

Universidad ORT Uruguay

Facultad de Ingeniería

CompraCerca

Entregado como requisito para la obtención del título de Licenciado
en Sistemas

Agustín Taruselli - 143113

Alejandro Mamchur - 173658

Ignacio Peirano - 151322

Tutor: Amalia Álvarez

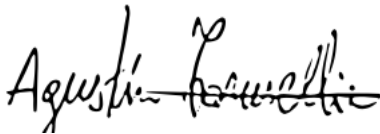
2019

Declaración de autoría

Nosotros, Agustín Taruselli, Alejandro Mamchur e Ignacio Peirano, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano.

Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el Proyecto final de carrera de Licenciatura en Sistemas.
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad.
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra.
- En la obra, hemos acusado recibo de las ayudas recibidas.
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros.
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Agustín Taruselli – Setiembre de 2019



Alejandro Mamchur - Setiembre de 2019



Ignacio Peirano - Setiembre de 2019

Agradecimientos

Queremos dedicar esta sección para agradecer a todas aquellas personas que por su colaboración, apoyo y guía nos permitieron potenciar y enriquecer la realización de este proyecto.

En primer lugar queremos mencionar a nuestra tutora Amalia Álvarez quien nos apoyó y estuvo en constante disposición con el equipo durante todo el transcurso del proyecto, aportando una visión crítica y objetiva al momento de definir los requerimientos, valor del producto, la metodología de trabajo y otros aspectos vitales para el éxito del proyecto.

Además queremos agradecer al licenciado Javier Álvarez por su colaboración como experto de marketing al momento de evaluar, aportar y validar el valor del producto.

También al ingeniero en sistemas Tomás Ferrero que si bien era ajeno al proyecto, con buena disposición aportó su conocimiento y experiencia en el desarrollo de este documento.

A Rafael Betancur, quien al comienzo del proyecto nos aportó su experiencia y conocimiento a la hora de madurar y entender el valor del producto.

Finalmente a familiares y amigos que siempre estuvieron presentes para apoyarnos desde que emprendimos la carrera.

A todos ellos un fuerte agradecimiento,

Agustín, Ignacio y Alejandro

Abstract

Compra Cerca es un emprendimiento que surge como una solución a la necesidad de encontrar un producto en una tienda física cercana a la ubicación del usuario brindando una aplicación para facilitar esta tarea. La aplicación tiene como objetivos principales que los compradores puedan encontrar el producto que buscan en los comercios más cercanos a su ubicación y que los comercios logren más visibilidad y por consiguiente aumentar el flujo de clientes a su local.

El foco del proyecto es el desarrollo de una aplicación móvil para los compradores y vendedores y un sistema web para los administradores. La aplicación permite a los compradores buscar los productos que necesita y poder ver en el mapa los comercios que pueden tener este producto. Por otra parte los comercios podrán cargar ofertas que serán mostradas a los compradores en sus búsquedas. El sistema web para los administradores permite consultar datos estadísticos y administrar usuarios de la aplicación.

Se decidió utilizar *ionic* para el desarrollo del *frontend* para lograr realizar una aplicación web que sea *responsive* (que se adapte a diferentes tamaños de pantalla) y multiplataforma ya que apuntamos a que se utilice en dispositivos móviles. Para el *backend* se utilizó *.Net framework* y el lenguaje *C#* ya que esta tecnología permite cubrir las necesidades del proyecto y además es en la que más experiencia tiene el equipo.

Uno de los desafíos más importantes que enfrentó el equipo en el desarrollo del proyecto fue el algoritmo que sugiere las categorías de los productos durante la búsqueda que ingresa el comprador.

La metodología de trabajo utilizada fue híbrida, utilizando prácticas tradicionales para la construcción del producto y una adaptación de SCRUM para la gestión del proyecto. En la etapa final del desarrollo se realizaron pruebas con usuarios potenciales del sistema para obtener retroalimentación de los usuarios finales. Se realizaron además, internamente, pruebas funcionales de la aplicación para evaluar el grado de cumplimiento de los requerimientos del producto.

Durante el desarrollo del proyecto los integrantes del equipo adquirieron conocimientos sobre nuevas tecnologías y aprendieron a enfrentar y resolver desafíos e imprevistos mediante el trabajo en equipo y la colaboración.

Palabras Clave

Universidad ORT Uruguay, C#, MVC, Entity framework, Geofencing, Angular, Ionic.

Glosario

A

API: La interfaz de programación de aplicaciones, conocida también por la sigla API, en inglés, application programming interface, es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

B

Backend: es la parte del software que procesa la entrada desde la interfaz de usuario que interactúa con el usuario.

Burndown: Gráfica que representa cómo se van completando los puntos de las tareas de un sprint

C

Commodity: *Bien o producto que tiene valor o utilidad, y un muy bajo nivel de diferenciación o especialización.*

Commit: Confirmar un conjunto de cambios provisionales de forma permanente.

D

Deploy:

E

E-commerce: consiste en la distribución, venta, compra, marketing y suministro de información de productos o servicios a través de Internet. Originalmente el término se aplicaba a la realización de transacciones mediante medios electrónicos, como por ejemplo el intercambio electrónico de datos

F

Frameworks: Un framework, entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Frontend: es la parte del software que interactúa con los usuarios.

G

H

I

IEEE: Instituto de Ingeniería Eléctrica y Electrónica es una asociación mundial de ingenieros dedicada a la normalización y el desarrollo en áreas técnicas.

J

Javascript: Lenguaje de programación de tipo Scripting.

K

L

M

MVP: Sigla que proviene del inglés (Minimum Viable Product), que significa un producto con suficientes características para satisfacer a los clientes iniciales, y proporcionar retroalimentación para el desarrollo futuro.

N

O

P

PYMES: Pequeña y medianas empresas

Q

QA: Sigla que proviene del inglés (Quality Assurance), el aseguramiento de la calidad es el conjunto de actividades planificadas y sistemáticas aplicadas en un sistema de gestión de la calidad para que los requisitos de calidad de un producto o servicio sean satisfechos.

R

RF: Requerimiento Funcional

RNF: Requerimiento No Funcional

Responsive: El diseño web responsive o adaptativo es una técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos. Desde ordenadores de escritorio a tablets y móviles

S

Sprint Planning: es una de las ceremonias dentro del marco de trabajo Scrum.

Scrum: Marco de trabajo para desarrollo ágil de software.

Scrum Master: Término del inglés que en español se traduce a facilitador de proyectos, es la figura que lidera los equipos en la gestión ágil de proyectos.

Scripting: Lenguajes de programación que no necesita ser compilado, es decir solo necesitan ser interpretados.

SQA: Sigla que proviene del inglés (Software Quality Assurance), que en español se traduce a las actividades para el aseguramiento de la calidad del software.

Story points: Los puntos de la historia o Story Points (SPs) son una unidad de medida para expresar o estimar el esfuerzo general que sería necesario para fabricar un elemento de un producto o cualquier otra parte de la producción.

T

Taskboards: Término del inglés que en español se traduce a tablero de tareas, es una herramienta utilizada por individuos, equipos u organizaciones para representar el trabajo y su camino hacia la finalización.

TDD: Test-driven development es una práctica de ingeniería de software que involucra otras dos prácticas: Escribir las pruebas primero y Refactorización. Para escribir las pruebas generalmente se utilizan las pruebas unitarias.

Tester: Los probadores de software (también conocidos como **testers**, su denominación en inglés) planifican y llevan a cabo pruebas de software de los ordenadores para comprobar si funcionan correctamente.

Testing: Las pruebas de software (en inglés *software testing*) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada.

U

URL: es una sigla del idioma inglés correspondiente a Uniform Resource Locator (Localizador Uniforme de Recursos). Se trata de la secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro del entorno de Internet para que puedan ser localizados.

UI: Sigla que proviene del inglés (User Interface), que en español se traduce a Interfaz de usuario.

User Stories: Representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos.

V

W

X

Y

Z

Contenido

Declaración de autoría	2
Agradecimientos	3
Abstract	4
Palabras Clave.....	5
Glosario	6
1. INTRODUCCIÓN.....	14
1.1. Introducción al proyecto	14
1.2. Composición del documento	14
1.3. Proceso de selección de la idea.....	15
1.4. Propósito, motivación y alcance del proyecto	16
1.5. Equipo de trabajo	16
2. DESCRIPCIÓN DEL PROYECTO	18
2.1. Objetivo del proyecto	18
2.2. Interesados.....	19
2.3. Descripción del problema.....	20
2.3.1. Desde la perspectiva de los consumidores	20
2.3.2. Desde la perspectiva de los comerciantes	20
2.4. Descripción de la solución	21
2.4.1. Funcionalidades de la solución propuesta.....	22
3. PROCESO Y CICLO DE VIDA	24
3.1. Ciclo de vida	24
3.1.1. Ciclo de vida prototipado evolutivo	24
3.2. Metodología de gestión de proyectos.....	25
3.3. Roles	27
4. INGENIERÍA DE REQUERIMIENTOS.....	28
4.1. Proceso de análisis de requerimientos.....	28
4.1.1. Relevamiento del problema.....	28
4.1.2. Análisis	29
4.1.3. Especificación.....	29
4.1.4. Validación	30
4.2. Selección de funcionalidades para MVP.....	31
4.3. Requerimientos Funcionales	33
4.4. Requerimientos no funcionales	36
4.4.1. Seguridad	36
4.4.2. Disponibilidad	36
4.4.3. Usabilidad	36
4.4.4. Performance	37
4.4.5. Eficiencia	37
4.4.6. Modificabilidad	37

5. ARQUITECTURA.....	38
5.1. Introducción	39
5.2. Vista global del sistema.....	39
5.3. API Rest	40
5.4. Patrón MVC	41
5.5. Patrones Composite y Fachada	42
5.5.1. Composite.....	42
5.5.2. Fachada.....	42
5.6. Angular CLI	43
5.7. Objetivos de calidad.....	44
5.7.1. Disponibilidad	44
5.7.2. Seguridad	44
5.7.3. Modificabilidad	44
5.7.4. Performance	45
5.7.5. Usabilidad	45
5.8. Comunicación del sistema con Mercadolibre	46
5.9. Diagramas del sistema	46
5.9.1. Diagrama de paquetes, vista de alto nivel.....	46
5.9.2. Catálogo de entidades	48
5.9.3. Decisiones de diseño	49
5.9.4. Diagrama de Secuencia.....	50
5.9.5. Vista de Despliegue.....	51
6. PROCESO DE DESARROLLO	52
6.1. Características del equipo	52
6.2. Responsabilidades	52
6.3. Estrategia de desarrollo	53
6.3.1. Requerimientos Incluidos en el MVP	55
7. GESTIÓN DEL PROYECTO	56
7.1. Herramientas utilizadas para la gestión del proyecto	57
7.1.1. Planificación de tareas.....	57
7.1.2. Estimación y registro de esfuerzo.....	61
7.1.3. Seguimiento de las tareas.....	63
7.2. Riesgos	64
7.3. Gestión de las comunicaciones	66
7.3.1. Comunicación interna.....	67
7.3.2. Comunicación hacia la tutoría	67
7.3.3. Comunicación hacia el socio de marketing.....	67
7.3.4. Plan de comunicaciones	68
7.4. Métricas de esfuerzo.....	69
7.4.1. Esfuerzo total (en horas) por recurso	69
7.4.2. Esfuerzo de los recursos por iteración.....	70
7.4.3. Esfuerzo por rol.....	72
7.4.4. Esfuerzo total (horas) por Sprint.....	73
7.4.5. Esfuerzo por rol por Sprint.....	74
8. GESTIÓN DE CALIDAD	76

8.1.	Objetivos de calidad a nivel de producto	76
8.2.	Objetivos de calidad a nivel del proceso de desarrollo	77
8.3.	Planificación de SQA	77
8.4.	Actividades de SQA	78
8.4.1.	Selección de estándares	78
8.4.2.	Revisiones del proyecto	81
8.4.3.	Revisiones internas	81
8.4.4.	Verificaciones	82
8.4.5.	Validaciones	83
8.4.6.	Controles sobre los repositorios	85
8.5.	Testing	86
8.5.1.	Planificación de las pruebas	86
8.5.2.	Ejecución de las pruebas	88
8.6.	Métricas de Testing	89
1.1.1.	Cantidad de defectos reportados por iteración	89
1.1.2.	Cantidad de defectos reportados por funcionalidad	90
1.1.3.	Defectos reportados vs defectos corregidos	91
1.1.4.	Defectos reportados vs Casos ejecutados	92
9.	GESTIÓN DE LA CONFIGURACIÓN	92
9.1.	Elementos de configuración	93
9.2.	Herramientas de gestión	93
9.3.	Gestión de versiones	94
9.3.1.	Versionado de código de fuente	94
9.3.2.	Versionado de la documentación	96
9.4.	Respaldos físicos	97
10.	LECCIONES APRENDIDAS	98
11.	CONCLUSIONES	100
12.	REFERENCIAS	101
13.	ANEXOS	103
13.1.	Anexo 1 – Análisis de la Situación Actual	103
13.2.	Anexo 2 - Acceso a créditos por parte de las PYMES en Uruguay	104
13.3.	Anexo 3 - Uso potencial del crédito de las PYMES en marketing/publicidad	104
13.4.	Anexo 4 - Resultado encuesta de compras online	105
13.5.	Anexo 5 – Canvas	107
13.6.	Anexo 6 – Investigación de Mercado	107
13.6.1.	Problemas detectados de las alternativas de marketing	107
13.6.2.	Problemas relacionados a los consumidores finales	109
13.6.3.	Oportunidades	109
13.7.	Anexo 7 – Análisis tecnológico <i>Geofencing</i>	110
13.7.1.	¿Qué es <i>Geofencing</i> ?	110
13.7.2.	¿Por qué <i>Geofencing</i> ?	110
13.7.3.	¿Por qué es aplicable a nuestro negocio?	110
13.8.	Anexo 8 – Carta de Experto	111

13.9.	Anexo 9 – Especificación de Requerimientos.....	113
13.9.1.	Alcance del producto	113
13.9.2.	Descripción general.....	113
13.9.3.	Requerimientos funcionales	113
13.9.4.	Requerimientos no funcionales	121
13.10.	Anexo 10 – Heurísticas de Nielsen	122
13.11.	Anexo 11 – Resultados de encuestas a compradores.....	124
13.12.	Anexo 12 – Resultados de encuestas a vendedores	126
13.13.	Anexo 13 - <i>Mocks</i> creados en Figma	131
13.14.	Anexo 14 - Planilla de selección de tecnologías	134

1. INTRODUCCIÓN

1.1. Introducción al proyecto

El presente documento tiene como objetivo describir el proyecto de grado "CompraCerca", que se ha realizado como cumplimiento al requisito para la obtención del título de Licenciatura en Sistemas de la Universidad ORT Uruguay.

El grupo de alumnos que realizó este proyecto estuvo integrado por Alejandro Mamchur, Ignacio Peirano y Agustín Taruselli. El proyecto comenzó en el mes de marzo de 2019 y culminó en el mes de setiembre de 2019.

La tutoría fue realizada por Amalia Álvarez, quien guió, orientó y dio seguimiento al grupo y al avance del trabajo el transcurso del proyecto.

En el mes de febrero el equipo de alumnos comenzó a participar en el proceso de selección y validación de las ideas de proyecto de manera conjunta con la Universidad ORT. Durante el primer mes se analizaron diferentes alternativas e ideas potenciales de emprendimientos que se fueron descartando por su viabilidad como negocio más allá del ámbito puramente académico. Este proceso terminó en la presentación de un anteproyecto que fue validado y aprobado por la Universidad ORT.

1.2. Composición del documento

El presente documento está compuesto por las siguientes secciones que se describen y justifican el trabajo realizado por el equipo para llevar adelante el proyecto.

- **Descripción del proyecto:** en este capítulo se detalla el objetivo del proyecto, se identifican los interesados y se describe el problema y la solución propuesta por el equipo para resolver el problema identificado.
- **Proceso y ciclo de vida:** en este capítulo se explica el ciclo de vida elegido para el proyecto y se enuncian los motivos de la selección del mismo. Se explica y justifica además la metodología de trabajo a aplicar y los roles que intervienen en el proyecto.
- **Ingeniería de requerimientos:** aquí se detalla el proceso de ingeniería de requerimientos por el que pasó el equipo de trabajo, las técnicas y métodos utilizados para relevar y validar los requerimientos funcionales y no funcionales. Se incluyen además los requerimientos relevados.
- **Arquitectura:** en esta sección se explica la arquitectura que se utilizará para el proyecto, se explican las decisiones de diseño tomadas, patrones arquitectónicos y de diseño y se incluyen diferentes diagramas que dan un mejor entendimiento del sistema desde diferentes puntos de vista.
- **Proceso de desarrollo:** este capítulo describe características de los integrantes del equipo de trabajo, los roles que tomaron durante el proyecto, la manera en que se trabajó durante el proyecto y las funcionalidades incluidas en el alcance del producto final.

- **Gestión del proyecto:** esta sección habla sobre las herramientas que se utilizan para la gestión de las tareas, planificación, seguimiento del avance, identificación de los riesgos y la gestión de las comunicaciones. Se incluyen además algunas métricas de interés.
- **Gestión de la calidad:** este capítulo trata sobre las tareas realizadas durante el proyecto que tienen como fin el aseguramiento de la calidad. Se detallan los objetivos de calidad a nivel de producto y de procesos, la planificación de las actividades a realizar para el aseguramiento de la calidad, los estándares a utilizar, las revisiones que se realizarán y lo respectivo al *testing* (planificación y ejecución de pruebas).
- **Gestión de la configuración:** este capítulo detalla de qué manera se lleva adelante la gestión de los elementos de configuración, los elementos a tener en cuenta, se justifica la selección de los repositorios de código y documentación y cómo se gestionan estos repositorios.
- **Lecciones aprendidas:** se incluyen en este capítulo todas las lecciones que fue aprendiendo el equipo de trabajo durante el transcurso del proyecto que se consideran que dejan una enseñanza para futuros proyectos y que son fruto de errores cometidos o de aspectos identificados a mejorar.
- **Conclusiones:** esta sección incluye las conclusiones que se sacaron al final del proyecto, respecto al producto construido, la manera de trabajar, experiencias y situaciones por las que tuvo que pasar el equipo de trabajo durante el desarrollo del proyecto.

1.3. Proceso de selección de la idea

En febrero de 2019 los integrantes del equipo comenzaron a reunirse y a participar en el taller de emprendimientos que brinda la Universidad ORT para evaluar las ideas a presentar para su aprobación. Luego de algunas semanas de trabajo en las que se analizaron y discutieron varias opciones y potenciales ideas el equipo se decantó por dos opciones finales, una que implicaba el desarrollo de un sistema para una empresa que se presentó en la feria de proyectos que realiza la Universidad ORT y el emprendimiento propio.

Se realizaron algunas reuniones con la empresa de la feria de proyectos para tener una idea más clara de los requerimientos y desafíos que presentaba esta opción y al mismo tiempo se trabajó con la idea del emprendimiento en el taller brindado por la Universidad ORT.

Llegado el momento de presentar formalmente las opciones para su aprobación, el equipo se decidió por el emprendimiento como primera opción ya que presentaba desafíos tecnológicos interesantes y además presentaba una motivación extra para los integrantes, la idea de trabajar en un emprendimiento propio.

1.4. Propósito, motivación y alcance del proyecto

El propósito del proyecto es crear una solución que conecte comercios y consumidores cercanos para favorecer así la adquisición de productos/servicios en tiendas físicas.

El análisis de la situación actual de las PYMES (ver [Anexo 1 - Análisis de la Situación Actual](#)) y los problemas detectados en sus alternativas disponibles para aumentar su público, sumado a las necesidades de los clientes, permitieron detectar las siguientes oportunidades a atacar con nuestra propuesta:

- Lograr disminuir la brecha entre la intención de compra y una compra concretada de parte de los clientes.
- Lograr una alternativa abordable en lo económico para las empresas.
- Conseguir una alternativa con resultados tangibles a corto plazo para lograr la credibilidad de la tecnología.
- Optimizar el tiempo de los clientes.

Como consumidores nos hemos enfrentado a la situación de necesitar un producto/servicio, buscar en internet los comercios que lo ofrecen y descubrir posteriormente que teníamos un comercio más cercano a nuestra ubicación que no aparecía en los resultados de búsqueda.

Por lo tanto, decidimos crear una herramienta que mejore esta búsqueda, mostrando todos los comercios cercanos y no solamente los más conocidos, que son los que generalmente aparecen en los resultados de los buscadores. Con esta herramienta los consumidores accederán con mayor rapidez al producto que buscan y los comercios ganarán visibilidad ante los clientes, aumentando potencialmente sus ventas.

Para cumplir con lo anteriormente mencionado nos basamos en la aplicación de las prácticas y técnicas de ingeniería de software adquiridas durante la carrera.

En cuanto al alcance, se propuso realizar una aplicación web que será utilizada por los compradores y vendedores desde sus dispositivos móviles y además un sistema web que será utilizado por los administradores del sistema para realizar tareas de administración.

1.5. Equipo de trabajo

Para la asignación de roles y responsabilidades de cada integrante del equipo se tuvieron en cuenta habilidades y preferencias personales de manera de que cada uno de los integrantes se sienta cómodo en las tareas a desempeñar y logre así desempeñarse de la mejor manera durante el transcurso del proyecto.

Integrante/Rol	Gerencia de proyecto	Ing. Requerimientos	SQA	Arquitectura	SCM	Desarrollo	Testing
Alejandro				X		X	
Ignacio		X			X	X	
Agustin	X		X				X

Tabla 6.1 – Roles

Cada integrante, además de cumplir con el rol principal asignado en cada etapa del desarrollo del proyecto, brindó soporte y apoyo a las tareas de los demás.

2. DESCRIPCIÓN DEL PROYECTO

En el siguiente capítulo se dará una descripción general del proyecto; incluyendo los objetivos, los interesados, la descripción del problema y del sistema que lo resuelve.

2.1. Objetivo del proyecto

A comienzos del proyecto el equipo se planteó ciertos hitos a cumplir que si bien no están directamente relacionados con el objetivo del producto, colaboran al éxito del proyecto como tal.

Los hitos definidos son los siguientes:

- Cumplir los cronogramas y plazos establecidos.
- Respetar la metodología de trabajo definida.
- Cumplir con las responsabilidades del rol que se toma en los diferentes momentos del proyecto.
- La calidad del trabajo debe ser acorde a un proyecto de grado.
- Aplicar los conocimientos de ingeniería de software adquiridos durante la carrera para contribuir al éxito del proyecto.

El equipo es responsable de realizar seguimiento al cumplimiento de los hitos y de corregir lo necesario en caso de incumplir con alguno.

2.2. Interesados

Ante el comienzo de un proyecto de software es importante identificar los interesados en el mismo para tener claro quienes tienen influencia (ya sea positiva o negativa) y también el poder que tiene cada uno de los interesados sobre el proyecto.

Es clave considerar a los interesados con mayor influencia para que sus intereses sean considerados durante el desarrollo del proyecto y lograr así satisfacer sus expectativas para poder finalizar de manera exitosa.

Los interesados se clasifican por el grado de interés y el poder que tienen sobre el proyecto.

Interesado	Grado de interés	Poder sobre el proyecto	Comentarios
Equipo de trabajo	Alto	Alto	El equipo de trabajo es el principal interesado en conseguir el éxito del proyecto ya que el objetivo académico es el principal a cumplir y estará determinado por la aprobación.
Tutora	Alto	Alto	La tutora del proyecto tiene un interés positivo sobre el mismo ya que el resultado académico estará fuertemente relacionado a su desempeño en la tutoría del equipo.
Universidad ORT	Alto	Alto	La Universidad es uno de los principales interesados en el éxito del proyecto ya que provee el marco para el desarrollo del mismo.
Usuarios	Medio	Medio	Los potenciales usuarios del sistema, tanto consumidores como comercios se verán beneficiados por el uso de la aplicación pero al mismo tiempo el éxito/fracaso proyecto no significa ningún riesgo para ellos.
Socio de marketing	Medio	Medio	La empresa de marketing que se utiliza durante el proyecto en las etapas de requerimientos y pruebas de aceptación tiene interés en el éxito del proyecto ya que una vez finalizado el mismo se hará una prueba piloto con un cliente proporcionado por esta empresa y eso puede resultar beneficioso desde un punto de vista comercial.

Tabla 7.1 – Interesados

2.3. Descripción del problema

El análisis que se realizó de la situación actual nos llevó a entender que el problema que estamos intentando resolver tiene tres perspectivas que deben ser tenidas en cuenta. Por un lado están los consumidores, que serán los principales usuarios de la aplicación para realizar búsquedas de productos, por otro lado los comercios que aparecerán en los resultados de las búsquedas de los consumidores y por último el negocio en sí mismo y el reto de generar un emprendimiento monetizable que resuelva los problemas y/o explote las oportunidades encontradas en las perspectivas antes mencionadas.

2.3.1. Desde la perspectiva de los consumidores

Los consumidores deben invertir tiempo buscando activamente los productos que desean. Sin embargo, la vorágine del día a día hace que las personas tengan poco tiempo y eviten dedicárselo a una compra, salvo que el producto sea una necesidad básica. La alternativa de realizar la compra online de un producto muchas veces no es adecuada porque dependiendo del producto, puede ser necesario verlo personalmente o probarse un talle en particular como es el caso de la ropa, para determinar si cumple con lo que busca el consumidor o no.

Otro problema, es la falta de conocimiento de puntos de venta que serían mucho más convenientes para su ubicación, evitando traslados que muchas veces hace que se descarte una compra.

Los consumidores en su mayoría terminan no comprando el producto que están buscando en las plataformas de *e-commerce* (por ejemplo Mercadolibre). Esto en general se debe a que el punto de retiro del mismo queda lejos de su ubicación o no le es cómodo ir a retirarlo. También los usuarios manifiestan haber estado en la situación de enterarse demasiado tarde que pasaron por un punto de venta de un producto que estaban buscando y no supieron de que el mismo se encontraba allí y valoran en segundo lugar al momento de decidirse por un producto la facilidad para hacerse con él, es decir, la comodidad para retirarlo o recibirlo. Esta investigación se puede ver en el [Anexo 6 - Investigación de mercado](#).

2.3.2. Desde la perspectiva de los comerciantes

Los comercios menos conocidos, que generalmente son PYMES, usan diferentes alternativas de marketing para lograr mayor visibilidad de su negocio y llegar así a un mayor número de potenciales clientes.

Las alternativas actuales de marketing más utilizadas son:

- 1- Hacer campaña de marketing con alguna empresa especializada.
- 2- Publicitar en redes sociales.
- 3- Vender en plataformas de *e-commerce* (como por ejemplo Mercadolibre).

Las anteriores alternativas cuentan con algunos problemas que, a nuestro entender, la solución que proponemos resuelve.

Los problemas identificados para estas tres alternativas son los siguientes:

- Los costos de hacer una campaña de marketing para un pequeño o mediano comercio suelen resultar elevados para sus posibilidades, tal cual se refleja la encuesta de MIEM donde se muestran datos acerca del acceso al crédito para el año 2017 y las actividades en las que se invierte dicho crédito (ver [Anexo 2 - Acceso a créditos por parte de las PYMES en Uruguay](#) y [Anexo 3 - Uso potencial del crédito de las PYMES en marketing/publicidad](#)).
- Las redes sociales más populares como Facebook e Instagram, ofrecen la opción de realizar publicaciones pagas o gratuitas, siendo muy utilizadas por las pequeñas y medianas empresas en todo el mundo. Esto se debe a la baja inversión que suponen y el alcance ilimitado que podría tener en todo el mundo. Tener éxito en este tipo de medios no es fortuito y para nada sencillo, se necesita creatividad y personas capacitadas en esta área y recursos para lograr un perfil exitoso en las redes. Para más información consultar [Anexo 6 – Investigación de Mercado](#).
- Como tercera y no menos frecuentada alternativa se encuentran las plataformas de *e-commerce*, fundamentalmente MercadoLibre en nuestro país, utilizado como mecanismo de venta e interacción con clientes. Aquí es difícil lograr diferenciación ya que las empresas compiten con otras más grandes que tienen una mayor cantidad de ventas realizadas para los mismos artículos, teniendo ubicaciones preferenciales en las búsquedas. Es en estas modalidades donde radica otro de los grandes problemas que detectamos y que buscamos disminuir, como lo es la alta brecha entre intención de compra y compra concretada de productos en plataformas de *e-commerce*. Esta brecha se incrementa aún más cuando los productos no son un “*commodity*”. Está comprobado que cuanto más se incrementa la diferenciación o especialización entre los productos similares, la brecha también experimenta dicho aumento. Consultar más información y referencias en [Anexo 6 – Investigación de Mercado](#).

2.4. Descripción de la solución

Se pretende desarrollar una aplicación web que permita buscar un producto por su nombre exacto o una categoría a la que pertenece, desplegando los resultados de comercios que ofrecen dicho producto en un mapa priorizados por cercanía a la ubicación del usuario. Se buscará disparar el impulso de compra [13] de los clientes por encontrarse cerca del comercio que ofrece el producto que está buscando.

La solución apunta a cubrir las necesidades de tres grupos de usuarios:

- Compradores: Son los clientes que utilizan la aplicación para buscar los productos.
- Vendedores: Son los comerciantes cuyos comercios están registrados en el sistema y aparecen (eventualmente) en los resultados de las búsquedas de los compradores.

- Administradores: Los usuarios administradores del sistema, en un principio serán exclusivamente los integrantes del emprendimiento.

2.4.1. Funcionalidades de la solución propuesta

Perspectiva comprador

Descripción de las principales funcionalidades:

- El comprador deberá registrarse y autenticarse para poder acceder al sistema.
- Dentro del sistema podrá realizar búsquedas de un producto por nombre exacto o por la categoría a la que pertenece.
- Los comercios que se muestran como resultados de las búsquedas se desplegarán en el mapa, siempre y cuando caigan dentro del radio de distancia de búsqueda configurable por parte del usuario.
- Para cada resultado el comprador podrá consultar los datos que el comercio expone y hasta ver sus ofertas si el comercio ha publicado alguna.
- Cada búsqueda podrá ser guardada por el usuario, pasando a estar disponible dentro de su historial de búsquedas.
- El usuario podrá retomar sus búsquedas cuando lo necesite.
- En caso de querer seguir buscando un producto, podrá marcar la búsqueda como activa en segundo plano. De esta manera, ya no necesitará tener abierta la aplicación y en caso de estar cerca de un comercio que tiene el producto buscado recibirá una notificación informando de la situación.

Perspectiva comercio

Descripción de las principales funcionalidades:

- Los comercios deberán registrarse y autenticarse para operar en el sistema. Deberán proveer datos como nombre, teléfono, dirección, categoría y sub-categorías de los productos que vende.
- Podrán ingresar ofertas completando un formulario de campos obligatorios.
- Tendrán un panel donde verán estadísticas de su comercio, cantidad de estrellas por publicación de ofertas y sus productos más buscados.

Perspectiva administrador CompraCerca

Descripción de las principales funcionalidades:

- El administrador deberá autenticarse para acceder al módulo de administración.
- La página principal mostrará un panel de control del sistema donde se podrá visualizar información de la aplicación en tiempo real. Se desplegará el número de compradores, el número de comercios y la cantidad de búsquedas en paneles para

una rápida visualización. Se mostrará una tabla de acciones diarias sobre el sistema, identificando el usuario, fecha y tipo de acción. Se permitirá filtrar por estos campos.

- Podrá acceder a una lista de compradores y comercios del sistema, pudiendo realizar acciones sobre dicha lista como dejar inactivo sus cuentas en caso de ser pertinente.

3. PROCESO Y CICLO DE VIDA

El siguiente capítulo tiene como objetivo describir la metodología de trabajo aplicada, el ciclo de vida de desarrollo empleado y la división de tareas por roles.

3.1. Ciclo de vida

Consideramos que el avance del proyecto tenía que ser iterativo con el objetivo de identificar fallas y oportunidades de mejoras en las etapas más tempranas del proyecto. Consideramos un enfoque evolutivo con el objetivo de construir/extender el sistema sobre lo ya desarrollado.

3.1.1. Ciclo de vida prototipado evolutivo

En este ciclo de vida se agrupan las tareas en pequeñas etapas repetitivas. Consta de diversas etapas de desarrollo, cada incremento comienza con el análisis de los requerimientos y finaliza con el desarrollo de dichos requerimientos obteniendo como resultado un prototipo del sistema que agrega funcionalidad al final de cada evolución.

En la siguiente imagen se muestran las tareas que se realizaron en las evoluciones del prototipado. Se pueden realizar tantas evoluciones como se entienda necesario.

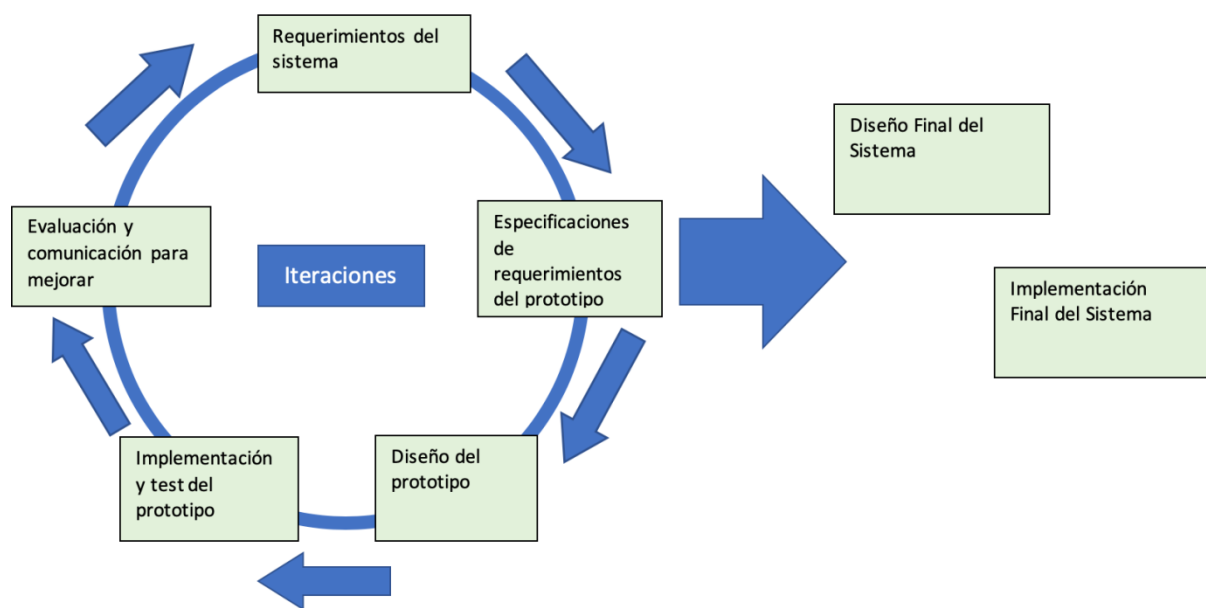


Imagen 8.1 – Ciclo de Vida

Nuestro proyecto constó de cuatro evoluciones donde se realizaron tareas de relevamiento y especificación de requerimientos, diseño de las pantallas e implementación de las funcionalidades.

- Iteración 1: Prueba de concepto para definir las tecnologías a utilizar de manera que se puedan cubrir los requerimientos relevados.

- Iteración 2: Desarrollo de los requerimientos especificados para la evolución.
- Iteración 3: Continuación del desarrollo de los requerimientos especificados para la evolución, partiendo del prototipo de la evolución anterior.
- Iteración 4: Continuación del desarrollo de los requerimientos especificados para la evolución, partiendo del prototipo de la evolución anterior.

Cada evolución coincidió con un *sprint* de trabajo y una vez finalizadas estas iteraciones se continuó trabajando en tres iteraciones posteriores para el desarrollo de la documentación del proyecto.

3.2. Metodología de gestión de proyectos

Debido a la naturaleza del proyecto que fue un emprendimiento, no contábamos con requerimientos bien definidos de un cliente concreto sino con un conocimiento global de los objetivos que fueron surgiendo y definiéndose conforme se avanzó con el proyecto.

Otro aspecto que estuvo presente fueron los cortos plazos de desarrollo (Ver [Planificación](#)) en relación a otros proyectos de mayor duración que pueden llegar a un año o más. Por lo tanto, se hizo imprescindible contar con funcionalidad implementada lo más tempranamente posible, logrando así unidades funcionales que nos pudiesen servir para aprender y poder definir nuevos requisitos y evitar así grandes modificaciones en etapas más tardías del proyecto. Por los motivos antes mencionados consideramos que una metodología ágil es el enfoque apropiado, dentro de las posibles metodologías ágiles la metodología *SCRUM*.

Los beneficios de implementar una metodología ágil son que las entregas son en forma temprana y predecible, ya que los entregables se hacen de manera rápida y frecuente. Otro motivo por el cual optamos por la metodología ágil es que decidimos trabajar con el ciclo de vida llamado prototipación evolutiva el cual se puede alinear con los *sprints* de *SCRUM*. Los procesos ágiles tienen como ventaja que se permite cambiar prioridades y refinar el *backlog* del producto. Estos cambios se pueden incorporar a la siguiente iteración, de manera que puedan ser introducidos en pocas semanas. Otra característica importante es que permite identificar qué es lo más importante para el negocio y entregar dichas características que brindan mayor valor al negocio, en nuestra situación aquellas funcionalidades que aportan mayor valor a nuestra aplicación.

Los siguientes puntos fueron de especial importancia al momento de seleccionar la metodología de trabajo:

- La corta duración del proyecto sumado a la cantidad reducida de integrantes a cargo del proyecto (tres personas), la capacidad de administrar cambios y entregar funcionalidad fue limitada, sin mencionar que tuvimos que madurar y definir los requerimientos conforme avanzó el proyecto.
- Definimos dos MVP, siendo el primero aquel que consideramos que entregaba el mayor valor de nuestra propuesta.

- Fue necesario dividir tareas y seguir una clara metodología de trabajo, asignando las responsabilidades a los integrantes con mejores aptitudes para lograr los objetivos de trabajo.

A continuación se mencionan aquellas prácticas que consideramos fundamentales para el aseguramiento del cumplimiento de los objetivos de trabajo y que transcurrieron durante el proyecto:

- Priorización y evaluación de objetivos periódica, en cada iteración se priorizaron las funcionalidades evaluando aquellas que entregan el mayor valor al sistema, la complejidad de su desarrollo y sus dependencias con otras funcionalidades.
- Reuniones de equipo, se coordinaron reuniones entre los miembros del equipo presenciales tres veces por semana, la duración de cada reunión fue en promedio de unas tres horas aproximadamente para coordinar avances, revisar el trabajo pendiente, estimar el esfuerzo requerido para completarlo, evaluar y medir riesgos presentados y repartición de tareas entre otros temas.

3.3. Roles

Durante el transcurso del proyecto los integrantes del equipo cumplieron diferentes roles de acuerdo a la etapa y las necesidades del momento.

Los roles se asignaron de acuerdo a la experiencia y preferencia de cada uno, para lograr en la medida de lo posible que todos trabajen cómodos y se desempeñen de la mejor manera.

Rol	Responsabilidades del rol
Gerente de proyecto	Planificación general. Elaboración y actualización del plan de proyecto. Seguimiento de tareas. Obtención y análisis de métricas del proyecto.
Responsable de Arquitectura	Elaboración de la arquitectura. Seguimiento respecto a que se respete la arquitectura en el desarrollo. Soporte al equipo de desarrollo.
Responsable de Requerimientos	Relevamiento de requerimientos. Validación de los requerimientos. Especificación de requerimiento. Mantener los requerimientos actualizados.
Responsable de SQA	Elaborar el plan de SQA. Definición del plan de pruebas. Obtención y análisis de métricas de calidad. Elaborar <i>template</i> para casos de prueba. Elaborar <i>template</i> para reporte de defectos.
Responsable de SCM	Definir repositorio de documentos. Definir repositorio de código. Mantener los repositorios. Controles sobre los repositorios.
Desarrollador	Diseño de la solución. Implementación. Revisiones de código a otros desarrolladores.
Tester	Elaborar casos de prueba. Ejecutar casos de prueba. Reporte de defecto. Seguimiento de defectos.
Scrum Master	Responsable de asegurar el cumplimiento de la metodología de trabajo. Mantenimiento del tablero con las tareas. Realizar los cierres de sprint y sacar la información de interés.

Tabla 8.1 - Roles

4. INGENIERÍA DE REQUERIMIENTOS

Los requerimientos fueron definidos por la *IEEE* como la condición o capacidad que debe satisfacer o poseer un sistema o un componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.

A continuación, se detallarán las etapas del proceso de ingeniería de requerimientos por las cuales se transitó para obtener dichos requerimientos iniciales, tanto funcionales como no funcionales.

4.1. Proceso de análisis de requerimientos

El proceso de análisis de requerimientos consistió en cuatro etapas principales que se detallan a continuación.

4.1.1. Relevamiento del problema

Se utilizaron diferentes técnicas:

Tormenta de ideas

La primera técnica utilizada fue la de realizar una tormenta de ideas en conjunto todos los integrantes del proyecto. En esta instancia se buscó identificar problemas y mejoras que nuestro emprendimiento podría aportar sobre el mercado actual, utilizando la experiencia personal de cada uno y evaluando las alternativas disponibles del mercado.

Las preguntas más enigmáticas que surgieron fueron: ¿Qué comercios estarían interesados en trabajar con nosotros?, ¿Qué le ofreceríamos a dichos comercios?, ¿Los problemas que podemos pensar que tienen, ellos también los perciben?, ¿Vendrían a buscarnos a nosotros o nosotros deberíamos ir hacia ellos?, ¿Solución como complemento a alternativa del mercado o seríamos un emprendimiento independiente?.

Análisis de mercado

Con las preguntas planteadas en la tormenta de ideas se pasó a realizar un análisis del mercado y de los productos similares. El análisis se realizó desde dos puntos de vista: del comprador y de los comercios.

Para cada uno se realizó un listado de los atributos claves para cumplir su rol en el proceso de compra-venta y cómo poder influenciar positivamente en ellos. Para los compradores surgió la variable del tiempo de búsqueda y la proximidad de los negocios. En el caso de los comercios sobresalieron factores como el costo y la visibilidad en internet.

De la investigación realizada no se encontraron productos similares, sino que se encontraron coincidencia en algunas funcionalidades proporcionadas por dos referentes del mercado como son Google Maps y Mercadolibre.

Para más información de la investigación de mercado realizada consultar [Anexo 6 – Investigación de Mercado](#).

4.1.2. Análisis

Toda la información obtenida del relevamiento se unificó y procesó para obtener una descripción del sistema y sus prestaciones.

Una de las decisiones a la que se llegó fue que nuestra aplicación sería un híbrido entre Google y Mercado Libre, utilizando la potencia de Google Maps y sus mapas pero con el enfoque en la búsqueda de productos y servicios similar a como trabaja Mercado Libre y su estructura de categorías.

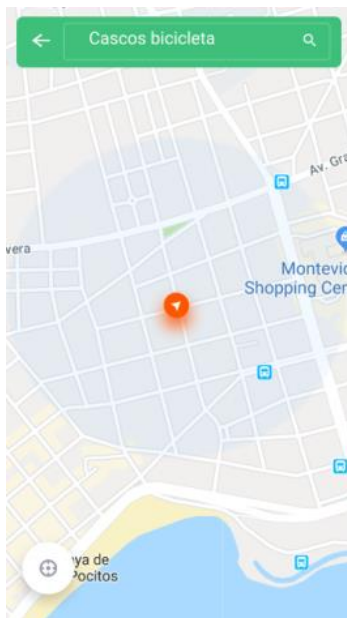
4.1.3. Especificación

Culminado el análisis se obtuvo una definición a alto nivel de los requerimientos del sistema. En esta etapa de especificación se procedió a escribir formalmente los requerimientos utilizando *User Stories* y *TDD* definiendo escenarios para la especificación de cada requerimiento. El documento con la especificación de los requerimientos detallado se encuentra en [Anexo 9 – Especificación de Requerimientos](#) y en la sección 8.3 donde se listan los requerimientos.

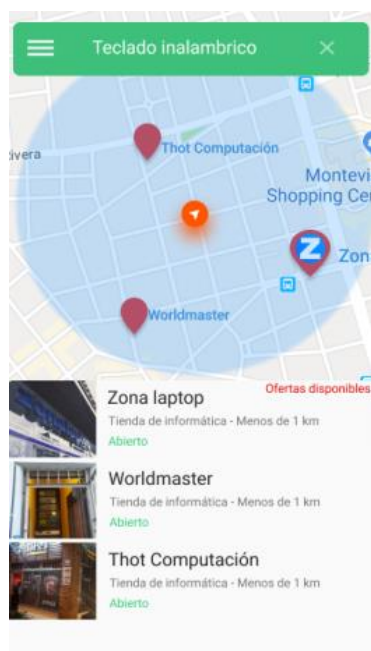
Durante la especificación de requerimientos se construyeron las posibles pantallas utilizando la herramienta Figma para la creación de *mocks*.

Se adjuntan a continuación algunas pantallas a modo de ejemplo, la totalidad de las pantallas creadas se encuentran en [Anexo 13 - Mocks creados en Figma](#).

Pantalla principal de la aplicación



Pantalla con búsqueda activa



Pantalla perfil comercio



4.1.4. Validación

Para la validación de los requerimientos se realizaron encuestas sobre los compradores y comercios. Estas encuestas permitieron validar que los requerimientos propuestos se ajustaban a las necesidades de los compradores y comercios.

El detalle completo de los resultados de la encuesta se encuentran en el [Anexo 11 – Resultados](#) de encuestas a compradores y [Anexo 12 – Resultados de encuestas a vendedores](#).

A grandes rasgos los resultados para las encuestas de los compradores en su mayoría entre 20-30 años fueron que:

- No suelen descargar aplicaciones.
- Solo un 7% no ha encontrado comercios en Google Maps frecuentemente.
- Suelen comprar en el shopping, como segunda opción online y por último en comercio cercano.
- Han cambiado su celular hace menos de un año.

En la encuesta de comercios en su mayoría pequeños o medianos los resultados arrojaron que:

- Buscan gastar lo mínimo en publicidad.
- Utilizan redes sociales y publicaciones gratuitas.
- La mayoría manifiesta la necesidad de atraer más clientes.
- La mayoría trabaja con ofertas.

Por otra parte, se realizó una entrevista con un experto de marketing, dueño de una empresa de marketing al cual se le mostraron las funcionalidades del sistema y cómo podrían mejorar las ventas de los comercios. El experto hizo una valoración muy positiva, se mostró muy interesado en el proyecto, asegurando de que si el sistema lograra los requerimientos propuestos podía ser de gran ayuda para su empresa.

Ver la carta de experto completa en [Anexo 8 – Carta de Experto](#).

Además de lo mencionado anteriormente, durante las tutorías se realizó una validación con la tutora como potencial usuaria de la aplicación que nos aportó otro punto de vista crítico a los requerimientos del sistema.

Se realizaron además, y como complemento, revisiones cruzadas de los requerimientos entre los integrantes del equipo como otra herramienta de validación.

4.2. Selección de funcionalidades para MVP

De todos los requerimientos identificados en el proceso de análisis de requerimientos, para el momento de la entrega del proyecto en la universidad, se propuso acotar el desarrollo a un conjunto de funcionalidades indispensables para que el sistema cumpla con su razón de ser. A dicho conjunto de funcionalidades le llamamos *MVP*. El *MVP* contendrá funcionalidades para la perspectiva de los consumidores y de los administradores de CompraCerca.

Funcionalidades del MVP

Perspectiva de Comprador

- Registro de usuario.
- *Login* de usuario.
- Búsqueda por nombre exacto.
- Búsqueda por categoría.
- Agrupación de marcadores cercanos.
- Consultar información de un comercio.

Perspectiva de Administrador CompraCerca

- *Dashboard* de control.
- Administrar compradores del sistema.
- Administrar comercios del sistema.

El orden de implementación y el alcance se decidieron en conjunto entre el equipo de proyecto y la tutora.

Para tomar esta decisión se consideraron los siguientes criterios:

- Necesidad de mostrar un producto en su primera versión que tenga las funcionalidades básicas para que el sistema tenga valor para los usuarios finales. Consiguiendo al momento de la entrega del proyecto un sistema con bases sólidas para continuar hacia la implementación de una segunda versión.
- El tiempo disponible de desarrollo llevó a optar por aquellas funcionalidades con baja curva de aprendizaje para su desarrollo, con conocimiento inicial sobre las tecnologías a aplicar y dónde el tiempo de implementación estimado no exceda tres *sprints*.
- La falta de comercios adheridos al proyecto llevo a escoger funcionalidades donde no sea imprescindible para su funcionamiento una gran masa crítica.
- La financiación nula del proyecto llevó a reducir los costos monetarios de la implementación y evitar desarrollos donde sea necesario compra de tecnologías, servicios, etc.

4.3. Requerimientos Funcionales

A continuación se listan los requerimientos funcionales del sistema. Se agruparon de acuerdo a la perspectiva del usuario (comprador, vendedor y administradores).

Perspectiva de Comprador

RF1 - Login de usuario

Descripción: Comprador deberá ingresar credenciales para poder ingresar al sistema. Por lo tanto, deberá estar previamente registrado.

RF2 - Registro de usuario

Descripción: Comprador se registra en el sistema. Se solicitan todos los datos necesarios para utilizar el sistema.

RF3 – Buscar producto por categoría

Descripción: Comprador podrá realizar una búsqueda por una categoría de productos en particular. Estas categorías se desplegarán como un árbol de dos niveles. Los resultados de la búsqueda se desplegarán en el mapa.

RF4 – Buscar producto por coincidencia exacta

Descripción: Comprador podrá realizar una búsqueda de un producto desde la lupa de búsqueda. Los resultados de la búsqueda se desplegarán en el mapa. Se priorizarán resultados de comercios que trabajen con Compra Cerca.

RF5 – Agrupación de marcadores cercanos

Descripción: Para una correcta visualización de los resultados en el mapa los mismos deberán agruparse de acuerdo al nivel de zoom en el mapa, de manera que no se solapen nunca.

RF6 - Consultar información del comercio

Descripción: Comprador podrá seleccionar uno de los resultados en el mapa y consultar toda la información disponible del comercio.

RF7 - Visualizar ofertas de comercio

Descripción: Comprador tendrá la posibilidad de ver ofertas que el comercio publique. Dichas ofertas se señalaran en el mapa junto con el marcador correspondiente al comercio.

RF8 – Administración de la cuenta

Descripción: Comprador podrá configurar la información de su perfil, así como configurar su rango de búsqueda actual y la información que se permite mostrar a terceros.

RF9 - Recordar búsqueda

Descripción: Comprador podrá guardar sus búsquedas para poder acceder a sus resultados en otro momento.

RF10 - Histórico de búsquedas

Descripción: Comprador podrá acceder a una lista con todas sus búsquedas guardadas. Al seleccionar una de ellas se desplegará dicha búsqueda en el mapa.

RF11 - Mostrar ruta a comercio

Descripción: Comprador al seleccionar un comercio en particular y ver el perfil del mismo tendrá la posibilidad de buscar la ruta más cercana para llegar al comercio.

RF12 - Contactar comercio

Descripción: Comprador al seleccionar un comercio en particular y ver el perfil del mismo tendrá la posibilidad de contactarse telefónicamente con el comercio.

RF13 - Búsqueda activa

Descripción: Comprador tendrá la posibilidad de dejar una búsqueda activa en segundo plano. En caso de pasar a una distancia configurable de un comercio que ofrezca el producto que está buscando, se le enviará una notificación al comprador y un aviso también al comercio.

Perspectiva de Comercio**RF14 - *Dashboard* de comercio**

Descripción: Comercios tendrán la posibilidad de ver información acerca del sistema que sea de su interés. Los productos más buscados en el mercado y aquellas búsquedas que generaron coincidencia con el comercio.

RF15 - Ingreso de oferta

Descripción: Comercios tendrán la posibilidad de subir sus ofertas al sistema. Dichas ofertas deberán completarse en un formulario donde se solicitarán datos obligatorios.

Cada comercio tendrá una cantidad específica de ofertas que podrá subir de acuerdo a su plan de contrato.

RF16 - Auditar oferta

Descripción: Comercios tendrán la posibilidad de auditar la cantidad de visualizaciones que tuvo su oferta en el sistema y cuantos compradores la marcaron de su interés. A su vez, podrá modificar la información de la oferta.

RF17 - Notificar a comercio de un usuario potencial cerca

Descripción: Comercios recibirán un mensaje en su bandeja de mensajes y una alerta por cada usuario que cruce la frontera establecida para la tienda.

Perspectiva de Compra Cerca (Administración)

RF18 - *Dashboard* de control

Descripción: *Dashboard* en la pantalla principal de los administradores de CompraCerca donde se mostrará información sobre el uso de la aplicación y un log de acciones realizadas en el sistema. Deberá generarse un archivo plano por día con las acciones realizadas sobre el sistema.

RF19 - Administrar compradores del sistema

Descripción: Lista de usuarios del sistema donde se podrá ver la información de cada usuario y tomar acciones como bloquear un comprador en particular.

RF20 - Administrar comercios del sistema

Descripción: Lista de comercios del sistema donde se podrá ver la información de cada uno y tomar acciones como bloquear un comercio en particular.

4.4. Requerimientos no funcionales

Los requerimientos no funcionales se garantizarán con las decisiones de arquitectura que se explicarán en detalle en el capítulo de [Arquitectura](#).

4.4.1. Seguridad

RNF1 - Identificación y autenticación de usuarios

Descripción: Los usuarios deberán autenticarse en el sistema para poder acceder a las funcionalidades de la aplicación. Este requerimiento deberá cumplirse para los distintos roles de la aplicación.

RNF2 - Utilización de *token* para llamadas http

Descripción: Todas las llamadas a la *API* del *backend* deberán contener en el encabezado de la llamada, un *token* de identificación, de lo contrario se rechazará la solicitud.

RNF3 - Protección de URLs

Descripción: Todas las páginas de la aplicación y sus respectivas *URLs* se deberán proteger para evitar accesos externos.

RNF4 - Protección de información sensible

Descripción: La información de los usuarios no estará visible por defecto. Será decisión del usuario que información compartir públicamente.

4.4.2. Disponibilidad

RNF5 - Disponibilidad del sistema

Descripción: La solución debe ofrecer niveles de servicio del 97% donde la disponibilidad y recuperación de fallos sea garantizada e inmediata.

4.4.3. Usabilidad

RNF6 - Utilización de estándares de diseño

Descripción: Deberá hacerse hincapié en respetar los estándares de diseño para móviles al momento de realizar la interfaz de usuario.

RNF7 - *Responsive*

Descripción: Interfaz deberá visualizarse correctamente en distintos tipos de dispositivos en que se ejecute la aplicación.

RNF8 - Facilidad de acceso

Descripción: La mayoría de las funcionalidades podrán ser accedidas desde el menú principal.

RNF9 - Estética adaptativa

Descripción: La interfaz de usuario deberá adaptar los estilos de los elementos que la componen según el tipo de dispositivo donde se ejecuta la aplicación.

4.4.4. Performance

RNF10 - Tiempo de respuesta

Descripción: El sistema deberá responder a todos los pedidos del usuario en un tiempo menor a 3 segundos promedio.

4.4.5. Eficiencia

RNF11 - Optimización en el uso de recursos

Descripción: El sistema deberá cumplir los tiempos máximos de respuesta en dispositivos móviles de gama media y baja. Se considerarán a estos sistemas como las principales plataformas en las que correrá la aplicación.

4.4.6. Modificabilidad

RNF12 - Implementación a futuro de aplicación nativa

Descripción: La implementación deberá contemplar el pasaje a corto plazo a una aplicación nativa con el menor esfuerzo posible.

5. ARQUITECTURA

El objetivo del siguiente capítulo es el de especificar la arquitectura del sistema. Definimos un marco, una estructura para que por medio de un conjunto de patrones y abstracciones podamos interactuar con el código fuente del sistema.

Al momento de diseñar la arquitectura del sistema, para lograr que la misma esté alineada con nuestros objetivos (requerimientos) y las restricciones dadas por el sistema que pretendemos implementar, consideramos relevantes los siguientes puntos:

- **Mantenibilidad:** Dado la naturaleza de nuestro proyecto que es un emprendimiento, no tenemos una especificación definida de lo que debe hacer el sistema sino que fuimos especificando los requerimientos del sistema en forma iterativa y en base a los formularios de encuesta. Esto implica que nuestro sistema está en constante evolución y cambio en comparación a un sistema basado en metodologías tradicionales, por lo que el sistema debe tener una arquitectura pensada en la mantenibilidad y en la flexibilidad de poder realizar cambios.
- **Tecnología:** Nos basamos en tecnologías en las cuales el equipo contaba con relativa experiencia y/o familiaridad, con el objetivo de mitigar riesgos por una curva de aprendizaje muy elevada, que pudiese causar que los objetivos queden fuera de alcance, o no lograr implementar con éxito las tecnologías elegidas ya sea por falta de experiencia o por no haber seleccionado las adecuadas al problema. Este punto también refiere a las tecnologías necesarias para correr el sistema.
- **Calidad:** Diseñar una arquitectura de sistema que logre cubrir de forma eficiente y eficaz todos los requerimientos funcionales y no funcionales del sistema. Para ello se aplicaron patrones, y decisiones de diseño que contribuyan a la calidad del diseño.

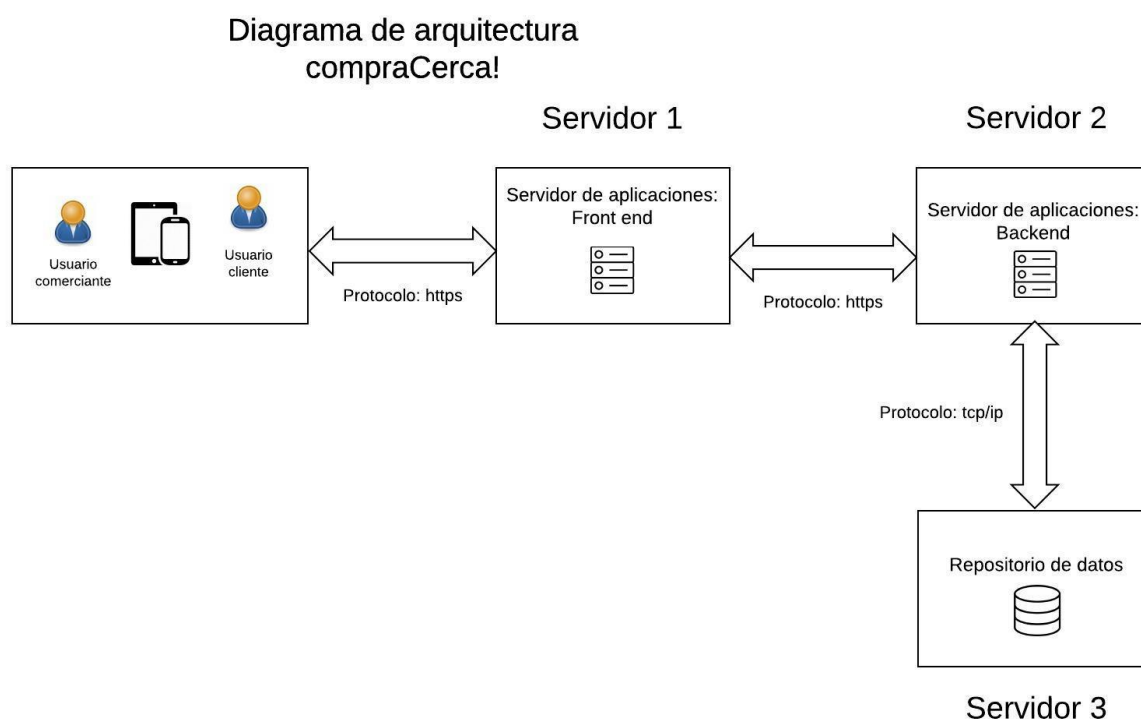
5.1. Introducción

Para poder comunicar adecuadamente el diseño y especificación del sistema como una vista global de alto nivel, se presentará la arquitectura del sistema en dos grandes áreas. Por un lado el *backend* que fue desarrollado en base a una arquitectura *web API* en .NET Core, en conjunto con determinados patrones de diseño como *MVC*, *Fachada*, y *Composite*, y por otro el *frontend* basado en Angular. Finalmente se detallarán los objetivos de calidad y cómo la arquitectura lo soporta para luego presentar los diagramas de componentes, de paquetes, de clase, de comunicación, y de despliegue.

5.2. Vista global del sistema

Tal como se detalla en [Anexo 14 – Planilla de selección de tecnologías](#), optamos por tener nuestro sitio web dentro del servidor y la infraestructura de la empresa, pues decidimos tener una mayor personalización en nuestro ambiente de instalación. Si bien actualmente no se almacenan datos sensibles de los clientes, creemos que en el futuro si habrá, por lo que este tipo de instalación nos dará mayor seguridad sobre los datos al estar en la organización.

De todas maneras, es una característica deseable a futuro poner nuestro sistema en la nube pues esto permitiría que sea escalable logrando así mejorar la disponibilidad del servicio y poder atender una mayor carga sobre el servidor.



En la figura anterior se muestran como se comunican los diferentes componentes. El usuario desde un *browser* realiza la petición *http* del sitio web a el servidor de aplicaciones, este a su vez se comunica por protocolo *http* al *backend* del sistema que se encuentra en otro servidor de aplicaciones, el *backend* luego se comunica con el repositorio de datos por medio del protocolo MySQL.

El servicio de *hosting* se compone de:

- **Servidor de aplicaciones:** En el cual se encuentra el *frontend* y *backend* en el cual se encuentra toda la lógica de la aplicación.
- **Repositorio de datos:** Servidor en cual se encuentra los datos usados por la aplicación.

5.3. API Rest

Decidimos que toda la lógica para que el sistema funcione sea implementada por medio de una arquitectura *API Rest*. A continuación se enumeran las ventajas de utilizarla y cómo esas ventajas favorecen a la arquitectura del sistema:

- Separación cliente/servidor ya que al ser sistemas independientes que se comunican por medio de un protocolo de intercambio, tenemos la ventaja de trabajar en forma independiente el *frontend* (cliente) del *backend* (servidor). Ya que para el cliente es indistinto en que tecnología se encuentra desarrollado y a su vez para el servidor también es indiferente que va a hacer el cliente con los datos que le proporcionó. Mientras no se rompa el contrato en cómo se comunican, el *frontend* y el *backend* se puede evolucionar y refactorizar el sistema sin causar impactos en un sistema o el otro. Una *REST API* nos permite tener tantos *frontends* como necesitemos pudiendo trabajar sobre el mismo *backend*. Esta flexibilidad nos permite tener mayor libertad de elegir la tecnología de desarrollo.
- Se obtiene mejor performance que en el desarrollo de un sitio tradicional, pues cuando se hace una solicitud por lo general la respuesta del servidor es la información de la página como el *html*, *css* y *javascript*. En cambio en una *REST API* la respuesta es siempre información en formato plano que requieren tiempos de transferencia menores, luego es responsabilidad del cliente en mostrar los datos.
- Requiere menos recursos del servidor, no mantiene estado, no requiere memoria por lo que se pueden atender más peticiones. Al no tener que escribir la página, tiene menos procesamiento que realizar.

5.4. Patrón MVC

Decidimos implementar este patrón ya que favorece la mantenibilidad del sistema al separar en capas el modelo de datos de la presentación al usuario. Este modelo tiene tres capas diferenciadas. El modelo, la vista y el controlador. El módulo del modelo hace referencia a toda la lógica de nuestro negocio, el módulo de la vista es el que tiene la responsabilidad de mostrar los datos pero este no aplica pues fue desarrollado como otro proyecto, y por último el controlador que tiene la responsabilidad de comunicar el *backend* con el *frontend*.

Las ventajas que identificamos por implementar este patrón son:

- Fácil comunicación entre módulos, que al ser un estándar es simple de utilizar y entender facilitando el desarrollo de la aplicación.
- Modularización en el desarrollo, la programación se divide en módulos los cuales tienen responsabilidades bien definidas. Esto mantiene una alta cohesión en el módulo.
- Los módulos favorecen el bajo acoplamiento lo cual facilita la mantenibilidad y la capacidad de poder extender el sistema.
- Se favorece la usabilidad, al haber una clara separación de la vista de lo que es el modelo, pues es posible modificar la vista para facilitar la interpretación de los datos al usuario sin necesidad de que afecte a la lógica del sistema.

5.5. Patrones Composite y Fachada

En el desarrollo del sistema nos enfrentamos a problemas que por sus características buscamos resolverlos a través de patrones de desarrollo. Elegimos el patrón que consideramos más adecuado a nuestras necesidades buscando que facilite posibles futuros cambios, minimizar el riesgo por un mal diseño y poder comunicar adecuadamente nuestra implementación del sistema.

5.5.1. Composite

Dentro de los requerimientos funcionales tuvimos que definir un árbol de categorías para que el usuario pueda navegar a través de categorías de productos o servicios que busca. Identificamos que el patrón Composite se ajusta mejor a resolver esta necesidad, ya que permite construir objetos complejos a partir de otros más simples y que son similares entre sí. Esto se logra gracias a la composición recursiva y a la estructura de árbol. En el caso de las categorías, son objetos complejos que están constituidas por otras categorías. Las categorías padres e hijas son similares en cuanto a la información que almacenan, por ejemplo nombre de la categoría e identificador. Este patrón permite obtener una implementación sencilla, que por medio de poca codificación podemos tener un código más sencillo y claro.

5.5.2. Fachada

Es un patrón de diseño estructural que tiene como objetivo reducir la complejidad de un sistema al subdividirlo en sistemas, minimizando la comunicación y dependencias entre ellos. En nuestra implementación se logró al abstraer cada entidad y crear el modelo [MVC](#) anteriormente mencionado para poder agregar, editar y obtener datos de cada uno de ellos en forma independiente disminuyendo así la comunicación y dependencia entre entidades de negocio. La comunicación entre módulos se implementó por medio de interfaces lo que permitió ocultar la complejidad que posee cada módulo. Esto lo hace más independiente, portable y reusable.

5.6. Angular CLI

Decidimos usar para el *frontend* el marco de trabajo Angular, dado los cortos plazos para entregar un sistema funcional decidimos elegir una tecnología que tenga un amplio soporte de comunidad y documentado, otra ventaja es que no necesita licencia paga para usarse.

Otro motivo por el cual se optó por esta tecnología es que Angular provee un marco de trabajo en el cual nos acerca a la programación orientada a objetos, facilitando organizar el código, tener una arquitectura clara de trabajo al igual que contar con elementos adicionales que son necesarias para comunicar el *frontend* con el *backend*, realizar *testing* y gestionar dependencias entre otras cosas.

Los componentes que se pueden crear con Angular favorecen el desarrollo ágil del sitio web además de permitir la reutilización del componente en otra parte del proceso.

5.7. Objetivos de calidad

En base al análisis de requerimientos funcionales y no funcionales del sistema establecimos un conjunto de atributos de calidad que consideramos que deben estar presentes a lo largo de todo el proyecto.

A continuación se enumeran y se detalla cómo la arquitectura soporta dichos atributos:

5.7.1. Disponibilidad

Realizamos un adecuado manejo de excepciones y validaciones a nivel del *backend* y *frontend* para que el usuario tenga una experiencia ininterrumpida con el sistema. El *backend* al ser *stateless* no guarda en memoria estados, lo que nos permite mayor capacidad de procesamiento. La separación del *backend* del *frontend* y la base de datos permite mayor fiabilidad pues son sistemas que corren en ambientes separados, además nos brinda la posibilidad de replicar dichos sistemas para aumentar la disponibilidad del servicio.

5.7.2. Seguridad

Para acceder al sistema deberá *loguearse* previamente.

Para el *logueo* se le solicita al usuario su nombre de usuario y su contraseña.

Durante las tareas de relevamiento identificamos diferentes roles de usuarios (administrador, comprador y vendedor) que cuentan con diferentes accesos y permisos al sistema. Los roles vienen dados si el usuario se registra como vendedor o comprador, siendo el rol de administrador reservado para el equipo de proyecto. El rol del administrador es el único que puede realizar cambios sobre las entidades del sistema, al igual que tener acceso y control sobre todos los usuarios registrados. Para garantizar la seguridad y el acceso seguro al sistema realizamos la autenticación por medio de OAuth 2.0 que es un *framework* de autorización nativo de .Net que nos provee de toda la infraestructura necesaria para un manejo adecuado de la autenticación y autorización al sistema de CompraCerca. Es un estándar usado por la industria que cuenta con soporte de la comunidad de software libre además de estar debidamente documentado, por lo que su implementación y utilización resulta relativamente sencillo.

5.7.3. Modificabilidad

Al momento del desarrollo tomamos la decisión de programar en torno a módulos siendo cada módulo lo más atómico posible, de forma de que cada uno contenga solo lo necesario para cumplir con su cometido. Esto tiene como objetivo minimizar el impacto del cambio en una pequeña área del código.

La comunicación entre módulos la realizamos por medio de interfaces, separando el “que hace” del “cómo lo hace”, esto potencia la modificabilidad y extensibilidad del sistema a través del acoplamiento a nivel de interfaz y no de implementación.

A su vez, separamos la responsabilidad de los componentes al haber una separación del *backend* de lo que es el *frontend*, y del servidor de datos. El componente *frontend* desconoce el origen de datos pues es recibido por el controlador (*backend*). A su vez el *backend* desconoce la visualización del sistema pues esta es responsabilidad del *frontend*. El servidor de datos desconoce la manipulación de datos pues es responsabilidad del *backend*, a la vez que el *backend* desconoce el almacenamiento de datos pues accede y recibe los datos del servidor de datos. Esto favorece el bajo acoplamiento minimizando el impacto del cambio.

5.7.4. Performance

La performance viene dada por la velocidad de comunicación de los componentes y el volumen de datos que son necesarios de procesar. Obtuvimos tiempos correctos de respuesta del sitio, para ello realizamos la búsqueda de productos, categorías y servicios, y en respuesta obtuvimos en forma casi instantánea los resultados de las búsquedas.

5.7.5. Usabilidad

Hemos desarrollado el sistema buscando estar alineados a las heurísticas de Nielsen. Partimos de un diseño “estándar” por ser un *frontend* basado en las *GIS*, en el cual predomina la visualización de un mapa. Todas las acciones del sistema, se realizan en pocos pasos (no más de tres pulsaciones), menús y barras de navegación simples, con uso de iconos auto-descriptivos.

Dado los cortos plazos de desarrollo decidimos apoyarnos en el uso de la librería Bootstrap. Mediante el uso de Bootstrap para el diseño y estilos del sitio favorecemos notablemente la usabilidad para el usuario.

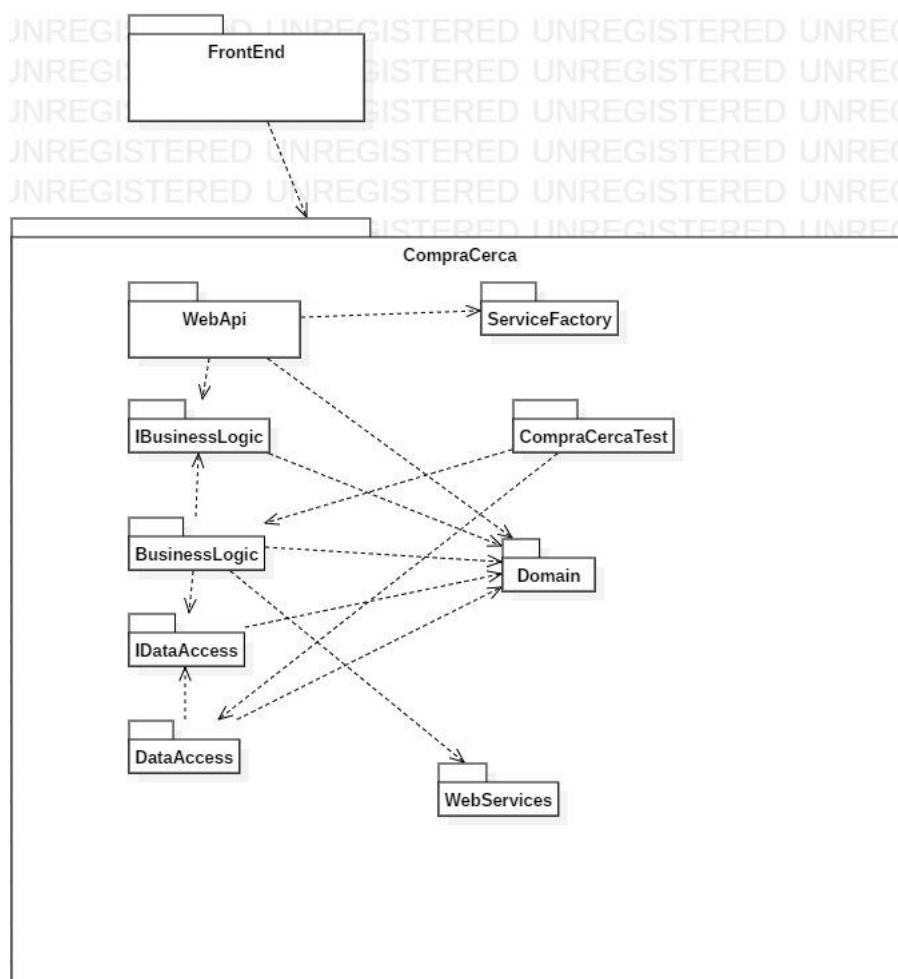
5.8. Comunicación del sistema con MercadoLibre

Uno de los requerimientos funcionales del sistema (ver requerimientos) es la búsqueda de un producto por descripción. En esta primera instancia, de los negocios sólo conocemos a qué categoría de productos o servicios trabaja, por lo que tenemos que ubicar a esa búsqueda de producto en una categoría de servicio o de producto. Para ello tomamos la decisión de comunicarnos con la *API* de MercadoLibre que es libre de licencia para que haga la predicción de catalogar el nombre de producto/servicio. Para lograr esta implementación se creó un paquete concreto que se comunica con la *API* por medio de un cliente *HTTP*, se arma la sentencia *SQL* en el cual viaja el texto del producto que el usuario busca, en respuesta se obtiene un *JSON* que es serializado a un modelo que tenemos definido en el sistema. Tomamos el nombre de categoría que arrojó la mejor coincidencia, el porcentaje de coincidencia es recibido en el objeto *JSON* que devuelve la *API* de MercadoLibre.

5.9. Diagramas del sistema

El objetivo del siguiente apartado es el de exponer en mayor detalle cómo se diseñó nuestra solución por medio de la presentación de diagramas.

5.9.1. Diagrama de paquetes, vista de alto nivel



Catálogo de componentes del sistema

Elemento	Responsabilidad
Frontend	Capa de presentación, en donde el usuario interactúa con el sistema. Para poder llevar a cabo esta implementación se utiliza la tecnología Angular Cli.
CompraCerca	Este módulo contiene toda la lógica del <i>backend</i> de la solución.
Web API	Este paquete está conformado por todas las clases necesarias para desarrollar el patrón MVC. En este módulo se encuentran todos los controladores que se comunican con el <i>frontend</i> . Los controladores usan el módulo IBusinesslogic y toman decisiones dependiendo de la petición del usuario como por ejemplo la información que le retorna al usuario. Además los controles de seguridad se implementan en esta capa autenticando y autorizando al usuario según el rol que posea.
IBusinessLogic	Este módulo tiene la responsabilidad de definir el contrato que debe tener la capa de negocio, actúa de nexo entre la <i>API</i> y la implementación de la capa de negocio (BusinessLogic), permitiendo bajar el acoplamiento, mejorar la mantenibilidad y poder implementar <i>reflection</i> e inyección de dependencias.
BusinessLogic	Contiene todas las operaciones del negocio del sistema, está conformada por diversos módulos que se encargan de garantizar la operación correcta de cada una de las operaciones del sistema. Se encuentran todas las validaciones necesarias al realizar cada acción sobre el sistema.
IDataAccess	Tiene la responsabilidad de definir el contrato que deberá tener la capa de acceso a datos, actúa de nexo entre la capa de negocio y la capa concreta de acceso a datos (DataAccess), el objetivo de esta capa es bajar el acoplamiento, mejorar la mantenibilidad y poder implementar <i>reflection</i> e inyección de dependencias.
DataAccess	Contiene toda la lógica necesaria para el mantenimiento de las entidades, baja alta y modificación de entidades. Además se encuentra la definición del contexto de la persistencia del sistema.
ServiceFactory	Contiene todas las operaciones necesarias para poder hacer la inyección de dependencias y <i>reflection</i>
CompraCercaTests	Se encuentran todas las pruebas automáticas realizadas sobre la lógica del DataAccess y de la BusinessLogic

WebServices	Se encuentra toda la lógica necesaria para comunicarse con la <i>API</i> de mercado libre para obtener la predicción de la categoría
Domain	En este módulo se encuentran todas las definiciones de entidades que serán persistidas en la base de datos SQL.

5.9.2. Catálogo de entidades

Se mencionan las principales entidades del sistema que son persistidas en la base de datos con una breve descripción:

Catálogo de las principales entidades del sistema

Elemento	Descripción
Business	Encapsula toda la información pertinente al negocio como dirección, latitud, longitud, dirección, entre otros datos. Además se relaciona a la entidad categorías pues ese comercio puede trabajar con una o más categorías de productos o servicios
Category	Encapsula toda la información pertinente a la clasificación de productos o al de un servicio.
User	Encapsula toda la información pertinente a un usuario registrado como nombre, correo, entre otros datos. Además de asociarse a uno o más roles (usuario cliente, usuario comerciante) y a uno o más comercios de tener el rol usuario comerciante.

5.9.3. Decisiones de diseño

El siguiente apartado tiene como objetivo describir y mencionar las principales características del sistema en cuanto a su diseño.

El *backend* se encuentra dividido y estructurado en capas (MVC), lo cual como se mencionó en el capítulo [MVC](#) favorece la mantenibilidad, el bajo acoplamiento y la independencia en responsabilidades, mitigando el impacto al cambio.

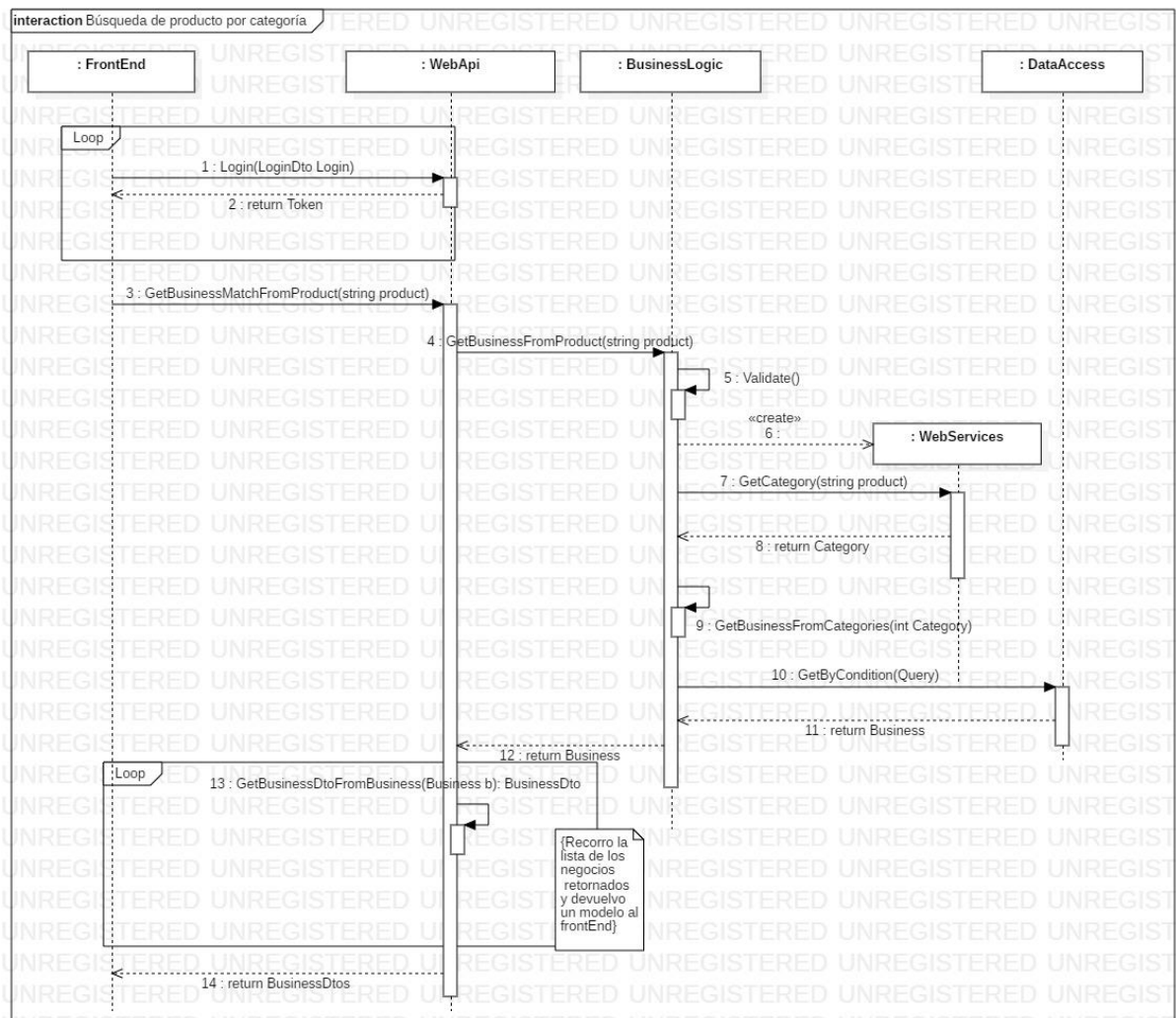
Asignamos las *primary key* de las entidades “Categoría” para que tengan un sentido semántico, el dígito 0 en la *primary key* es un dígito reservado que actúa como separador, las categorías padre no contienen el 0. Por ejemplo la categoría “Tecnología” es una categoría padre y tiene como *primary key* el 1, en cambio “Celulares y telefonía” es subcategoría de “Tecnología”, pues tiene de *primary key* 101, el 0 es el dígito separador, lo que se encuentra antes del 0 nos indica a cuál categoría pertenece, “Tecnología” en este caso y lo que precede al 0 corresponde a su numeración relativa. Esto tiene como ventaja que la búsqueda de categorías de productos o sub-categorías sea rápida pues se comparan contra la *primary key* en la base de datos, y que a nivel de *Backend* la búsqueda sea más sencilla e intuitiva.

Se hace uso de la inyección de dependencia y de *reflection* con el objetivo de facilitar y aislar las pruebas unitarias, definir explícitamente las dependencias de una clase facilitando así el buen diseño (principio de responsabilidad única, SRP),

Tal como se explicó en capítulo de los patrones [Composite](#) y [Fachada](#) se resolvieron problemas concretos como diseñar un árbol de categorías y ocultar complejidad del sistema respectivamente con el objetivo de favorecer la mantenibilidad, bajar el acoplamiento y aumentar la cohesión del sistema. Además se implementó el patrón Repository con el fin de disminuir la complejidad del acceso a datos y de mantenibilidad por si en un futuro se cambia el motor de base de datos.

Se implementa el patrón *Factory* a nivel del paquete de DataAccess con el objetivo de delegar la responsabilidad de crear un contexto de base de datos, esto nos permite por ejemplo al realizar la pruebas, crear un contexto de base de datos en memoria para realizar las pruebas unitarias o el de crear un contexto para acceder al repositorio de datos cuando el sistema se encuentra en ejecución.

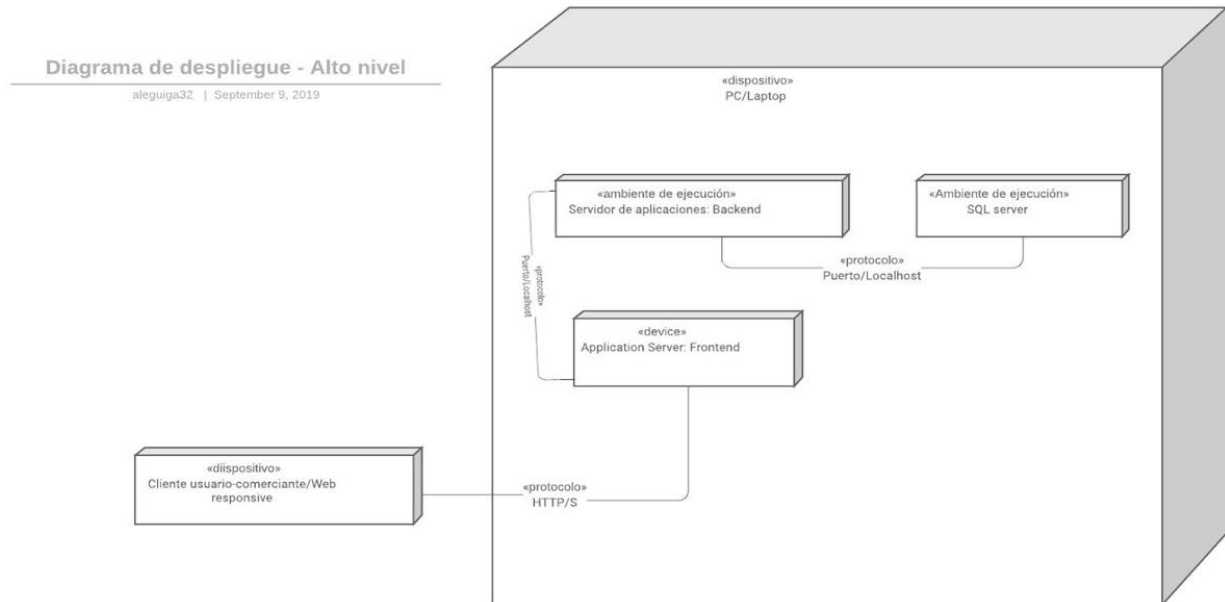
5.9.4. Diagrama de Secuencia



Como se puede observar en el diagrama el usuario comprador se *loguea* primero si ya no lo hizo para luego buscar un producto por descripción, para ello se hace uso del paquete “Web service” que resuelve la predicción de la categoría del producto al usar la API de MercadoLibre. Esta API nos devuelve un listado de categorías como resultado y su correspondiente porcentaje de coincidencia, nos quedamos con el de mayor porcentaje de coincidencia. Una vez lograda la predicción buscamos todos los negocios que trabajan con esa categoría, para luego convertir dicha respuesta en un modelo para el *frontend*.

5.9.5. Vista de Despliegue

A continuación se presenta el diagrama de despliegue en el cual se presenta el mapeo de las unidades de software a elementos en el ambiente en el cual van a ser ejecutados.



6. PROCESO DE DESARROLLO

El objetivo de esta sección es explicar la forma de trabajo adoptada y consensuada por parte del equipo de desarrollo del proyecto.

6.1. Características del equipo

El equipo del proyecto está compuesto por 3 integrantes Agustín Taruselli, Alejandro Mamchur e Ignacio Peirano, los cuales trabajan en el rubro informático hace ya varios años. En el caso de Agustín, se desempeña en el área de QA de una empresa que desarrolla aplicaciones móviles para el exterior, acumulando ya varios años en este rol. Trabajó previamente 8 años en IBM adquiriendo una gran experiencia y conocimiento en su área. Su rol y visión apartada en el día a día del desarrollo provee una mirada desde una perspectiva distinta al proyecto, algo que consideramos sumamente importante. Alejandro por su parte, trabaja en una empresa de telefonía multinacional, desempeñando el rol de desarrollador sintiéndose a gusto tanto en *backend* como *frontend*. Este conocimiento amplio que posee, nos aportó una visión integral del producto a desarrollar. Su capacidad de liderazgo y entusiasmo fueron fundamentales en el proyecto. Por último, Ignacio trabaja como desarrollador en una empresa nacional que tiene como clientes a la mayoría de los entes públicos. Sus fortalezas están en el trabajo diario con lenguajes *scripting* que son un pilar del sistema y su pasión por el diseño de interfaces.

6.2. Responsabilidades

En cuanto al desarrollo de la aplicación, la arquitectura, el *backend* y la base de datos de la aplicación tuvo como principal responsable a Alejandro Mamchur. Encargado de aplicar patrones de arquitectura para cumplir con los atributos de calidad del software que se propusieron implementar las capas de negocio de la aplicación.

Agustín trabajó en el *backend*, en la investigación de metodologías y herramientas para la traducción de una frase que el usuario ingresa en la barra de búsqueda a una categoría de productos o servicio preexistente en el sistema. También realizó el modelo de la base de datos de acuerdo a las entidades de negocio que se identificaron.

Ignacio se encargo del *frontend*. Para su desarrollo trabajó utilizando en conjunto los *frameworks* Angular y Ionic. Aquí su principal tarea fue el diseño de la *UI*, el manejo del mapa de la aplicación y el desarrollo de las llamadas a las *API* del *backend* para consumir su información.

En cuanto a la gestión del proyecto y la documentación Agustín fue el líder de esta área. Desempeñando el rol de *Scrum Master* y siendo experto en este marco de trabajo, entrenó y guió al equipo en la forma de trabajar y fue el encargado de la utilización de herramientas de *Scrum* como *TaskBoards*.

6.3. Estrategia de desarrollo

Lo primero que tuvimos que realizar fue la división de roles para la construcción del sistema. Para esto se tomó en cuenta el conocimiento inicial de cada integrante y su predisposición para efectuar cada tarea. Una vez definidos estos roles, se procedió a la investigación por separado de las tecnologías y herramientas de trabajo necesarias para cumplir con el objetivo de cada integrante.

Culminada la investigación, se realizaron reuniones entre los integrantes del proyecto para lograr un consenso, donde cada integrante tuvo que explicar su investigación y convencer a los demás integrantes de cuál es la mejor solución.

Llegado a un común acuerdo, cada integrante procedió formarse y estudiar aquellos conocimientos que no tenían para comenzar formalmente en el desarrollo con bases sólidas y no cometer errores o decisiones que pudieran comprometer el éxito del proyecto. En esta etapa se procedió a una retroalimentación constante del grupo en apoyo con la visión de la tutora, la ayuda de personas afines al grupo que pudieran dar su recomendación acerca de cómo resolver los problemas que ya estaban planteados o qué herramienta utilizar para minimizar el tiempo de desarrollo, para lograr un producto de mayor calidad. Dentro de esta etapa surgió por ejemplo la idea de utilizar el *framework* Ionic por recomendación de un compañero de trabajo de Agustín. A pesar de que ya se había definido la utilización de Angular, Ionic que utiliza Angular provee componentes web que facilitan el diseño y no solo eso ya que permite sin ningún esfuerzo la creación de aplicaciones híbridas [12], con posibilidad de crear aplicaciones nativas.

Dentro de esta etapa también se definió en el *frontend* si se utilizaba el *plugin* de Google Maps para *Angular* o la *API Javascript* de Google Maps. Esta decisión fue difícil y se optó por la *API javascript* por la cantidad de información en la web por encima del riesgo de alguna incompatibilidad que pudiera surgir.

Con los requerimientos del sistema ya priorizados y definido el *MVP* de funcionalidades del sistema para la primera etapa de desarrollo, se procedió a definir la duración en días de cada sprint y colocar las funcionalidades a realizar dentro de cada sprint.

Con esta planificación, se realizó un *Sprint Planning* al comienzo de cada sprint donde se fraccionaron las funcionalidades en tareas, cada una con su estimación en *Story Points* realizada por todos los integrantes del proyecto.

Se realizó la *Daily Meeting* todos los días del proyecto. Algunas veces se realizó remoto a través de *Whatsapp* o llamada de *Hangout* y otras veces se realizó presencial en casa de Alejandro que sirvió como laboratorio base para el equipo.

Una vez finalizado el *sprint*, se procedió a una instancia en la que cada integrante contó cómo se sintió durante el *sprint*, se hizo un balance de los objetivos cumplidos y se propusieron variantes en la metodología como por ejemplo la necesidad de juntarse personalmente más días.

A continuación se podrá apreciar una representación de cómo se trabajó a partir del primer *sprint*. La reunión de retrospectiva, se efectuó a la noche del día de finalización de los *sprints*. En la primera imagen se observan los *sprints* relacionados puramente al desarrollo del software. Las tareas de *testing* se efectuaron en paralelo al desarrollo, realizándose durante el desarrollo de un *sprint* las pruebas del sprint anterior.

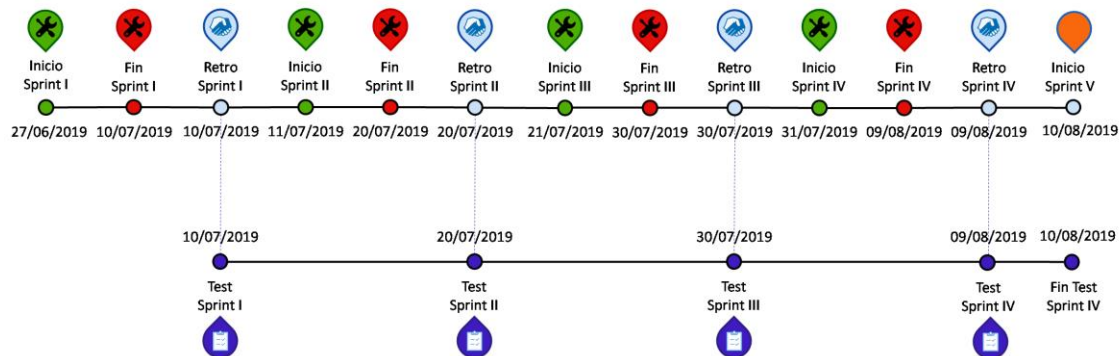


Imagen 11.1 – Sprints de desarrollo

Culminadas las iteraciones de construcción del *software*, se pasaron a aquellas relacionadas a los test de regresión, cargando la base de datos con datos reales del entorno y documentación. En este período se realizaron entregas periódicas de la documentación a la tutora.

Se mantuvo la utilización de *Scrum* en estas etapas debido a que facilitaba la división de las tareas y el tiempo era acotado para todas las tareas que debían realizarse.

El 2 de Setiembre, durante el *sprint 7*, se le entregó a la tutora una versión preliminar del documento final para que se pudieran pulir detalles o algún elemento faltante. El 9 de Setiembre, cuando finalizó el *sprint 7*, se realizó una segunda entrega preliminar, esta vez con el documento completo en cuanto a su contenido.

El trabajo en esta etapa se representa en la próxima imagen.

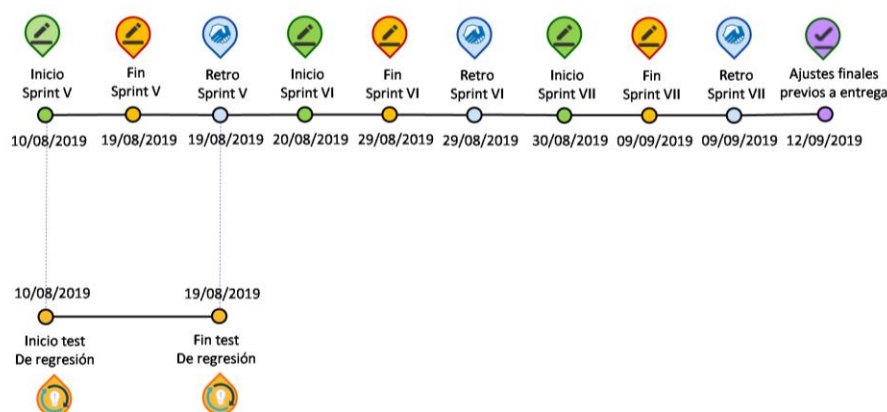


Imagen 11.2 – Sprints de documentación

6.3.1. Requerimientos Incluidos en el MVP

Se listan a continuación los requerimientos que se incluyeron en esta primera etapa

ID Req.	Requerimiento	% Implementación
RF1	Login Usuario	100
RF2	Registro de Usuario	100
RF3	Buscar producto por categoría	100
RF4	Buscar producto por coincidencia exacta	100
RF5	Agrupar marcadores cercanos	100
RF6	Consultar información de comercio	100
RF18	Dashboard de Control	100

Los requerimientos no implementados en esta primera etapa serán desarrollados en futuras etapas de desarrollo, una vez completada la etapa académica del proyecto.

7. GESTIÓN DEL PROYECTO

Se incluirán en este capítulo las tareas realizadas asociadas a gestionar el proyecto, incluyendo actividades de gestión de riesgos, esfuerzo, comunicaciones y las herramientas utilizadas para llevar a cabo dichas actividades.

Para llevar adelante el proyecto se eligió un ciclo de vida de prototipación evolutiva y la metodología ágil *SCRUM* (ver capítulo [Proceso y Ciclo de Vida](#) para más detalles respecto a esto). El trabajo a realizar se dividió en *sprints* que comenzaron siendo de dos semanas para la investigación y selección de herramientas y se cambiaron a diez días en etapa de desarrollo.

Consideraciones generales que se tomaron en cuenta en etapa de planificación del proyecto:

- Ninguno de los integrantes del equipo contaba con experiencia previa en la tecnología elegida para el desarrollo del *frontend*.
- Los tres integrantes del equipo cuentan con empleos de horario completo y cursan materias en paralelo con el proyecto.
- Dos de los tres integrantes del equipo no tienen experiencia laboral con *SCRUM*.
- Parte del equipo no contaba con experiencia previa en *Gitflow*.

Dados los puntos anteriores es que se optó por comenzar el trabajo por un primer sprint de investigación de herramientas, tecnologías y lectura sobre la metodología seleccionada para trabajar durante el proyecto. De esta manera se logró nivelar el conocimiento del equipo en puntos clave para el éxito del proyecto.

7.1. Herramientas utilizadas para la gestión del proyecto

Se describen a continuación las herramientas utilizadas durante el proyecto que sirvieron de soporte a la gestión de las actividades y tareas realizadas.

7.1.1. Planificación de tareas

Se utilizaron *sprints* de duración fija para controlar el avance del tiempo el proyecto. Al trabajar con una metodología ágil, las tareas a realizar durante los *sprints* no se planificaron en un cronograma sino que se utilizó un tablero de *SCRUM*.

Se utilizó la herramienta Trello (<https://trello.com/b/KlmbA9mP/compra-cerca>) que permite crear un tablero de trabajo donde se agregan las historias de usuario y las tareas. Mediante este tablero es posible realizar un seguimiento del avance del trabajo a lo largo de cada iteración de manera gráfica y clara para todos los integrantes del equipo.

Se muestra a continuación un ejemplo de cómo se veía el tablero de trabajo durante la ejecución del segundo *Sprint*

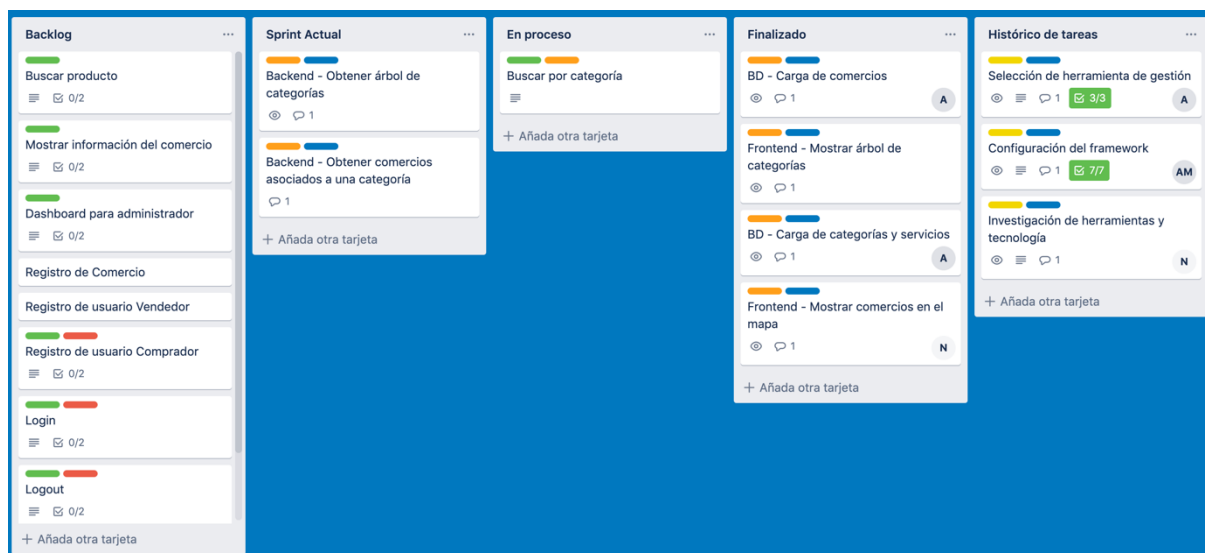


Imagen 12.1 – Sprint Dashboard

Explicación de las columnas del tablero:

- **Backlog:** Contiene el Product *Backlog*. En esta columna se encuentran todas las historias de usuario y tareas sin completar que se crearon como resultado del relevamiento de requerimientos que aún no fueron incluidas en un *sprint*, permanecen en esta columna hasta que sean incluidas en un *sprint* durante la planificación.
- **Sprint Actual:** Contiene el *Sprint Backlog*. En esta columna se encuentran todas las tareas que se incluyeron en la iteración pero que aún no las tomó ningún integrante del equipo para trabajar en ellas.
- **En proceso:** Aquí se encuentran las tareas sobre las que se está trabajando actualmente.

- **Finalizado:** Tareas que fueron completadas.
- **Histórico de tareas:** Tareas completadas en *sprints* anteriores, al cerrar el sprint se mueven las tareas a esta columna.

Como complemento al tablero de Trello, se creó una planilla que incluye la planificación de fechas para cada sprint en cuanto a duración (fecha inicio, fecha fin), el registro de puntos estimados a completar y completados reales y una gráfica con el *Burndown* que permite ver de forma clara cómo fue el avance del trabajo durante el *sprint* (ejemplo a continuación).

Planificación macro de los *sprints*

Sprint	Inicio	Fin	Comentarios
1	27/06/2019	10/7/2019	Configuración, investigación de herramientas y preparación de ambientes
2	11/7/2019	20/7/2019	Desarrollo
3	21/7/2019	30/7/2019	Desarrollo
4	31/7/2019	9/8/2019	Desarrollo
5	10/8/2019	19/8/2019	Documentación y pruebas de regresión
6	20/8/2019	29/8/2019	Documentación y corrección de defectos
7	30/8/2019	8/9/2019	Documentación

Tabla 12.1 – Planificación Macro Sprints

Ejecución del Sprint

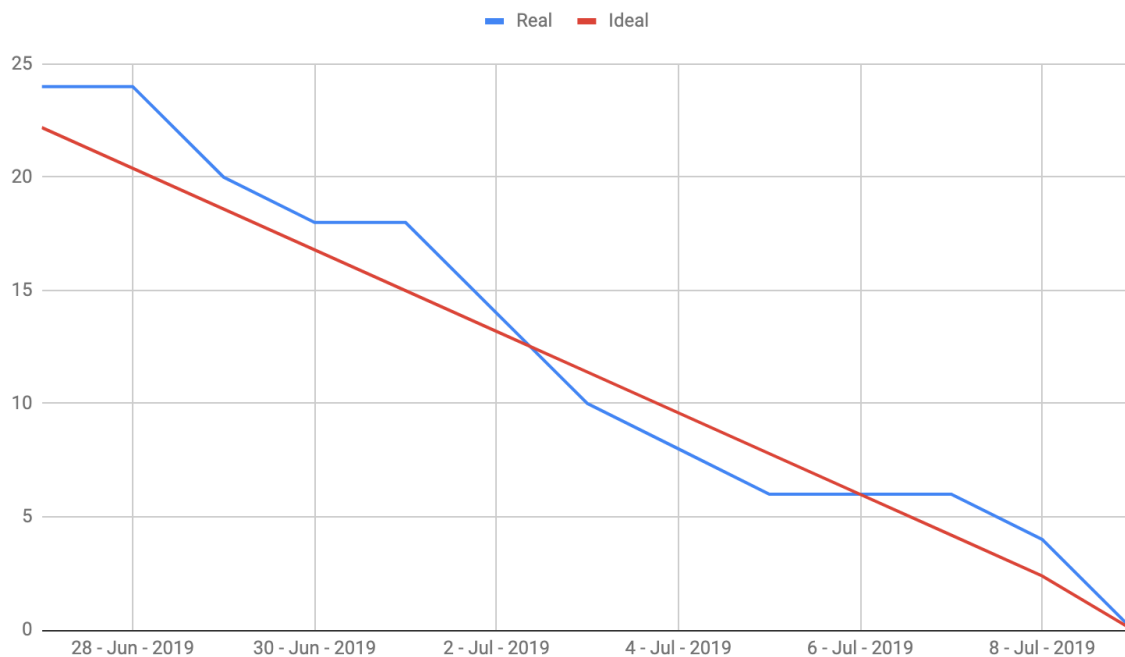
Para mostrar de una manera gráfica lo mencionado anteriormente se agregan los datos de la ejecución del *sprint* 1, en el que se completaron todas las tareas incluidas y la ejecución del *sprint* 2 como ejemplo de un caso en que no se logra cumplir todas las tareas.

Sprint 1

SP estimados	24
SP completados	24
Carry Over	0

- SP Estimados: representa la cantidad de *Story Points* estimados a completar en el *sprint*
- SP Completados: representa la cantidad de *story points* realmente completados durante el sprint
- Carry Over: corresponde a los puntos que quedan sin completar al final del *sprint*

Burndown chart



Gráfica 12.1 – Burndown Chart Sprint 1

En la gráfica se muestra la evolución del trabajo durante el sprint, la cantidad de puntos reales completados vs lo ideal para que la distribución del trabajo sea igual durante todo el sprint

Tablero al cierre del *sprint*

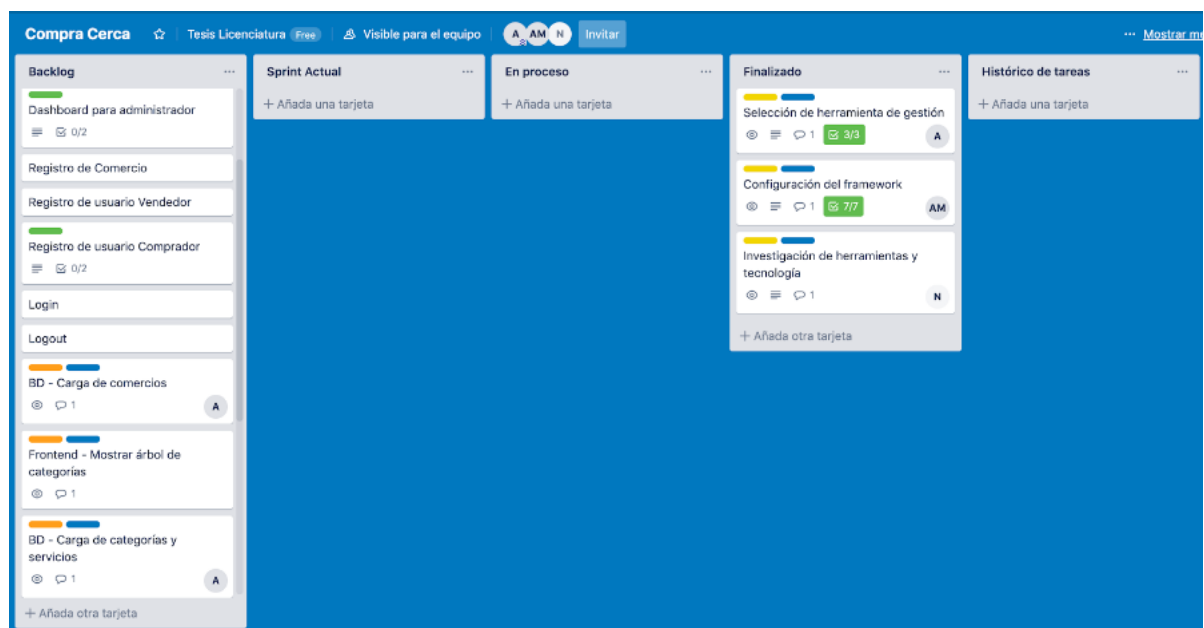


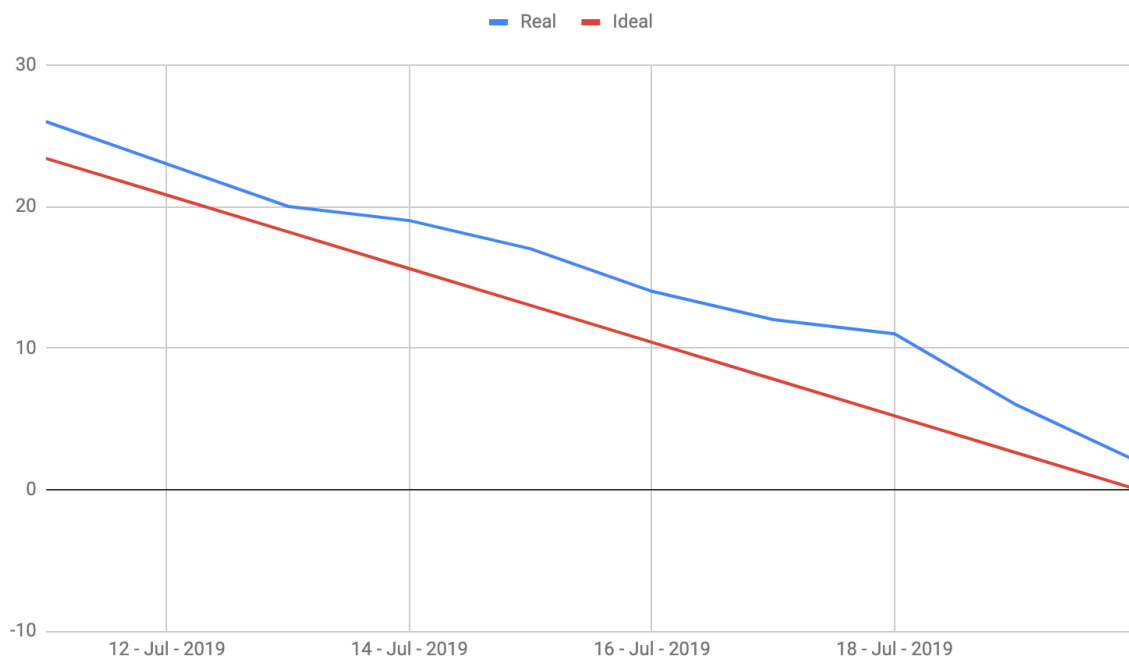
Imagen 12.2 – Sprint Dashboard al cierre del sprint 1

En este *sprint* se completaron todas las tareas planificadas, pero esto no siempre es así, ya que por ejemplo en el segundo *sprint* no se llegaron a completar todas las tareas, lo que genera un *carry over* para el siguiente.

Sprint 2

SP estimados	26
SP completados	24
Carry Over	2

En el *burndown chart* se puede apreciar que los puntos reales completados no llegan a tocar la línea



Gráfica 11.2 – Burndown Chart Sprint 2

El tablero al final del segundo *sprint* queda con tareas en proceso

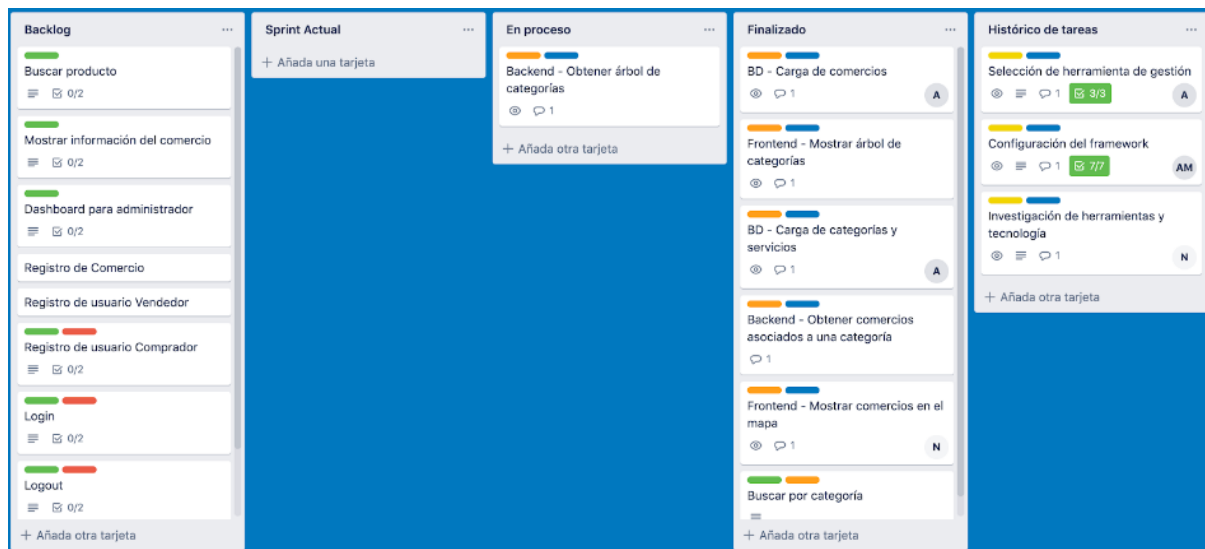


Imagen 12.3 – Sprint Dashboard al cierre del sprint 2

7.1.2. Estimación y registro de esfuerzo

Técnica para la estimación de las tareas

La estimación de las historias de usuario se realizó utilizando *Story Points* como medida de esfuerzo necesario para que la tarea sea completada.

Durante la *Planning Meeting* del *Sprint* se estimaron las tareas utilizando la técnica de “*Planning Poker*” que consiste en que cada integrante del equipo otorgue un valor en *Story Points* a la tarea que se está estimando utilizando la sucesión de Fibonacci (0, 1, 2, 3, 5, 8, 13, 21, 34...). Una vez que todos los integrantes del equipo votaron, si todos concuerdan con el valor puesto se le asigna el valor a la tarea, sino se debe discutir hasta llegar a un consenso.

Registro del esfuerzo

Entendemos por esfuerzo las horas dedicadas por parte de los integrantes del equipo a tareas del proyecto. Para el registro de esta información se creó una planilla en el repositorio del proyecto con el fin de que cada integrante registre el tiempo que dedica al proyecto diariamente, especificando la tarea que realizó y más información de interés estadístico.

De acuerdo a lo sugerido en tutoría y se estimaba un aproximado de 20hrs semanales de dedicación por integrante.

Se muestra a continuación un ejemplo de la planilla de registro de horas utilizada.

Categoría	Sub-categoría	Horas	Fecha	Integrante
Reuniones con tutora		3	22/04/2019	Equipo
Reuniones con tutora		3	29/04/2019	Equipo
Reuniones con tutora		3	06/05/2019	Equipo
Reuniones de equipo		6	08/05/2019	Equipo
Reuniones con tutora		3	13/05/2019	Equipo
Reuniones de equipo		6	15/05/2019	Equipo
Reuniones con tutora		3	20/05/2019	Equipo
Reuniones de equipo		6	22/05/2019	Equipo
Gestión	Documentación	2	02/06/2019	Agustin
Arquitectura	Investigación	2.5	02/06/2019	Alejandro
Requerimientos	Validación	2.5	02/06/2019	Ignacio
Gestión	Documentación	2	02/06/2019	Agustin
Arquitectura	Investigación	2.5	02/06/2019	Alejandro
Requerimientos	Documentación	2.5	02/06/2019	Ignacio
Reuniones de equipo		2	03/06/2019	Agustin
Reuniones de equipo		2	03/06/2019	Alejandro
Reuniones de equipo		2	03/06/2019	Ignacio
Gestión	Seguimiento	1	03/06/2019	Agustin

Tabla 12.2 – Registro de horas

7.1.3. Seguimiento de las tareas

El seguimiento del avance de las tareas se realizó utilizando el tablero mencionado en la sección de [planificación de las tareas](#) del *sprint*.

El tablero del *sprint* contiene las tareas en las que se está trabajando y permite tener una idea clara del estado de avance general del trabajo realizado.

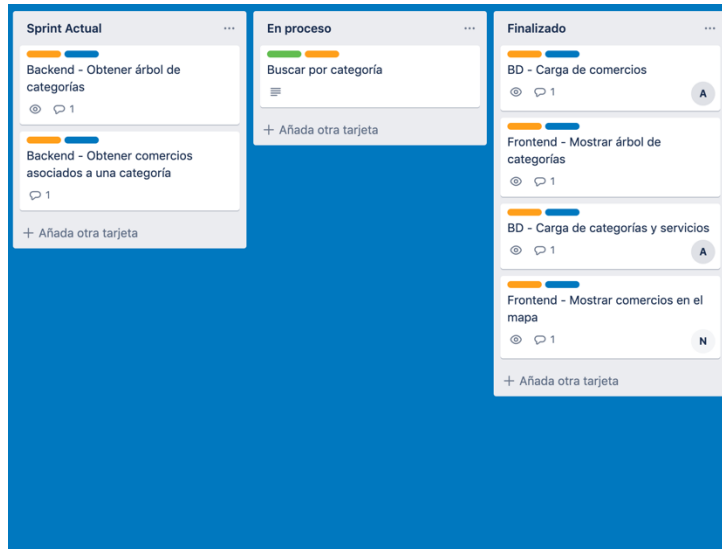


Imagen 12.4 – Tareas en proceso

El flujo definido para las tareas es el siguiente:

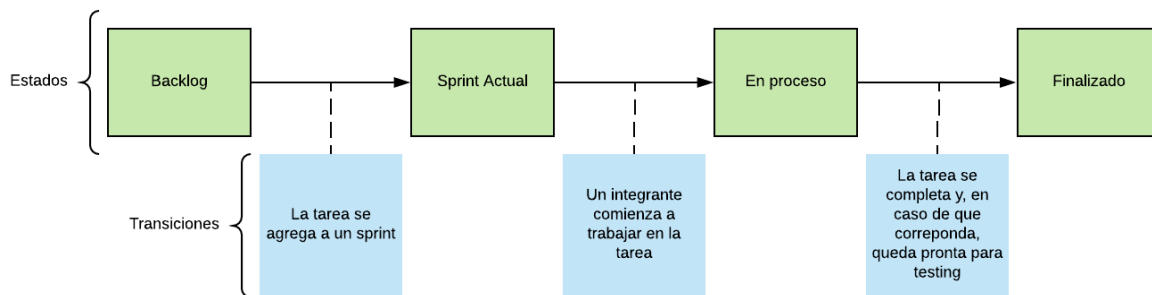


Imagen 12.5 – Flujo de estados de las tareas

Si una tarea no llega a completarse durante el sprint, se agrega al siguiente como tarea en proceso y los *Story Points* asociados a la misma se cuentan como *carry over*.

7.2. Riesgos

A continuación se listan los riesgos identificados para el proyecto.

Se planificaron mediciones semanales de los de mayor magnitud para determinar a tiempo las acciones a tomar.

Riesgo	Impacto (I)	Prob. (P)	Magnitud = I x PO	Plan de Respuesta	Plan de contingencia
Trabajar con tecnología con la que no se tiene experiencia (Mediano plazo)	3	0.8	2.4	Disponer de toda la información posible sobre cada una de las alternativas entre las que es posible escoger y las consecuencias que cada una podría conllevar respecto a los objetivos marcados.	Decidir la opción que tenga el menor grado de dificultad. Elegir formas de implementación en que el equipo tenga experiencia previa.
Mala estimación del esfuerzo (Mediano plazo)	3	0.8	2.4	Hacer estimaciones considerando todos los elementos disponibles y utilizar técnicas formales de estimación para minimizar el error.	Dedicación extra de tiempo para poder cumplir con las especificaciones.
Degradación de la calidad del software por recortes de actividades (Largo plazo)	3	0.6	1.8	Dar prioridad en el sprint aquellas tareas que son clave en la calidad del entregable final.	Realizar horas extras para minimizar la cantidad de actividades recortadas.
Cambios en	3	0,4	1.2	Ante cambios en los requerimientos se crearán las tareas necesarias para	Evaluar el cambio y determinar el impacto del mismo en el proyecto. Una vez analizado

requerimientos (Inmediato)				cumplirlos y se incluirán en futuros sprints de acuerdo a su prioridad.	se podrá determinar la mejor manera de llevarlo a cabo
Enfermedades (Inmediato)	4	0,2	0.8	Cada integrante del equipo está al tanto del progreso del resto. Están capacitados para dominar la situación en la que se encontraba su compañero.	Priorizar las tareas pendientes y reasignarlas de acuerdo a su importancia para que alguno de los integrantes del equipo las pueda finalizar
Pérdida/Robo/Falla de equipos de trabajo (Inmediato)	3	0,2	0.6	Todos los cambios que cada integrante realice, deben ser persistidos de manera online (por ejemplo: guardarlos en una nube).	Conseguir un equipo alternativo para que todos los integrantes puedan continuar con su trabajo
Mala comunicación entre los integrantes del equipo. (Medianoplazo)	3	0.4	0.6	<i>Feedback</i> constante acerca del proyecto y su avance. Toda la información de avance en la nube disponible para ser consultada.	Reunión entre los integrantes del equipo para corregir cualquier comportamiento o método que interfiera con los objetivos del proyecto
Impacto: 0 - Ninguno. 1 - Marginal. 2 - Poco importante. 3 - Importante (puede retrasar el proyecto). 4 - Crítica (puede detener el proyecto). 5 - Catastrófica (fracaso del proyecto). Probabilidad de ocurrencia: 0.0 - no probable. 0.2 - poco probable.					

0.4 - probable. 0.6 - muy probable. 0.8 - altamente probable. 1.0 - se convierte en problema	
---	--

Tabla 12.3 – Riesgos

Para definir qué riesgos se evalúan y se les realiza seguimiento con mayor prioridad nos basamos en la magnitud y a qué plazo se manifestarían.

- Alto: Magnitud alta (≥ 2) y plazo de ocurrencia mediano o inmediato.
- Medio: Magnitud media (≥ 1 y < 2) y plazo de ocurrencia mediano o inmediato.
- Bajo: Magnitud baja (< 1) y plazo de ocurrencia largo.

Riesgos a los que se catalogaron como “Alto” le debe realizar seguimiento con frecuencia semanal.

- **Trabajar con tecnología con la que no se tiene experiencia.** Se realizaron pruebas de concepto previo al comienzo del desarrollo y se consideró esta dificultad en la estimación de los tiempos de las tareas antes de incluirlas dentro de un *sprint*.
- **Mala estimación de los tiempos.** Se revisó el avance de las tareas a través del tablero del *sprint* diariamente para identificar desvíos importantes en el esfuerzo estimado para cada tarea a realizar.

Riesgos a los que se catalogaron como “Medio” le debe realizar seguimiento una vez por *sprint*.

- **Cambios en los requerimientos.** La metodología de trabajo mitigó este riesgo en gran medida.

No se le realizó seguimiento periódico a los riesgos catalogados como “Bajo”.

- **Mala comunicación entre los integrantes del equipo.**
- **Pérdida/Robo/Falla de equipos de trabajo.**
- **Enfermedades de integrantes del equipo.**
- **Degradación de la calidad del software por recortes de actividades.**

7.3. Gestión de las comunicaciones

Establecer una buena comunicación entre los interesados es clave para lograr un proyecto exitoso. Por lo anterior es que se definieron ciertos canales y medios de comunicación tanto para la comunicación interna del equipo como para la comunicación con otros interesados externos.

7.3.1. Comunicación interna

Se acordó con el equipo de proyecto reunirse dos veces por semana de manera presencial para realizar una puesta a punto y coordinar y planificar tareas.

Por otra parte se mantuvo comunicación diaria a través de mensajería vía Hangouts, Whatsapp y correo electrónico.

La reunión diaria de avance (ceremonia correspondiente a la metodología utilizada) se realizó mediante Whatsapp.

7.3.2. Comunicación hacia la tutoría

La comunicación con la tutora se realizó vía mail para el envío de documentos, reuniones presenciales de seguimiento y a través de Whatsapp para la coordinación de reuniones.

7.3.3. Comunicación hacia el socio de marketing

Se coordinaron reuniones mediante Whatsapp para realizar validación de requerimientos y vía correo electrónico para el intercambio de documentos (por ejemplo, la carta de experto).

7.3.4. Plan de comunicaciones

A continuación se detalla cómo fue la comunicación entre los integrantes del equipo, con el tutor, el cliente y las herramientas que se utilizarán.

En la siguiente tabla se indica el responsable de comunicar la información a lo largo del proyecto, el medio por el que se hará, los destinatarios y la frecuencia.

Qué se comunica	Responsable	Destinatarios	Medio/Herramienta	Frecuencia
Plan de proyecto	Gerente del proyecto	Equipo de desarrollo y tutoría de proyecto	- Reuniones - Vía mail - Repositorio	Cuando se elabora el plan
Avance del proyecto	Equipo de proyecto	Equipo de desarrollo y tutoría de proyecto	- Reuniones - Vía mail - Repositorio	Semanalmente
Avance diario	Equipo de desarrollo	Equipo de desarrollo	- Reuniones - Vía mail - Chat (Hangouts, Whatsapp)	Diario
Revisiones formales	Equipo de proyecto	Tutoría	Entrega formal con boleta	04/04/19 - Anteproyecto 18/06/19 - Revisión formal
Cambios en el proyecto	Equipo de proyecto	A las partes interesadas	- Reuniones - Vía mail	Cada vez que se requieran cambios
Finalización del proyecto	Gerente del proyecto	Todas las partes	Comunicación oficial	Cuando finaliza el proyecto

Tabla 12.4 – Plan de comunicaciones

7.4. Métricas de esfuerzo

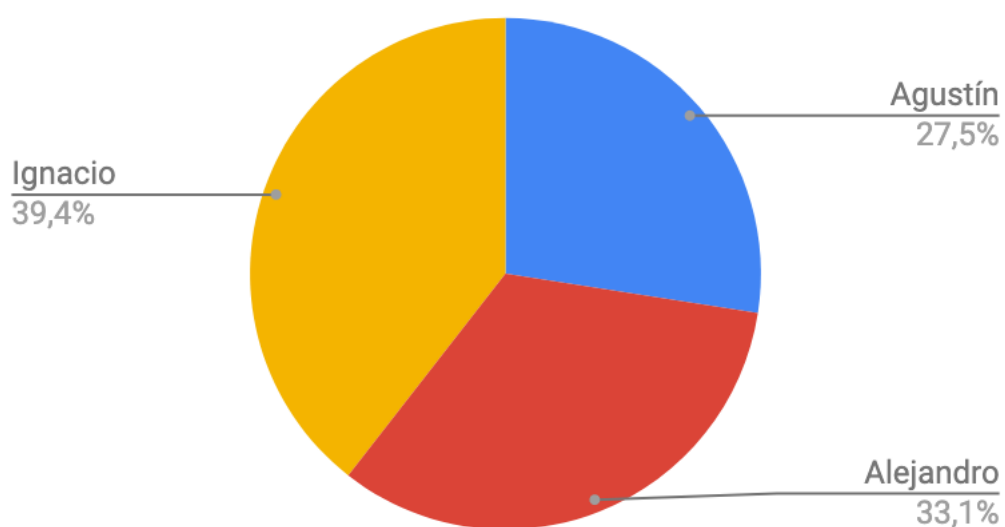
Se presentan a continuación las métricas obtenidas a partir del registro de esfuerzo realizado durante el transcurso del proyecto de los tres integrantes del equipo cumpliendo los diferentes roles que tomaron según el momento del proyecto. Las métricas consideradas excluyen el esfuerzo dedicado a documentar el proyecto como tal, por lo anterior es que se toman en cuenta los *Sprints* 1 al 4.

Se detalla el esfuerzo dedicado por recurso al proyecto en general y por *Sprint*. El esfuerzo por rol en general y por *Sprint*.

7.4.1. Esfuerzo total (en horas) por recurso

En el siguiente gráfico se aprecia la distribución total del esfuerzo dedicado por los integrantes del equipo entre los *Sprints* 1 y 4.

Horas dedicadas en total por recurso



Gráfica 12.3 – Horas totales por recurso

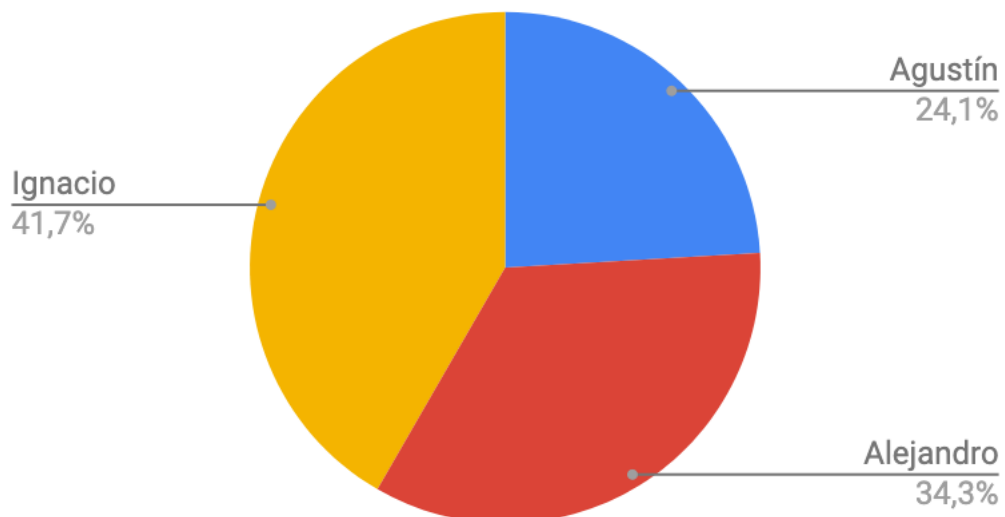
Dado a que los *Sprints* tomados en cuenta tuvieron como principal tarea el desarrollo, es esperable que la dedicación de los integrantes encargados del desarrollo fuera mayor a la del integrante del equipo que era responsable de la gestión y aseguramiento de la calidad.

7.4.2. Esfuerzo de los recursos por iteración

En el siguiente gráfico se muestra la distribución del esfuerzo dedicado por los integrantes del equipo en cada uno de los *Sprints* entre los *Sprints* 1 y 4.

Sprint 1

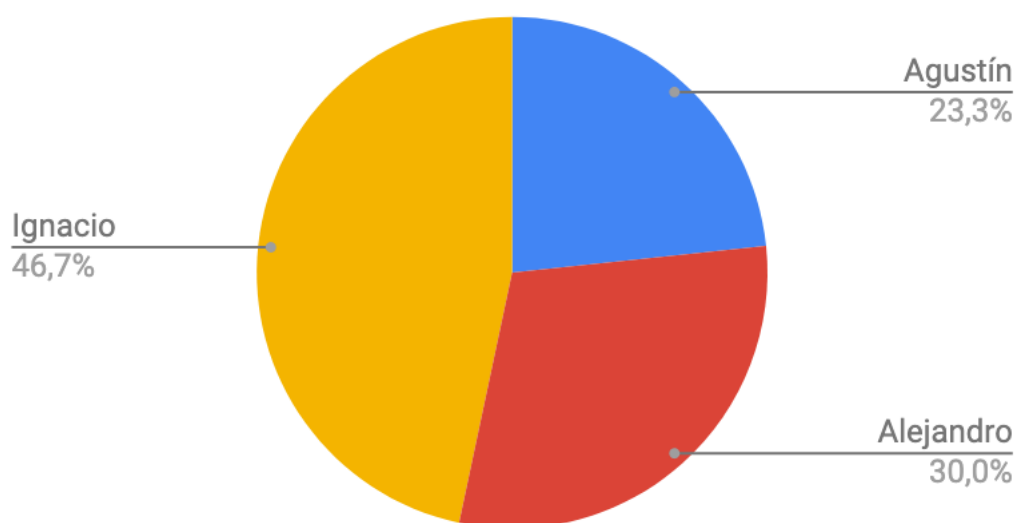
Sprint 1



Gráfica 12.4 – Esfuerzo recursos por iteración SP 1

Sprint 2

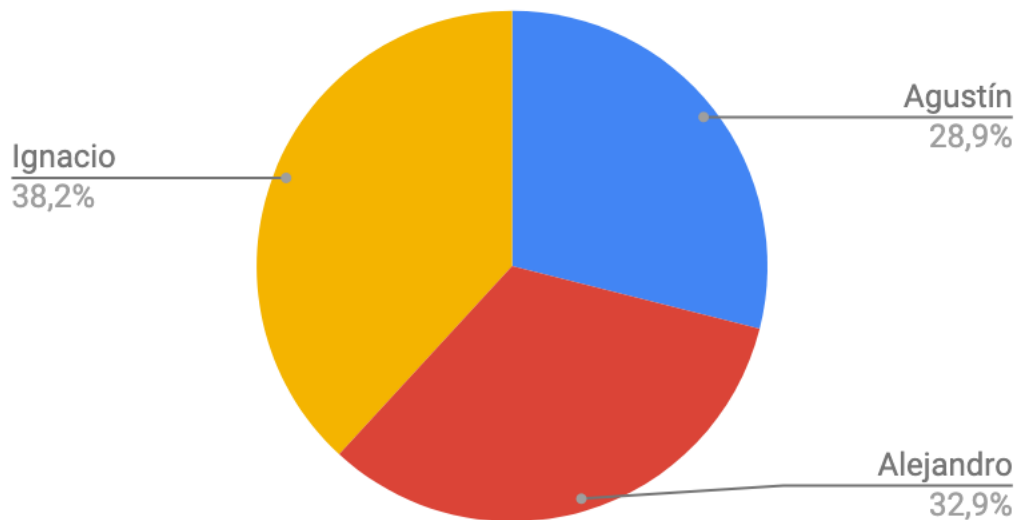
Sprint 2



Gráfica 12.5 – Esfuerzo recursos por iteración SP 2

Sprint 3

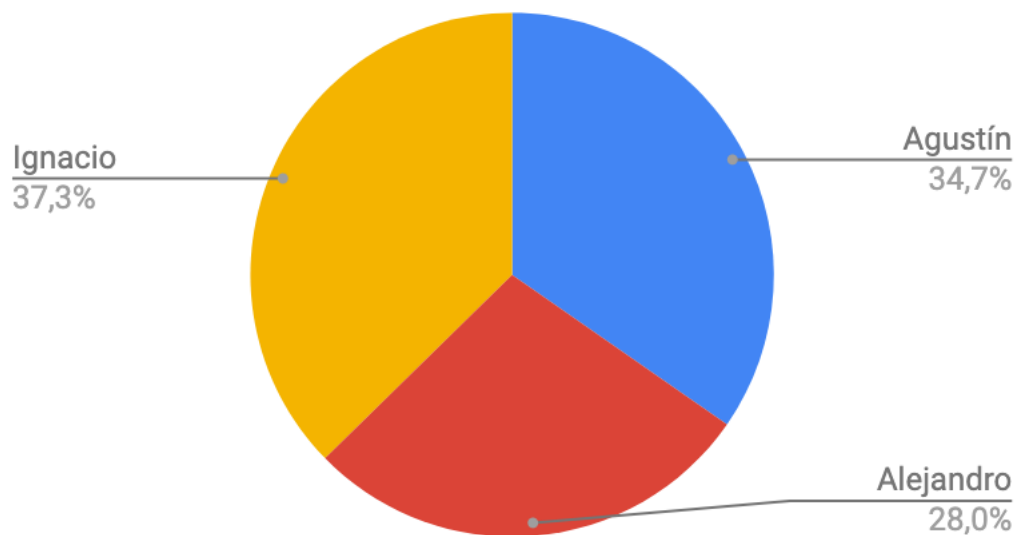
Sprint 3



Gráfica 12.6 – Esfuerzo recursos por iteración SP 3

Sprint 4

Sprint 4



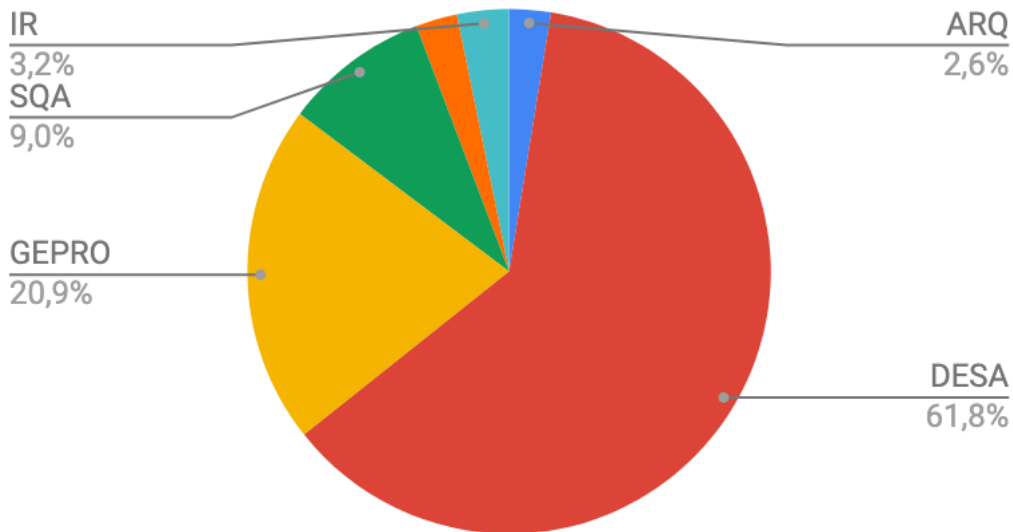
Gráfica 12.7 – Esfuerzo recursos por iteración SP 4

De los gráficos anteriores se observa que la distribución del esfuerzo se mantuvo en su proporción con el correr de los Sprints. En el *Sprint 4* la dedicación del responsable de aseguramiento de la calidad aumentó con respecto a los anteriores *Sprints* ya que se sumaron horas de pruebas de regresión.

7.4.3. Esfuerzo por rol

En la gráfica a continuación se muestra el esfuerzo de cada uno de los integrantes del equipo fueron tomando diferentes roles según las necesidades del proyecto.

Esfuerzo por rol



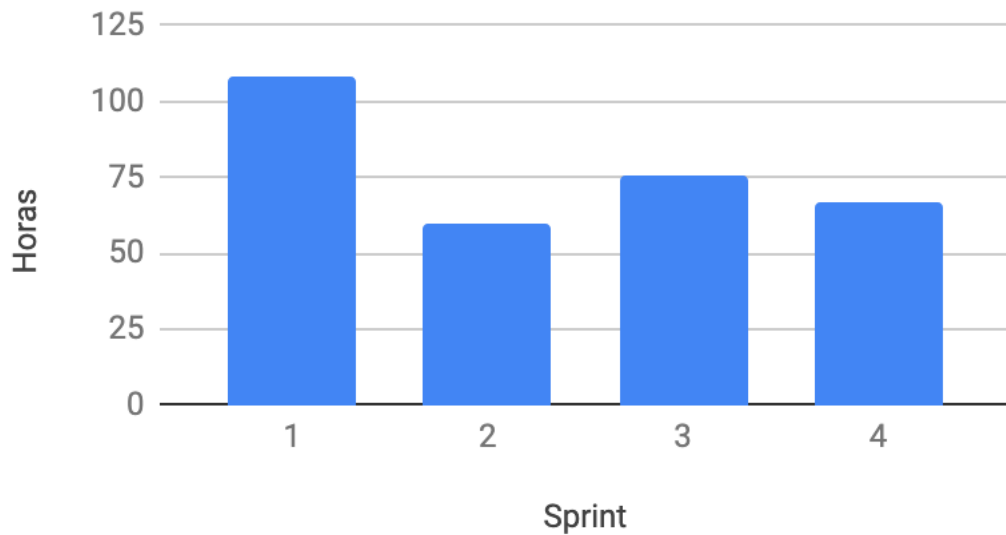
Gráfica 12.8 – Esfuerzo recursos por rol

La principal tarea en esta etapa del proyecto es la de desarrollo, por ello es que el rol DESA es quien toma protagonismo sobre los demás roles. Las tareas de IR y ARQ que se realizan en esta etapa no son muchas ya que se limitan a ajustes y mejoras por el lado de la arquitectura y actualización y relevamiento por parte de los requerimientos (el ciclo de vida de prototipado evolutivo que se seleccionó requiere que así sea durante todas las iteraciones). Las tareas de SQA cobran especial importancia en la ejecución de las pruebas funcionales al final de cada sprint y en las pruebas de regresión que se ejecutan al final del *Sprint 4*.

7.4.4. Esfuerzo total (horas) por Sprint

El siguiente gráfico muestra las horas dedicadas por parte del equipo de trabajo a cada uno de los *sprints*.

Esfuerzo por sprint



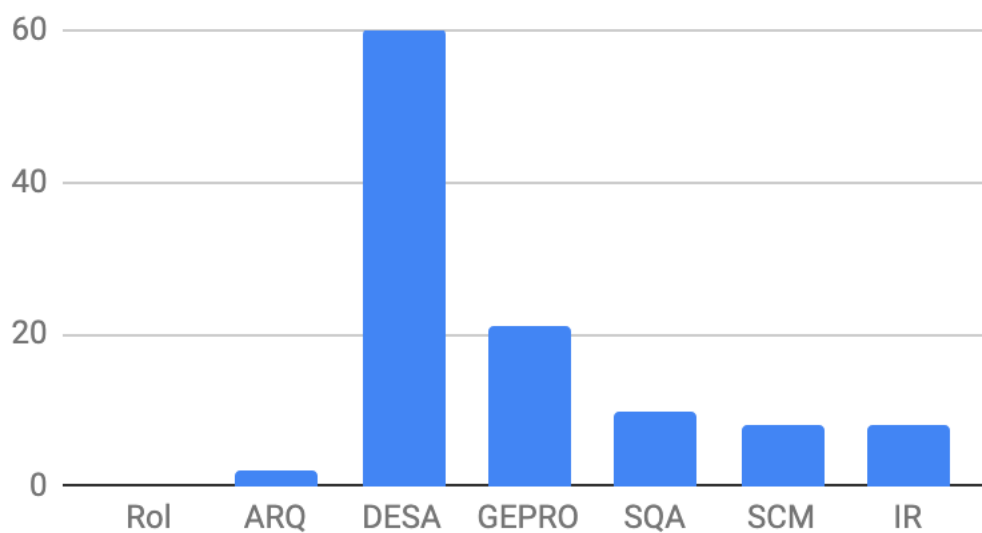
Gráfica 12.9 – Esfuerzo total en horas por sprint

Se puede apreciar una distribución bastante uniforme del esfuerzo para los *Sprints* 2, 3 y 4. La diferencia con el primer *Sprint* radica en que en éste se realizaron tareas de investigación de herramientas, configuración, pruebas de concepto y preparación de ambientes de trabajo que no fueron necesarias en los siguientes.

7.4.5. Esfuerzo por rol por Sprint

En las siguientes gráficas se muestra de qué manera se distribuyó el trabajo necesario de cada rol en los cuatro *Sprints* analizados.

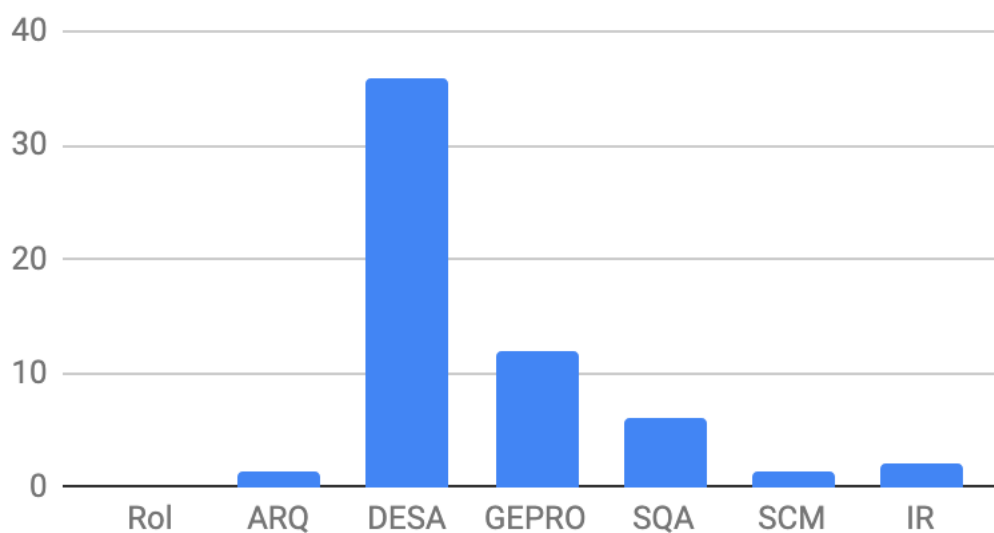
Sprint 1



Esfuerzo por rol Sprint 1

Gráfica 12.10 – Esfuerzo por rol por sprint SP1

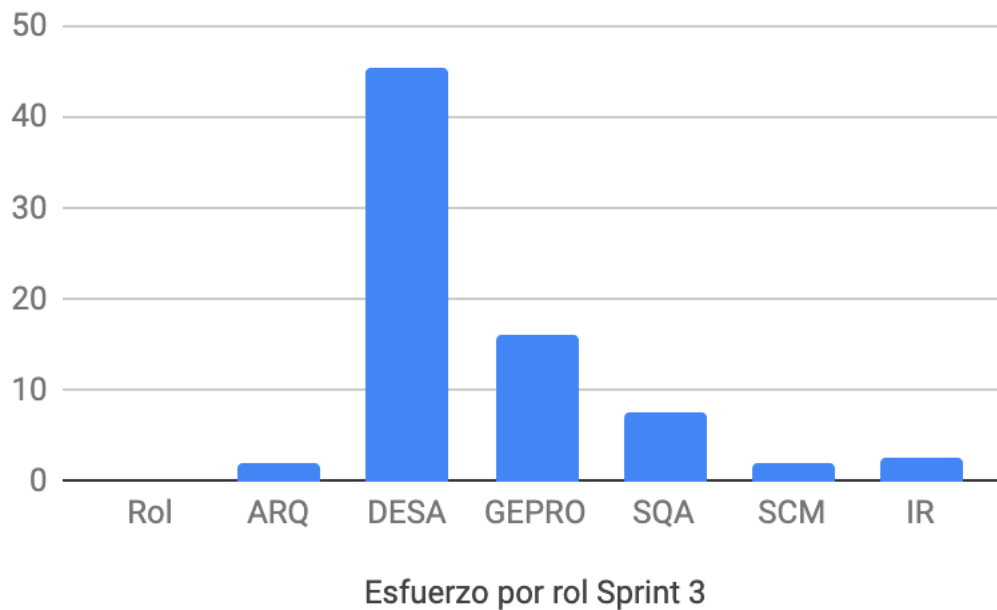
Sprint 2



Esfuerzo por rol Sprint 2

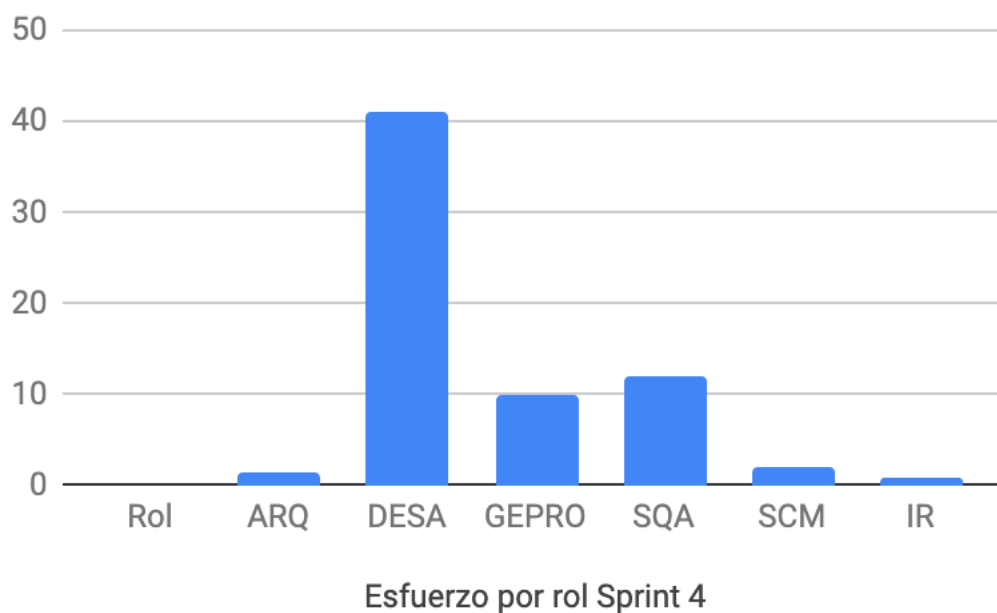
Gráfica 12.11 – Esfuerzo por rol por sprint SP2

Sprint 3



Gráfica 12.12 – Esfuerzo por rol por sprint SP3

Sprint 4



Gráfica 12.13 – Esfuerzo por rol por sprint SP4

La dedicación de cada rol a lo largo de los cuatro *Sprints* se mantuvo constante, con una notoria diferencia en el rol de DESA debido a que el foco de estos *Sprints* era el desarrollo de la aplicación. Se puede observar además el aumento del esfuerzo de SQA en el cuarto *Sprint*, esto se debe a la ejecución de pruebas de regresión que no estuvieron presentes en los anteriores.

8. GESTIÓN DE CALIDAD

Para asegurar la calidad tanto del proyecto como del producto a desarrollar es que se establecieron los estándares de calidad a cumplir y la forma de satisfacerlos. Estos estándares surgen de experiencias previas de otros proyectos, de los objetivos del proyecto, así como también del tiempo, costo y recursos disponibles.

La intención de estas tareas es la de encontrar posibles errores/defectos lo antes posible para evitar su propagación y así evitar costos innecesarios.

Se realizaron revisiones que permitan saber si lo que se está construyendo cumple con los requisitos funcionales y no funcionales especificados, si el código es eficiente para cumplir con requisitos de performance, si las pruebas diseñadas cubren las funcionalidades de los sistemas y si la documentación generada es correcta.

Las actividades asociadas al aseguramiento de la calidad se realizaron en todas las etapas del proyecto. Estas actividades fueron revisiones, *checklists* y también revisiones cruzadas entre los integrantes del proyecto.

Los defectos detectados en revisiones e inspecciones fueron formalizados en un documento/herramienta para luego ser distribuidos entre los integrantes del equipo y también a los posibles interesados con el fin de mantenerlos informados.

8.1. Objetivos de calidad a nivel de producto

En cuanto al producto a construir, los objetivos de calidad que se definieron son los siguientes:

- El producto de software deberá de cumplir con el alcance definido para la entrega en un 100%. Todos los requerimientos funcionales y no funcionales incluidos para esta primera etapa deben estar implementados.
- En caso de contar con defectos sin resolver al momento del cierre del proyecto, éstos no pueden ser de severidad alta, es decir, que impidan el normal funcionamiento de la aplicación y de sus funcionalidades principales.

8.2. Objetivos de calidad a nivel del proceso de desarrollo

La calidad del proceso de desarrollo incide directamente en la calidad del producto resultante del mismo, por lo tanto se realizaron las siguientes actividades para poder certificar que el proceso cumple con los estándares de gestión y documentación establecidos.

- Contar con un plan de calidad adecuado a la realidad del proyecto.
- Realizar seguimiento de la planificación para asegurar el cumplimiento de las actividades allí definidas y corregir desviaciones en tiempo y forma.
- Seguir la metodología de trabajo definida para el proyecto.
- Establecer indicadores que permitan medir la calidad del proceso y revisarlos de manera periódica.

8.3. Planificación de SQA

Las actividades de aseguramiento de la calidad se realizan a lo largo de todo el ciclo de vida del proyecto. Algunas actividades son específicas de una etapa del proyecto y otras se realizan durante todo el proyecto.

Actividades que se realizan al inicio del proyecto:

- Definición de estándares de documentación.
- Definición de estándares de codificación.
- Definición de planes.

Actividades que se realizan previo a comenzar una iteración:

- Definición del alcance y objetivo de la iteración.
- Establecimiento de los criterios de aceptación de la iteración.

Actividades que se realizan durante una iteración:

- Seguimiento de la planificación (avance).
- Pruebas sobre el software construido.
- Revisiones cruzadas de código y documentación.
- Reuniones internas del equipo y con el cliente o responsable del producto.

Actividades que se realizaron al finalizar una iteración:

- Reunión de cierre de la iteración (revisión de métricas y resultados para realizar correcciones).
- Evaluación interna del desempeño el equipo.
- Validación del entregable con el cliente o responsable de producto.
- Lecciones aprendidas.

Actividades que se realizaron durante todo el proyecto

- Pruebas.
- Documentación.
- Aplicar los estándares definidos.
- Relevar métricas.
- Revisiones.

8.4. Actividades de SQA

Las actividades comprendidas dentro de SQA apuntan a la selección de estándares tanto para el software como para la documentación, verificar el cumplimiento de los mismos. Estas actividades abarcan también revisiones internas de los productos de *software* a medida que se construyen, así como de la documentación resultante del proceso y la selección y recolección de métricas durante el proyecto.

8.4.1. Selección de estándares

Los estándares a seguir durante el proyecto fueron divididos en tres grandes grupos, Documentación, Código y Usabilidad.

Documentación

Para la documentación del proyecto se tomaron como estándares y lineamientos a seguir las restricciones impuestas por la Universidad ORT para la entrega del proyecto final.

Documento/Ítem	Estándar a seguir	Comentarios
Documento del proyecto	Documento 302 de ORT	Aplica a la documentación general del proyecto y pautas a seguir para la entrega final
Diagramas del sistema	UML	Todos los diagramas que aplican para describir la solución (clases, paquetes, despliegue, componentes) siguen lo establecido por el

		lenguaje UML
Nombre de los archivos y documentos	Nombrar el archivo por el nombre que lo describe y explicitar la versión al final "DocumentoEjemplo_vX.X"	Las versiones X.0 se generan para entregar. Las intermedias serán X.X mientras se está trabando

Tabla 13.1 –Estándares de documentación

Código

Para lograr que el código sea uniforme y entendible se tuvieron en cuenta, además de las convenciones específicas de los lenguajes utilizados, las buenas prácticas incorporadas durante la carrera en la asignatura de Diseño de Aplicaciones y en el libro “Clean Code” de Robert C. Martin:

- Nombres con sentido: Usar nombres que revelen las intenciones, que se puedan pronunciar, que se puedan buscar, nombres de clases en singular y representativos, nombres de métodos que sean verbos.
- Funciones: tamaño reducido, hacer una sola cosa, un nivel de abstracción por función, nombres descriptivos, no recibir más de tres parámetros.
- Comentarios: Evitar comentarios en el código.

Usabilidad

En cuanto a la usabilidad del sistema, decidimos tomar como referencia al momento de diseñar las pantallas las “10 Reglas heurísticas de usabilidad” de Jakob Nielsen ya que es un referente de la industria en cuanto a usabilidad:

- Visibilidad del estado del sistema.
- Relación entre el sistema y el mundo real.
- Control y libertad del usuario.
- Consistencia y estándares.
- Prevención de errores.
- Reconocimiento antes que recuerdo.
- Flexibilidad y eficiencia de uso.
- Estética y diseño minimalista.
- Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores.
- Ayuda y documentación.

El detalle de cada una de las reglas se encuentra en el [Anexo 10 – Heurísticas de Nielsen](#)

8.4.2. Revisiones del proyecto

Se realizaron revisiones internas, validaciones y verificaciones que apuntan a mejorar la calidad del proyecto en general.

8.4.3. Revisiones internas

Durante el proyecto se realizaron revisiones internas de los productos a medida que se va avanzando.

Para cada producto se definió el responsable, criterio y momento en el que se realizará la revisión correspondiente

Producto	Criterio	Responsable	Momento de revisión
Plan de Proyecto	Revisión interna, estándar de documentación	Gerente de proyecto	Cuando se genera una nueva versión
Plan de Calidad	Revisión interna, estándar de documentación	Responsable de SQA	Cuando se genera una nueva versión
Documentación de los requerimientos	Revisión interna, estándar de documentación	Responsable de ingeniería de requerimientos	Cuando se genera una nueva versión
Producto de software	Requerimientos, estándares de codificación	Equipo de desarrollo	Al finalizar el desarrollo
Documentación para entrega	Revisión cruzada de las secciones	Equipo de proyecto	Cuando se agrega una nueva sección

Tabla 13.2 – Revisión de productos

8.4.4. Verificaciones

Las verificaciones realizadas durante el proyecto buscan asegurar que los productos que se crearon cumplen con los niveles de calidad establecidos.

Las verificaciones varían de acuerdo al producto que se encuentra bajo verificación. Se listan a continuación las verificaciones realizadas para los diferentes productos:

- **Especificación de requerimientos:** Se verificó que los requerimientos relevados fueran especificados de forma correcta. El detalle de este proceso de relevamiento y validación de los requerimientos del sistema se encuentra en el capítulo [Ingeniería de requerimientos](#).
- **Plan de testing:** Se verificó que el alcance de las pruebas cubra las funcionalidades que están dentro del alcance del desarrollo de esta instancia.
- **Producto de software:** Se verificó el cumplimiento con los requerimientos incluidos en el alcance. Esta verificación se hizo mediante la ejecución de los casos de prueba definidos.
- **Documentación de la arquitectura del sistema:** Se verificó, además de la correctitud del documento en general, que se satisfagan los atributos de calidad para cada uno de los requerimientos no funcionales especificados.

En la siguiente tabla se resume qué producto se verificó, quién era el responsable del producto y de la verificación y el momento en que se llevó a cabo la misma

Producto	Responsable producto	Responsable de la verificación	Momento de la verificación	Comentarios/Defectos
Especificación de requerimientos	Responsable de ing. De requerimientos	Responsable de SQA	Al generar una nueva versión	Los defectos se reportaron mediante comentarios en el documento
Plan de testing	Responsable de SQA	Responsable de SQA	Al generar una nueva versión	Los defectos se reportaron mediante comentarios en el documento
Producto de software	Equipo de desarrollo	Responsable de testing	Al final de cada iteración	Los defectos se reportaron en una planilla creada para tal propósito
Documentación de la arquitectura	Responsable de arquitectura	Responsable de SQA	Al generar una nueva versión	Los defectos se reportaron mediante comentarios en el documento

Tabla 13.3 – Verificación de productos

8.4.5. Validaciones

Para la validación de los productos generados durante el proyecto se coordinó con expertos en cada área dependiendo del producto a validar.

Como salida a estas validaciones se obtuvieron comentarios y sugerencias de mejora que fueron aplicados en las siguientes versiones de los productos.

En la siguiente tabla se listan las validaciones realizadas a los productos generados durante el proyecto

Producto	Responsable producto	Responsable de la validación	Método/Técnica de validación	Momento de la validación
Especificación de requerimientos	Responsable de ing. De requerimientos	Experto en marketing	Reunión presencial	Una vez definidos los requerimientos en el alcance
Producto de software	Equipo de desarrollo	Responsable de testing con usuarios potenciales	Reunión con los usuarios y demo de las funcionalidades	Al final de las pruebas de regresión
Documentación de la arquitectura	Responsable de arquitectura	Arquitecto de empresa en la que trabaja uno de los integrantes del equipo	Reunión presencial	Al generar la versión final del documento

Tabla 13.4 – Validación de productos

Se realizaron además dos encuestas para validar los requerimientos y la tecnología seleccionada para el proyecto. Dichas encuestas apuntaron a obtener información desde los dos usuarios principales del sistema, los compradores (usuario final que buscará los productos en la aplicación) y lo vendedores (dueños de los comercios registrados en el sistema).

- **Encuesta a compradores:** Las preguntas incluidas buscaban determinar si nuestro público objetivo tiende a realizar búsquedas de productos y compras a través de sus dispositivos móviles, si tienen preferencia sobre compras en línea o en tiendas físicas de determinados productos, si se habían enfrentado a la problemática que intentamos resolver y la preferencia a aplicativos nativos o utilizar aplicaciones web que sean multiplataforma.
- **Encuesta a vendedores:** Las preguntas incluidas buscaban determinar si los comercios que consideramos nuestro público objetivo expresaban la necesidad de atraer más clientes a sus locales, cómo les han resultado sus campañas de marketing, por qué medios realizan las publicidades y campañas y la valoración que le darían a la posibilidad de exponer ofertas en línea.

Los resultados de las encuestas nos brindaron información que apoya la iniciativa y el propósito y además colaboraron en la toma de decisiones en la selección de tecnologías para el desarrollo del proyecto.

El detalle de las preguntas y los resultados de las encuestas se encuentran en los anexos [Anexo 11 - Resultado de encuestas a compradores](#) y [Anexo 12 - Resultado de encuestas a vendedores](#).

8.4.6. Controles sobre los repositorios

Tal como se detalla en el capítulo de [Gestión de la Configuración](#) el versionado de documentación y de código fue uno de los puntos importantes a gestionar durante el desarrollo del proyecto.

Para asegurar el cumplimiento de lo establecido para este punto se realizaron las siguientes acciones en el transcurso del proyecto.

Repositorio de documentación

- Versionado de documentos: Cada documento creado deberá tener la versión correspondiente explicitada.
- Nomenclatura para el nombramiento de los documentos: Respetar el formato establecido para nombrar los archivos.
- Ubicación de los archivos en los repositorios correctos: Cada documento deberá estar guardado en la carpeta contenedora correcta.

Repositorio de código

- Respetar las prácticas establecidas en *GitFlow*: Ver referencia [1]

Al comienzo el equipo tuvo algún inconveniente con el flujo de trabajo establecido para el manejo de las versiones y controles sobre las mismas, pero conforme avanzó el proyecto esta metodología de trabajo se incorporó de manera casi natural a los integrantes, logrando así tener un control sobre las versiones de los productos en los que se trabajaba y se evitaron problemas relacionado a la pérdida de información por errores en las versiones o faltante de archivos físicos.

8.5. Testing

El producto de software se somete a una etapa de *testing* para asegurar el cumplimiento con los requerimientos definidos y poder determinar el nivel de calidad.

8.5.1. Planificación de las pruebas

La ejecución de las pruebas sobre las funcionalidades desarrolladas se realizó al final de cada *sprint* y luego del último *sprint* de desarrollo se realizaron además pruebas de regresión.

A lo largo del proyecto se realizaron diferentes tipos de pruebas, de modo de verificar el cumplimiento de los requerimientos tanto funcionales como no funcionales.

A continuación se listan los tipos de pruebas incluidas en el *testing* del producto:

Pruebas unitarias

Las pruebas unitarias las ejecutaron los desarrolladores durante la construcción de cada módulo, ya que se trabajó con TDD (*Test Driven Development*).

La ejecución pruebas unitarias verifican que lo que se está agregando no agregan errores al código ya existente.

Se tomó como regla interna del equipo el no subir código al repositorio para el cual no estén pasando OK todos los *tests* unitarios.

Pruebas de integración

Estas pruebas permiten verificar que los módulos funcionan correctamente cuando se integran.

Las pruebas de integración se realizaron conjuntamente con las pruebas funcionales del sistema y tuvieron vital importancia en la integración del *backend* con el *frontend*.

Pruebas de seguridad

Respecto a la seguridad, las pruebas realizadas fueron principalmente sobre el módulo de *login* y se realizaron mediante la ejecución de los casos de prueba funcionales.

Los tiempos acotados y la falta de capacitación del equipo en esta área no permitieron que se profundice más sobre el test de seguridad de la aplicación.

Pruebas funcionales

Se definieron casos de prueba funcionales para las funcionalidades incluidas en el *MVP* con el objetivo de asegurar, desde un punto de vista funcional, el cumplimiento de los requerimientos del sistema.

Se utilizó un *template* para la especificación de los casos de pruebas y además para el reporte de los defectos encontrados.

Se adjunta a continuación un ejemplo de la planilla de especificación de casos de prueba y un ejemplo de la planilla de reporte de defectos.

Casos de prueba

# Caso	Propósito del caso	Resultado esperado	Resultado Obtenido
Búsqueda por Producto			
1	Verificar que al realizar una búsqueda para la que existen resultados en el rango, se muestran los comercios coincidentes en el mapa	Se muestran en el mapa todos los comercios que cumplen con la condición de búsqueda	pendiente
2	Verificar que al realizar una búsqueda para la que NO existen resultados en el rango, se muestra un mensaje que indica que no hay resultados	Se muestra un mensaje que indica que no hay resultados para la búsqueda realizada	pendiente
3	Verificar que al realizar una búsqueda para la que NO existen resultados (no hay comercios registrados que ofrezcan el producto), se muestra un mensaje que indica que no hay resultados	Se muestra un mensaje que indica que no hay resultados para la búsqueda realizada	pendiente
4	Verificar que no se permiten realizar búsquedas por producto con el campo de producto vacío	El botón de buscar no se habilita hasta que no se ingresa un dato en el campo de producto	pendiente
5	Verificar que al realizar una búsqueda por Productos se muestran los comercios registrados en el sistema y los resultados de Google Maps	En los resultados de búsqueda se muestran los comercios registrados en el sistema y los sugeridos por Google Maps	pendiente

Tabla 13.5 – Ejemplo de caso de prueba

Reporte de defectos

# defecto	Reportado por	Funcionalidad	Descripción	Severidad	Estado	Comentarios
1	ATaruselli	Búsqueda por Producto	No se muestra mensaje informativo si no existen resultados para la búsqueda realizada	2 - Medio	Nuevo	Caso #2 Caso #3
2	ATaruselli	Búsqueda por Categoría	No se muestra mensaje informativo si no existen resultados para la búsqueda realizada	2 - Medio	Nuevo	Caso #7 Caso #8
3	ATaruselli	Ver Perfil Comercio	Si el campo no tiene información para mostrar no se carga con un mensaje descriptivo, debería decir "información no proporcionada por el comercio"	3 - Bajo	Nuevo	Caso #21

Tabla 13.6 – Reporte de defectos

8.5.2. Ejecución de las pruebas

El responsable de *SQA* tomó el rol de *tester* tanto para la definición de los casos de prueba como para la ejecución de los mismos.

La documentación asociada a la ejecución de los casos de prueba y el reporte de defectos encontrados se almacenó en una carpeta específica en el repositorio que era accedida por todos los miembros del equipo.

Se definió la siguiente escala para asignar la severidad de un defecto al reportarlo:

- **Alta:** Es un defecto que impide continuar con la ejecución de las pruebas.
- **Medio:** No cumple con los requerimientos pero no se generan fallas graves en el sistema.
- **Bajo:** Sugerencias de mejora, defectos de cosmética, mensajes poco claros, disposición de la información en la pantalla.

Por otra parte, se definieron los siguientes estados para los defectos reportados:

- **Nuevo:** Se reportó el defecto pero no ha sido tomado por un desarrollador para resolverlo.
- **Rechazado:** El defecto reportado no es tal.
- **En desarrollo:** Un desarrollador está trabajando para corregir el defecto.
- **Corregido:** El desarrollador corrige el defecto y está pendiente de que sea testeado.
- **Cerrado:** El responsable de *testing* verificó que el defecto fue corregido.

La ejecución de las pruebas funcionales se realizó al finalizar cada *sprint*.

8.6. Métricas de Testing

En esta sección se muestran las métricas obtenidas a partir del análisis de los resultados de la ejecución de los casos de prueba definidos para esta instancia.

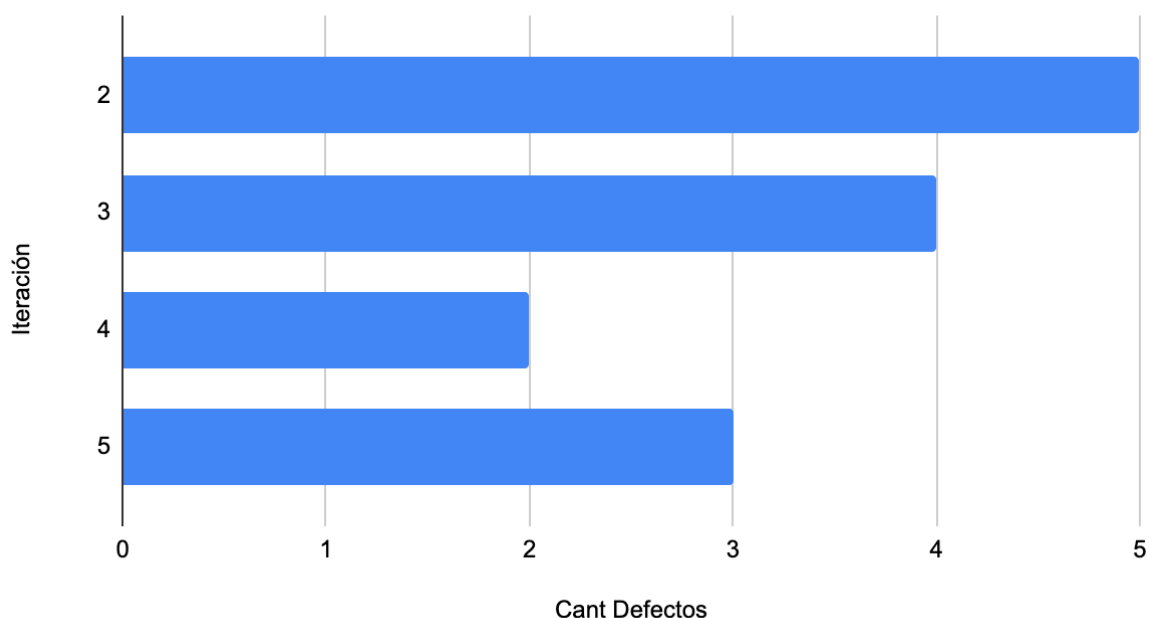
Se ejecutaron los 27 casos de prueba definidos que cubren las funcionalidades desarrolladas y se realizó una prueba de regresión luego de finalizar el último *sprint* de desarrollo.

1.1.1. Cantidad de defectos reportados por iteración

Al final de cada iteración de desarrollo se ejecutaron los casos de prueba asociados a las funcionalidades que se desarrollaron en esa iteración.

Comparando la cantidad de defectos por iteración podemos tener una visión de cómo fue evolucionando la calidad de la aplicación a lo largo del proyecto.

Cantidad de defectos por iteración



Gráfica 13.1 – Defectos por iteración

En las primeras dos iteraciones fue donde se encontraron la mayor cantidad de defectos y esto coincide con la implementación de las funcionalidades más complejas (Búsqueda por producto y Búsqueda por categoría).

El desglose de los defectos reportados por iteración, teniendo en cuenta la severidad de los mismos, es el siguiente:

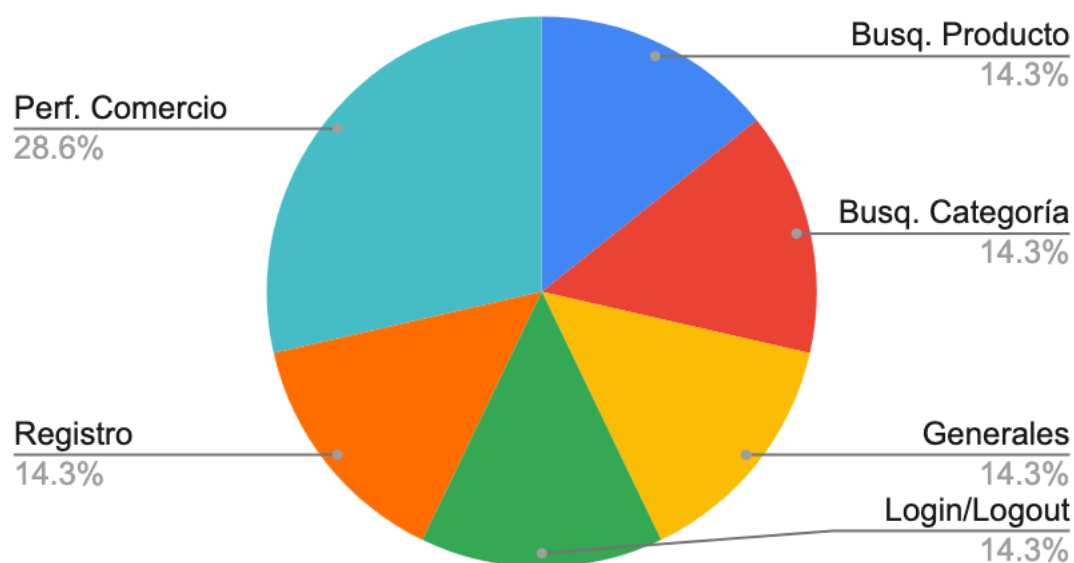
Iteración	Alto	Medio	Bajo
2	0	4	1
3	2	2	0
4	2	0	0
5	1	2	0

Tabla 13.7 – Defectos por iteración y severidad

1.1.2. Cantidad de defectos reportados por funcionalidad

Al analizar la cantidad de defectos reportados por funcionalidad se pueden tomar medidas correctivas a futuro respecto al foco que se tiene que poner en cada funcionalidad en nuevas etapas de *testing*. Una funcionalidad para la que se reportan muchos defectos requiere un foco de atención tanto desde el punto de vista de desarrollo como de *testing*.

Defectos por funcionalidad



Gráfica 13.2 – Defectos por funcionalidad

El desglose de los defectos reportados por cada funcionalidad, teniendo en cuenta la severidad de los mismos, es el siguiente:

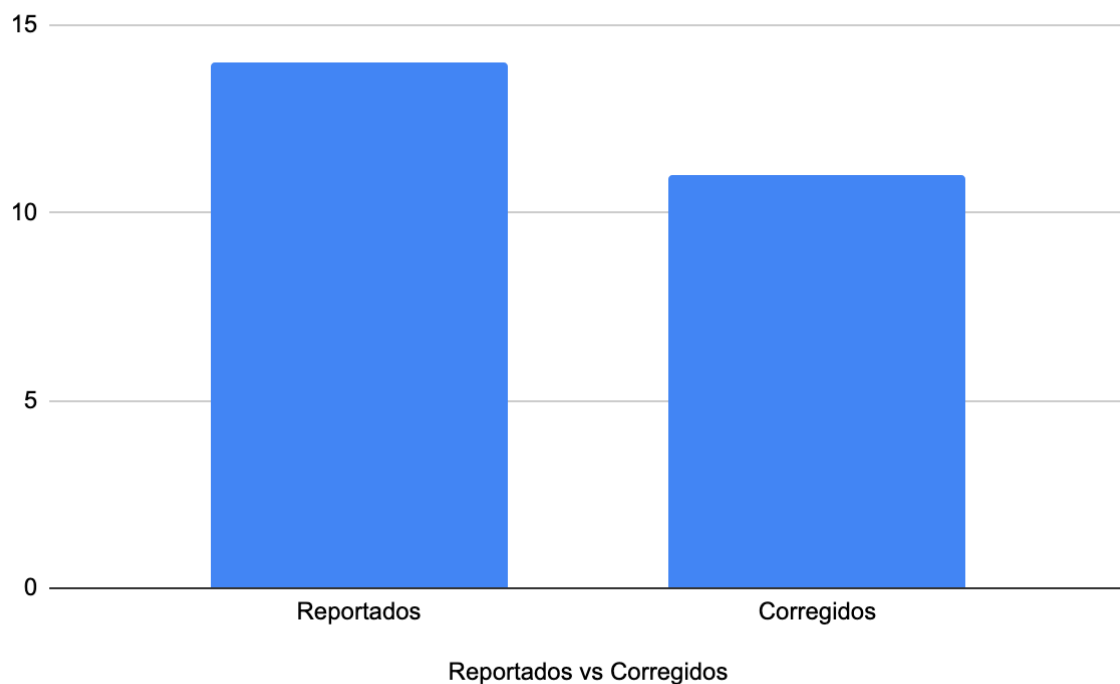
Funcionalidad	Alto	Medio	Bajo
Búsq. Producto	0	2	0
Búsq. Categoría	0	2	0
Generales	0	2	0
Login/Logout	2	0	0
Registro	2	0	0
Perf. Comercio	1	2	1

Tabla 13.8 – Defectos por funcionalidad y severidad

1.1.3. Defectos reportados vs defectos corregidos

Al cierre del proyecto se corrigieron todos los defectos de severidad alta que fueron reportados en etapa de *testing*, quedando dos defectos de severidad media por corregir previo a la defensa del proyecto.

Dichos defectos corresponden a mensajes de error que no se muestran de una manera clara al usuario.



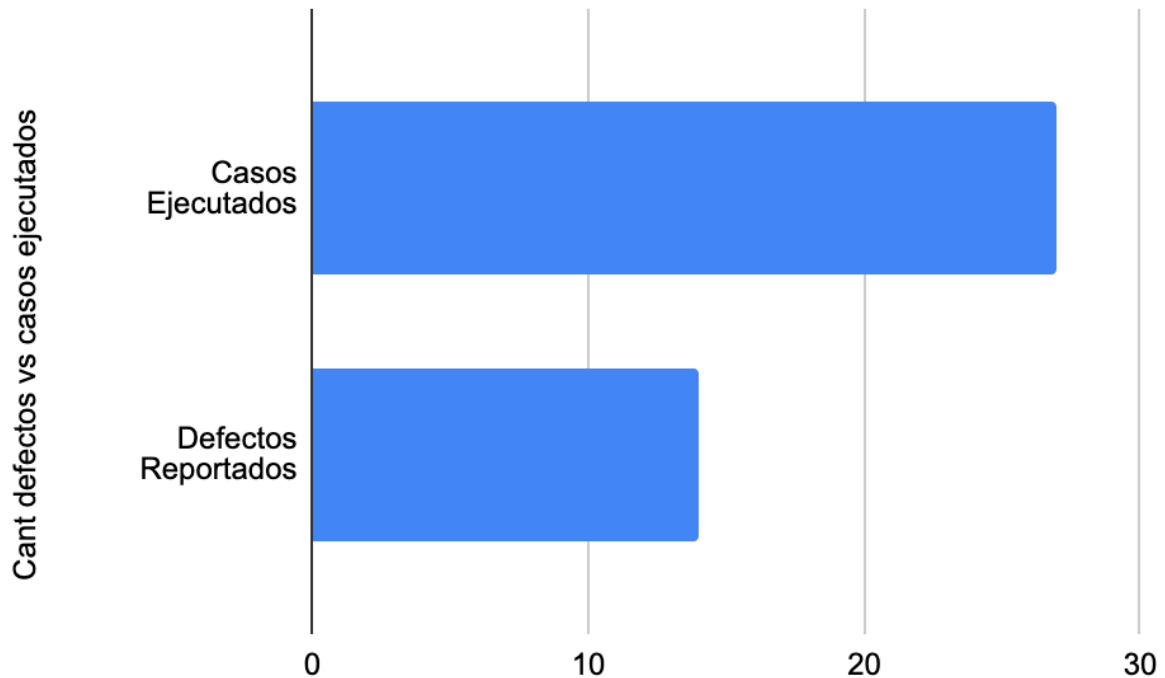
Gráfica 13.3 – Defectos reportados vs corregidos

La razón por la que no se corrigieron el 100% de los defectos es que algunos prioritarios se reportaron en la etapa de regresión (última etapa de *testing*) y los de severidad más baja se dejaron para futuro ya que no afectan de manera crítica al funcionamiento de la aplicación.

1.1.4. Defectos reportados vs Casos ejecutados

La comparación de la cantidad de defectos reportados contra la cantidad de casos de prueba ejecutados sirve para tener una idea de la efectividad del *testing*.

En este caso de 27 casos de prueba ejecutados se reportaron 14 defectos, por lo que entendemos que el *testing* fue efectivo ya que se reportó un defecto cada dos casos de prueba aproximadamente.



Gráfica 13.4 – Defectos reportados vs casos ejecutados

9. GESTIÓN DE LA CONFIGURACIÓN

La Gestión de la Configuración (SCM) identifica, organiza y controla las modificaciones del software y la documentación del proyecto a través del ciclo de vida.

Actividades de SCM que nos interesarán son:

- Identificar elementos de configuración del *software*.
- Definir herramientas de gestión a utilizar en el desarrollo.
- Elegir herramientas y tecnologías para el desarrollo del *software*.
- Control de versiones de documentos y *software*.

9.1. Elementos de configuración

En la construcción del sistema los elementos que componen toda la información producida como parte del proceso de ingeniería de *software* se dividen en 2 grandes ramas:

- Los elementos de *software* que componen todos los archivos que componen el *software* en todas sus capas.
- Los elementos de documentación que abarcan todos los archivos que conforman la documentación del proyecto.

Por restricciones de tiempo, no se realizó una identificación de estos elementos formal en tablas como suele realizarse. De todas maneras, las herramientas de gestión utilizadas que se explicarán a continuación permiten tener un registro de cada elemento y sus respectivas versiones.

9.2. Herramientas de gestión

Para poder realizar un correcto seguimiento del proyecto hemos elegido las siguientes herramientas:

- **GITHUB:** es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones GIT. Se utiliza principalmente para la creación de código fuente de programas de computadora. El *software* que opera *GitHub* fue escrito en *Ruby on Rails*.
- **Google Drive:** Repositorio de información que permite almacenar y editar documentos de diversos formatos. Para acceder solo se necesita un navegador web y conexión a internet. Se asume que no se necesita capacitación para usar las herramientas, debido al perfil de los integrantes del equipo.

9.3. Gestión de versiones

9.3.1. Versionado de código de fuente

Para el versionado del código se aplicó GitFlow [1], la metodología de trabajo con GITHUB consistió en crear una rama para el desarrollo por parte de cada integrante del equipo. De esta manera se logró desacoplar los cambios y que los mismos no afecten en el desarrollo de los demás integrantes. Una vez que cada desarrollo era testeado se pasaban los cambios a la rama central denominada Development donde se integraban las funcionalidades finalizadas.

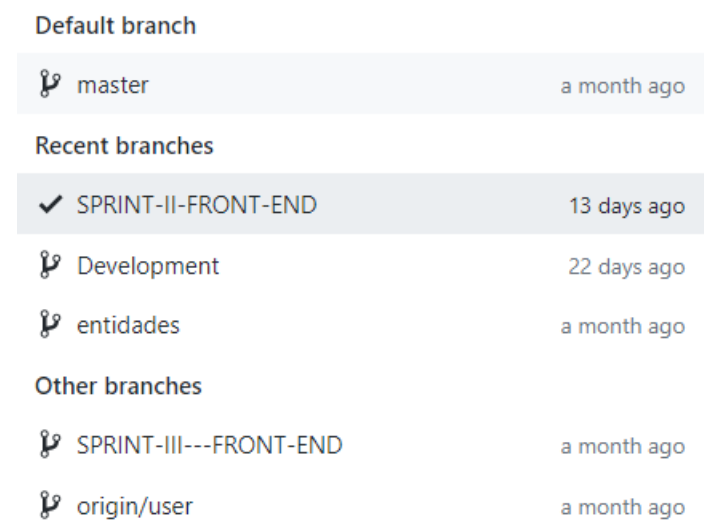


Imagen 14.1 – Repositorio de código Github

En la raíz del proyecto se crearon 2 subcarpetas cada una conteniendo la solución para el *backend* y otra para el *frontend*. Cada una de estas carpetas conteniendo su propio archivo *Gitignore*. En este archivo se especifican aquellos archivos que no deberán ser compartidos entre las distintas instalaciones locales ya que contienen metadatos que pueden generar conflictos en distintas máquinas.



Imagen 14.2 – Repositorio de código carpetas

Para el intercambio de información con el repositorio se utilizó *GitHub Desktop*. El mismo permite fácilmente trabajar con las funcionalidades de *GitHub* a nivel gráfico sin necesidad de utilizar líneas de comando.

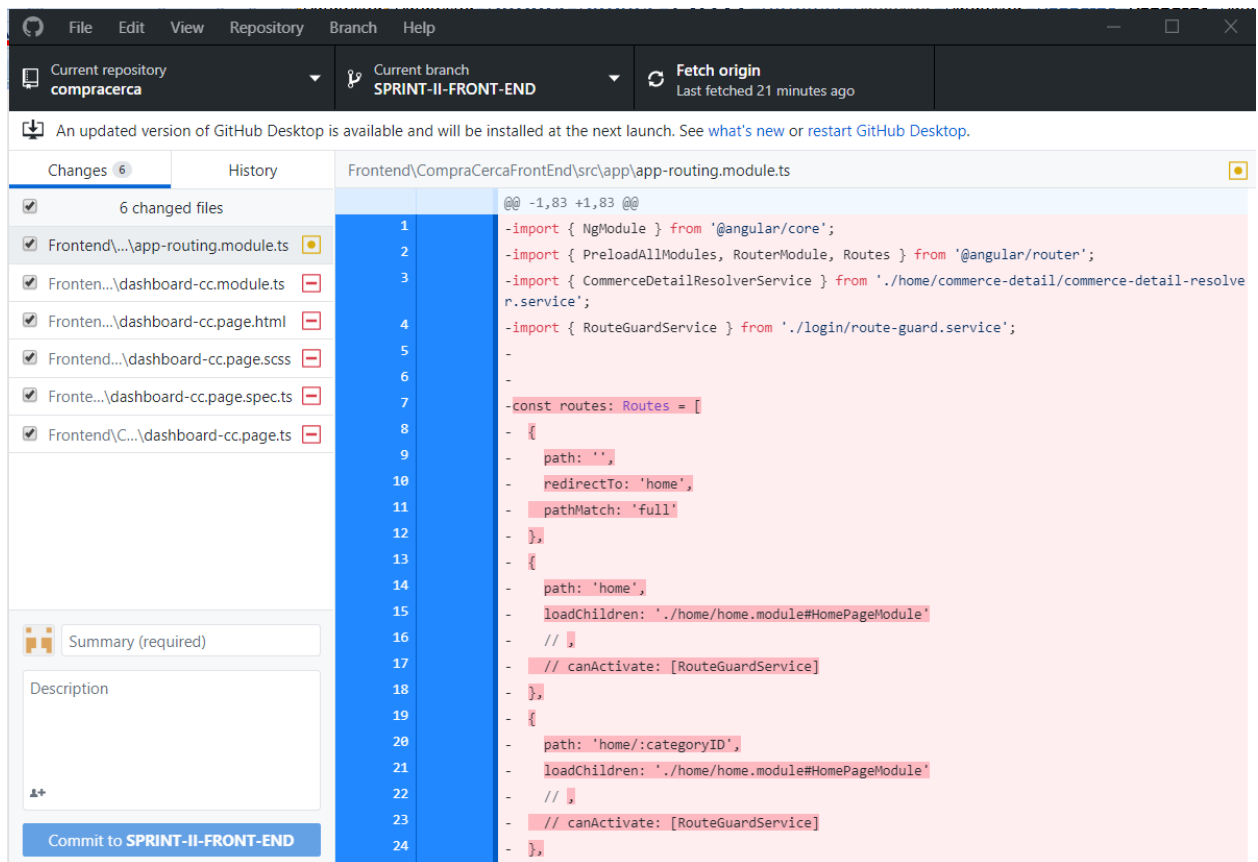


Imagen 14.3 – *Github Desktop*

Se tomó como regla que cada vez que se realiza un *commit* del código fuente u otro archivo, se deben realizar los comentarios descriptivos acerca de que consisten los cambios. A su vez, se trató de realizar *commits* diarios para evitar acumulación de información lo cual genera que la descripción sea muy genérica. A su vez, los *commits* diarios ayudan a tener un registro del desarrollo a lo largo de los *sprints*.

9.3.2. Versionado de la documentación

Para el versionado de la documentación se eligió *Google Drive* y su manejo de archivos y carpetas en la nube. Esta herramienta permite por un lado la colaboración y la modificación de los documentos en tiempo real por más de un usuario a la vez. A su vez, permite el versionado al agregar archivos con idéntico nombre y formato. Esto facilitó la escritura de documentos ya que si no debería registrarse en el propio documento las versiones. Tiene compatibilidad con todos los tipos de archivos utilizados en la documentación del sistema y permite acceder fácilmente a cada documento simplemente con acceso a internet. Esto es importante ya que facilitó las reuniones con la tutora y flexibilidad a la hora del trabajo, ya que no siempre en las reuniones de equipo los integrantes contaban con su ordenador personal.

La estructura de carpetas en la cual se segmentó la documentación es la siguiente:

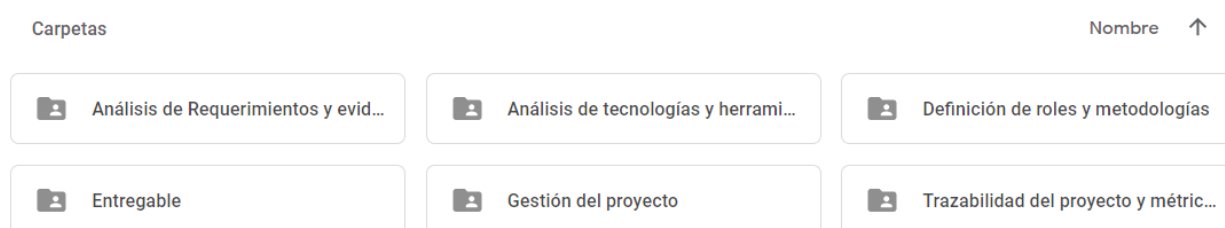


Imagen 14.3 – Repositorio de documentos

- **Análisis de requerimientos y evidencias:** Almacena los documentos de ingeniería de requerimientos y herramientas de relevamiento y validación utilizadas.
- **Análisis de tecnologías y herramientas:** Documentos enfocados a la selección de tecnologías y herramientas, su justificación y comparación de pros y contras de cada una para la toma de decisiones.
- **Definición de roles y metodologías:** Documentos enfocados a describir la forma de trabajo elegida y su justificación. Además se definen los distintos roles que serán necesarios para el correcto desarrollo del proyecto.
- **Entregable:** Carpeta donde se gestionan todos los documentos que se deberán entregar durante el transcurso del proyecto.
- **Gestión del proyecto:** Aloja los documentos que conforman el plan de proyecto.
- **Trazabilidad de proyecto y métricas:** Contiene los documentos referentes a las planificaciones de las tareas a realizar y el registro de horas del equipo.

9.4. Respaldos físicos

Para una mayor seguridad se realizaron respaldos periódicos del código y la documentación en discos físicos. La razón es un posible acceso indebido en una cuenta de mail, que puede ocurrir ya que a veces las sesiones quedan activas al utilizarlas en ámbitos como facultad o trabajo.

En el caso del código se respaldó por acceso indebido al repositorio o algún error humano en el manejo de los archivos.

10. LECCIONES APRENDIDAS

En el transcurso del proyecto nos hemos enfrentado a problemas o situaciones que antes no nos habíamos enfrentado en el transcurso de la carrera o en el trabajo donde algunas tareas se delegan a otras personas. Estas situaciones nos llevaron a tomar decisiones que, en algunos casos, pasado el tiempo creemos podrían haberse resuelto de una mejor manera. En otras, fue acertada la determinación pero en definitiva el objetivo de este capítulo es resumir estas lecciones para bien o mal que afectaron el transcurso del proyecto.

INGENIERÍA DE REQUERIMIENTOS

Creemos nos llevó mucho tiempo definir los requerimientos del sistema. De esto aprendimos que hubiese sido mejor conversar con más expertos en el área, que nos hubiera permitido saltar varios escalones en el proceso de toma de decisiones que tuvimos que atravesar. También hubiese acertado los tiempos obtener asesoramiento en cuanto a las tecnologías a utilizar, ya que como se señaló previamente en este documento, uno de los *frameworks* utilizados para el desarrollo del *frontend* surgió de una sugerencia que recibió en el trabajo uno de los integrantes del proyecto. Por lo general, optamos por resolver e investigar por nuestra cuenta, pero entendemos hoy que en algunos casos a veces es importante la experiencia de otros, como base para empezar a investigar. Más en proyectos como éste donde el tiempo es acotado.

ESTIMACIONES EN STORY POINTS

Dada la poca experiencia en el trabajo con metodologías ágiles de parte de dos integrantes del proyecto, en algunos casos la estimación en *Story Points* pudimos comprobar luego que fue insuficiente. La utilización de tecnologías nuevas también exacerbó dicha insuficiencia de tiempo para completar tareas. Esto provocó atrasos en la construcción del sistema, que complicaron el cronograma de trabajo e hicieron tener que postergar o descartar algunos ítems del mismo.

PREVISIÓN DE TIEMPOS DE INTEGRACIÓN DEL TRABAJO

En la estimación de los tiempos, no se dio la suficiente relevancia a integración de los desarrollos en *backend* y *frontend*. Esto llevó a atrasos en el cronograma y tener que dedicar menos tiempos a otras tareas que considerábamos importantes. A futuro entendemos fundamental planear tareas de integración para cada funcionalidad que requiera de desarrollo en más de una capa de la aplicación.

MEJORAR PRUEBAS DE SEGURIDAD

Planificar y realizar pruebas de seguridad formales ya que por falta de conocimiento y la utilización de información sensible que maneja el sistema entendemos deberían ser imprescindibles para una puesta en producción.

PRUEBAS DE PERFORMANCE

Realizar pruebas exhaustivas de performance para asegurar el correcto funcionamiento del sistema en casos de aumentar la masa crítica de comercios y compradores.

REUNIONES PRESENCIALES

Fue necesario aumentar el número de reuniones presenciales entre los miembros del equipo de desarrollo. Las integraciones de los desarrollos individuales de cada integrante fueron más eficientes con todos los implicados en el desarrollo presentes. Al comienzo del proyecto, las integraciones se realizaron por un miembro asignado, lo que demandó más tiempo del esperado.

PRIORIZAR NECESIDADES

Antes de comenzar el desarrollo se realizó un diseño de la interfaz que buscó asemejarse lo mayor posible a la realidad de un sistema funcionando, esto llevó mucho esfuerzo. Luego a la hora de ponernos a desarrollar y tener que implementar toda la interfaz real del programa, entendemos podríamos haber dedicado menos tiempo a los prototipos ya que al no tener que vender el producto, lo importante hubiera sido que se tenga la información necesaria para desarrollar la interfaz sin demasiado detalle.

Lo mismo ocurrió a nivel de la documentación ya que se empezó a utilizar *Latex*, que entendemos que es la herramienta que podría entregarnos el mejor resultado. Sin embargo, el tiempo de aprendizaje necesario para utilizarlo nos hizo descartarlo. Se perdió más de un día de trabajo que podría haber sido dedicado a la lista de pendientes que para ese entonces teníamos.

ARQUITECTURA

La arquitectura se redujo por un tema de alcance. Fuimos muy ambiciosos, apuntando a un concepto mucho más maduro para lo que necesitábamos y para el tiempo desarrollo acotado. Hoy podemos afirmar que necesitábamos ser más efectivo que pretender buscar la perfección.

RIESGOS

No se realizó un correcto seguimiento de los riesgos durante los *sprints*, lo que impidió poder obtener suficiente información estadística que contribuya a realizar correcciones y medir la evolución de los mismos a lo largo del proyecto.

11. CONCLUSIONES

Concluida la primera etapa del proyecto que se cierra con la entrega en la universidad, podemos afirmar que se cumplieron con los objetivos planteados al comienzo de este documento.

Como resultado del proyecto se obtuvo una solución de software conformada por un conjunto de funcionalidades definidas como imprescindibles para la primera versión del producto. Estas funcionalidades se dividen en dos módulos, uno de compradores y otro de administración.

El módulo de compradores permite a los usuarios definir un radio de búsqueda y encontrar los comercios que venden un producto/servicio dentro de dicho radio. A su vez, permite consultar toda la información disponible de los comercios. Las búsquedas podrán realizarse por nombre del producto ingresado o por el árbol de categorías que el sistema ofrece.

Por su parte, el módulo de administración tiene un panel de control con información clave del negocio como contador de usuarios y comercios registrados así como la cantidad de búsquedas diarias. Permite ver la información de todos los usuarios y comercios del sistema.

Dentro de los puntos que consideramos tienen más importancia dentro de un proyecto de esta naturaleza son el detectar cuáles son las necesidades reales de los posibles usuarios que utilizarán día a día nuestro sistema. En nuestro caso, al no interactuar con un cliente que nos diga, lo que espera y qué es lo que no espera de él, se buscó justificar lo más tangible posible los beneficios económicos, laborales, y de cualquier otra índole que se piensan alcanzar con el sistema completamente terminado.

La restricción de tiempo fue uno de los principales desafíos a resolver por fuera del problema de negocio propiamente dicho. En base a esto, las metodologías de trabajo de *Scrum* se adaptaron muy bien a nuestras necesidades. Los roles fueron bien elegidos y se logró llegar al alcance de producto definido. La elección de tecnologías como Angular e Ionic resultó acertada y contribuyeron también a alcanzar las metas en tiempo y forma.

Estamos convencidos que la experiencia fue muy enriquecedora a nivel académico y personal sobre todas las cosas. Por un lado, tuvimos que aprender y estudiar muchas tecnologías y aplicar dichos conceptos adquiridos. Se nos presentaron obstáculos y dificultades que nos obligaron a tomar decisiones en conjunto, donde no siempre existió el consenso total y por tanto la comunicación fue vital para seguir adelante.

12. REFERENCIAS

- [1] MIEM, 'Encuesta Nacional de mipymes', 2017. [Online]. Available: https://www.miem.gub.uy/sites/default/files/encuesta_nacional_de_mipymes_industriales_y_de_servicios_2017_0.pdf [Mar 15, 2019]
- [2] Carolina Samsing, 'Por qué el marketing tradicional está obsoleto', Febrero 2018. [Online]. Available: <https://blog.hubspot.es/marketing/porque-el-marketing-tradicional-esta-obsoleto> [Mar 16, 2019]
- [3] Maunamedia, '65 datos y estadísticas de redes sociales para considerar en 2019', Febrero 2019. [Online]. Available: <https://www.maunamedia.com/estadisticas-de-redes-sociales/#.XXbHwuKJJPY> [Mar 16, 2019]
- [4] ElEmpresario.mx, 'Las redes sociales que prefieren las pymes para dar a conocer su negocio', Diciembre 2018. [Online]. Available: <https://mba.americaeconomia.com/articulos/notas/las-redes-sociales-que-prefieren-las-pymes-para-dar-conocer-su-negocio> [Mar 19, 2019]
- [5] Rodrigohm.com, 'Estadísticas del Uso de Hashtags en Instagram', 2018. [Online]. Available: <https://rodrigohm.com/estadisticas-del-uso-de-hashtags-en-instagram/> [Mar 20, 2019]
- [6] Spatially, 'How To Target Customers On The Go', Febrero 2018. [Online]. Available: <https://www.spatially.com/blog/geofencing-demystified> [Mar 22, 2019]
- [7] Sandy Skrovan, 'Why most shoppers still choose brick-and-mortar stores over e-commerce', Febrero 2017. [Online]. Available: <https://www.retaildive.com/news/why-most-shoppers-still-choose-brick-and-mortar-stores-over-e-commerce/436068/> [Mar 23, 2019]
- [8] Cydney Hatch, 'Ecommerce Statistics You Should Know', Marzo 2019. [Online]. Available: <https://www.disruptiveadvertising.com/ppc/ecommerce/2018-ecommerce-statistics/> [Mar 25, 2019]
- [9] Guest Post, ' <https://ducttapemarketing.com/growing-business-with-geofencing/>', Marzo 2019. [Online]. Available: <https://ducttapemarketing.com/growing-business-with-geofencing/> [Mar 28, 2019]
- [10] Snigdha, 'What is Geofencing and how it can help you grow your small business?', Marzo 2019. [Online]. Available: <https://www.appypie.com/top-geofencing-tools-best-suited-for-local-marketing> [Mar 28, 2019]
- [11] R. C. Figueroa, 'Revista Escuela de Administración de Negocios', Núm 52, Pags. 118-135. Setiembre – Diciembre 2014
- [12] Ionic Inc., 'What is Hybrid App Development?', 2019. [Online]. Available:

<https://ionicframework.com/resources/articles/what-is-hybrid-app-development>
[Jun 23, 2019]

[13] Brainfodder, 'The science behind an impulsive purchase', Dicembre 2017. [Online]. Available: <https://brainfodder.org/science-of-impulse-purchases/> [Mar 25, 2019]

[14] Rebecca Lee White, 'How the marketing Funnel works from top to bottom', Marzo 2016. [Online]. Available: <https://trackmaven.com/blog/marketing-funnel-2/> [Mar 17, 2019]

13. ANEXOS

13.1. Anexo 1 – Análisis de la Situación Actual

El Instituto Nacional de Estadística (INE) relevó en su último estudio [1] que las PYMES representan el 94% de las empresas del Uruguay. En síntesis, el número total de empresas con actividad en el país es de más de 170.000 empresas a la fecha.

Las pequeñas y medianas empresas del Uruguay tienen problemas para llegar al público y lograr así aumentar sus ventas y popularidad. Esta dificultad, es tratada hoy en día con alternativas que como se analizarán más adelante presentan inconvenientes que no permiten que sus resultados sean completamente satisfactorios. La falta de capital, planeamiento y visión llevan a tomar decisiones que afectan el funcionamiento global de las empresas.

Una investigación del CEI (Centro extensionismo industrial) detectó que las pymes en el país tienen dificultades para expresar sus demandas porque no tienen claras cuáles son sus necesidades ni el orden de las mismas. A su vez, se constató que el 80% de las pequeñas empresas, carecen en líneas generales de empleados con formación universitaria lo que hace difícil para ellos detectar las posibilidades que brinda la tecnología y la ciencia ya que se necesita cierto conocimiento científico.

De todas maneras, aproximadamente el 50% manifestó el interés por la innovación tecnológica para lograr crecimiento y mejora de la performance.

En esta línea, el estado fomenta dichas actividades de innovación, Rafael Mendive, titular de MIEM expresó que el trabajo del ministerio es generar condiciones junto con todo el resto del sistema de transformación productiva y de competitividad para que cada vez sea más posible en Uruguay la innovación, la incorporación tecnológica y la unión entre la academia la empresa y el Estado

Específicamente en materia de marketing las PYMES del Uruguay tienen como opciones contratar a una empresa de marketing/publicidad y las nuevas alternativas como son vender sus productos en plataformas de *e-commerce* (como puede ser Mercadolibre) o bien hacer uso de las redes sociales e intentar llegar así a dar su marca/productos a conocer mediante publicaciones en plataformas como Facebook, Instagram o Twitter.

13.2. Anexo 2 - Acceso a créditos por parte de las PYMES en Uruguay

Datos proporcionados por la última encuesta nacional de PYMES [1] efectuada por MIEM en 2017 en nuestro país indicó que las pymes presentan condiciones de acceso al crédito restringidas. Solo el 8% de las empresas cuentan con acceso pleno al crédito debido a que pudieron acceder a un préstamo y no presentaron ningún indicador de dificultad. El 19% accedió pero con dificultad, el 20% -unas 30 mil- se encuentran excluidas o autoexcluidas del crédito y el 53% de las pymes no son usuarias.

La siguiente tabla muestra la forma de acceso a crédito de las PYMES en Uruguay.

	Accede al crédito	No accede al crédito	Total
No presenta dificultad	8% Acceso pleno	53% No usuaria	61%
Presenta dificultad	19% Acceso con dificultades	20% Excluida o autoexcluida	39%
Total	27%	73%	100%

13.3. Anexo 3 - Uso potencial del crédito de las PYMES en marketing/publicidad

	2017
Compra de materia prima/capital de trabajo	26
Reforma de local/mejora infraestructura	13
Compra de vehículo	12
Compra de maquinaria, equipos y licencias	11
Endeudamientos varios	8
Capital de giro	7
Compra de local/depósito	6
Otros	16
Total	100

Estos indicadores revelan la escasez de recursos orientados al marketing. Sin embargo, la encuesta revela que un 40% de las pymes han realizado alguna innovación con el objetivo de mejorar el desempeño de la empresa en los últimos dos años.

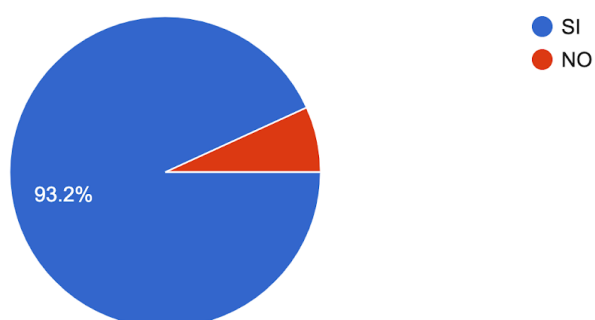
13.4. Anexo 4 - Resultado encuesta de compras online

Para comprender mejor la situación actual y además poder determinar con mayor exactitud si la solución pensada se ajusta a las necesidades se realizó una encuesta online que arrojó los siguientes resultados:

Link a la encuesta: <https://forms.gle/9v1MKd9WhSZJmLJq5>

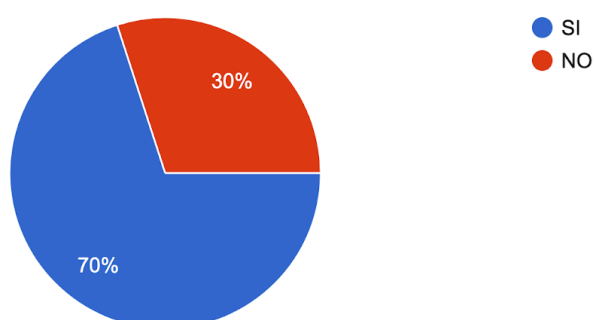
¿Has entrado a Mercadolibre con la intención de comprar un producto y decidiste luego no hacerlo?

44 responses



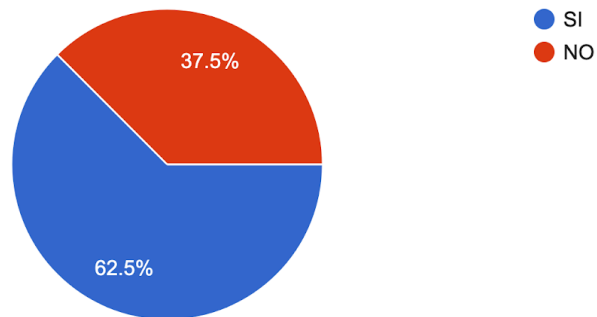
¿Has decidido no comprar un producto online porque te quedaba lejos el retiro?

40 responses



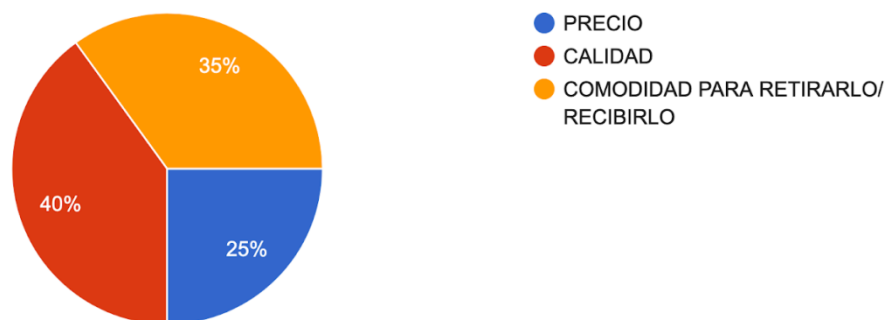
¿Te ha pasado de darte cuenta de que podrías haber comprado un artículo cuando estabas demasiado lejos del local donde lo venden?

40 responses



Qué valoras más a la hora de comprar un producto?

40 responses



13.5. Anexo 5 – Canvas

Socios Clave Empresa/agencia de marketing Experto en diseño y UI	Actividades Clave Desarrollo y mantenimiento de la aplicación Registro y mantenimiento de los comercios y sus datos Difusión de la aplicación Lograr cantidad masiva de usuarios/clientes Interacción con los clientes y usuarios	Propuesta de Valor Desarrollar una aplicación móvil que permita acercar a los clientes a los comercios mediante notificaciones que se enviaran a los usuarios cuando se encuentren cerca del negocio que vende el producto que están buscando.	Relación con Clientes Contacto vía mail y redes sociales Comentarios en Google Play Store Contacto presencial con los comercios	Segmento de Clientes Pequeños y medianos comercios que necesiten aumentar flujo de clientes y ganar visibilidad en el mercado Usuarios de aplicaciones móviles que desean comprar productos usando tecnología
	Recursos Clave Red de clientes y usuarios Equipo de desarrollo y mantenimiento del sistema Plataforma en la nube Equipo de soporte		Canales de Distribución/Comunicación Google Play Store Aplicación web en los equipos de los administradores Redes sociales Medios de socio de marketing	
Costos Almacenamiento en la nube Horas de desarrollo, mantenimiento y soporte Visitas a clientes			Ingresos Fee fijos y variables a empresas (clientes) de acuerdo al servicio prestado Venta de información generada por el uso de la aplicación Venta de publicidad en la aplicación	

13.6. Anexo 6 – Investigación de Mercado

Consideramos importante realizar un estudio del mercado y las alternativas que este le provee tanto a clientes como a comercios. Esta investigación fue el punto de partida para plantear una solución a las problemáticas encontradas en ambos bandos, tanto de comercios como de consumidores finales. A su vez, fue importante para segmentar los comercios a los que apuntaría el proyecto, así como el público objetivo que se vería beneficiado por el uso de la aplicación.

13.6.1. Problemas detectados de las alternativas de marketing

Los costos de hacer una campaña de marketing para un pequeño o mediano comercio suelen resultar elevados para sus posibilidades, tal cual se refleja la encuesta de MIEM [1] donde se muestran datos acerca del acceso al crédito para el año 2017 y las actividades en las que se invierte dicho crédito. La vieja escuela del marketing, analizada por el blog español Hubspot [2] donde se describe los beneficios de las nuevas tecnologías adaptadas a las promociones para el consumidor final. Por otro lado, detalla riesgos de las metodologías tradicionales donde muchas veces hasta que la campaña está completada en un 90% no se detecta que el mensaje y la ejecución sencillamente no tenían sentido para el consumidor, por lo que existe la posibilidad de desperdiciar grandes cantidades significativas de dinero, tiempo y energía en una campaña y producto cuyo resultado final no tenía un impacto en la conciencia del consumidor.

Facebook e Instagram se han convertido en algunas de las redes sociales favoritas de las personas en todo el mundo, pero especialmente para las pequeñas y medianas empresas (pymes), ya que es a través de ellas que han podido dar a conocer su negocio y expandirse a

otras regiones. Esto es por el gran alcance que tiene entre las audiencias y la relativamente baja inversión que supone.

Estudios de Latinoamérica muestran que la presencia en Facebook es casi total y aproximadamente la mitad tiene perfil en Instagram [3]. Podemos tomar como referencia el estudio realizado en México por el sitio web mba.americaeconomia.com [4] acerca de la estrecha relación entre las PYMES y las redes. Un relevamiento preliminar realizado por el equipo de proyecto reveló que empresas de insumos tecnológicos de nuestro país se alinean con las estadísticas para Facebook e Instagram.

Las modalidades de llegar al público en redes sociales se dividen en publicaciones pagas y gratuitas.

Por un lado, las publicaciones gratuitas en las redes sociales no resultan muy efectivas para hacer conocida la empresa si no se tiene una masa crítica considerable, ya que es baja la probabilidad de que usuarios de la red social lleguen a dicha publicación si no siguen previamente a la página. La utilización de *hashtags* es una opción en estos casos, sin resultados significativos a nivel de publicaciones [5].

Por otra parte, para las publicaciones pagas las estadísticas actuales sobre este tipo de publicidad no son alentadoras. Ha habido un aumento del 61,5% en el gasto en redes sociales de publicidad, lo que ha provocado que los usuarios sean bombardeados con una gran variedad de mensajes promocionales de los vendedores. Es por esta razón que el porcentaje de *clicks* sobre estos anuncios (CTR) de hoy solo se mantiene en un mero 0.5% en todos los formatos y ubicaciones de anuncios. Estos datos fueron proporcionados por el blog de tecnología “spatially” en su publicación llamada “geofencing-desmitificado” [6].

En ambos casos es necesario contenido creativo y estratégico que permita transformar en forma continua un suscriptor en un potencial cliente.

Conseguir tráfico masivo a estas páginas se sabe que no es un hecho fortuito sino que requiere de un gran trabajo, tiempo dedicado e inversión.

Como tercera y no menos frecuentada alternativa se encuentran las plataformas de e-commerce, fundamentalmente MercadoLibre en nuestro país, utilizado como mecanismo de venta e interacción con clientes. Aquí es difícil lograr diferenciación ya que compiten con empresas más grandes que tienen una mayor cantidad de ventas realizadas para los mismos artículos, teniendo ubicaciones preferenciales en las búsquedas.

Es en estas modalidades donde radica otro de los grandes problemas que detectamos y que buscamos disminuir, como lo es la alta brecha entre intención de compra y compra concretada de productos en plataformas de e-commerce. Esta brecha se incrementa aún más cuando los productos no son un “commodity”. Está comprobado que cuanto más se incrementa la diferenciación o especialización entre los productos similares, la brecha también experimenta dicho aumento. Esto aumenta la chance de que el cliente busque una tienda física y no el comercio electrónico para efectuar su compra.

En relación a esto, resulta interesante mencionar estudios acerca de la psicología de los compradores como el publicado por Retaildive [7] en donde se analiza la prevalencia aún del comercio en tienda física. Aquí se señala que la capacidad de ver, tocar y sentir los productos, sumado a el placer de llevar los artículos a casa de inmediato puede hacer elegir a una persona preferir comprar en una tienda en vez de comprar el mismo producto online.

Sumado a esto, un estudio publicado a fines de 2018 por la empresa DisruptiveAdvertising [8], en los Estados Unidos constató que:

El 56% de los compradores dijo que la aparición de costos inesperados es la razón por la que se van sin completar su compra.

Los compradores en línea abandonan sus carritos de compras el 68% del tiempo.

Con una tasa promedio de abandono del 68%, los sitios de comercio electrónico podrían estar perdiendo \$ 3 mil millones al año.

Un estudio de PayPal reveló que el 43% de los compradores abandonaron los carritos de compras porque los gastos de envío eran demasiado altos.

De 2015 a 2017, la tasa promedio de abandono de carros a nivel mundial aumentó de 71% a 77 %.

Se estima que el 23% de los usuarios abandonan su carrito de compras si tienen que crear una nueva cuenta de usuario.

Se calcula que el 54% de los compradores accede a adquirir productos que queden en los carritos de compra si esos productos se ofrecen posteriormente a un precio más bajo.

Una política de devoluciones inconveniente disuade al 80% de los compradores.

13.6.2. Problemas relacionados a los consumidores finales

Los consumidores deben invertir tiempo buscando activamente los productos que desean. Sin embargo, la vorágine del día a día hace que las personas tengan poco tiempo y eviten dedicárselo a una compra, salvo que el producto sea una necesidad básica.

La alternativa online como se señaló anteriormente muchas veces no es adecuada para el tipo de producto deseado.

Otro problema, es la falta de conocimiento de puntos de venta que serían mucha más convenientes para su ubicación, evitando traslados que muchas veces hace que se descarte una compra.

Los consumidores en su mayoría terminan no comprando el producto que están buscando en las plataformas de e-commerce (en este caso Mercadolibre). Esto en general se debe a que el punto de retiro del mismo queda lejos de su ubicación o no le es cómodo ir a retirarlo. También los usuarios manifiestan haber estado en la situación de enterarse demasiado tarde que pasaron por un punto de venta de un producto que estaban buscando y no supieron de que el mismo se encontraba allí y valoran en segundo lugar al momento de decidirse por un producto la facilidad para hacerse con él, es decir, la comodidad para retirarlo o recibirlo. Esta información fue obtenida de una encuesta online realizada a 40 personas de entre 20 y 50 años. (Ver anexo “Resultado encuesta de compras online”)

13.6.3. Oportunidades

De acuerdo al análisis de la situación actual de las PYMES y los problemas detectados en sus alternativas disponibles para aumentar su público y las necesidades de los clientes se detectan las siguientes oportunidades a atacar con nuestra propuesta:

- Lograr disminuir la brecha entre intención de compra y una concretada de parte de los clientes.

- Lograr una alternativa abordable en lo económico para las empresas.
- Conseguir una alternativa con resultados tangibles a corto plazo para conseguir la credibilidad de la tecnología.
- Conseguir apoyos del estado para expandir el mercado a medida que la credibilidad crezca y se tengan casos de éxito.
- Optimizar el tiempo de los clientes.
- Traer una tecnología en expansión y de poca utilización en nuestro país.

13.7. Anexo 7 – Análisis tecnológico *Geofencing*

13.7.1. ¿Qué es *Geofencing*?

Es una tecnología que establece límites virtuales en un área geográfica real. Así, las marcas/empresas pueden fijar un radio de interés dentro del que podrán poner en marcha acciones en cualquier dispositivo móvil que tenga activado el GPS. Esto significa que las compañías podrían enviar notificaciones y mensajes a los usuarios que traspasen esos límites geográficos previamente definidos.

13.7.2. ¿Por qué *Geofencing*?

Este es un mundo competitivo, por decir lo menos, se ha vuelto muy importante para las empresas de hoy ser creativas y descubrir nuevas formas de captar la atención de los clientes y mantenerlos comprometidos.

Geofencing es una de las ideas más exclusivas que permiten a las empresas proporcionar una variedad de contenido a un usuario en función de su ubicación geográfica en tiempo real. Cuando decimos dispositivo, incluimos todo tipo desde una computadora de escritorio, un teléfono móvil o incluso un reloj inteligente.

En el Uruguay es una herramienta poco utilizada para el marketing por lo que vemos una oportunidad real de tener éxito.

13.7.3. ¿Por qué es aplicable a nuestro negocio?

Mejoras en la eficiencia de publicaciones

De acuerdo con las estadísticas de la empresa “Plotprojects”[9], *Geofencing* aumenta el uso de las aplicaciones en más del 200%, lo que lleva a un alto compromiso de los clientes y una mayor probabilidad de realizar una venta en nuestro caso. A su vez, la tasa de *clicks* de páginas que utilizan la tecnología es quince veces más alta que la de las notificaciones *push*. Se ha comprobado que las notificaciones de *Geofencing* aumentan la cantidad de personas que realmente hacen “*click*” en la notificación promocional que una empresa envía. La explicación a esto es que la notificación pasa a tener un por qué detrás. La persona que la reciba es porque manifestó interés en la temática a la que refiere dicha notificación.

Aumentar la presencia/reconocimiento de la empresa por parte de los consumidores

Las grandes empresas tienen la ventaja de ser conocidas por los consumidores y por si fuera poco los motores de búsqueda hacen que figuren en primera plana. Con esto, las pequeñas y medianas empresa encuentran problemas para mostrarse y ganar notoriedad. Geofencing puede ser una gran ventaja para las pequeñas empresas que buscan aumentar la presencia de la marca, ya que los consumidores serán alertados de comercios según su ubicación y ya no aplicará la priorización por renombre.

Es una manera rentable de comercializar

Geofencing no sólo proporciona a la empresa una sincronización de marketing perfecta, sino que también es una publicidad increíblemente rentable. La mayoría de las pequeñas empresas no tienen un presupuesto para marketing y publicidad, pero aún necesitan estas herramientas para hacer crecer su negocio. No supera el presupuesto porque una pequeña empresa puede crear su propio canal de marketing en lugar de pagar uno a través de una agencia de publicidad o marketing. El uso de *Geofencing* permite a las pequeñas empresas competir directamente con las grandes empresas por una fracción del precio. Cualquiera que sea el presupuesto, existe una amplia gama de opciones que cualquier pequeña empresa puede pagar.

13.8. Anexo 8 – Carta de Experto



Montevideo, 26 de Marzo de 2019

A quien corresponda, Como director de la empresa, Alvarez Latallada - Marketing, se deja constancia de que el día 26 de Marzo de 2019 nos reunimos con Alejandro Mamchur, Ignacio Peirano y Agustin Taruselli, estudiantes de la Universidad ORT Uruguay, quienes manifestaron estar desarrollando su proyecto de final de carrera.

En este marco, intercambiamos impresiones respecto a la conveniencia de la implementación de un sistema/aplicación para la solución de eliminar el gap existente entre el deseo de compra de un potencial cliente y la efectivización de la misma mediante el aumento del flujo de clientes hacia los comercios. Esto no solo beneficiaría a los comercios, sino que le daría además a los usuarios una herramienta práctica para acceder a productos deseados sin tener la necesidad de buscarlos activamente.

Cómo referentes del sector de Marketing consideramos que dicha propuesta es de valor para esta industria por aportar una solución al proceso de compra y del proceso de búsqueda. Lo más importante que percibimos es la automatización del proceso de búsqueda, brindándole al consumidor la oportunidad de encontrar el producto que busca al

alcance de su mano y al punto de venta le permite generar oportunidades de ventas latentes en concretas. Además de la funcionalidad general, creemos que la posibilidad de ser notificados de que el producto buscado se encuentra a “x” metros de distancia genera la intención de compra por impulso y al día de hoy vemos pocas herramientas que generen un impulso parecido.

Adicionalmente, expresamos nuestra disponibilidad de dar apoyo al proyecto a lo largo del desarrollo del mismo, y participar, de ser conveniente, en las diversas instancias de relevamiento y pruebas de producto que se requieran realizar.

Atentamente,

Javier Alvarez Latallada Mtr. Marketing Deportivo Experto en Neuromarketing y Marketing Digital

Tel: (+598) 96 102 114 Mail: javier@alvarezlatallada.com

13.9. Anexo 9 – Especificación de Requerimientos

13.9.1. Alcance del producto

Compra Cerca fue pensada para brindar soluciones tanto a compradores como a comercios. El estudio realizado de ambas perspectivas permitió especificar un conjunto de funcionalidades que brinden una mejora notoria a las actividades realizadas tanto por consumidores como comercios al desempeñar su rol en el proceso de compra y venta de productos y servicios.

El sistema se enfocará por un lado en mejorar las búsquedas de productos, disminuyendo el tiempo de búsqueda y buscando puntos de venta próximos a la ubicación del usuario. Los comercios que trabajen con CompraCerca tendrán la posibilidad de ganar visibilidad, aumentar el flujo de clientes, publicar ofertas, obtener información relevante del mercado. Debido a restricciones de tiempos, para la primera instancia de desarrollo se escogieron aquellas funcionalidades básicas e imprescindibles para la utilización del sistema.

El sistema tendrá para la perspectiva de los compradores los módulos de registro y *login* de los usuarios, búsqueda de productos/servicios por categoría y búsqueda por texto, la visualización de resultados de comercios en el mapa y su información.

Las funcionalidades referentes a la perspectiva de los comercios se desarrollarán en la segunda etapa de implementación.

Las funcionalidades para la administración de CompraCerca se desarrollarán en una primera versión y se continuarán en la segunda etapa.

13.9.2. Descripción general

Los requerimientos listados a continuación describen las funcionalidades propuestas para los 3 tipos de usuarios detectados de la aplicación. Por un lado, los usuarios compradores podrán buscar productos y encontrar puntos de venta más adecuados de acuerdo a su ubicación.

Los comercios podrán publicar ofertas en el sistema y aparecer resaltados en los resultados de búsqueda que realizan los compradores. Tendrán además un panel para el monitoreo del mercado y estadísticas acerca del funcionamiento de su comercio.

Por último los administradores podrán administrar y auditar los recursos del sistema y tomar acciones en caso que les sea pertinente.

13.9.3. Requerimientos funcionales

RF1 - Login de usuario

Descripción: Comprador deberá ingresar credenciales para poder ingresar al sistema. Por lo tanto, deberá estar previamente registrado.

Historia de usuario: Como usuario necesito ingresar mis credenciales para acceder al sistema.

Escenarios:

Usuario acceso correcto

Dado un usuario registrado

Cuando se ingresa el *login* y contraseña correcta y presiona el botón de loguearse

Entonces se despliega el home de la aplicación.

Usuario acceso incorrecto

Dado un usuario registrado

Cuando se ingresa un *login* y/o contraseña incorrecta

Entonces se muestra en el formulario de *login* el o los campos con error.

RF2 - Registro de usuario

Descripción: Comprador se registra en el sistema. Se solicitan todos los datos necesarios para utilizar el sistema.

Historia de usuario: Como usuario quiero poder registrarme en el sistema para luego poder acceder al mismo.

Escenarios:

Registro usuario datos correctos

Dado un usuario en el formulario de registro

Cuando se ingresa los datos correctamente

Entonces se despliega mensaje de éxito y se redirige a la pantalla de *login*.

Registro usuario datos incorrectos

Dado un usuario en el formulario de registro

Cuando se ingresa los datos incorrectos o incompletos

Entonces muestran errores en los campos que no cumplen las validaciones.

Registro usuario duplicado

Dado un usuario en el formulario de registro

Cuando se ingresa un *login* de usuario que coincide con un usuario ya existente

Entonces se despliega un mensaje indicando la situación que impide el registro.

RF3 – Buscar producto por categoría

Descripción: Comprador podrá realizar una búsqueda por una categoría de productos en particular. Estas categorías se desplegarán como un árbol de dos niveles. Los resultados de las búsquedas se desplegarán en el mapa.

Historia de usuario: Como usuario quiero poder navegar por el árbol de categorías provistas por el sistema para realizar una búsqueda por una categoría específica de producto.

Escenarios:

Búsqueda por categoría exitosa

Dado un usuario que navega el árbol de categorías y elige una categoría en particular

Cuando existen resultados de comercios dentro del rango de búsqueda del usuario
Entonces se despliega en el mapa los marcadores referenciando a los comercios resultados.

Búsqueda por categoría sin resultados dentro del rango

Dado un usuario que navega el árbol de categorías y elige una categoría en particular
Cuando no existen resultados de comercios dentro del rango de búsqueda del usuario
Entonces se muestra un modal donde el usuario confirmara si desea ampliar el rango de búsqueda.

Búsqueda por categoría sin resultados totales

Dado un usuario que navega el árbol de categorías y elige una categoría en particular
Cuando no existen resultados de comercios en todo el mapa
Entonces se muestra un mensaje informativo de que no se obtuvieron resultados.

RF4 – Buscar producto por coincidencia exacta

Descripción: Comprador podrá realizar una búsqueda de un producto desde la lupa de búsqueda. Los resultados de la búsqueda se desplegarán en el mapa. Priorizar resultados de comercios que trabajen con Compra Cerca.

Historia de usuario: Como usuario quiero poder buscar un producto ingresando un texto en la barra de búsqueda de la aplicación.

Escenarios:

Búsqueda por texto exitosa

Dado un usuario que escribe un texto en la lupa de búsqueda y presiona buscar
Cuando existen resultados de comercios dentro del rango de búsqueda del usuario
Entonces se despliega en el mapa los marcadores referenciando a los comercios resultados.

Búsqueda por texto exitosa sin resultados dentro del rango

Dado un usuario que escribe un texto en la lupa de búsqueda y presiona buscar
Cuando no existen resultados de comercios dentro del rango de búsqueda del usuario
Entonces se muestra un modal donde el usuario confirmará si desea ampliar el rango de búsqueda.

Búsqueda por texto exitosa sin resultados totales

Dado un usuario que escribe un texto en la lupa de búsqueda y presiona buscar
Cuando no existen resultados de comercios en todo el mapa
Entonces se muestra un mensaje informativo de que no se obtuvieron resultados.

RF5 – Agrupación de marcadores cercanos

Descripción: Para una correcta visualización de los resultados en el mapa los mismos deberán agruparse de acuerdo al nivel de *zoom* en el mapa, de manera que no se solapen nunca.

Historia de usuario: Como usuario quiero poder visualizar correctamente todos los resultados de búsqueda en el mapa.

Escenarios:

Resultados de búsqueda solapados

Dados varios comercios resultados de las búsquedas

Cuando los mismos se solapan para el nivel de zoom de la vista

Entonces marcadores deberían agruparse en elemento que indique que dentro se encuentran una cantidad n de resultados.

Resultados de búsqueda separados

Dados varios comercios resultados de las búsquedas

Cuando los mismos no solapan para el nivel de zoom de la vista

Entonces todos los resultados deberían estar a la vista.

RF6 - Consultar información del comercio

Descripción: Comprador podrá seleccionar uno de los resultados en el mapa y consultar toda la información disponible del comercio.

Historia de usuario: Como usuario quiero consultar la información de los comercios que me interesan de mis búsquedas.

Escenarios:

Consultar información de comercio regular

Dado un usuario que selecciona un marcador sin indicación especial en el mapa

Cuando haga *click* en el botón de ver perfil

Entonces se despliegan todos los datos disponibles del comercio

Consultar información de comercio que trabaja con Compra Cerca

Dado un usuario que selecciona un marcador resaltado en el mapa

Cuando hace *click* en el botón de ver perfil

Entonces se despliegan todos los datos disponibles del comercio, incluyendo las ofertas si es que las tiene.

RF7 - Visualizar ofertas de comercio

Descripción: Comprador tendrá la posibilidad de ver ofertas que el comercio publique.

Dichas ofertas se señalarán en el mapa junto con el marcador correspondiente al comercio.

Historia de usuario: Como usuario quiero ver las ofertas que publican los comercios. En caso de que sean de mi interés quiero poder guardarlas.

Escenarios:

Visualizar ofertas de comercio

Dado un usuario que selecciona un marcador con indicación especial en el mapa

Cuando hace *click* en el botón de ver perfil y el comercio tiene ofertas disponibles

Entonces se despliega perfil del comercio con un botón especial para acceder a las ofertas.

Las mismas se desplegarán en una nueva pantalla que podrá hacer *scroll* horizontalmente.

Marcar oferta de interés

Dado un usuario que selecciona un marcador con indicación especial en el mapa
Cuando hace *click* en el botón de ver perfil y el comercio tiene ofertas disponibles e ingresa a ver las ofertas
Entonces se muestra un ícono de estrella para agregar la oferta a mi bandeja de favoritas.

RF8 – Administración de la cuenta

Descripción: Comprador podrá configurar la información de su perfil así como configurar su rango de búsqueda actual y la información que se permite mostrar a terceros.
Historia de usuario: Como usuario quiero administrar mis datos, mis búsquedas y la privacidad de la información.

Escenarios:

Administración de la cuenta

Dado un usuario logueado en el sistema
Cuando selecciona desde el menú hamburguesa el icono de administración
Entonces se despliega formulario editable con toda su información en otra sección una barra que permite aumentar o disminuir el rango de búsqueda y por último una sección de privacidad donde administrará los permisos sobre sus datos.

Administración de la cuenta persistir datos erróneos

Dado un usuario logueado en el sistema
Cuando selecciona desde el menú hamburguesa el ícono de administración y selecciona la opción de modificar datos y se ingresa algún dato que no cumple con validaciones
Entonces se despliega mensaje de error en aquellos campos que presentan problema

Administración de la cuenta persistir datos correctos

Dado un usuario logueado en el sistema
Cuando selecciona desde el menú hamburguesa el ícono de administración y selecciona la opción de modificar datos y se ingresa algún dato que no cumple con validaciones
Entonces se despliega mensaje de error en aquellos campos que presentan problema

RF9 - Recordar búsqueda

Historia de usuario: Como usuario quiero guardar una búsqueda y sus resultados para acceder a los mismos en otro momento.

Escenarios:

Recordar búsqueda correcta

Dado un resultado de búsqueda desplegado en el mapa
Cuando el usuario presiona para guardar búsqueda
Entonces se despliega mensaje informativo acerca de que la búsqueda fue guardada

RF10 - Histórico de búsquedas

Historia de usuario: Como usuario quiero ver la lista de búsquedas guardadas y acceder alguna de ellas en caso que quiera.

Escenarios:

Ver histórico de búsqueda.

Dado un usuario logueado con búsquedas guardadas

Cuando el usuario selecciona el ícono de histórico

Entonces se despliega un modal con las búsquedas del usuario

Selección de búsqueda del histórico con conexión.

Dado un usuario logueado con conexión y búsquedas guardadas

Cuando el usuario selecciona el ícono de histórico y selecciona un ítem de la lista

Entonces se pasa a la vista de mapa, mostrando los resultados para la ubicación actual

Selección de búsqueda del histórico sin conexión.

Dado un usuario logueado sin conexión y búsquedas guardadas

Cuando el usuario selecciona el ícono de histórico y selecciona un ítem de la lista

Entonces se pasa a la vista de mapa, mostrando los resultados para la ubicación actual

RF11 - Mostrar ruta a comercio

Descripción: Comprador al seleccionar un comercio en particular y ver el perfil del mismo tendrá la posibilidad de buscar la ruta más cercana para llegar al comercio.

Historia de usuario: Como usuario quiero que el sistema me muestre la ruta más corta para llegar a un comercio de mi interés de acuerdo a mi ubicación actual.

Escenarios:

Mostrar ruta a comercio correcto

Dado un usuario logueado que se encuentra en la pantalla de perfil de un comercio

Cuando el usuario selecciona el ícono de mostrar ruta a comercio

Entonces se despliega en el mapa la ruta más conveniente para el comprador de acuerdo a su posición actual.

RF12 - Contactar comercio

Descripción: Comprador al seleccionar un comercio en particular y ver el perfil del mismo tendrá la posibilidad de contactarse telefónicamente con el comercio.

Historia de usuario: Como usuario quiero contactar un comercio para realizar una consulta acerca de un producto que vende.

Escenario:

Contactar comercio *mobile*

Dado un usuario logueado desde un celular y se encuentra en la pantalla de perfil de un comercio.

Cuando el usuario selecciona el ícono de teléfono

Entonces redirige al comprador a la funcionalidad teléfono de su celular con el número telefónico del comercio pre-cargado.

RF13 - Búsqueda activa

Descripción: Comprador tendrá la posibilidad de dejar una búsqueda activa en segundo plano. En caso de pasar a una distancia configurable de un comercio que ofrezca el producto que está buscando, se le enviará una notificación al comprador.

Historia de usuario: Como usuario quiero que el sistema continúe mi búsqueda incluso cuando no esté con la aplicación abierta y me notifique en caso de encontrarme cerca de un comercio que tenga lo que estoy buscando.

Escenarios:

Persistir búsqueda activa

Dado un usuario logueado con una búsqueda activa en el mapa

Cuando el usuario selecciona la opción de persistir búsqueda activa

Entonces el programa deberá mantener dicha búsqueda activa de acuerdo a la posición del comprador.

Búsqueda activa encuentra comercio

Dado un usuario logueado con una búsqueda activa

Cuando el usuario cruza la cerca virtual de un comercio

Entonces se envía una notificación al comprador indicando que tiene un comercio cerca para el producto o servicio que está buscando.

RF14 - Dashboard de comercio

Descripción: Comercios tendrán la posibilidad de ver información acerca del sistema que sea de su interés. Los productos más buscados en el mercado y aquellas búsquedas que generaron coincidencia con el comercio.

Historia de usuario: Como comercio quiero ver información del mercado y mis publicaciones para utilizar el sistema como una herramienta para la toma de decisiones.

RF15 - Ingreso de oferta

Descripción: Comercios tendrán la posibilidad de subir sus ofertas al sistema. Dichas ofertas deberán completarse en un formulario donde se solicitarán datos obligatorios.

Cada comercio tendrá una cantidad específica de ofertas que podrá subir de acuerdo a su plan de contrato.

Historia de usuario: Como comercio quiero crear una oferta para atraer compradores a mi tienda.

Escenarios:

Ingreso de una oferta válida.

Dado un comercio logueado en el sistema

Cuando ingresa en la funcionalidad de Ingreso de oferta

Entonces se despliega un formulario con datos para completar.

Sin saldo para cargar oferta.

Dado un comercio logueado en el sistema

Cuando comercio supera la cantidad máxima de ofertas que puede realizar

Entonces la funcionalidad en el menú del programa se mostrará apagada y en caso de que se haga *click* sobre ella se desplegará un pop up informativo

RF16 - Auditar oferta

Descripción: Comercios tendrán la posibilidad de auditar la cantidad de visualizaciones que tuvo su oferta en el sistema y cuantos compradores la marcaron de su interés. A su vez, podrá modificar la información de la oferta.

Historia de usuario: Como comercio quiero poder auditar las ofertas publicadas y modificar datos de la oferta.

RF17 - Notificar comercio usuario potencial cerca

Descripción: Comercios recibirán un mensaje en su bandeja de mensajes y una alerta por cada usuario que cruce la frontera establecida para la tienda.

Historias de usuario: Como vendedor quiero recibir una notificación cuando un cliente potencial ingresa dentro del radio establecido.

Escenarios:

Comercio recibe notificación

Dado un comprador logueado en el sistema y con una búsqueda activa

Cuando comprador cruza la cerca virtual del comercio

Entonces comercio recibe en una notificación y se ingresará un registro en su bandeja de mensajes.

RF18 - *Dashboard* de control

Descripción: *Dashboard* en la pantalla principal de los administradores de compra cerca donde se mostrarán información sobre el uso de la aplicación y un log de acciones realizadas en el sistema. Dicho archivo tendrá posibilidad de descargarse.

Historias de usuario: Como administrador de Compra Cerca quiero poder acceder rápidamente a información relevante de la aplicación.

Escenarios:

Comercio recibe notificación

Dado un comprador logueado en el sistema y con una búsqueda activa

Cuando comprador cruza la cerca virtual del comercio

Entonces comercio recibe en una notificación y se ingresará un registro en su bandeja de mensajes.

RF19 - Administrar compradores del sistema

Descripción: Lista de usuarios del sistema donde se podrá ver la información de cada usuario y tomar acciones como bloquear un comprador en particular.

Historias de usuario: Como administrador de Compra Cerca quiero ver la lista de compradores del sistema, filtrar por nombre de usuario y bloquear un comprador si fuera necesario.

RF20 - Administrar comercios del sistema

Descripción: Lista de comercios del sistema donde se podrá ver la información de cada uno y tomar acciones como bloquear un comercio en particular.

Historias de usuario: Como administrador de CompraCerca quiero ver la lista de comercios del sistema, filtrar por nombre de comercio y bloquear un comprador si fuera necesario.

13.9.4. Requerimientos no funcionales

Seguridad

RNF1- Identificación y autenticación de usuarios

Descripción: Los usuarios deberán loguearse en el sistema para poder acceder a las funcionalidades de la aplicación. Este requerimiento debe cumplirse para los distintos roles de la aplicación.

RNF2 - Utilización de *token* para llamadas http

Descripción: Todas las llamadas a la *API* del backend deberán contener en el encabezado de la llamada un *token* de identificación, de lo contrario se rechazará la solicitud.

RNF3 –Protección de URLs

Descripción: Todas las páginas de la aplicación y sus respectivas URL se deberán proteger para evitar accesos externos.

RNF4 - Protección de información sensible

Descripción: La información de los usuarios no estará visible por defecto. Será decisión del usuario que información compartir públicamente.

Disponibilidad

RNF5 - Disponibilidad del sistema

Descripción: La solución debe ofrecer adecuados niveles de servicio donde la disponibilidad y recuperación de fallos sea garantizada e inmediata.

Usabilidad

RNF6– Utilización de estándares de diseño

Descripción: Deberá hacerse hincapié en respetar los estándares *mobile* para el diseño de la interfaz de usuario.

RNF7 - *Responsive*

Descripción: Interfaz deberá contemplar la utilización del sistema desde un ordenador PC o desde un teléfono *mobile*.

RNF8 - Facilidad de acceso

Descripción: La mayoría de las funcionalidades podrán ser accedidas desde el menú principal directamente. No deberá ser necesario ingresar a una sección de la aplicación para saber que contiene.

RNF9 - Estética adaptativa

Descripción: La interfaz de usuario deberá respetar los estilos de UI dependiendo del dispositivo en el que se utilice la aplicación.

Performance

RNF10 - Tiempo de respuesta

Descripción: El sistema deberá responder a todos los pedidos del usuario en un tiempo menor a tres segundos promedio, trabajando en un sistema con requisitos mínimos para utilizar la aplicación y niveles aceptables de conexión a internet.

Eficiencia

RNF11 - Optimización en el uso de recursos

Descripción: El sistema deberá funcionar correctamente en móviles de gama media y baja. Se considerarán estos sistemas los principales plataformas en las que correrá la aplicación ya que las encuestas realizadas indicaron que en nuestro país son este tipo de móviles los predominantes.

Modificabilidad

RNF12 - Implementación a futuro de aplicación nativa.

Descripción: El sistema en primera instancia se creó como una aplicación web. Sin embargo, alguna de las funcionalidades que se prevén a futuro se adaptan de mejor manera a las características y prestaciones que brindan las aplicaciones nativas tanto de *Android* y como de *iOS*. Por lo tanto, la implementación y las tecnologías elegidas deberán contemplar el pasaje a corto plazo a una aplicación nativa.

13.10. Anexo 10 – Heurísticas de Nielsen

Las siguientes son las 10 reglas heurísticas sobre usabilidad de Jakob Nielsen:

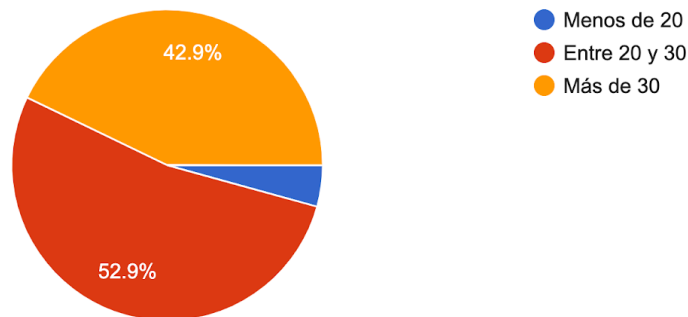
1. **Visibilidad del estado del sistema:** el sistema siempre deberá mantener informados a los usuarios de lo que está ocurriendo, a través de retroalimentación apropiada dentro de un tiempo razonable.
2. **Relación entre el sistema y el mundo real:** el sistema deberá hablar el lenguaje de los usuarios mediante palabras, frases y conceptos que sean familiares al usuario, más que con términos relacionados con el sistema. Seguir las convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.
3. **Control y libertad del usuario:** hay ocasiones en que los usuarios elegirán las funciones del sistema por error y necesitarán una “salida de emergencia” claramente marcada para dejar el estado no deseado al que accedieron, sin tener que pasar por una serie de pasos. Se deben apoyar las funciones de deshacer y rehacer.

4. **Consistencia y estándares:** los usuarios no deberán cuestionarse si acciones, situaciones o palabras diferentes significan en realidad la misma cosa; seguir las convenciones establecidas.
5. **Prevención de errores:** mucho mejor que un buen diseño de mensajes de error es realizar un diseño cuidadoso que prevenga la ocurrencia de problemas.
6. **Reconocimiento antes que recuerdo:** se deben hacer visibles los objetos, acciones y opciones. El usuario no tendría que recordar la información que se le da en una parte del proceso, para seguir adelante. Las instrucciones para el uso del sistema deben estar a la vista o ser fácilmente recuperables cuando sea necesario.
7. **Flexibilidad y eficiencia de uso:** la presencia de aceleradores, que no son vistos por los usuarios novatos, puede ofrecer una interacción más rápida a los usuarios expertos que la que el sistema puede proveer a los usuarios de todo tipo. Se debe permitir que los usuarios adapten el sistema para usos frecuentes.
8. **Estética y diseño minimalista:** los diálogos no deben contener información que es irrelevante o poco usada. Cada unidad extra de información en un diálogo, compite con las unidades de información relevante, disminuyendo su visibilidad relativa.
9. **Ayudar a los usuarios a reconocer,** diagnosticar y recuperarse de errores: los mensajes de error se deben entregar en un lenguaje claro y simple, indicando en forma precisa el problema y sugerir una solución constructiva al problema.
10. **Ayuda y documentación:** incluso en los casos en que el sistema pueda ser usado sin documentación, podría ser necesario ofrecer ayuda y documentación. Dicha información debería ser fácil de buscar, estar enfocada en las tareas del usuario, con una lista concreta de pasos a desarrollar y no ser demasiado extensa.

13.11. Anexo 11 – Resultados de encuestas a compradores

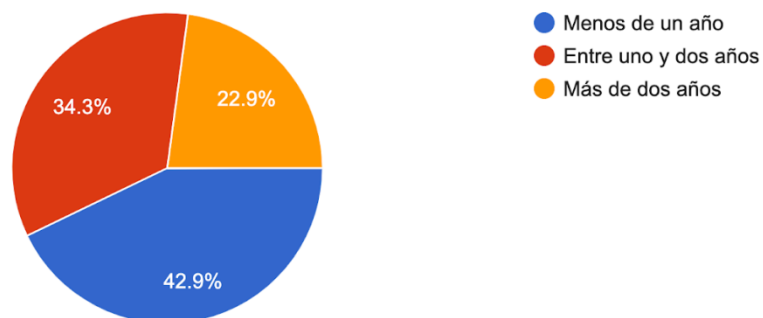
1 - ¿Qué edad tienes?

70 responses



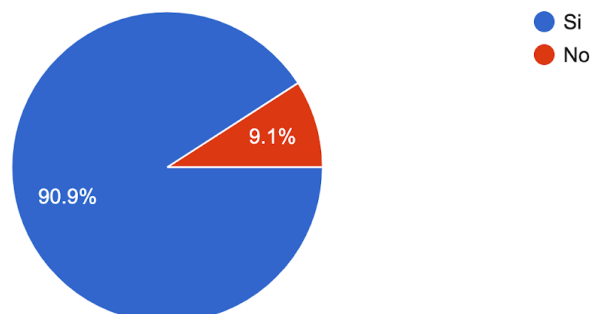
2 - ¿Hace cuánto tiempo cambiaste tu teléfono celular?

70 responses



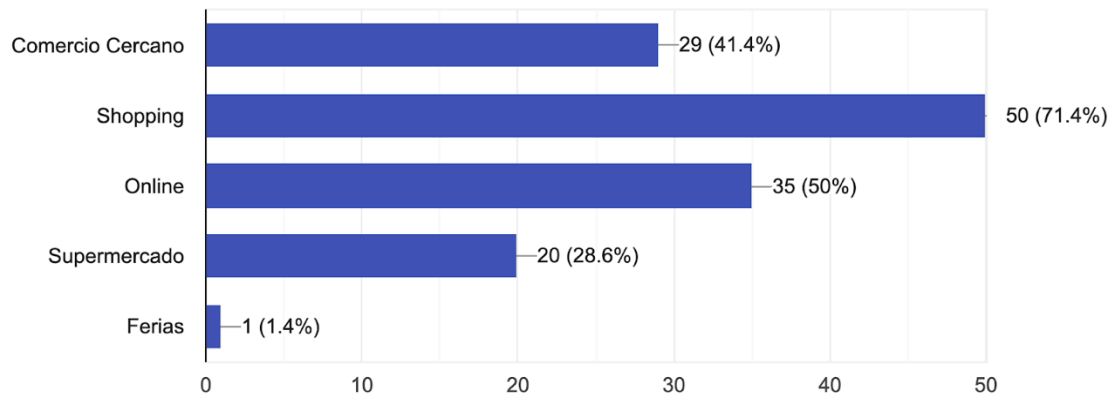
3 - ¿Cuenta con paquete de datos móviles en su celular?

66 responses



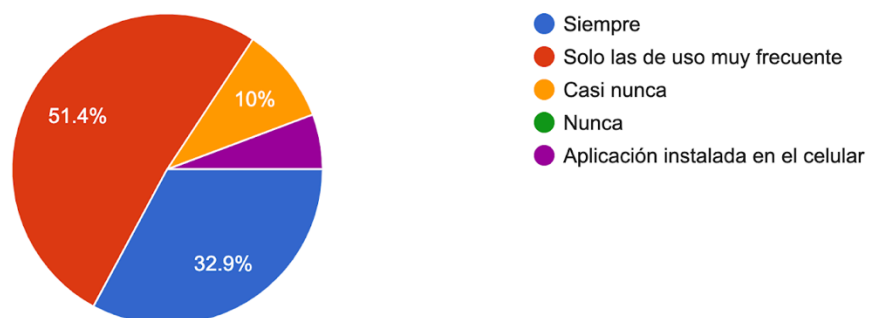
4 - ¿Dónde sueles comprar productos (calzados, ropa, tecnología, bazar, ferretería, etc.) y servicios (tintorería...etc.) Seleccionar las dos más frecuentes

70 responses



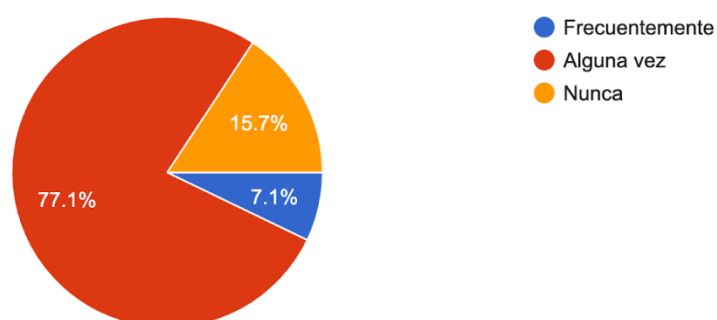
5 - Suele descargar aplicaciones en su celular?

70 responses



6 - ¿Le ha sucedido de buscar un comercio en Google/ Google Maps y no lo encontró?

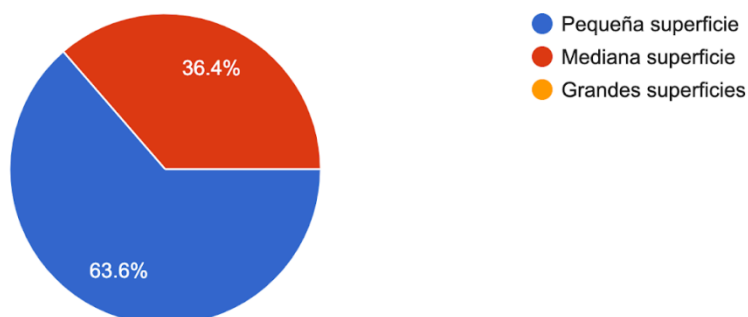
70 responses



13.12. Anexo 12 – Resultados de encuestas a vendedores

1 - ¿Qué tamaño tiene su comercio?

11 responses



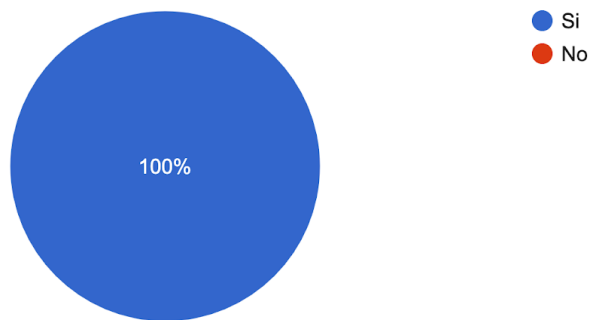
2 - ¿En qué rubro/categoría trabaja su comercio?

11 responses

Indumentaria deportiva
Ferretería
Rotisería
Repuestos de motos
Zapatería
Peluquería
Juguetes
Librería
Farmacia
Repuestos de motos y bicicletas
Heladería

3 - ¿Entiende que tiene la necesidad de captar más clientes?

11 responses



4 - ¿Qué medios utiliza actualmente para atraer clientes a su comercio?

11 responses

Promociones en instagram
Instagram
Publicidad en redes sociales, publicidad SEM publicidad en medios tradicionales, promociones de productos
Folleto
Facebook
Redes sociales y ofertas
Folletería
No estoy haciendo nada actualmente
Promociones en el local y folletos

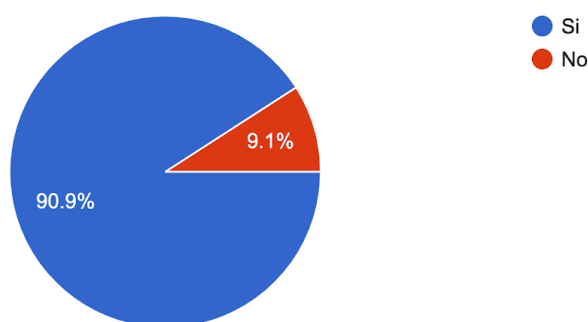
5 - ¿De los medios mencionados en la pregunta anterior, cuál es el que más utiliza y por qué?

11 responses

Publicidad en redes sociales y publicidad SEM debido a su bajo costo y alta eficiencia, tanto en volumen de impactos como en segmentación
Folletos porque es lo mas económico
Facebook porque es gratis
Redes sociales porque es gratuito
Instagram porque es gratuito y llega a muchos usuarios
Instagram es sin costo y es sencillo de usar
Siempre lo hice así y no es muy caro
Instagram porque es gratis y lo ven todos mis seguidores
cuando hago promociones, reparto folletos por el barrio
Promociones en el local porque imprimir los folletos demora y es mas caro
Instagram porque es gratuito y me promociono con muchos seguidores

6 - ¿Su comercio cuenta con ofertas/promociones habitualmente?

11 responses



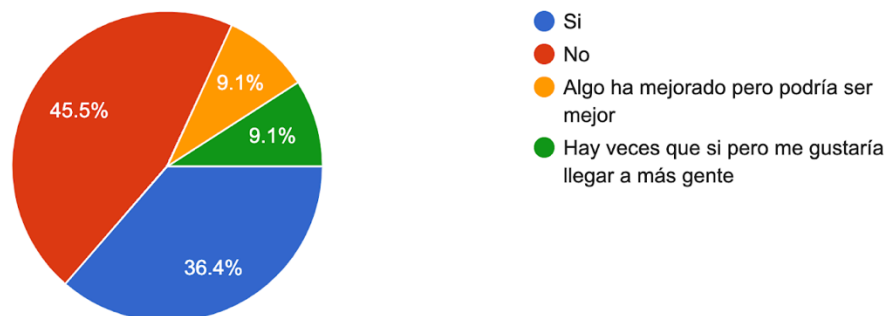
7 - De ser así ¿De qué manera da a conocer actualmente las ofertas de su comercio?

11 responses

Redes sociales
Cartelería en el local y en los folletos
Por facebook y en los pizarrones del local
En las redes sociales y con folletos en el local
En instagram y en el local
Instagram
Cuando tengo promociones las pongo en los folletos y en el local
Instagram y lo comento en el local
Cuando llegan los clientes les comento
Pongo las ofertas en el local
Las pongo en mi cuenta de instagram

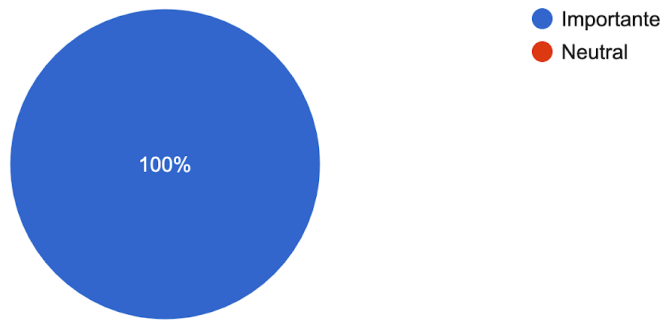
8 - ¿Considera que su estrategia actual para publicitar ofertas le da el resultado esperado?

11 responses



9 - ¿Cómo valoraría la posibilidad de poder poner ofertas online?

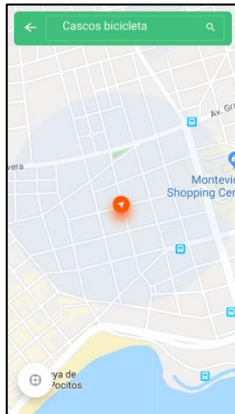
11 responses



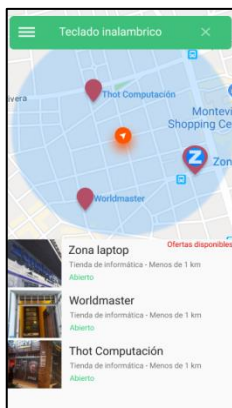
13.13. Anexo 13 - Mocks creados en Figma

Los siguientes *mockups* se definieron como herramienta de validación de los requerimientos funcionales propuestos, los cuales sirvieron además para la construcción del *frontend* de la aplicación.

Pantalla del *home* de la aplicación



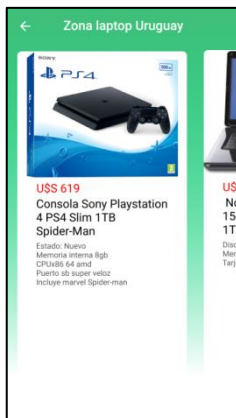
Pantalla con búsqueda activa



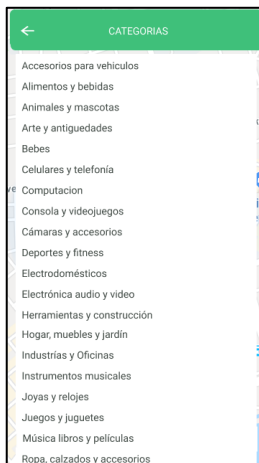
Pantalla perfil comercio



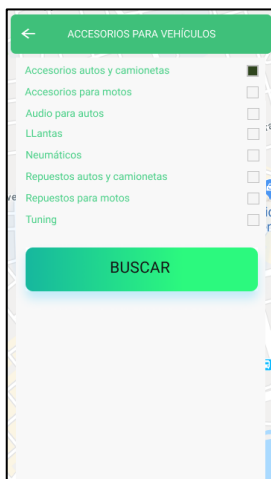
Ver ofertas de un comercio



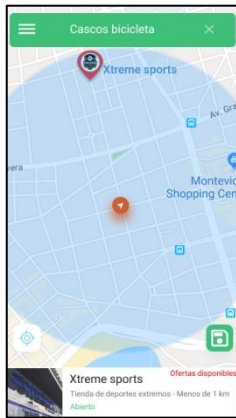
Árbol de categorías sin desplegar



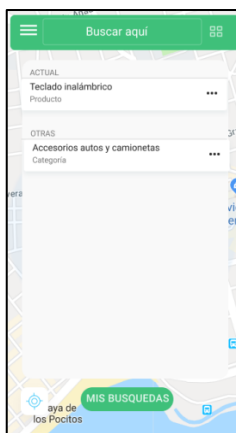
Árbol de categorías desplegado



Búsqueda con botón para guardar búsqueda



Histórico de búsquedas y submenú al tocar los tres puntos de cada ítem

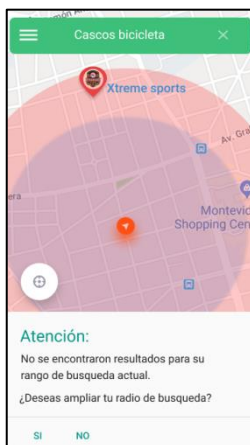


Mostrar

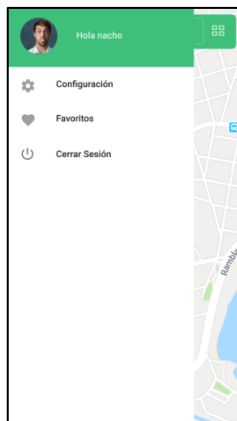
Buscar en segundo plano

Eliminar

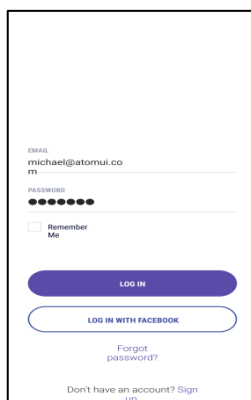
Sugerencia de aumentar radio de búsqueda en caso de no encontrar resultados



Menú principal desplegado



Pantalla de login



13.14. Anexo 14 - Planilla de selección de tecnologías

Web(Responsive)									
Front end					Backend				
Tecnología	Descripción	Costo	Curva de aprendizaje	Conclusiones	Tecnología	Descripción	Costo	Curva de aprendizaje	Conclusiones
Javascript	Es un lenguaje de programación interpretado que permite ejecutar lógica del lado del cliente	Gratis	Bajo, el equipo cuenta con los fundamentos básicos	Nos favorece el uso de esta tecnología al ser un lenguaje interpretado; no es necesario compilar la solución, permite en forma dinámica procesar la información del lado del usuario antes de enviar a servidor, portable y liviano. Es un	.Net Core	Es una plataforma para el desarrollo de software que lanzada por Microsoft permite desarrollar aplicación	Visual studio 2019: community gratis, el resto tiene evaluación gratuita, el	Bajo y medio pues un integrante trabaja con dicha tecnología y los restantes	Consideramos que es ventajoso el uso de esta tecnología por las siguientes razones: * Posee una curva de aprendizaje

				lenguaje muy sencillo. Es rápido, por lo tanto tiende a ejecutar las funciones inmediatamente. Cuenta con múltiples opciones de efectos visuales. Es soportado por los navegadores más populares y es compatible con los dispositivos más modernos, incluyendo iPhone, móviles y PS3. Es muy versátil, puesto que es muy útil para desarrollar páginas dinámicas y aplicaciones web. Es una buena solución para poner en práctica la validación de datos en un formulario. Es multiplataforma, puede ser ejecutado de manera híbrida en cualquier sistema operativo móvil. Es el único lenguaje que permite trabajar modo FullStack en cualquier tipo de desarrollo de programación.	es y sistemas independientes de la arquitectura física y del sistema operativo sobre el que se ejecuta. Desde el punto de vista comercial, .NET es la alternativa por parte de Microsoft en el sector de los desarrollos web para competir con la plataforma Java de Oracle Corporation, así como con los diversos frameworks basados en PHP. Al tratarse de una plataforma de propósito general, se pueden realizar tanto desarrollo	visual studio code es gratis	es lo vierno en la materia de aplicaciones que se enseña en la facultad	e muy rápida. * Integra el diseño e implementación de formularios de Windows. * Permite usar con facilidad la plataforma de los sistemas Windows, dado que tiene acceso prácticamente total a la API de Windows, incluidas librerías actuales. * Es uno de los lenguajes de uso más extendido, por lo que resulta fácil encontrar información, documentación y fuentes para los proyectos. * Fácilmente extensible mediante librerías
--	--	--	--	---	--	-------------------------------------	--	--

					s web, como programas de escritorio o aplicaciones para dispositivos móviles. Además favorece el desarrollo en multiplataforma, como por ejemplo el que una misma aplicación pueda correr indistintamente en los diferentes sistemas operativos móviles como IOS, Android o Windows Phone garantizando la comunicación entre los diferentes dispositivos.		DLL y componentes ActiveX de otros lenguajes. * Posibilita añadir soporte para ejecución de scripts, VBScript o JScript, en las aplicaciones mediante Microsoft Script Control. * Tiene acceso a la API multimedia de DirectX (versiones 7 y 8). También está disponible, de forma no oficial, un componente para trabajar con OpenGL 1.1: [7] * Existe una versión, VBA, integrada en las aplicaciones de Microsoft
--	--	--	--	--	--	--	---

										Office, tanto Windows como Mac, que permite programar macros para extender y automatizar funcionalidades en documentos, hojas de cálculo, bases de datos (access). * Si bien permite desarrollar grandes y complejas aplicaciones, también provee un entorno adecuado para realizar pequeños prototipos rápidos.
Angular	Angular es un framework desarrollado para javascript desarrollado por google, su finalidad es la de facilitar el desarrollo	Gratis	Medio, si bien el equipo no cuenta con experiencia previa, contamos con la página oficial de angular en la cual cuenta	Facilita implementar el comportamiento de la UI, manejo de mapas, servicios.	Aprendizaje de la estructura de proyecto angular.	Sql server	Es un sistema de gestión de base de datos relacional, desarrollado por Microsoft	Gratis	Bajo pues en el transcurso de la carrera de licenciatura en sistemas hemos tenido	Concluimos que el uso de esta tecnología nos favorece al brindarnos escalabilidad, seguridad y

	de aplicaciones web SPA y además el de brindar herramientas para trabajar con los elementos de una web de forma simple y óptima		con documentación extensa y detallada al igual que ejemplos					que usar en reitrad as ocasion es esta tecnología al igual de contar con experiencia del ámbito de trabajo	estabilida d, sporta transaccio nes		
Html 5	Lenguaje utilizado por angular e ionic para la construcción de la UI	Gratis	Bajo, el equipo cuenta con los fundamentos básicos	Provee el medio de presentación de los datos para los sitios web							
Bootstrap	Bootstrap es una Librería multiplataforma usada para dar estilos a aplicaciones web	Gratis	Medio, si bien el equipo no cuenta con experiencia previa, su manejo es sencillo pues es elegir el estilo que se adecúe mas al diseño que se desea. Cuenta con un amplio soporte de la comunidad de open	Nos favorece ampliamente su uso pues nos podemos desentender del diseño y de la elección de estilos, aplicando estándares de diseños ya aceptados por la industria							

			source al igual que un sitio web oficial que cuenta con extensa documentación y ejemplos							
TypeScript	Lenguaje utilizado por angular que luego se interpreta a javascript	Gratis	Bajo, el equipo cuenta con los fundamentos básicos	Nos favorece en escalabilidad del código en un "Interface oriented development", cuenta con amplias herramientas y comunidad open source, como por ejemplo la comunidad que empezó Microsoft. Types ha demostrado promover la calidad del código y su entendibilidad. Incrementa la agilidad en el refactoring	Aprendizaje de la sintaxis.					

				del código							
ionic frame work	Ionic es la plataforma a que facilita la creación de aplicaciones móviles con tecnologías web para los programadores web. La plantilla de Ionic permite a los desarrolladores crear aplicaciones móviles en diferentes plataformas que pueden instalarse en teléfonos con Android e iOS	Gratis	Medio, de a pesar de que el equipo no tiene experiencia trabajando con esta tecnología, se cuenta con sitio web oficial con toda la documentación y ejemplos necesarios	Nos favorece el uso de este Framework al simplificar el desarrollo de la UI respetando el estilo android	Aprendizaje de la estructura del proyecto y sus componentes web. Como se integra con angular.						
				Conclusión final: consideramos que las tecnologías a emplear que se adecuan mas a nuestras necesidades, es decir las que							

				mitigan mas los riesgos, nos permiten lograr el alcance estipulado en tiempo y en forma son todas las tecnología s que se enumeran en el cuadrante web(respo nsive), front end por todos los motivos que se enumerar on respectiva mente en cuanto a su curva de aprendizaj e, costo y conclusion es que obtuvimos por su posible uso.							
--	--	--	--	--	--	--	--	--	--	--	--

Mobile							
ios				Android			
Descripcio n	Costo	Curva de aprendizaj e	Conclusion es	Descripcio n	Costo	Curva de aprendizaje	Conclusion es
Desarrolla r una aplicación nativa que corra en el	Es gratis desarrollar en la plataforma de ios pero	Alta pues ningun integrante tiene experienci	Descartamo s desarrollar el front end para ios pues un solo	Desarrolla r una aplicación nativa para el	Es gratis el compilador para desarrollar en android	Media, si bien el equipo cuenta con relativa	Descartamo s desarrollar en android pues el equipo no

sistema operativo ios de apple	para lanzar un producto al mercado es necesario registrarse y publicar la app como desarrollador y eso tiene un costo de 100 dolares al año, además es necesario adquirir una mac para poder desarrollar en xcode y dichos equipos son costosos.	a trabajando o con esta tecnología , uno de los integrantes realizó un curso online completo hace dos años	miembro del equipo de proyecto contaba con experiencia desarrollando para ios y por tener la limitante de no poder desarrollar en xcode al no tener mac.	sistema operativo Android	(Android studio), al igual que publicar la aplicación en play store, basta con registrarse como desarrollador en la cuenta de google	experiencia en programación orientada a objetos, ningún miembro del equipo desarrolló antes una aplicación nativa para android	contaba con experiencia previa desarrollando aplicaciones nativas para android siendo la curva de aprendizaje media. Con la limitación del tiempo para aprender dicha tecnología
--------------------------------	--	--	--	---------------------------	--	--	--