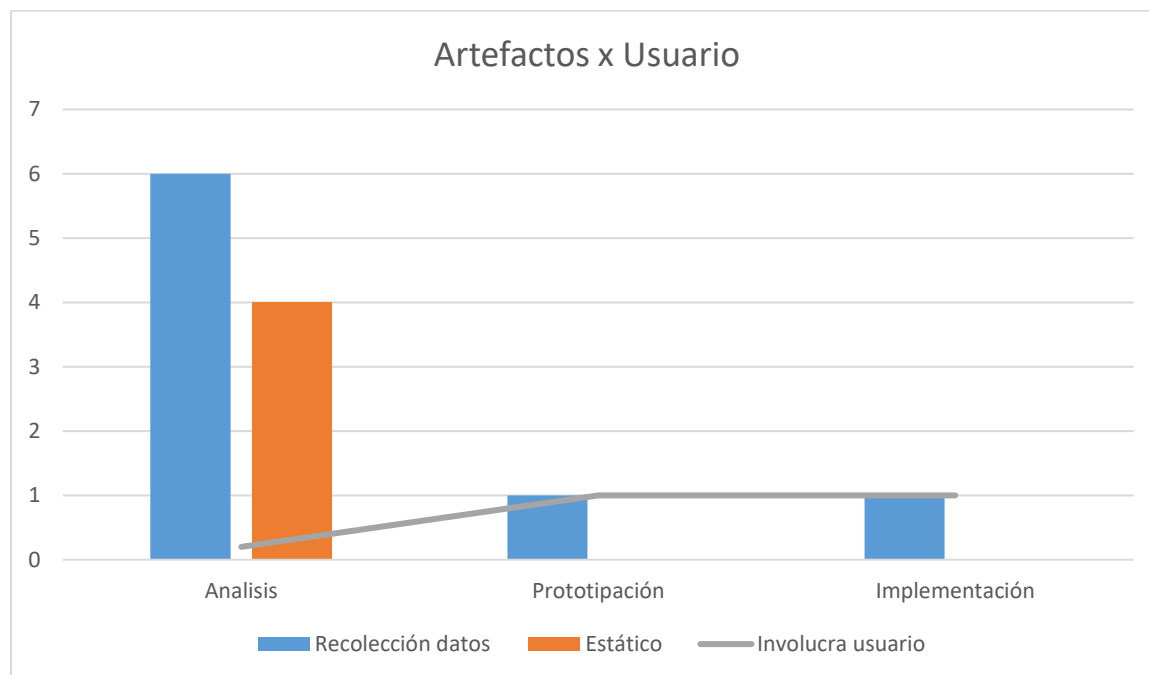


Ilustración 41 Resultado de Artefactos por Usuario

De acuerdo con el gráfico anterior, se logra evidenciar que en todas las Fases del modelo participó por lo menos una interacción con el usuario en Artefactos que generaban recolección de datos, donde se hizo valioso obtener información de primera mano con las validaciones con actividades cortas y efectivas para ir prototipando o modelando el producto lo mas cercano al usuario

objetivo. Por otro lado, se logra evidenciar que se agregaron Artefactos en su mayoría en la Fase de Análisis que es donde se define el que desea el usuario por medio de un producto, en esta etapa es muy importante obtener la mayor información posible debido a que se tienen actividades como definición del modelo del sistema, requerimientos, funcionalidades requeridas a alto nivel, alcance del producto.

Otros de los resultados en el proceso es la facilidad con que se logran aplicar los artefactos ya sea que lleguen del mundo del software o del Design Thinking convergidos en el modelo de adaptación ya que son complementarios en el momento de generar los respectivos entregables generando valor a un proyecto de desarrollo de software con documentación técnica necesaria, fases rápidas en ejecución, centrado en el usuario para cumplir con sus respectivas necesidades por medio de validaciones en cada una de las fases. Por otro lado es de considerar tener artefactos estáticos y recolección de información ya que apoyan en la toma de decisiones entre lo definido, las incertidumbres, oportunidades de mejora para madurar e incrementar el producto de software por cada una de las iteraciones.

11.4 Principios de la adaptación de procesos de software con Design Thinking

A continuación se mencionan los principios para lograr la adaptación entre los procesos de software con el *Design Thinking*:

- Centrado en el usuario.
- Interacciones cortas y livianas en actividades.
- Validación del usuario en cada una de las fases.
- Documentación técnica suficiente, generando artefactos estáticos y logrando entender el producto.
- Elaboración de producto con alcances mínimos a bajo costo.
- Fallar en cada una de las fases es bueno con involucramiento del usuario.
- Recolección de información para la toma de decisiones.
- Cultura de agilismo.

11.5 Guía de aplicabilidad del modelo

A continuación, se describe una guía de aplicabilidad del modelo definido, el cual es basado en los procesos de desarrollo de software y los procesos de la metodología *Design Thinking* convergiendo ambos mundos en un modelo práctico y adaptativo para entrega de productos tangibles al usuario final, remitido de forma rápida a bajo costo y con una propuesta de valor en el mercado.

- **Identificación de necesidad:** Se debe tener claro que para iniciar se debe pensar en lo DESEABLE por el usuario para llegar a una posible solución, en este enfoque de

adaptabilidad se debe eliminar la premisa del cómo voy hacer sin tener claro el problema y necesidad del usuario objetivo. Para llegar a este punto se trae gráfico para mejor entendimiento de la circulación de prioridades por dónde iniciar en encontrar las necesidades a un problema, sin perder el foco en el valor agregado y creando un producto tangible como solución de software. Ver Ilustración 6 del documento.

- **Identificación Usuarios:** Para el modelo definido es importante tener claro los posibles usuarios que se deben involucrar en el proceso, para llegar a una adecuada recolección de información.
 - **Usuario(s) objetivo:** Mencionado en el capítulo 7.
 - **Usuario(s) experto:** Mencionado en el capítulo 7.
 - **Usuario validación producto:** Es aquel usuario que está involucrado en la fase de validación y/o lanzamiento del producto, no necesariamente debe ser el mismo usuario objetivo donde por población o segmentación de usuarios se requiere validar la solución.
 - **Desarrollador:** Rol que ejecuta la solución de software convirtiéndola en producto tangible.
 - **Analista de información:** Rol que tiene como funcionalidades levantar modelo de dominio, requerimientos, diseño de prototipos, entre otros procesos de desarrollo de software.
 - **Diseñador de producto:** Es importante tener este rol en el equipo ya que al adaptar ambos mundos de Design Thinking y procesos de desarrollo de software suma relevancia en aplicar los diferentes artefactos por medio de encuestas, entrevistas, recolección de información, canvas, benchmark y lluvia de ideas centrados en el usuario.
 - **Otros roles técnicos necesarios:** Si se requiere iniciar con un proceso robusto o por el tamaño del proyecto para aplicar el modelo, se puede considerar ingresar arquitectos de software, gerente(s) de proyectos, entre otros roles importantes, como nota importante, es relevante considerar que el producto se debe lanzar con alcances mínimos y bajo costo, de acuerdo con el principio cinco definido con anterioridad.
- **Artefactos:** Este punto de la guía hace parte fundamental de los entregables en el modelo definido ya que cuenta con artefactos tanto estáticos y recolección de información, para

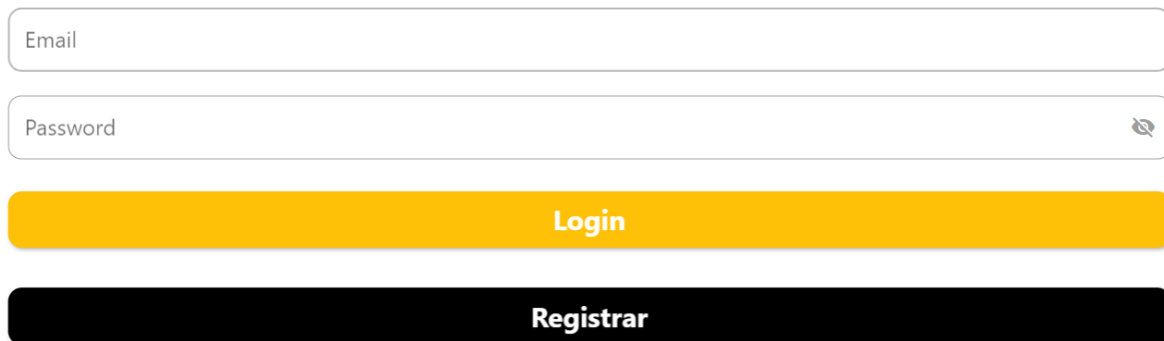
Llegar un alcance de producto validado con el usuario, a continuación se nombran algunos de ellos para ser tenidos en cuenta. Mencionados en el capítulo 8.

- Lluvia de ideas.
 - Encuesta.
 - Entrevista.
 - Requerimientos.
 - Resultados trabajo campo.
 - Canvas (Patrones de información) Ver Ilustración 16 para mayor detalle.
 - Canvas de negocio.
 - Benchmark.
 - Diagrama actividades.
 - Diagrama arquitectura.
-
- **Adaptación producto:** Es importante validar el prototipo del producto con métodos baratos que permitan que el usuario tenga una aceptabilidad del producto de software a lanzar y madurar en una población de usuarios objetivos, teniendo en cuenta los siguiente tips:
 - Aplicar encuestas con los mockups de las funcionalidades futuras que va a tener el producto de software.
 - Definir un alcance corto con lanzamientos de producto.
 - Aplicar encuestas de satisfacción sobre el mismo producto de software para obtener retroalimentación inmediata e ir incrementando el producto con las recomendaciones del usuario.
 - Fallar barato es una de las claves para incrementar el producto lo más cercano al usuario.
 - Tomar nota de las oportunidades de mejora por cada ciclo finalizado del modelo.

12. RESULTADO DEL DESARROLLO PROTOTIPO FINAL

A continuación, se mostrarán algunas de las funcionalidades principales del desarrollo del prototipo.

Login



Email

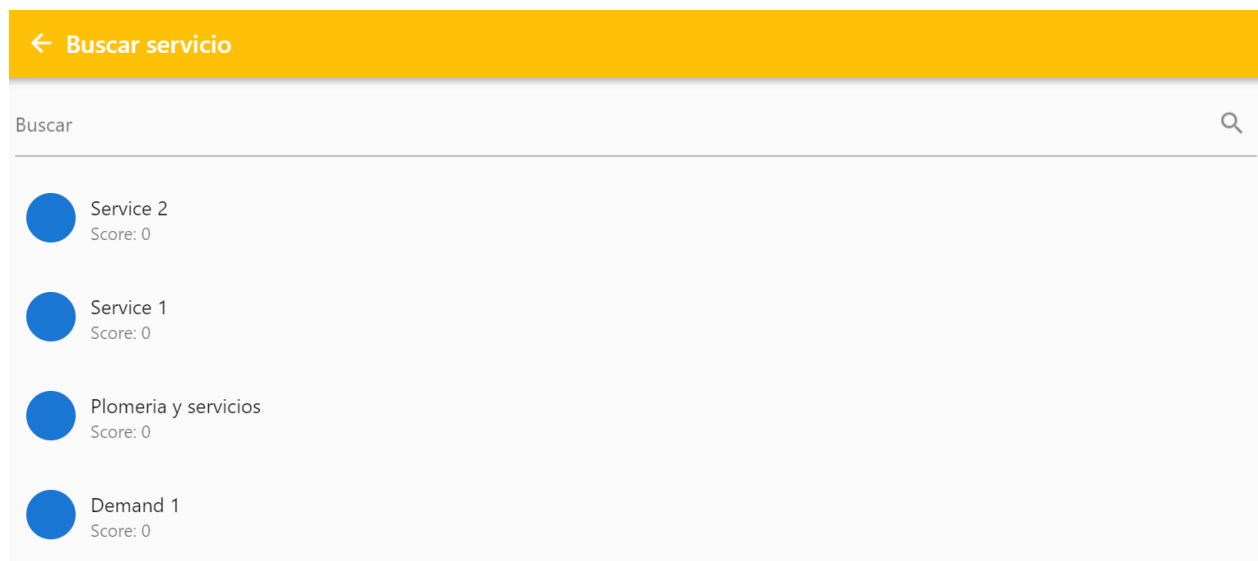
Password

Login

Registrar

Ilustración 42 Login

En la Ilustración se muestra la interfaz principal de la aplicación, para dar acceso a usuarios creados o registrar los usuarios.



← Buscar servicio

Buscar

- Service 2
Score: 0
- Service 1
Score: 0
- Plomeria y servicios
Score: 0
- Demand 1
Score: 0

Ilustración 43 Buscar servicio

En la Ilustración se muestra la opción para buscar los servicios cargados en el sistema por los usuarios que ofertan el sistema.

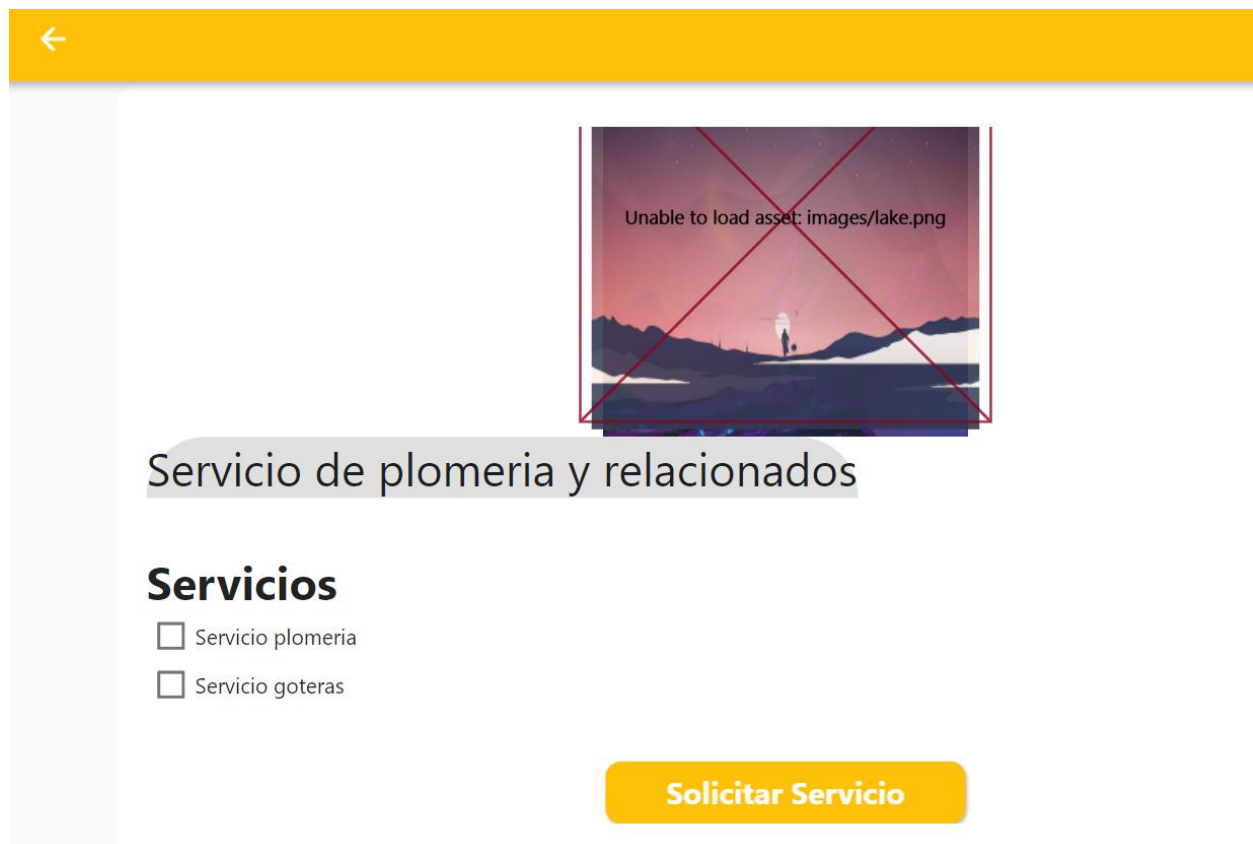
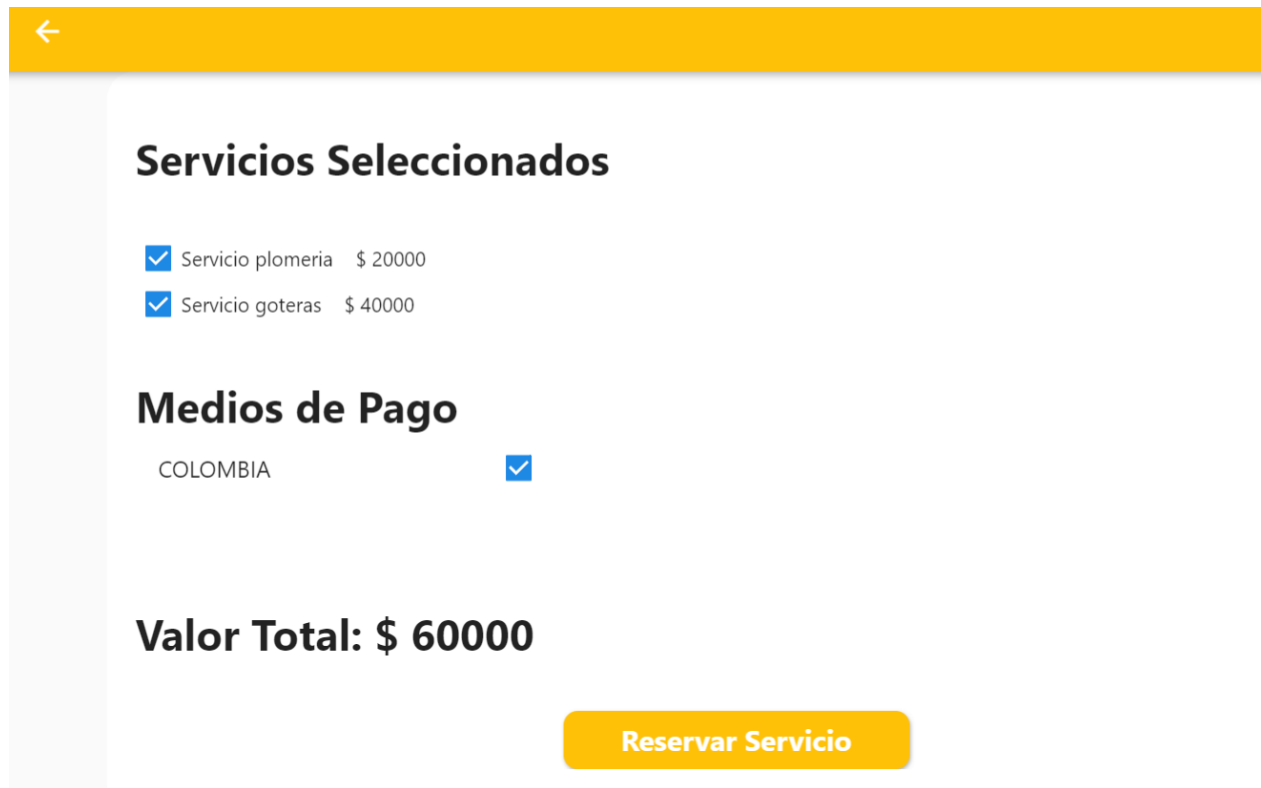


Ilustración 44 Información del servicio

En la Ilustración anterior, se carga la información de los servicios ofertados por un usuario, donde se brinda la posibilidad de seleccionar uno o muchos servicios de este.



←

Servicios Seleccionados

- Servicio plomeria \$ 20000
- Servicio goteras \$ 40000

Medios de Pago

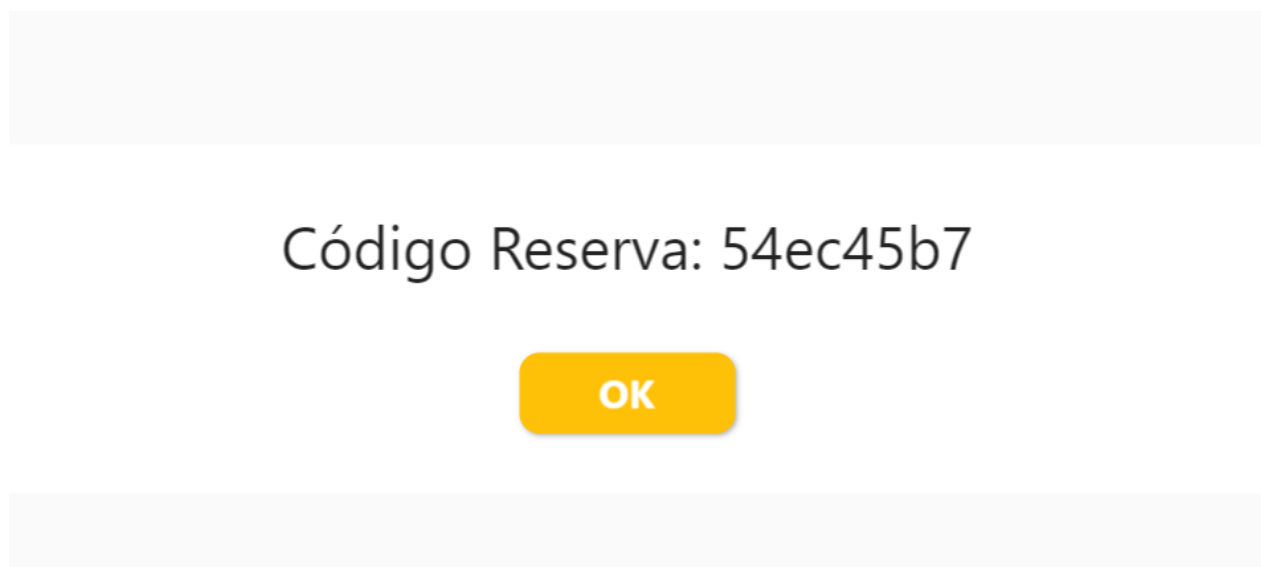
COLOMBIA

Valor Total: \$ 60000

Reservar Servicio

Ilustración 45 Reservar servicio

En la Ilustración anterior, se muestra el valor total del servicio, y el medio de pago disponible antes de dar clic sobre el botón reservar.



Código Reserva: 54ec45b7

OK

Ilustración 46 Generación código de reserva

En la Ilustración anterior, se muestra la generación del código de reserva con el cual se realizará trazabilidad del servicio hasta cerrar el ciclo.

13. PRUEBAS REALIZADAS AL PROTOTIPO

En la siguiente sesión se explica la ejecución de las pruebas en el “Back” de la aplicación para asegurar la correcta integración con la aplicación que va consumir cada una de las APIS.

```

3 @SpringBootTest
3 class ServiceDemandApplicationTests {
3
1  @Test
2  void contextLoads() {
3      int leftLimit = 48; // numeral '0'
4      int rightLimit = 122; // letter 'z'
5      int targetStringLength = 6;
6      Random random = new Random();
7
8      String generatedString = random.ints(leftLimit, rightLimit + 1)
9          .filter(i -> (i <= 57 || i >= 65) && (i <= 90 || i >= 97))
10         .limit(targetStringLength)
11         .collect(StringBuilder::new, StringBuilder::appendCodePoint, StringBuilder::append)

```

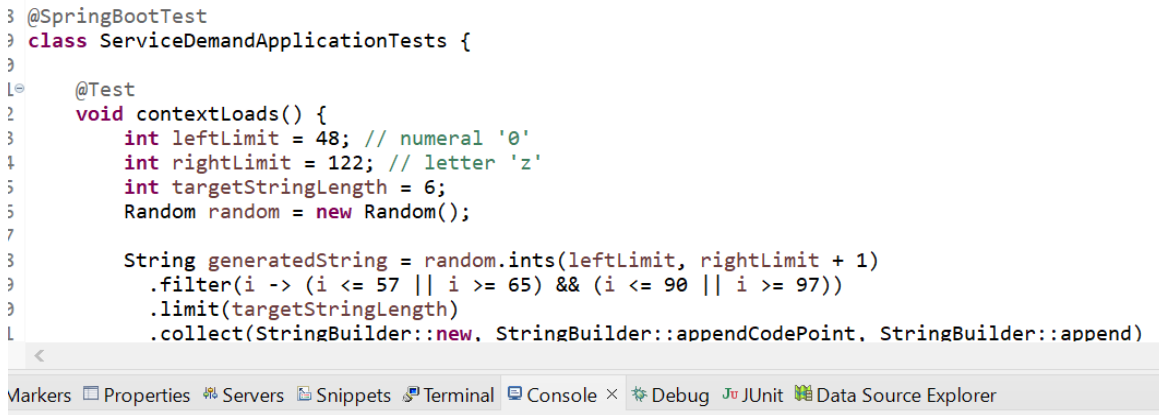
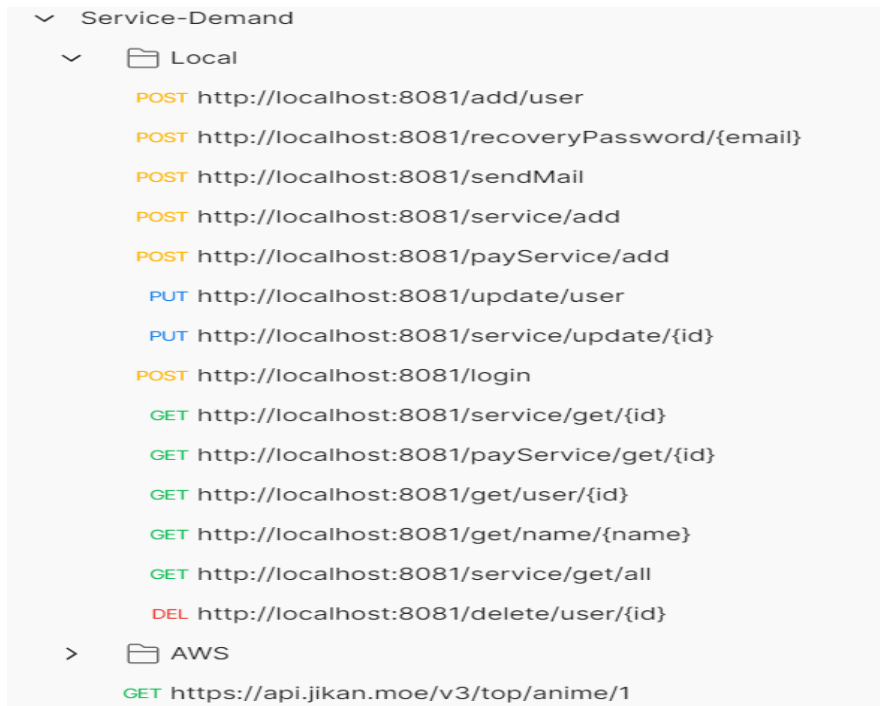


Ilustración 47 Pruebas código back

De acuerdo con la Ilustración anterior, se desarrolla prueba para que en el momento de la compilación «Maven» se ejecute.



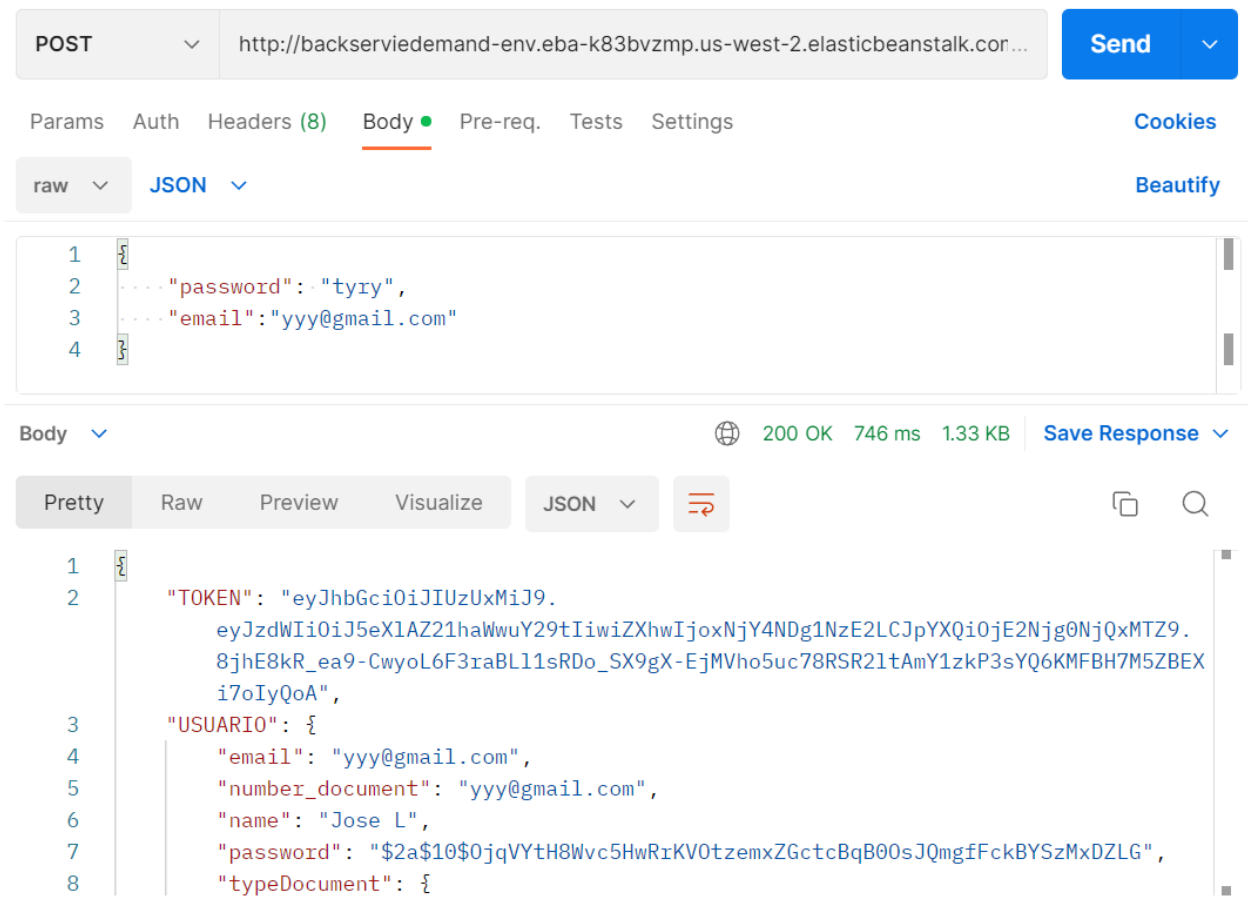
```

Service-Demand
├── Local
│   ├── POST http://localhost:8081/add/user
│   ├── POST http://localhost:8081/recoveryPassword/{email}
│   ├── POST http://localhost:8081/sendMail
│   ├── POST http://localhost:8081/service/add
│   ├── POST http://localhost:8081/payService/add
│   ├── PUT http://localhost:8081/update/user
│   ├── PUT http://localhost:8081/service/update/{id}
│   ├── POST http://localhost:8081/login
│   ├── GET http://localhost:8081/service/get/{id}
│   ├── GET http://localhost:8081/payService/get/{id}
│   ├── GET http://localhost:8081/get/user/{id}
│   ├── GET http://localhost:8081/get/name/{name}
│   ├── GET http://localhost:8081/service/get/all
│   └── DEL http://localhost:8081/delete/user/{id}
└── AWS
    ├── GET https://api.jikan.moe/v3/top/anime/1

```

Ilustración 48 Consumo de APIS

De acuerdo con la Ilustración anterior, se mapean las APIS desarrolladas para que puedan ser verificadas con la respuesta esperada.



The screenshot shows a REST client interface with a POST request to `http://backserviedemand-env.eba-k83bvzmp.us-west-2.elasticbeanstalk.cor...`. The request body is a JSON object with `"password": "tyry"` and `"email": "yyy@gmail.com"`. The response is a 200 OK status with a response time of 746 ms and a size of 1.33 KB. The response body is a JSON object containing a JWT token, user information, and document details.

```
1 {
2   ... "password": "tyry",
3   ... "email": "yyy@gmail.com"
4 }
```

Body 200 OK 746 ms 1.33 KB Save Response

```
1 {
2   "TOKEN": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ5eX1AZ21haWwY29tIiwiaXNhwIjoxNjY4NDg1NzE2LCJpYXQiOiJlMjNjg0NjQxMTZ9.8jhE8kR_ea9-CwyoL6F3raBL1sRDo_SX9gX-EjMVho5uc78RSR21tAmY1zkP3sYQ6KMFbH7M5ZBEXi7oIyQoA",
3   "USUARIO": {
4     "email": "yyy@gmail.com",
5     "number_document": "yyy@gmail.com",
6     "name": "Jose L",
7     "password": "$2a$10$0jqVYtH8Wvc5HwRrKV0tzemxZGctcBqB00sJQmgfFckBYSzMxDZLG",
8     "typeDocument": {
```

Ilustración 49 Seguridad JWT APIS

En la Ilustración anterior, se demuestra la seguridad en las APIS con tecnología JWT (Json Web Token), donde permite no solo permite la conexión a las APIS con el Token autorizado, sino también evita conexiones inseguras.

14. CONCLUSIONES

- Gracias a la definición del modelo de adaptación de los procesos de desarrollo de software y de la metodología *Design Thinking* se logró llegar a los resultados, principios y guía del modelo, implementando el prototipo de desarrollo de una aplicación que permite dar a conocer la oferta y demanda de empleos para oficios varios en la ciudad de Cali.
- Se logró identificar mediante el modelo definido la adaptabilidad de la metodología *Design Thinking* en los proyectos de desarrollo de software determinando las fases, actividades, artefactos de ambos procesos que tengan una alta cohesión y obteniendo un modelo liviano en ejecución y con efectividad de entregable, para un adecuado desarrollo de software.
- Mediante la aplicabilidad del modelo se logró aplicar las fases de *Design Thinking* de empatizar, definir, testear, idear, prototipar con fases centradas en el usuario, ofreciendo un factor clave de validación en cada una de ellas.
- La plataforma como prototipo desarrollado beneficiará a la comunidad de la ciudad de Cali con un despliegue mayor, logrando conectar la oferta con la demanda en el sector de oficios varios con un valor diferencial en el mercado.
- El modelo de adaptación definido beneficiará a las personas que deseen implementar proyectos de desarrollo de software, aplicando los principios y guía definida como conclusión del resultado de la investigación, basado en una metodología que permitió implementar el desarrollo de un prototipo de la aplicación de oferta y demanda de servicios en el sector de oficios varios.
- La metodología *Design Thinking* permitió evidenciar la importancia de la herramienta como método de desarrollo y ejecución de nuevas plataformas que impacten de forma positiva al usuario, puesto que facilita el proceso de ejecución, la reducción de costos, la efectividad de la información y la accesibilidad en su uso.

14.1 Trabajos futuros:

A continuación, se mencionan trabajos futuros que se pueden relacionar al trabajo realizado.

- Estudio de mercado aplicando *Design Thinking*.
- Trabajo complementario para fortalecer funcionalidades y maduración del prototipo, aplicando modelo diseñado.
- Realizar trabajo de validaciones y pruebas del prototipo aplicando modelo diseñado.

15. REFERENCIAS BIBLIOGRÁFICAS

- [1] “Empleo informal y seguridad social - Históricos.” <https://www.dane.gov.co/index.php/estadisticas-por-tema/mercado-laboral/empleo-informal-y-seguridad-social/empleo-informal-y-seguridad-social-historicos> (accessed Sep. 05, 2021).
- [2] S. Speranza, “Mind Garage”, Accessed: Aug. 31, 2022. [Online]. Available: <https://platzi.com/cursos/design-thinking/>
- [3] “Información básica sobre la COVID-19.” <https://www.who.int/es/news-room/questions-and-answers/item/coronavirus-disease-covid-19> (accessed Aug. 31, 2022).
- [4] “Qué son las metodologías ágiles y cómo pueden ayudar | Salesforce - Blog de Salesforce.” <https://www.salesforce.com/mx/blog/2021/12/que-son-metodologias-agiles-y-como-pueden-ayudar-a-tus-equipos-de-trabajo.html> (accessed Aug. 31, 2022).
- [5] “Nicho de mercado - Qué es, definición y concepto | 2022 | Economipedia.” <https://economipedia.com/definiciones/nicho-de-mercado.html> (accessed Aug. 31, 2022).
- [6] “SWEBOK - Guide v4.0,” *The Jerusalem Talmud, First order: Zeraim, Tractate Berakhot*, pp. 39–155, 2013, doi: 10.1515/9783110800487.39.
- [7] “About The Standish Group - The Standish Group.” <https://www.standishgroup.com/about> (accessed Aug. 31, 2022).
- [8] “Descifrando las metodologías ágiles - Decide.” <https://decidesoluciones.es/descifrando-las-metodologias-agiles/> (accessed Sep. 05, 2021).
- [9] “El valle de la muerte de proyectos TI en sector público — El Escritorio de Alejandro Barros : El Escritorio de Alejandro Barros.” <https://www.alejandrobarrros.com/el-valle-de-la-muerte-de-proyectos-ti-en-sector-publico/> (accessed Apr. 19, 2021).
- [10] “Cuál es la diferencia entre Agile, Lean Startup y Design Thinking.” <https://www2.deloitte.com/es/es/blog/todo-tecnologia/2021/diferencias-agile-lean-startup-design-thinking.html> (accessed Sep. 05, 2021).
- [11] “Design Thinking.” <https://hbr.org/2008/06/design-thinking> (accessed May 15, 2021).
- [12] M. Vianna, Y. Vianna, E. K. Adler, B. Lucena, and B. Russo, *Design Thinking. Innoación en negocios*. 2011.
- [13] “Design Thinking en Español.” <https://www.designthinking.es/inicio/> (accessed Mar. 17, 2021).
- [14] “5 empresas que han triunfado gracias al Design thinking - Artículo | CEEI Elche | EmprenemJunts.” <https://ceeielche.emprenemjunts.es/?op=8&n=16097> (accessed Mar. 19, 2021).
- [15] ““Mi Guía’: plataforma de prestación de servicios que apoya formalización de empleo en Medellín.” <https://www.mintic.gov.co/portal/inicio/Sala-de-Prensa/Noticias/14708:Mi-Guia-plataforma-de-prestacion-de-servicios-que-apoya-formalizacion-de-empleo-en-Medellin> (accessed Jun. 07, 2021).
- [16] D. Fernández Astudillo, A. T. Peña Prias, I. de Sistemas, dfernandez@javerianacali.edu.co, and apena72@javerianacali.edu.co, “Diseño e implementación de Aplicativo móvil para

- solicitud de clases al instante.” *reponameVitela Repos. Inst. PUJ*, Jun. 2019, Accessed: Mar. 19, 2021. [Online]. Available: <http://vitela.javerianacali.edu.co/handle/11522/11992>
- [17] L. Martínez, G. Antonia, and T. Okamura, “El Design Thinking aplicado en el desarrollo de un Sistema de Información , permite incrementar la satisfacción de los operarios al reducir los tiempos de atención de Capital Humano El Design Thinking aplicado en el desarrollo de un Sistema de Información,” 2021.
- [18] “Entorno Global y Desempeño Económico del Valle del Cauca-Esteban Piedrahíta”.
- [19] A. F. L. Bedoya, “No Title,” <https://miro.com/app/board/uXjVO7OPJfo=/>, 2022.
<https://miro.com/app/board/uXjVO7OPJfo=/>
- [20] “AWS | Servicio de base de datos gestionada NoSQL (DynamoDB).”
https://aws.amazon.com/es/dynamodb/?trk=74365dcd-ec2d-40b2-a201-8d654d09350d&sc_channel=ps&sc_campaign=acquisition&sc_medium=ACQ-P%7CPS-GO%7CBrand%7CDesktop%7CSU%7CDatabase%7CDynamoDB%7CLATAMO%7CES%7CText&s_kwid=AL!4422!3!590500029988!p!!g!!dynamodb&ef_id=Cj0KQCjwhqaVBhCxARIsAHK1tiOm4jRBvuB84wcFryJkuYtUnuTLS6m4bezy38a1deOWxWiEGeZhh5QaAt9xEALw_wcB:G:s&s_kwid=AL!4422!3!590500029988!p!!g!!dynamodb (accessed Jun. 20, 2022).
- [21] D. De Paula, “Design Thinking in Practice : Understanding Manifestations of Design Thinking in Software Engineering,” pp. 1059–1069, 2019.

ANEXOS

Anexo 1 entrevista de profundidad

Entrevista Profundidad

Hola soy Andres Felipe López, estudiante de posgrado de Maestría en Ingeniería de Software de la Pontificia Universidad Javeriana Cali, Investigando la metodología Design Thinking cómo se adapta en los procesos de Desarrollo de Software enfocado en los servicios de Oferta y Demanda que presentan un índice de informalidad de empleos alto en la región del Valle del Cauca.

* Obligatorio

* Este formulario registrará su nombre, escriba su nombre.

1. ¿Cuál es su nombre? *

2. ¿Cuál es su edad? *

3. ¿Cómo se compone su núcleo familiar?

4. ¿Cuál es su ocupación? *

5. ¿Cuéntanos cómo es su día a día en breves palabras? *

6. La ocupación en la que te desempeñas ¿Cómo es el flujo de demanda por parte de los clientes? *

- Baja
- Media
- Alta

7. ¿Cuéntanos cómo es el proceso tuyo para adquirir un nuevo cliente? *

8. ¿Cuál es la plataforma de comunicación por el que te contactan? *

- Whatsapp
- Ninguna
- Voz a voz
- Facebook
- Instagram
-

Otras

9. ¿Cuál de las siguientes alternativas consideras lo difícil de adquirir clientes? *

Las plataformas de comunicación no son lo suficientemente efectivos

La economía de la región no pasa por buen momento

Ocupación poco requerida

Otras

10. ¿Cuál de las siguientes opciones se acercan a las necesidades de tu ocupación? *

Dar a conocer el trabajo realizado por medio de una herramienta efectiva

Cómo manejo las plataformas de comunicación están bien

Reforzar los medios de comunicación actuales de manera mas efectiva

Otras

11. En este mundo de adaptación al cambio, ¿Consideras que tu ocupación debe dar un siguiente paso para aumentar la demanda de los clientes? *

Si

No

12. ¿Ayúdanos a mencionar soluciones similares a la planteada para los servicios de oferta y demanda? *

13. ¿En caso de plantearse una solución de ofrecer servicios de Oferta y Demanda cómo te parece el mejor escenario de una plataforma? *

- Página web
- Aplicación móvil para Android
- Chat automático por Whatsapp
- Aplicación móvil para Apple

Otras

14. Agradeciéndole por tu tiempo ya para finalizar, ¿Qué características debería tener la plataforma de servicios de Oferta y Demanda? *

- Publicar los trabajos realizados
- Ranking de calificación de sus servicios
- Costos de sus servicios ofrecidos
- Chat de conversación
- Calificación de clientes
- Trazabilidad de la orden de trabajo
- Recomendaciones de sus servicios ofrecidos a los clientes
-
- Otras

Anexo 2 Requerimientos funcionales

A continuación, se listan los requerimientos funcionales del sistema captados de la exploración de la información de acuerdo a las necesidades de los usuarios manifestadas mediante las entrevistas y encuestas.

CATEGORIA	PROPUESTA REQUISITOS REFINADOS	1. Priorización in/out	2. Priorización tres niveles (A = Alta, M = Medio, B = Baja)
RNF-ACCESIBILIDAD-001	El sistema deberá ser en ambiente mobile para que pueda ser utilizado desde cualquier lugar	IN SECCION A Y B	A
RNF-INTEGRACION-001	El sistema deberá permitir realizar pagos por medio de la aplicación en los diferentes medios de pago disponibles	IN SECCION A Y B	M
RF-USABILIDAD-001	El sistema deberá mostrar al usuario una interfaz facil de interpretar, usable para que el usuario disponga de todas las funcionalidades de la aplicación	IN SECCION A Y B	M
RF-SERVICIOS-001	El sistema deberá permitir al usuario cliente ver la trazabilidad de los servicios obtenidos	IN SECCION A Y B	B

RF-SERVICIOS-002	El sistema deberá permitir al usuario ofertante del servicio ver la trazabilidad de los servicios prestados.	IN SECCION A Y B	M
RF-COMUNICACION-001	El sistema deberá permitir chat o conversacion sobre los servicios ofrecidos a los diferentes usuarios del sistema (Cliente, Ofertante)	IN SECCION A Y B	M
RNF-COMUNICACION-002	El sistema deberá permitir integrarse con las APIS de Whatsapp para la mensajería instantanea para integrar la comunicación entre los usuarios	IN SECCION A Y B	B
RF-SERVICIOS-003	El sistema deberá permitir anunciar los servicios por medio de diferentes filtros cómo mejor servicio calificado, precio, más solicitado	IN SECCION A Y B	M
RF-CONFIGURACION-001	El sistema deberá permitirle al usuario administrador configurar las diferentes parametrizaciones en el sistema cómo inactivación de servicios, usuarios.	IN SECCION A Y B	A
RF-CALIFICACION-001	El sistema deberá permitirle al usuario ofertante calificar al cliente cómo pago oportuno, comunicación amigable	IN SECCION A Y B	M
RF-CALIFICACION-002	El sistema deberá permitirle al usuario cliente calificar al usuario ofertante del servicio cómo prestación oportuna del servicio	IN SECCION A Y B	M
RF-CONTROL ACCESO-001	El sistema deberá permitirle a los diferentes usuarios del sistema registrarse en el sistema con sus respectivos datos	IN SECCION A Y B	A
RF-CONTROL ACCESO-002	El sistema deberá permitirle al usuario loguearse en el sistema con el respectivo usuario y contraseña asignada	IN SECCION A Y B	A
RF-USABILIDAD-003	El sistema deberá permitir calificar la funcionalidad del sistema y realizar comentarios para mejora continua de la plataforma	IN SECCION A Y B	A
RF-USABILIDAD-004	El sistema deberá permitir guardar información off-line sobre transacciones críticas de la operación de la plataforma cómo guardar pagos, servicios, logueo	IN SECCION A Y B	B
RF-USABILIDAD-005	El sistema deberá permitir cargar contenido digital de los diferentes servicios ofertados en la plataforma	IN SECCION A Y B	A
RF-USABILIDAD-006	El sistema deberá desplegar la información o detalle de cada uno de los servicios ofertados cómo el precio, descuentos, características, ubicación geográfica	IN SECCION A Y B	A

RF-USABILIDAD-007	El sistema deberá permitir compartir los servicios en los diferentes medios o plataformas soportadas	IN SECCION A Y B	B
RF-CONTROL ACCESO-003	El sistema deberá solicitar información de condiciones de uso y avisos de privacidad	IN SECCION A Y B	A
RF-CONTROL ACCESO-004	El sistema deberá validar la identidad de los usuarios del sistema registrados	IN SECCION A Y B	A
RF-SERVICIOS-004	El sistema deberá permitir al usuario cliente buscar la oferta que requiera utilizar de sus servicios y abrir contacto con el usuario ofertante	IN SECCION A Y B	A
RF-SERVICIOS-005	El sistema deberá permitir al usuario cliente solicitar un servicio de acuerdo a los criterios de búsqueda	IN SECCION A Y B	A
RF-SERVICIOS-006	El sistema deberá permitir al usuario ofertante reservar el servicio solicitado por el usuario cliente y no estar disponible para otro servicio hasta liberar el servicio	IN SECCION A Y B	A
RF-SERVICIOS-007	El sistema deberá permitir al usuario cliente cuando solicite el servicio desplegar los diferentes medios de pago disponibles en la plataforma, sin este paso no es posible finalizar la transacción	IN SECCION A Y B	A
RF-SERVICIOS-008	El sistema deberá permitir al usuario cliente declinar la transacción del servicio donde el servicio quedará disponible de nuevo para que puede ser tomado por otro usuario ofertante	IN SECCION A Y B	A
RF-SERVICIOS-009	El sistema deberá permitir a los usuarios (Cliente, Ofertante), mostrar los estados del servicio confirmada la transacción (Pendiente, En proceso, Entregado, Finalizado)	IN SECCION A Y B	A
RF-SERVICIOS-010	El sistema deberá permitirle al usuario ofertante cambiar los estados del servicio (En proceso, Entregado, Finalizado)	IN SECCION A Y B	A
RF-SERVICIOS-011	El sistema deberá permitirle al usuario cliente recibir el código de la reserva cuando se confirme la transacción de la reserva en el sistema cuando finaliza el último proceso del medio de pago	IN SECCION A Y B	A

Tabla 5 Definición de requerimientos funcionales

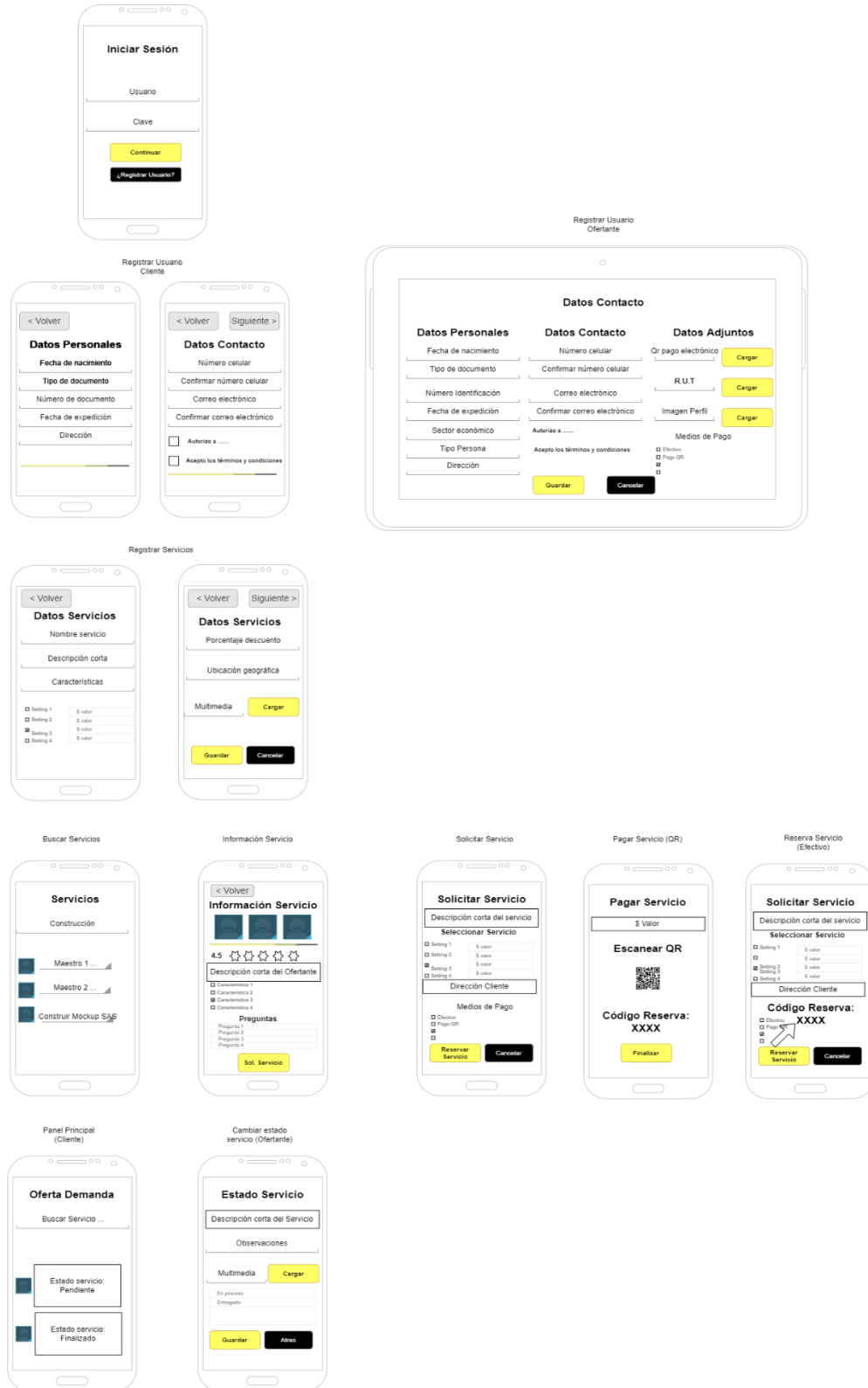
Anexo 3 Requerimientos no funcionales

A continuación, se describen cada uno de los requerimientos no funcionales con su respectivo ambiente, estímulo, respuesta y requerimientos asociados.

CATEGORÍA	PROPUESTA DE REQUISITOS NO FUNCIONALES
RNF - ACCESIBILIDAD - 001	El sistema deberá desarrollarse en ambiente Android para que pueda ser utilizado desde cualquier lugar.
RNF DISPONIBILIDAD - 0001 (Conectividad por medio de dispositivo)	Recurso: Conectividad recursos Estímulo: Detección de una falla de conectividad con el servidor central Artefacto: Servidor central Ambiente: S.O Linux activo, acciones cómo : Entorno de red y tráfico de red Respuesta: Dejar un rastro de la falla, notificar la falla, operar en modo degradado. Medida de la respuesta: 99% REQF asociado:
RNF DISPONIBILIDAD - 0002 (Sincronización de datos en dispositivo)	Recurso: Sincronización datos en dispositivo Estímulo: Sincronización de datos sin afectar rendimiento en dispositivo Artefacto: Dispositivo móvil Ambiente: Dispositivo móvil activo, acciones cómo : Latencia en red Respuesta: Actualizar registros completados. Medida de la respuesta: 99% REQF asociado:
RNF SEGURIDAD - 0001 (Autenticación usuario)	Recurso: Autenticación de usuario Estímulo: Envío datos del usuario Artefacto: Dispositivo móvil Ambiente: Dispositivo móvil activo, acciones cómo : Ataque en el sistema central Respuesta: Información cifrada, encolar transacción, respuesta satisfactoria al cliente. Medida de la respuesta: Respuesta no mayor al cliente de 1 minuto para recuperar sistema sino declinar transacción REQF asociado:
RNF SEGURIDAD - 0002 (Generar pagos electrónicos)	Recurso: Generar pagos electrónicos Estímulo: Un pago produce un evento inesperado de transacción Artefacto: Servidor central activo, sistema de pagos activo, latencia en red Ambiente: Operación normal Respuesta: Notificar al usuario inmediatamente el estado de la transacción. Medida de la respuesta: Notificar al usuario latencia de red. REQF asociado:

<p>RNF DESPLIEGUE- 0001 (Nueva versión del SW)</p>	<p>Recurso: Nueva versión del SW Estímulo: Nueva versión Artefacto: Nuevo módulo de software Ambiente: Despliegue parcial por segmento de usuarios Respuesta: Incorporar nuevo componente de software con obtención de resultados y fallas obtenidas Medida de la respuesta: Trazabilidad del proceso del nuevo módulo de software. REQF asociado:</p>
<p>RNF RENDIMIENTO- 0001 (Petición Usuario)</p>	<p>Recurso: Petición Usuario Estímulo: Evento esporádico Artefacto: Componente de software consultar servicios Ambiente: Carga máxima Respuesta: Sistema no genera respuesta Medida de la respuesta: Tiempo de recuperación máximo de un minuto para envío de respuesta. REQF asociado:</p>

Anexo 4 Prototipación



Anexo 5 Entrevista con experto

Entrevista a experto

Hola soy Andres Felipe López, estudiante de posgrado de Maestría en Ingeniería de Software de la Pontificia Universidad Javeriana Cali, Investigando la metodología Design Thinking cómo se adapta en los procesos de Desarrollo de Software enfocado en los servicios de Oferta y Demanda que presentan un índice de informalidad de empleos alto en la región del Valle del Cauca.

¿Cuál es su nombre?

Diana Isabel Riveros Ospina

¿Cargo que desempeñas?

Directora de innovación y emprendimiento en la Universidad Javeriana Cali

¿Cuéntanos un poco de tu experiencia con la metodología Design Thinking?

Yo soy diseñadora industrial de base, la metodología nace con un lenguaje que se asemeja a la manera de cómo se venia manejando desde el diseño se hace un lenguaje un poco más simple para poder comunicarse entre diferentes disciplinas. Llevo ya 15 años trabajando en este tema con la base del Design Thinking en ese sentido he dado clases, cursos y desde el centro de Innovación y emprendimiento está metodología es cómo digámoslo es el algoritmo base para todos los procesos de formación, desarrollo de tecnologías, y construcción de nuevos negocios desde la incubadora.

Pongámoslo de está manera: El diseño es el primer paso y luego debemos de trabajar por la implementación para chequear que somos innovadores.

¿Cómo observas la economía actual en la región del Valle del Cauca?

Es una economía muy compleja muy basada en pocas industrias que son grandes pero que no necesariamente con un despliegue de progreso en todos los ámbitos de la región, la actitud emprendedora cayó según el último estudio del GEM donde no se está generando empleo, innovación; hay una generación de pesimismo, pues entonces nuestra misión cómo Universidad es tratar de luchar contra eso y generar un tipo de emprendimiento más sofisticado, más de base tecnológica para poder salir de este tema que evidentemente la pandemia, situación social, nos ha pegado muy duro por todos estos factores la situación económica está compleja.

¿A pesar de todas estas dificultades crees que se cambien de una manera u otras las metodologías o se deben adaptar a las nuevas necesidades de la economía?

No, el pensamiento de diseño me parece que sigue estando vigente sobre todo porque particularmente el Design Thinking habla de diseño centrado en las personas (User Center Design), (Human Center Design), entonces la economía e inclusive la ecología todo está centrado en las personas entonces la metodología sigue siendo válida.

¿Cómo crees que se pueda generar una propuesta de valor enfocado en el tema de investigación de Oferta y Demanda?

Entendiendo muy bien que es parte de la metodología, es entendiendo muy bien a los diferentes actores quién está en la oferta y quién está en la demanda y tratándoles de ofrecer una propuesta de valor diferente a las que ya están que les haga muy fácil la vida. Digamos que una clave para generar propuestas de valor exitosas es apelar a esa naturaleza humana que quieren todo más fácil, rápido, sin dolor, cómodo. Hay cosas ya en este tema de conexión de Oferta y Demanda ya existen varios desarrollos validarlos muy bien, entender donde está ese factor diferencial que podríamos generar nosotros desde acá.

¿De acuerdo con tu visión, cómo observas la informalidad en la región del Valle del Cauca?

Mira que la informalidad es gigante en el Valle del Cauca, entonces allí hay un potencial porque hay mucha oferta y de verdad que mucha demanda es decir la gente confía mucho en su amigo que tiene el negocio y se dan muchas cosas por Whatsapp, Instagram, aquí en el Valle del Cauca. Con eso dicho si uno trata de formalizar un poquito eso puede generar una barrera de ingresos que es lo que ha pasado con las aplicaciones que ya existen, porque la gente empieza a asociar formalidad desde la tecnología con costos, compromisos; el emprendedor hay veces puede y hay veces no puede cómo el emprendimiento es por necesidad, mucha gente tiene empleo y emprendimiento entonces necesita no comprometerse con algunas fechas, horarios para manejar su tiempo, yo veo que hay un mercado grande potencial pero si hay muchos aspectos de la cultura tanto de la oferta y la demanda que hay que resolver con una propuesta de valor y me parece super interesante lo que haz hecho.

¿Los usuarios por medio de las encuestas manifiestan mucho una app que englobe las comunicaciones, conexión de oferta y demanda?

Muchos de los usuarios con poder adquisitivo que les gusta tener este tipo de servicios son mayores y no son tan buenos en tecnologías y el otro punto es que cada vez que hay un desarrollo de app yo no quiero otra app, entonces entre más sencillo, liviano, mucha gente no tiene datos también cómo para tener en cuenta esos datos que pueden afectar la propuesta.

¿Cómo crees que se puede ir involucrando una plataforma en una primera Fase?

Las aplicaciones por lo regular son muy difíciles de monetizar es lo que siempre decimos aquí en la incubadora por que la gente llega con la idea de hacer una aplicación, resulta que la aplicación es simplemente un medio de comunicación, la gente que espera hacer negocio con la aplicación no está teniendo en cuenta cómo por ejemplo las grandes aplicaciones como Uber, Aws, Tinder tienen un negocio detrás que no tiene nada que ver con ingresar a la aplicación, acceso ni nada del funcionamiento de la aplicación. Entonces hay es que centrarse en cómo por detrás pongo ese modelo de negocio con monetización quien me va a pagar y porque por aparecer en donde, y hay otra cosa que debes de tener en cuenta es el tema de las tendencias, por ejemplo, las redes sociales que están funcionando muy bien son las de video. Por ejemplo Tik Tok se ha llevado al resto de redes sociales. Nose si haz visto una cuenta en Tik Tok que se llama @elpridelcentro es una persona que va al centro de Cali la gente le escribe donde encontrar cosas por ejemplo zapatillas, camisetas, gorras etc. La persona va a esos lugares y hace los reportajes mostrando los negocios entonces es una forma de conectar Oferta con Demanda de una manera muy sencilla y es gratis porque ya está en Tik Tok de una forma efectiva. Entonces es un cruce de tendencia con la tecnología que es de muy fácil acceso y una propuesta de valor que está buscando conectar oferta con demanda entonces para que lo tengas en cuenta con el modelo de monetización. Otro es punto importante para las aplicaciones que funcionan por monetización sólo cuando tienen millones de usuarios en el proceso es capturar usuarios en un principio.

¿Cómo se lograría enfocar esta idea con el Design Thinking en darle una propuesta de valor al mercado?

Revisar muy bien a los usuarios y ver cómo se les cierra las brechas en lo esencial enfocado en los problemas.

¿Es recomendable no iniciar en aplicaciones por suscripción o este tipo de aplicaciones por pago de uso de acuerdo a lo que venimos validando?

Yo no seria el usuario para validar este punto, pero con lo validado desde punto con la metodología aplicada con los usuarios que ya has tenido acercamientos a parte siendo ese usuario tipo que estas buscando. Pero pues la recomendación es esa uno normalmente no paga por acceder a estas aplicaciones, imaginémonos es cómo un directorio entonces uno no paga por utilizar esto, uno paga es por el servicio. Entonces cómo por ejemplo un Rappi tanto los que buscamos cómo los que ofrecen pues pagamos para que la aplicación exista entonces en ese caso hay un porcentaje que va para la aplicación pero nuevamente eso depende de haber generado muchísimos usuarios primero. Allí viene luego los inconvenientes ya la gente baja su uso porque el porcentaje es muy alto para cubrir el servicio de la aplicación, y esto pasa porque ellos quieren crecer y no basta con el porcentaje que se está obteniendo.

¿Cómo se podría controlar el tema financiero para que los usuarios no realicen transacciones por aparte si el enfoque no estaría en la monetización y se fidelice la plataforma?

Bueno depende si tu negocio es la plataforma tienes que monetizar las transacciones porque sino seria cómo un servicio a la humanidad pues chévere pero en este caso no, bueno la otra posibilidad es que por ejemplo de un grupo de emprendedores que se unen para apoyar la idea o una empresa en particular que quiera hacer la conexión.

¿Por dónde te irías tú, para hacer un buen modelo de negocio?

Pues si esa fuera la idea de conectar la oferta y demanda tipo Rappi, pero entonces no mirando tanto en depender de esa cantidad de usuarios y más desde la empatía con el usuario que algo especial u oferta distinta, que plus tiene está plataforma versus las muchas que ya existen para eso. Porque por ejemplo ya existen unas para carpintero, amas de casa entre otras, de igual manera existen empresas que ya tienen sus propias aplicaciones, yo lo pensaría en el cómo volvería en una red de redes, una aplicación que cubre otras por ejemplo que a veces también funciona de esta manera.

Ya para finalizar y agradecerte por este tiempo, ¿Teniendo en cuenta algunos procedimientos o experiencia manejada con Frameworks para llevar un proyecto de este tipo que se tenga involucrado al usuario, la necesidad y la solución tecnológica que recomendarías para llevarlo a cabo?

Te recomiendo realizar muchas validaciones baratas en papel, Whatsapp, cosas muy rápidas, con una muy buena cantidad de usuarios e ir registrando en tu bitácora que aprendiste de cada una de esas validaciones; es decir que cambios deberías estar implementando muy ágilmente y luego con esos cambios vuelves y lanzas con otros usuarios la propuesta para ver que respuesta tiene entonces la clave del Framework es 1: Centrarse en el usuario que ya lo hablamos y 2: Validar con ese usuario por muchas veces e ir aprendiendo cada vez sacando unos Insights cada vez que te vayan puliendo tu producto de una manera muy ágil y muy barata donde no tengas que invertir ni mucho tiempo ni mucho dinero en eso hasta que tú ya veas que llegaste a un punto que esto ya se puede lanzar.

¿Existe alguna técnica o estrategia para obtener esos usuarios masivos?

Pues bueno hay plataformas de validación pero funcionan mucho al ingles entonces los usuarios son de otra parte, en Colombia recomendamos Whatsapp puedes mandar muchas cosas, la Universidad te sirve porque hay muchos usuarios diversos.

Anexo 6 Encuesta validación de prototipo

Validación de prototipación

Hola soy Andres Felipe López, estudiante de posgrado de Maestría en Ingeniería de Software de la Pontificia Universidad Javeriana Cali, Investigando la metodología Design Thinking cómo se adapta en los procesos de Desarrollo de Software enfocado en los servicios de Oferta y Demanda. Por medio de este cuestionario se pretende saber su opinión acerca del prototipo de la aplicación (diseño de la futura aplicación).

* Obligatoria

1

¿Cuál es su nombre? *

2

¿Cuál es su ocupación? *

3

Validación de prototipo para el Inicio de sesión. ¿Qué opinas de la interfaz de la pantalla?



- Buena
- Regular
- Mala

4

Considerando el registro de un usuario Cliente, ¿Cómo calificas la completitud de la información siendo 1 el más bajo, 5 Completo? *



- 1
- 2
- 3
- 4
- 5

5

Considerando el registro del usuario Ofertante (Ofrece los servicios), ¿Cómo calificas la completitud de la información siendo 1 el más bajo, 5 Completo?

*



- 1 2 3 4 5
-

6

Si usted fuera a ofrecer un servicio en una plataforma, ¿Cómo observas el contenido de la pantalla para crear un servicio y pueda ser visualizado por sus Clientes?

*



- Completo
- Regular
- Malo

7

¿Cómo calificas la facilidad de reserva de un servicio?

*



- 1 2 3 4 5
-

8

Considerando un mejor diseño de interfaz, ¿Que combinaciones tendrías en cuenta? *

	Verde, Negro, Blanco	Amarillo, Negro, Blanco	Azul, Negro, Blanco	Grande	Mediana	Pequeña	Otro
Color	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Letra	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Imagenes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9

Agradeciéndole por su valioso tiempo, ¿Qué comentarios o aportes adicionales agregarías en el diseño y las funcionalidades presentadas? *

