

Universidad ORT Uruguay

Facultad de Ingeniería

GPShop

Geolocalización Indoor - Outdoor

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

Federico Barceló - 180736

Alejandro Kwilman - 157772

Pablo Pereira - 210162

Makcim Silveira - 167981

Tutor: Oscar Andrés Sena Diego

2021

Declaraciones de autoría

Nosotros, Federico Barceló, Alejandro Kwilman, Pablo Pereira y Makcim Silveira, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos la 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.



Figura 1: Federico Barceló
07/10/2021



Figura 2: Alejandro Kwilman
07/10/2021



Figura 3: Pablo Pereira
07/10/2021

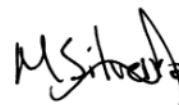


Figura 4: Makcim Silveira
07/10/2021

Agradecimientos

La elaboración y presentación del presente proyecto fue posible gracias a la buena predisposición, colaboración y ayuda de varias personas.

En primer lugar queremos agradecerle a nuestro tutor Oscar Sena que siempre estuvo dispuesto a ayudarnos, aconsejarnos y nos brindó su apoyo a lo largo de todo el proyecto.

También queremos expresar nuestra gratitud hacia el cliente Globant por la confianza en el equipo, a los desarrolladores de la librería y especialmente a Marcelo Santamaría, quien ha sido muy amable y generoso con su conocimiento, y nos ha brindado todo su apoyo durante el desarrollo de todo el proyecto y ha terminado siendo como un miembro más del equipo.

También agradecemos a Rafael Bentancur y Gastón Mousques quienes participaron en las instancias de revisión de este proyecto y han aportado con sus sugerencias y correcciones al desarrollo de la misma.

A Verónica Varela *UI/UX Expert* de Globant por su gran aporte en el desarrollo de la interfaz gráfica.

Agradecemos a Fabián Bozoglilanian *Team Agilist* de Globant por sus consejos y el aporte de su experiencia para la gestión del equipo y las responsabilidades.

Por ultimo y no menos importante, a todas las personas que han aportado de alguna u otra forma a este proyecto; familiares, amigos, entrevistados y demás personas quienes ofrecieron su tiempo para ayudarnos con las validaciones.

Abstract

El problema que se aborda trata de innovar en la utilización de las capacidades de los *smartphones* para la geolocalización, potenciando las mismas sin importar el ambiente en que se encuentre (espacios abiertos (*outdoor*) o cerrados (*indoor*), tales como centros comerciales, de salud, educativos, edificios de oficinas, etc.).

La idea surgió al observar que la emergencia sanitaria provocó que la utilización de mecanismos y dispositivos que permiten realizar actividades a distancia resulta imprescindible y además persistirá en el tiempo, puesto que cambió varios paradigmas de trabajo, negocios y relacionamiento social.

A tales efectos se desarrolló una aplicación que explota la posibilidad de geolocalización de un dispositivo, tanto en ambientes *indoor* como *outdoor* y su trayectoria, para resolver diversas necesidades y problemas para los usuarios de estos dispositivos, a la vez de generar valor para potenciales clientes que utilicen la aplicación, en diversos contextos de geolocalización.

En el caso de la tesis se optó por enfocarse en el contexto de un centro comercial (*shopping center*), poniendo de nombre a la aplicación GPSshop.

Un integrante del equipo planteó las ideas en la empresa donde trabaja, Globant. A través del intercambio, validación y ajuste de estas ideas, se logró despertar el interés de esta firma para utilizar un desarrollo interno de la empresa, consistente en una librería con funciones de georeferenciación y otras tecnologías, que posteriormente fueron utilizadas como base para realizar una plataforma que se adapte a cualquier contexto (ambiente) basada en el uso innovador (hacer algo que ya se hace pero de manera diferente) de geolocalización en interiores y exteriores, siendo de utilidad para diferentes mercados incluyendo a clientes actuales de la empresa.

El proyecto resultó un éxito, ya que Globant decidió utilizarlo internamente en el proceso de inducción de sus colaboradores, además de apreciar valor para incorporarlo a su cartera de soluciones para sus clientes.

Palabras Clave

Geolocalización, *indoor*, *outdoor*, navegación, *geofencing*, Android, iOS, *beacons*, plataforma, Laravel, *React native*, atributos de calidad, PHP, Docker, *CI/CD*, Laradock, usabilidad, gps, georeferenciación.

Índice general

1. Introducción	13
1.1. Estructura de la documentación	13
1.2. ¿De qué manera surge la idea del proyecto?	15
1.3. Objetivos	16
1.3.1. Del equipo	16
1.3.2. Del cliente	16
1.3.3. Del proyecto	17
1.3.4. Del producto	17
1.3.5. Académicos	17
1.4. Descripción del equipo	18
1.4.1. Experiencia y competencias del equipo	18
2. Descripción del problema	19
2.1. Contexto	19
2.1.1. Características del proyecto	19
2.1.2. Características del cliente	19
2.2. Propuesta de valor	20
2.2.1. Cliente	20
2.2.2. Interesados	20
2.3. Desafíos del proyecto	21
2.4. Tipos de usuario y necesidades	21
3. Descripción de la solución	22
3.1. Uso de la geolocalización de manera innovadora	22
3.2. Descripción general	23
3.3. <i>Benchmarking</i> de aplicaciones	24
3.4. Primera aproximación al producto	27
3.5. Alcance	31
3.6. Detalle sobre funcionalidades principales	33
3.6.1. Usuario cliente	33
3.6.2. Usuario tienda	36
3.6.3. Usuario Administrador	37

4. Ingeniería de Requerimientos	38
4.1. Proceso de ingeniería de requerimientos	38
4.2. Relevamiento	39
4.3. Análisis de requerimientos	40
4.4. Especificación de requerimientos	40
4.5. Definición de requerimientos	41
4.6. Validación de requerimientos	41
4.7. Product backlog	41
4.8. Requerimientos funcionales	44
4.9. Requerimientos no funcionales	45
5. Arquitectura de sistema	47
5.1. Descripción general de la arquitectura	47
5.1.1. Diagrama de paquetes	48
5.1.2. Diagrama de componentes	50
5.1.3. Diagrama de entrega	51
5.1.4. Diagrama de flujo	52
5.1.5. Diagrama de despliegue	53
5.1.6. Diagramas de (<i>Continuous Integration / Continuous Deployment</i>) (CI/CD)	56
5.1.7. Diagrama de base de datos	58
5.1.8. Módulos HM-10	59
5.1.9. Librería Geofence	64
5.2. Atributos de calidad	65
5.2.1. Adecuación funcional	65
5.2.2. Eficiencia	65
5.2.3. Compatibilidad	66
5.2.4. Usabilidad	66
5.2.5. Fiabilidad/confiabilidad	66
5.2.6. Seguridad	66
5.2.7. Mantenibilidad	67
5.2.8. Portabilidad	67
6. Desarrollo	69
6.1. <i>Frontend</i>	69
6.1.1. React Native	69
6.1.2. Lenguaje seleccionado	69
6.1.3. Migración de <i>JavaScript Bundler</i> de App	70

6.1.4.	Librería de Geolocalización	70
6.1.5.	Paradigma de programación	70
6.1.6.	Repositorio de la aplicación	71
6.1.7.	Estados Locales vs React Redux	72
6.1.8.	React Navigation	72
6.1.9.	Nuevas librerías y paquetes	73
6.1.10.	Autolinking	73
6.1.11.	Logo y Figma	73
6.2.	<i>Backend</i>	73
6.2.1.	Laravel	73
6.2.2.	Lenguaje seleccionado	73
6.2.3.	Composer	74
6.2.4.	Documentación	74
6.2.5.	Sistema de rutas	74
6.2.6.	Blade	74
6.2.7.	Eloquent	74
6.2.8.	QueryBuilder	74
6.2.9.	Artisan	75
6.2.10.	Laravel Socialite	75
6.2.11.	Laravel Nova	75
6.2.12.	Laradock	75
6.2.13.	Postman	75
6.3.	Beacons	76
6.4.	DevOps	79
7.	Gestión de proyecto	80
7.1.	Proceso de desarrollo	80
7.1.1.	Ciclo de vida	80
7.1.2.	Metodologías	81
7.2.	Etapas del proceso	89
7.2.1.	Planificación e Investigación	89
7.2.2.	Definición de requerimientos	91
7.2.3.	Diseño y prototipado	91
7.2.4.	Desarrollo	92
7.2.5.	Testing	92
7.2.6.	Documentación	92
7.2.7.	Release	93
7.3.	Estrategia de gestión	94

7.3.1.	Roadmap	94
7.3.2.	GANTT	95
7.3.3.	Planificación de roles y recursos	95
7.3.4.	Registro de actividades	96
7.3.5.	Plan de <i>releases</i>	97
7.3.6.	Acuerdos y convenciones	98
7.3.7.	Métricas	99
7.4.	Gestión de riesgos	103
7.4.1.	Plan de riesgos	103
7.4.2.	Riesgos identificados y mitigación respectiva	104
7.4.3.	Clasificación y evolución de riesgos respecto a su categoría	108
7.4.4.	Conclusión y lecciones aprendidas sobre riesgos	108
7.5.	Gestión de configuración	110
7.5.1.	Identificación de elementos de configuración	110
7.5.2.	Elección de herramientas	111
7.5.3.	<i>Software</i>	112
7.5.4.	Documentos	116
7.5.5.	Base de datos	119
7.5.6.	Elementos de UI/UX	119
7.6.	Gestión de calidad	120
7.6.1.	Objetivos para el aseguramiento de la calidad	120
7.6.2.	Plan de SQA	120
7.6.3.	Plan de calidad	121
7.6.4.	Revisiones	135
7.6.5.	Validaciones	136
7.6.6.	Defectos	138
7.6.7.	Verificaciones	139
7.6.8.	Pruebas unitarias	139
7.7.	Gestión de comunicación	140
7.7.1.	Interesados	141
7.7.2.	Canales de comunicación	141
7.7.3.	Plan de comunicación	143
7.8.	Gestión de incidentes	145
8.	Conclusiones y lecciones aprendidas	146
8.1.	Estado actual	146
8.2.	Opinión del cliente	147
8.3.	Conclusiones generales	149

8.3.1. Objetivos del equipo	149
8.3.2. Objetivos del cliente	149
8.3.3. Objetivos del proyecto	149
8.3.4. Objetivos del producto	149
8.3.5. Académicos	150
8.4. Lecciones aprendidas	151
8.5. Próximos pasos	152
9. Referencias bibliográficas	153
10. Anexos	164
10.1. Anexo 1 - Resúmenes <i>sprints</i>	164
10.1.1. <i>Sprint 1</i>	164
10.1.2. <i>Sprint 2</i>	166
10.1.3. <i>Sprint 3</i>	168
10.1.4. <i>Sprint 4</i>	170
10.1.5. <i>Sprint 5</i>	172
10.1.6. <i>Sprint 6</i>	174
10.1.7. <i>Sprint 7</i>	176
10.1.8. <i>Sprint 8</i>	178
10.1.9. <i>Sprint 9</i>	180
10.1.10. <i>Sprint 10</i>	182
10.1.11. <i>Sprint 11</i>	184
10.2. Anexo 2 - User persona	186
10.3. Anexo 3 - Prototipo	189
10.4. Anexo 4 - ESRE	199
10.4.1. Datos del proyecto	199
10.4.2. Propósito del documento	199
10.4.3. Requerimientos funcionales	200
10.4.4. Requerimientos no funcionales	238
10.5. Anexo 5 - Análisis de negocios	240
10.5.1. Introducción	240
10.5.2. Sobre de las 5 fuerzas competitivas de Porter	240
10.5.3. Las Fuerzas	241
10.5.4. Sobre el análisis PESTEL	244
10.5.5. Análisis de las 5 fuerzas competitivas de Porter	247
10.5.6. Conclusiones análisis 5 fuerzas	248
10.5.7. Análisis PESTEL	250

10.5.8. Conclusiones	253
10.5.9. Fuentes	254
10.6. Anexo 6 - Flujos principales aplicación	255
10.7. Anexo 7 - Gestión de Riesgos	262
10.7.1. Escala de Riesgos para Amenazas.	262
10.7.2. Escala de Riesgos para Oportunidades.	263
10.7.3. Riesgos evaluados en <i>Sprint</i> 0.	264
10.7.4. Riesgos evaluados en <i>Sprint</i> 1.	265
10.7.5. Riesgos evaluados en <i>Sprint</i> 2.	266
10.7.6. Riesgos evaluados en <i>Sprint</i> 3, parte I.	267
10.7.7. Riesgos evaluados en <i>Sprint</i> 3, parte II.	268
10.7.8. Riesgos evaluados en <i>Sprint</i> 4.	269
10.7.9. Riesgos evaluados en <i>Sprint</i> 5.	270
10.7.10. Riesgos evaluados en <i>Sprint</i> 6.	271
10.7.11. Riesgos evaluados en <i>Sprint</i> 7.	272
10.7.12. Riesgos evaluados en <i>Sprint</i> 8.	273
10.7.13. Riesgos evaluados en <i>Sprint</i> 9.	274
10.7.14. Riesgos evaluados en <i>Sprint</i> 10.	275
10.7.15. Riesgos evaluados en <i>Sprint</i> 11.	276
10.7.16. Riesgos evaluados en <i>Sprint</i> 12.	277
10.7.17. Plan de mitigación de riesgos.	278
10.7.18. Distribución de riesgos por sprint.	279
10.7.19. Magnitud por sprint y tipo de riesgos.	280
10.7.20. Magnitud por sprint y tipo de riesgos.	281
10.7.21. Mayor magnitud acumulada de tipo de riesgos en sprints.	282
10.8. Anexo 8 - Gestión de Calidad	283
10.8.1. Cobertura de pruebas automatizadas en SonarQube - <i>Frontend</i>	283
10.8.2. Reglas de calidad definidas para pruebas de SonarQube - <i>Frontend</i>	284
10.8.3. Cobertura de pruebas automatizadas en SonarQube - <i>Backend</i>	285
10.8.4. Reglas de calidad definidas para pruebas de SonarQube - <i>Backend</i>	286
10.9. Anexo 9 - Información Globant	287
10.9.1. Acerca de Globant	287
10.9.2. Ubicación geográfica	288
10.9.3. Estructura interna	289
10.9.4. Labs y Catalyst	291

10.10	Anexo 10 - Gestión de Incidentes	292
10.11	Anexo 11 - Módulos HM-10	293
10.12	Anexo 12 - <i>Business Canvas</i>	297

1. Introducción

1.1. Estructura de la documentación

Se brinda a grandes rasgos el contexto sobre el cual se desarrolló el proyecto, se explica cómo fue concebida la idea del mismo y cuáles son los objetivos de llevarla a cabo.

Descripción del problema

En esta sección se plantea el contexto del problema, cuál es el valor buscado por el cliente y diferentes *stakeholders*. Se detallan también los tipos de usuarios y desafíos encontrados en el problema.

Descripción de la solución

En esta sección se brinda de manera general la solución planteada, brindando también el alcance de proyecto.

Ingeniería de requerimientos

En esta sección se plantea como fue el trabajo realizado a nivel del proceso, análisis, especificación, definición, validación de los requerimientos funcionales y no funcionales.

Diseño arquitectónico

En esta sección se realiza la justificación sobre la elección tecnológica, se describe el diseño arquitectónico de la solución, evidenciándolo con diferentes diagramas.

Desarrollo

En esta sección se describe como fue el proceso de desarrollo para el *backend*, *frontend*, practicas relacionadas a *DevOps* como pueden ser *CI* y *CD*.

Gestión de proyecto

En esta sección se realiza la descripción completa sobre la gestión de proyecto, desde el proceso de desarrollo, etapas del proceso, y gestión de:

- Riesgos.
- Configuración.
- Calidad.
- Comunicaciones.
- Incidentes.

Conclusiones y lecciones aprendidas

En este capítulo se habla sobre las lecciones, conclusiones, opinión del cliente y próximos pasos a seguir.

1.2. ¿De qué manera surge la idea del proyecto?

El proyecto nace a partir de dos ideas planteadas durante las jornadas pautadas para realizar *brainstorming* ocurridas durante los momentos previos a la elección de la temática del proyecto a realizar, las cuales eran:

- Construcción de una red de máquinas expendedoras inteligentes construidas en base a *containers* de 40 pies, los cuales serían ubicados en diferentes puntos de la ciudad. Ofreciendo mediante una aplicación móvil las diferentes posiciones de las mismas, cada una con sus ofertas disponibles y ofreciendo la posibilidad de manejar publicidad. Pudiendo realizar análisis de mercado con el objetivo de mejorar las ventas dependiendo de la ubicación y productos disponibles, entre otros.
- Construir tótems de autoservicio inteligentes con especial énfasis en los emergentes y recientemente populares mercados de ofertas gastronómicas en diferentes puntos de la ciudad, donde los usuarios puedan realizar sus pedidos y recibir sugerencias personalizadas entre otros. Ofreciendo también la posibilidad de que las tiendas gastronómicas puedan ofrecer sus publicidades de manera directa en los mismos.

Con estas dos ideas en mente, comenzamos la búsqueda de patrocinadores que nos acompañen a llevarlas a cabo, entonces un integrante del equipo planteó las ideas en su trabajo (Globant).

De este intercambio surgió la idea final, el valorar que la emergencia sanitaria provocó que la utilización de mecanismos y dispositivos que permiten realizar actividades a distancia se potenció y al advertir esto, Globant llevó adelante un desarrollo interno, consistente en una biblioteca de funcionalidades de geolocalización, que entre otras cosas utiliza una tecnología innovadora para resolver la falta de fidelidad del GPS en ambientes cerrados, pero aún no tiene aplicaciones de utilización con esta librería, de tal forma que surgió la oportunidad de generar un producto que sirva como base para realizar una plataforma que se adapte a cualquier contexto de uso (centros comerciales, de salud, educativos, edificios de oficinas, etc.), siendo de utilidad en diferentes ambientes y mercados.

1.3. Objetivos

1.3.1. Del equipo

- Desarrollar un proyecto donde se integran todos los conocimientos adquiridos durante la carrera en una solución de alto impacto y gran nivel.
- Utilizar conocimientos adquiridos durante la etapa laboral, que permitieron refinar y aplicar los conocimientos teóricos adquiridos durante la carrera.
- Realizar un producto en colaboración con una empresa de renombre para adquirir experiencia en la utilización de las mejores prácticas en la Ingeniería de *Software* y Gestión de Proyectos, mediante la transferencia de conocimientos de la empresa hacia el equipo de proyecto.
- Inserción de la solución realizada en el mercado.
- Lograr un proyecto universitario con calificación de excelencia académica, y que pueda ser referenciado por futuras generaciones.
- Aprender sobre tecnologías no manejadas de forma cotidiana por los miembros, necesarias para poder realizar el producto de software.
- Trabajar en un equipo integrado por personas que cumplen diferentes roles.

1.3.2. Del cliente

- Obtener una plataforma que utilice como base el desarrollo realizado internamente sobre la geolocalización (librería de geolocalización y tecnologías de identificación de puntos georeferenciados), la cual será utilizada para integrarla a su cartera de productos.
- Posicionarse como una empresa que apoya tesis de grado y generar lazos entre la academia y la industria.
- Finalización de los estudios universitarios del *Glober*, con la obtención de una tesis de grado con la mayor calificación y reconocimientos universitarios posibles.

1.3.3. Del proyecto

El objetivo principal del proyecto es el desarrollo de una plataforma que utilice la geolocalización de una forma innovadora, partiendo del proyecto interno proporcionado por Globant, a quien le interesa explotar su potencial dentro de diferentes contextos y mercados. Integrándose con un *stack* tecnológico con alto grado de diversificación.

Con la idea planteada por el equipo de tesis, se va a lograr la integración de *beacons* con el fin de lograr geolocalización *indoor*, para obtener la plataforma base, que podrá ser integrada sin importancia del contexto en donde quiera ser utilizada.

1.3.4. Del producto

Llegar a un *Minimum Viable Product* (MVP) que se utilice en dispositivos móviles, específicamente aquellos que utilicen Android y iOS, que oficie como validador de la aplicación móvil que utiliza a geolocalización para interiores y exteriores.

El producto en cuestión se propone como principal objetivo exponer una nueva capa de servicios para ser visualizada por el usuario final en una aplicación móvil, que puede tener varios destinos:

- Marketing directo.
- Recorrido *outdoor/indoor*.
- *Business intelligence* y estadísticas.
- Control de flujo de personas.
- Logística y seguridad.
- En forma independiente del contexto (ambiente: centro comercial, de salud, educativo, etc.) donde se utilice.

1.3.5. Académicos

- Aplicación de todas las etapas del proceso de Ingeniería de *Software*.
- Desarrollar experiencia en metodologías de investigación y elaboración de tesis.
- Cumplir con todas las formalidades, requerimientos y solicitudes universitarias con el fin de lograr un nivel de aprobación de excelencia.

1.4. Descripción del equipo

1.4.1. Experiencia y competencias del equipo

El equipo se encuentra conformado por cuatro integrantes, cada uno tiene un enfoque tecnológico en diferentes áreas dentro del mundo de la Ingeniería, lo cual nos permite a la hora de realizar el proyecto, poder ordenar las cargas de trabajo de una manera más eficiente dependiendo de la experiencia de cada uno de los miembros.

Federico Barceló: Estudiante de Licenciatura en Sistemas, recibido de la carrera Técnico en sistemas operativos y redes dictada en la misma casa de estudios. Actualmente se encuentra trabajando como *DevOps Engineer* en Globant. Cuenta con experiencia en temas relacionados con infraestructuras híbridas, telecomunicaciones, migraciones de *datacenters* a entornos *cloud*, *continuous integration* (CI), *continuous deployment* (CD), *Platform as a Service* (PaaS), *Software as a Service* (SaaS) e *Infrastructure as a Service* (IaaS). También dicta un curso en la Universidad ORT Uruguay sobre metodologías *DevOps* y afines.

Alejandro Kwika: Estudiante de Licenciatura en Sistemas. Actualmente trabaja para Oracle NetSuite en el rol de Consultor Senior en el área de *advanced Customer Support*. También se desempeña como docente de la materia Taller de tecnologías en la Universidad ORT Uruguay.

Pablo Pereira: Estudiante de Licenciatura en Sistemas. Actualmente trabaja como Software Engineer en MercadoLibre. Cuenta con experiencia en programación competitiva, participando de múltiples competencias. Junto a su equipo son campeones de la última edición del Concurso de Programación de la Universidad ORT Uruguay.

Makcim Silveira: Estudiante de Licenciatura en Sistemas. Actualmente se desempeña como Consultor de Data y Analytics en Quanam. Cuenta con conocimiento de *Extract, Transform, Load* (ETL), *Extract, Load, Transform* (ELT) y *Data Pipelines*, *Data lakes*, *Data Visualization*, *Data Governance*, *Enterprise Data Warehouse*, *Big Data* y *Master Data Management* (MDM).

2. Descripción del problema

2.1. Contexto

Durante los últimos dos años, la movilidad de la sociedad se ha visto afectada por la pandemia que afecta a nivel mundial, llevando a tener que reinventarse a varios segmentos del mercado, con el fin de poder seguir realizando sus actividades en la llamada nueva normalidad y como consecuencia de esto, la necesidad de tener nuevas herramientas que se logren adaptar al nuevo estilo de vida instaurado por la crisis sanitaria. Esto llevó a una mayor utilización de los teléfonos inteligentes, explotando diversas capacidades de los mismos, por lo que el público usuario llegó a manejar con mayor habilidad los dispositivos, surgiendo la posibilidad de aumentar las aplicaciones basadas en geolocalización.

La aplicación tiene como objetivo brindar una ayuda guiada adaptable dependiendo el contexto en donde sea empleada.

2.1.1. Características del proyecto

La plataforma a obtener se caracteriza por ser un producto de *backend* multiusuario adaptable dependiendo el contexto en donde se quiera emplear. Pudiendo llegar a ser percibida como un ayudante virtual.

Como será explicado en los siguientes capítulos, existió la necesidad de enmarcar el proyecto en un solo contexto de uso, dadas las características temporales del proyecto.

Este contexto fue escogido luego de definir entre una serie de áreas e industrias que pudieran resultar de particular valor para el proyecto, cuales serían las más adecuadas para implementar, según su complejidad, impacto y comercialización.

Luego de un análisis de negocios optamos por la solución que involucró la implementación en centros comerciales. (anexo 10.5 análisis de negocio).

2.1.2. Características del cliente

El cliente del proyecto es Globant, empresa enfocada en brindar soluciones de alto nivel y escalables empleando tecnologías en el campo digital y cognitivo para

transformar las organizaciones en todos los aspectos a nivel mundial.

El nexos con la empresa fue a través del *IoT Platform Engineer* Marcelo Santamaria. Marcelo es parte del equipo interno de Labs en Globant, en el cual se encuentra alojado el programa Catalyst, cuyo cometido es impulsar la implementación interna de este tipo de desarrollos reutilizables. (anexo 10.10 para ver información en detalle sobre el cliente y su estructura).

2.2. Propuesta de valor

En esta sección se detallará el valor que aportará el proyecto a los diferentes actores identificados.

2.2.1. Cliente

Globant obtendrá un *MVP* que se integre con proyectos ya incubados dentro del equipo interno de Labs, en donde podrá terminar de validar los mismos con una prueba de campo.

La pretensión por parte del cliente es utilizar el proyecto a futuro en implementaciones de herramientas internas en la organización u ofrecerlo como producto a sus clientes.

2.2.2. Interesados

Se identifican varios interesados para el proyecto, a continuación se detalla el valor que aporta el proyecto a cada uno de ellos:

- Globant: Generar vínculo entre Universidad ORT Uruguay y la empresa para instancias de proyecto final, generación de nuevos casos de uso potenciales para la plataforma base de geolocalización.
- Universidad ORT Uruguay: Generación de un proyecto de grado, y el vínculo creado con Globant para las instancias de proyecto final.
- Centros comerciales: Creación de un nuevo canal diferenciado, para poder impactar a sus clientes de forma más eficaz, y ponerse en contacto; como fue investigado en el análisis de negocio, en el funcionamiento de las diferentes generaciones: *Baby Boomers*, *Generation X*, *Millennials* y *Centennials* a la hora de comprar y brindarles esa experiencia diferente a sus clientes. (anexo análisis de negocios 10.5)

- Clientes/Usuarios: Tener otro canal para aprovechar las ofertas de tiendas que realmente le interesan, en el momento que le pueden interesar y basada en su localización; Es decir en el momento y lugar adecuados.

2.3. Desafíos del proyecto

Como fue detallado a alto nivel anteriormente, el proyecto le implica al equipo y al cliente la tarea de integrar un *stack* de tecnologías con el fin de lograr obtener la aplicación en cuestión, en donde se debe destacar:

- Navegación de manera precisa tanto *indoor* como *outdoor*.
- Habilidad de comprensión certera sobre ciertos flujos de procesos para completar determinadas necesidades/actividades.

Como consecuencia de lo anterior más el trabajo a nivel académico, encontramos el desafío de realizar la gestión de un proyecto desde cero más el trabajo en conjunto con equipos del cliente.

2.4. Tipos de usuario y necesidades

Se encuentran tres tipos de usuarios principales, la clasificación de manera genérica nos lleva a identificarlos como usuarios cliente de la aplicación (anexo 10.2 para ver los *user* persona), usuarios tienda y usuarios administradores. La aplicación oficia como nexo entre todos.

Los usuarios cliente son aquellos que realizarán las acciones de consumo, compartido, *feedback* sobre los descuentos ofrecidos por las tiendas, buscando beneficiarse de manera directa por las promociones ofrecidas por las anteriores.

Los usuarios tienda son aquellos que buscan tener un canal de llegada alternativo y de manera directa con sus diferentes promociones sobre potenciales clientes que deambulan por el *shopping center*.

Los usuarios administrador son aquellos que pueden realizar las mismas funcionalidades que el usuario tienda pero además tienen la posibilidad de hacer tareas de administración sobre el sistema.

3. Descripción de la solución

3.1. Uso de la geolocalización de manera innovadora

Durante este apartado se realizará la justificación sobre el aporte innovador en el uso de una tecnología existente ya consolidada como lo es la geolocalización de una manera poco utilizada en la actualidad.

Es importante resaltar que la geolocalización es empleada en muchos ámbitos diariamente, desde el rubro de la logística, ubicación en tiempo real (*Waze*, *Google Maps*), aplicaciones para conocer gente (*Tinder*, *Happn*) o entretenimiento (*Pokémon Go*, *Swarm*), utilizando la misma con el fin de realizar *tracking* durante ciertos lapsos de tiempo en específico.

Dentro de la idea de plataforma que desarrollamos, se hace un fuerte foco en el uso de la ubicación en tiempo real de manera precisa, tanto *indoor* como *outdoor*, pero no con el fin de realizar un seguimiento por donde deambula el usuario, sino que cuando el mismo pasa por ciertos puntos estratégicos, podamos realizar ciertas acciones de nuestro interés, como el envío de notificaciones u obtener datos varios a partir del flujo de usuarios que transitan cerca de estos puntos para realizar *data analytics*.

La principal idea se centra en servir como una plataforma base para diferentes contextos de uso en donde se ve un foco común (geolocalización) desde el cual se pueden realizar diferentes acciones en torno al mismo, y apoyada en el uso de *beacons* (dispositivos de georeferenciación) para reducir la falta de fiabilidad del GPS en ambientes cerrados (*indoor*).

3.2. Descripción general

Como fue dicho en el apartado anterior, la aplicación a implementar busca funcionar como un asistente personalizado, apoyándose de manera directa con la geolocalización, de acuerdo con el análisis de mercado realizado donde se encontraron las condiciones adecuadas para la explotación comercial. (anexo 10.5 análisis de negocio) Es de esta forma y por más que puede utilizarse en múltiples contextos, que fue acordado con el cliente utilizar el contexto de grandes superficies, (shopping centers) para poder realizar una *Proof Of Concept* (POC) y en caso de ser satisfactorio, un futuro MVP.

El cliente deja disponible para el equipo, una librería como producto interno desarrollada sobre Swift, que realiza los cálculos de localización *indoor* (apoyándose con la utilización de *beacons*) y *outdoor*, la cual será utilizada como parte vital del desarrollo.

La idea del cliente brindando esta librería, además de realizar una POC sobre el desarrollo en un entorno real, es lograr activar un flujo interno dentro de una unidad empresarial denominada Labs, el cual termina llevando un *MVP* a la mesa de consideración como posible proyecto a ser implementado en clientes o internamente en la empresa.

3.3. *Benchmarking* de aplicaciones

Se realizó durante la etapa de especificación de requerimientos, un análisis sobre aplicaciones que de alguna manera se vinculan con nuestra idea, específicamente sobre cómo utilizan la geolocalización; no pretendemos de ninguna manera realizar algún tipo de integración con estas aplicaciones, sino analizarlas.



Interfase

- Mapa tipo google map.
- Newsfeed.
- Buscador.
- Menu principal debajo.
- Iconografía y poco texto.

Funcionalidades

- Buscar sitios de interés.
- Mostrar información sobre los sitios. Texto + fotos.
- Hacer Check In. Anunciarse.
- Inbox con noticias.
- Escribir reseñas.

Fortalezas y debilidades

- Buscador sencillo o navegar desde el mapa mismo.
- Incluye una sección de comunidad.

Figura 3.1: *Benchmarking* Swarm



Interfase

- Mapa tipo google map.
- Sujerencia de lugares favoritos o mas visitados.
- Buscador.
- Menú desplegable.
- Iconografía y poco texto.

Funcionalidades

- Buscar la mejor ruta para llegar de un punto a otro.
- Favoritos.

Fortalezas y debilidades

- Incluye audio asistido para no estar mirando la pantalla siempre.
- Poder compartir con la comunidad.
- Actualización por parte de los propios usuarios.

Figura 3.2: *Benchmarking* Waze



Interfaz

- Mapa tipo google map.
- Buscador.
- Menú desplegable.
- Iconografía y poco texto.

Funcionalidades

- Buscar sitios en mapa.
- Niveles de navegación dentro de una edificación identificando locales, etc.
- Mostrar información sobre las cosas dentro del sitio.
- Como llegar de un punto a otro dentro de una "edificación".
- Telemetry.
- Selección de idioma.
- Escaneo de QR.

Fortalezas y debilidades

- Niveles de navegación dentro de una misma estructura.
- Poder compartir.
- Poder obtener telemetry.
- Simple y claro.

Figura 3.3: *Benchmarking* Mapwize



My Disney Experience

Interfase

- Vista de mapa o lista.
- Menú principal debajo.
- Muestra la información de los lugares en tooltips y no en un menú desplegable como las otras.

Funcionalidades

- Buscar sitios en mapa.
- Comprar cosas en tiendas.
- Reservar tickets.

Fortalezas y debilidades

- Tiene una bienvenida con "First time help" donde te lleva de una forma muy amena por los pasos necesarios para que la app funcione de forma óptima. Como por ejemplo encender la localización o las notificaciones.
- Tiene mucha información pero un buen manejo de la misma.
- Observar las opciones en qué muestra los diferentes tipos de información tanto en los menús como en el mapa mismo.

Figura 3.4: *Benchmarking* My Disney Experience



¡appa!

Interfaz

- Notificación de nuevos descuentos.
- Newsfeed.
- Buscador.
- Menú principal debajo.
- Iconografía y poco texto.

Funcionalidades

- Explorar descuentos
- Mostrar información de descuentos por tienda.
- Hacer check-in. Anunciarse hacer reservas.
- Inbox con noticias.
- Agregar marcas y shoppings favoritos.

Fortalezas y debilidades

- Aplicación de descuentos y reservas.
- Incluye ofertas de parking gratis.
- No cuenta con geolocalización.

Figura 3.5: *Benchmarking* ¡appa!

3.4. Primera aproximación al producto

Para comprender mejor la solución, se presentan algunas pantallas de interacción del usuario cliente de la *app* que desea ubicar el centro comercial y una vez allí circula e interactúa con algunas diferentes posibilidades que brinda la *app* (el detalle completo se expone en el anexo 10.4)



Figura 3.6: Pantalla mapa *indoor*.

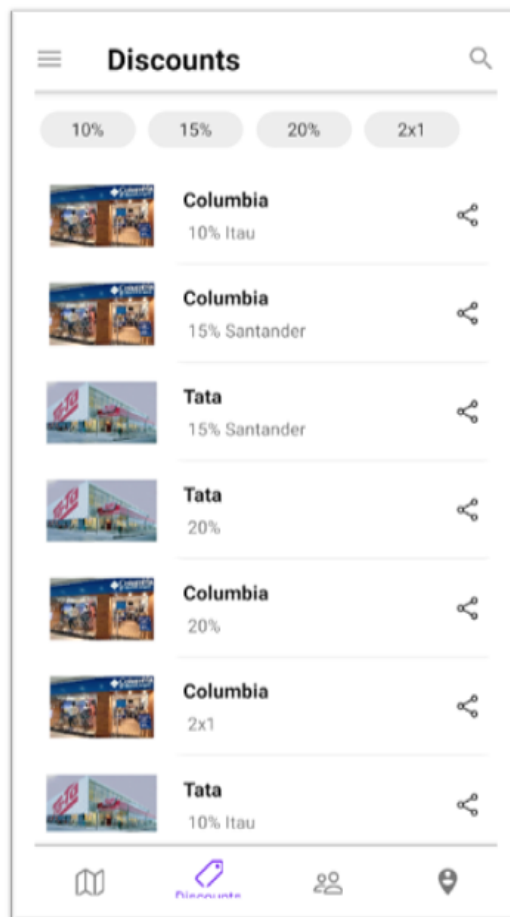


Figura 3.7: Pantalla de descuentos en función a preferencias.



Figura 3.8: Pantalla de descuentos en tiempo real en el mapa.

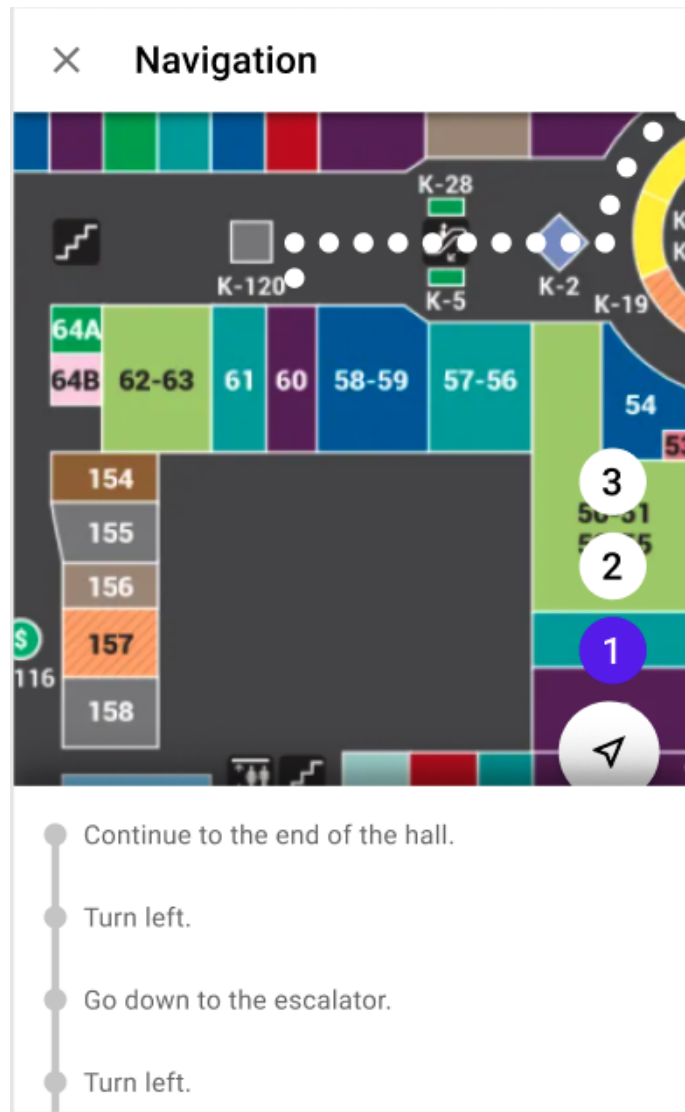


Figura 3.9: Pantalla de indicaciones para llegar a tienda.

3.5. Alcance

El cliente deja disponible para el equipo la librería llamada GeoFence encargada de realizar los cálculos de geolocalización, la cual se integra con la gran mayoría de *beacons* del mercado (ya sean propietarios, de terceros o de fabricación propia) para realizar los cálculos anteriores en los diferentes ambientes.

Se detallan a continuación las funcionalidades acordadas con el cliente, en donde se realizaron dos categorizaciones:

Mandatorias

Funcionalidades de carácter mandatorio, que deben de estar contempladas en su totalidad sobre la solución a implementar, las mismas son:

▪ *Community:*

- Lista de amigos.
- Agregar amigo.
- Invitar a usar la *app*.
- Invitar a un lugar.

▪ *Discounts:*

- Lista de descuentos con filtros.
- Buscador de descuentos.
- Encontrar descuentos en el mapa en tiempo real.
- Visualizar descuentos activos.
- Generación de QR para consumir descuento.
- Lectura de QR para efectivizar uso del descuento.

▪ *Explore:*

- Explorar mapa con tiendas.
- Buscador de tiendas.
- Hacer reseñas sobre tiendas.
- Agregar a favoritos una tienda.
- Obtención de información de la tienda.

- Obtención de descuentos de la tienda.
- **Login:**
 - Inicio de sesión con Google.
 - Inicio de sesión con Facebook.
- **QR:**
 - Scan de código QR para obtener información de las promociones.
- **User menu:**
 - Favoritos.
 - Configuración.
 - Descuentos compartidos.
 - Solicitudes de amistad pendientes.
 - Usuarios bloqueados.
 - Preferencias.
 - Ayudas de usuario.
 - Acerca de la *app*.
 - Cerrar sesión.

Deseadas

Funcionalidades que son deseadas por el cliente, que se van a empezar a realizar en caso de estar implementadas todas las funcionalidades del punto anterior, las mismas son:

- **Augmented reality:**
 - Realidad aumentada para visualizar productos o información de interés.
- **Descuentos:**
 - Indicaciones para llegar hasta tienda que ofrece el descuento.
- **Lugares:**
 - Invitar a un amigo a cierto lugar.
 - Compartir ubicación actual con un amigo.
- **RRSS:**
 - Posteo en redes sociales sobre descuentos.

3.6. Detalle sobre funcionalidades principales

Se realiza un detalle de las funcionalidades para los tres tipos de usuarios identificados en el capítulo anterior 2.4 - Tipos de usuarios y necesidades.

3.6.1. Usuario cliente

Acerca de la *app*

Tiene la posibilidad de ver información relevante sobre la aplicación.

Agregar a favoritos tienda

Tiene la posibilidad de poder a agregar sus tiendas de preferencia a su lista de favoritos.

Agregar amigo

Tiene la posibilidad de poder agregar como amigo a otro usuario que también utiliza la aplicación, mediante el envío de una solicitud de amistad.

Ayudas de usuario

Tiene la posibilidad de poder acceder a ciertas ayudas de usuario para evacuar de manera rápida dudas que puedan surgir con respecto a ciertos funcionamientos.

Buscador de descuentos

Tiene la posibilidad de utilizar un buscador para encontrar de manera rápida filtrando por letras sueltas o palabras ciertos descuentos.

Buscador de tiendas

Tiene la posibilidad de utilizar un buscador para encontrar de manera rápida filtrando por letras sueltas o palabras ciertas tiendas.

Cierre de sesión

Tiene la posibilidad de cerrar sesión con su cuenta.

Configuración

Tiene la posibilidad de realizar cambios sobre ciertas configuraciones como lo son sus preferencias, usuarios bloqueados, descuentos compartidos, solicitudes de amistad pendientes.

Encontrar descuentos en el mapa en tiempo real

Tiene la posibilidad de ser guiado en tiempo real desde su posición actual hasta la ubicación de la tienda que se encuentra ofreciendo la promoción/descuento.

Explorar mapa con tiendas

Tiene la posibilidad mediante un mapa *indoor* de poder explorar el mapa de los diferentes pisos de la gran superficie (*shopping*), con sus respectivas tiendas detalladas.

Favoritos

Tiene la posibilidad de poder agregar sus tiendas de preferencia a su lista de favoritos.

Generación de QR para consumir descuentos

Tiene la posibilidad de poder generar un código QR para poder hacer uso de la promoción/descuento que desea.

Hacer reseñas sobre tiendas

Tiene la posibilidad de poder realizar reseñas sobre las tiendas.

Inicio de sesión

Tiene la posibilidad de iniciar sesión mediante Google o Facebook.

Invitar a un lugar

Tiene la posibilidad de invitar a un amigo a una cierta ubicación dentro del *shopping*.

Invitar a usar la *app*

Tiene la posibilidad de invitar a sus amigos que no son usuarios de la aplicación, a que la utilicen.

Lista de amigos

Tiene la posibilidad de visualizar su lista de amigos.

Lista de descuentos con filtros

Tiene la posibilidad de visualizar los descuentos por filtros predeterminados por el sistema.

Obtención descuentos de la tienda

Tiene la posibilidad de obtener todos los descuentos ofrecidos por una tienda específica.

Obtención información de la tienda

Tiene la posibilidad de obtener información de relevancia sobre una tienda específica.

Scan código QR para obtener información de promociones

Tiene la posibilidad de escanear códigos QR de las tiendas para ver promociones/descuentos sobre ciertos productos específicos.

Visualizar descuentos activos

Tiene la posibilidad de visualizar todos los descuentos que tiene activos con los códigos de QR ya generados.

3.6.2. Usuario tienda

Las cuentas de este tipo de usuario son generadas por el administrador del sistema, quien también se encarga de brindar las credenciales de acceso. Este usuario es utilizado para realizar la administración sobre las tareas relacionadas con las tiendas que ofrecen promociones.

Acerca de la *app*

Tiene la posibilidad de ver información relevante sobre la aplicación.

Alta/Baja/Modificación promoción

Tiene la posibilidad de realizar alta, baja o modificación sobre promociones.

Cierre de sesión

Tiene la posibilidad de cerrar sesión con su cuenta.

Inicio de sesión

Tiene la posibilidad de ingresar con sus credenciales, brindadas por un administrador del sistema.

Lectura de QR para efectivizar uso del descuento

Tiene la posibilidad de realizar la lectura de los QR generados por los clientes.

Validación código QR descuento

Tiene la posibilidad de verificar la veracidad del código de QR correspondiente al descuento.

3.6.3. Usuario Administrador

Este usuario cuenta con las mismas funcionalidades que el usuario de la tienda, pero además puede realizar tareas de administración sobre el sistema como:

Alta/Baja/Modificación usuario tienda

Tiene la posibilidad de realizar alta, baja o modificación sobre tiendas.

Baja/Modificación usuario cliente

Tiene la posibilidad de realizar baja y modificación sobre usuario cliente.

Alta/Baja/Modificación entidades del sistema

Tiene la posibilidad de realizar alta, baja y modificación sobre entidades del sistema.

4. Ingeniería de Requerimientos

A continuación se muestran detalles de todo el proceso realizado para la recopilación y verificación de requerimientos tomando como base las funcionalidades acordadas con el cliente, según se describieron en el capítulo anterior. Se definen los requerimientos tanto funcionales como no funcionales que se acordó realizar para este proyecto.

4.1. Proceso de ingeniería de requerimientos

Para esta etapa, se realizó un estudio comparativo entre dos métodos existentes ya conocidos por los miembros del equipo para evaluar la viabilidad sobre cuál se adecuaba más a nuestro proyecto.

El primero en consideración fue el planteado por Pressman [1], con sus siguientes cinco etapas:

1. Reconocimiento del problema.
2. Evaluación y síntesis.
3. Modelización.
4. Especificación.
5. Revisión.

El segundo puesto a consideración es el planteado por Loucopoulos, con su planteo de tener en consideración tres aspectos fundamentales [2]:

1. Comprender el problema.
2. Describir formalmente el problema.
3. Obtener un acuerdo sobre la naturaleza del problema.

Lo cual nos lleva a simplificar el proceso a tres etapas con el fin de obtener en su totalidad los requerimientos:

1. Elicitación de requerimientos.
2. Especificación.
3. Validación.

Fue por decisión del equipo, la utilización del segundo método ya que el mismo encaja de mejor manera con nuestras necesidades y objetivos, además vamos a terminar obteniendo como parte del resultado el documento de especificación de requerimientos.

El procedimiento se inició mediante comunicaciones diarias con el cliente, en donde, a través del desarrollo de nuestra idea, se logró realizar una iteración en conjunto con algunas imposiciones del cliente, más el *benchmarking* realizado sobre soluciones que ofrecen propuestas similares de geolocalización.

Sobre estas últimas, se realizó un trabajo de investigación profundo en conjunto con la experta de *UI/UX* de Globant Verónica Varela a partir de la documentación disponible, acceso a aplicaciones, *reviews*, videos y otros archivos disponibles en línea.

Se realizó también una fuerte investigación sobre la integración de las tecnologías a utilizar, sobre todo, la integración de la librería provista por Globant para realizar la geolocalización.

Las tres etapas enumeradas en el segundo método fueron ejecutadas por parte del equipo mediante el trabajo que se detalla a continuación. Logrando obtener una certera especificación de los requerimientos.

4.2. Relevamiento

El relevamiento se realizó a partir de la idea base del equipo más las ideas por parte de cliente, con la finalidad de poder determinar los objetivos por cumplir.

Un punto a destacar durante esta etapa, fue que por factores externos al equipo (situación sanitaria), se generó un desafío a la hora de poder acordar reuniones por encontrarnos en pleno pico por la ola de COVID-19. De todos modos, se pudieron realizar diversas videollamadas con el cliente para poder realizar un buen intercambio de las ideas de ambas partes y poder llegar a buen puerto con el relevamiento de los requerimientos.

Una vez alcanzado un consenso sobre el relevamiento de requerimientos, realizamos entrevistas mediante la utilización de prototipos generados en conjunto con la experta de *UI/UX* con el fin de poder realizar una verificación y así comprobar que los requerimientos detectados anteriormente eran correctos; también el fin era ver si encontrábamos alguno no considerado anteriormente. (se brindan detalles del prototipo utilizado en el anexo 10.3).

4.3. Análisis de requerimientos

Dada por finalizada la etapa anterior, el equipo realizó su propio análisis en profundidad. Fueron realizadas videollamadas entre el equipo con la finalidad de poder realizar una evaluación en detalle sobre cada uno de los requerimientos relevados.

De esta forma logramos tener certeza sobre los factores relacionados como las características propias y restricciones que pudiesen afectar la etapa de desarrollo.

Obteniendo una clara realidad del problema, la solución y cuáles son las necesidades de cada tipo de usuario en específico. Mediante el apoyo de flujos realizados entre interacción de usuario con el sistema y las respuestas que el mismo debería devolver, a partir de esto se lograron terminar de pulir ciertos requerimientos no relevados de manera totalmente correcta anteriormente, siendo los mismos validados por parte del cliente.

Finalizado el análisis y una vez acordado que los requerimientos cumplieran con las siguientes características, pasamos con la etapa de especificación:

- No ambigüedad.
- Consistencia.
- Verificabilidad.
- Viabilidad.

4.4. Especificación de requerimientos

Una vez logrado tener un claro panorama sobre la totalidad del dominio, el equipo generó un documento de especificación de requerimientos (ESRE) el cual cuenta con las especificaciones de los requerimientos sobre el sistema, detallándose el cometido a nivel funcional pero sin abarcar de qué manera será realizado. (dirigirse al anexo 10.4 para ver el ESRE).

Finalizado el ESRE, se procedió a generar las historias de usuario en nuestro *product backlog* en función de cada requerimiento; esta definición nos sirvió para la gestión interna de nuestro desarrollo, generar entregables en función de cada *sprint* y lograr transparencia del trabajo realizado de cara al cliente.

4.5. Definición de requerimientos

Durante esta etapa, se realizó una planilla que luego fue ingresada en nuestro *product backlog*, con los siguientes ítems correspondiente a cada requerimiento:

- Título.
- Descripción.
- *Definition of ready.*
- *Definition of done.*
- Prioridad.
- Cantidad de *story points*.
- *Tags.*
- Riesgo.

4.6. Validación de requerimientos

Para realizar la validación de los requerimientos se realizaron prototipos en conjunto con la experta de *UI/UX* de Globant Verónica Varela con el fin de validar tanto con el cliente como con los potenciales usuarios; fue empleada adicionalmente la observación como método auxiliar mientras se realizaban las validaciones con los usuarios.

4.7. Product backlog

De las etapas anteriores logramos confeccionar nuestro *product backlog*, con un total de 186 *user stories*, en donde ciertas constituyen una *feature* y a su vez un conjunto de las anteriores pueden conformar una *epic*.

Para realizar el manejo de las *user stories* durante las diferentes iteraciones del proyecto, utilizamos la extensión Azure Boards de la plataforma Azure DevOps en

su formato de uso *Agile process* el cual termina empleando un tablero de *Kanban*.
[3] [4]

Dentro del tablero definimos los estados posibles para las tareas correspondientes a cada *user story* como:

- *New*
- *Backlog*
- *Blocked*
- *To Do*
- *In progress*
- *Review*
- *Done*

Lo cual nos permitió visualizar de manera rápida el estado correspondiente a cada *user story* asignada a los miembros del equipo.

Multigance Team

Backlog Analytics Epic/Feature Timeline + New Work Item View as Board Column Options

Order	Work Item Type	Title	State	Effort	Busin...	Value Area	Tags
1	Epic	EPIC - Registro y autenticación aplicación	New			Business	
	Feature	FEAT - Login	New			Business	
	User Story	US - Ingreso con google	Closed			Architectural	
	Task	Iniciar proyecto	Closed				
	Task	Investigar integración	Closed				
	Task	Implementar solución	Closed				
	Task	Investigar React-router	Closed				
	Task	Crear componentes iniciales	Closed				
	Task	Pair-programming con Alejandro	Closed				
	User Story	US - Pasar login con google a react-native para Android	Closed			Architectural	
	Task	Integración en App con login de Google compatible con RN	Closed				
	Task	Reemplazo de styling con elementos de RN por styled-components	Closed				
	User Story	US - Hacer login con Facebook y Twitter	Closed			Architectural	
	Task	Integración de login de Facebook en App	Closed				
	Task	Login con Twitter	Closed				
	Task	Pruebas e investigación de alternativas para implementación de Login de Twitter	Closed				
	Task	Migración de página de login a bare RN	Closed				
	Task	Bottom Tab Navigation	Closed				
	Task	Implementar Splash y Login/Signin navigation	Closed				
	Task	Cambiar color de status bar para simular full screen	Closed				
	Task	Linkar logout de drawer hacia splash screen	Closed				
	Task	Agregar iconos a las pantallas de navegación (Figma)	Closed				
	Task	Completar navegación entre diferentes logins	Closed				
	Task	Implementar landing screens	Closed				
	User Story	US - Terminar back login para redes social	Closed			Architectural	
	User Story	US - Lógica de login, clase de store relacionada, clase de API para endpoint	Closed			Architectural	
2	Epic	EPIC - Documentación entrega final	New			Business	

Figura 4.1: Ejemplo del *product backlog* para registro y autenticación en aplicación.

Taskboard Backlog Capacity Analytics + New Work Item Column Options

Sprint 9 Person: All

	New	Backlog	Blocked	To Do	In Progress	Review	Done
510 US - Gestion sprint Federico Barceló State: Closed							512 Tomar del .em versión de la app Unassigned State: Closed
522 Horas Federico Barce... State: Closed							
US - Backend Backend laravel arreglos varios							
US - Documentación Presentacion e informe avance							
US - Documentación Universitaria overleaf							
US - Detail empezar flujo de store							
US - Configuration agregar pantalla para ver my discounts							
US - Documentación Finalizar informe de avance							
US - Diagrama Modificaciones de diagrama de BD							
US - Figma Tercera iteración							
US - Backend Laravel Nova portal							
US - Community Terminar Community como esta en Figma							
US - Configuration agregar pantalla para ver shared discounts							

Figura 4.2: Tablero *Kanban* utilizado.

4.8. Requerimientos funcionales

A continuación se detallan los requerimientos funcionales pactados con el cliente como mandatorios, más algunos que se categorizan como deseados en caso de cumplir en totalidad con los anteriores.

Obligatorios:

Requerimiento Funcional - (RF) 1 - Explorar mapa: Como usuario *app mobile* quiero ser capaz de poder navegar en los mapas libremente.

RF2 - Buscador tiendas: Como usuario *app mobile* quiero ser capaz de poder buscar tiendas por nombre dentro del mapa.

RF3 - Hacer reseñas: Como usuario *app mobile* quiero ser capaz de poder dejar reseñas a una tienda.

RF4 - Agregar a favoritos: Como usuario *app mobile* quiero ser capaz de poder agregar una tienda a mis tiendas favoritas.

RF5 - Información de tienda: Como usuario *app mobile* quiero ser capaz de poder obtener la información relevante de una tienda.

RF6 - Descuentos de tienda: Como usuario *app mobile* quiero ser capaz de poder obtener los descuentos ofrecidos por una tienda.

RF7 - Compartir ubicación: Como usuario *app mobile* quiero ser capaz de poder compartir mi ubicación actual con algún amigo.

RF8 - Visualizar descuentos con filtros: Como usuario *app mobile* quiero ser capaz de poder visualizar descuentos filtrados.

RF9 - Buscar descuentos: Como usuario *app mobile* quiero ser capaz de buscar descuentos libremente.

RF10 - Visualizar descuentos en tiempo real: Como usuario *app mobile* quiero ser capaz de visualizar los descuentos en tiempo real en el mapa.

RF11 - Descuentos activos: Como usuario *app mobile* quiero ser capaz de visualizar los descuentos que tengo activados.

RF12 - Indicaciones para llegar a tienda que ofrece descuento: Como usuario *app mobile* quiero ser capaz de tener indicaciones para poder llegar a la tienda que ofrece el descuento de mi interés.

RF13 - Visualizar amigos: Como usuario *app mobile* quiero ser capaz de visualizar mi lista de amigos.

RF14 - Agregar amigo: Como usuario *app mobile* quiero ser capaz de agregar un usuario como mi amigo.

RF15 - Buscar usuario: Como usuario *app mobile* quiero ser capaz de buscar a otro usuario de la aplicación.

RF16 - Invitar a usar la app: Como usuario *app mobile* quiero ser capaz de poder invitar a un amigo no usuario de la aplicación a que la use.

RF17 - Generación de QR para descuento: Como usuario *app mobile* quiero ser capaz de poder generar un código QR para poder utilizar el descuento en la tienda que lo ofrece.

RF18 - Login: Como usuario *app mobile* o usuario *web* tienda quiero ser capaz de poder iniciar sesión en el sistema para poder utilizar la app.

RF19 - Configuración de usuario: Como usuario *app mobile* quiero ser capaz de poder realizar cambios en la configuración de mi perfil.

RF20 - Validación de QR activo: Como usuario *web* tienda quiero ser capaz de poder realizar validaciones sobre los QR para validar el descuento.

RF21 - Realizar Alta/baja/modificación promoción: Como usuario *web* tienda quiero ser capaz de poder realizar alta, baja o modificación de una promoción.

Deseados:

RF22 - AR: Como usuario quiero ser capaz de poder realizar visualizaciones mediante realidad aumentada de ciertos productos.

RF23 - RRSS: Como usuario quiero ser capaz de poder realizar posteos sobre los descuentos en diferentes redes sociales.

4.9. Requerimientos no funcionales

Requerimiento no funcional (RNF) 1 - Integración librería Geofence: Por imposición del cliente tenemos la restricción de integrar su librería (Geofence) para la geolocalización *outdoor* como *indoor*.

RNF2 - *App mobile* para iOS y Android: Por imposición del cliente tenemos la restricción de realizar el desarrollo de la aplicación tanto para dispositivos móviles iOS como Android.

RNF3 - API REST sobre Laravel: Por imposición del cliente tenemos la restricción de realizar nuestro *backend* de API REST sobre Laravel.

RNF4 - PostgreSQL y Redis: Por imposición del cliente tenemos la restricción de utilizar postgresQL como base de datos relacional y Redis para *caching*.

RNF5 - Idioma de codificación: Por imposición del cliente tenemos la restricción de codificar en idioma inglés.

RNF6 - Extensibilidad: Por imposición del cliente tenemos la restricción de que la adición de cualquier cambio sea sencilla y no involucre retrabajo.

RNF7 - Pruebas de código y cobertura: Por imposición del cliente tenemos la restricción de contar con automatización de pruebas tanto para *backend* como *frontend* y además lograr un nivel de cobertura por encima del 60 % total en función a líneas de código para el primero y por encima del 40 % para el segundo.

RNF8 - Flujos de acciones: Por imposición del cliente tenemos la restricción de tener flujos cortos sobre las funcionalidades del sistema.

RNF9 - Seguridad: Por imposición del cliente tenemos la restricción de contar con seguridad entre la comunicación de nuestro *frontend* y *backend* mediante autenticación por *token* más encriptación sobre datos sensibles almacenados en la base de datos.

RNF10 - *Backend* agnóstico: Por imposición del cliente tenemos la restricción de generar un *backend* que pueda ser utilizado sobre cualquier contexto en el que se quiera implementar la solución.

RNF11 - Uso de Material Design: Por imposición del cliente tenemos la restricción de utilizar Material Design para el desarrollo de la interfaz de la aplicación móvil.

5. Arquitectura de sistema

5.1. Descripción general de la arquitectura

La solución generada se encuentra conformada por tres componentes, como se muestra en el siguiente diagrama. En primer lugar, la capa de *frontend*, encargada de la parte visual y la interacción con los usuarios, desarrollada la *mobile app* en React Native y la *web app* apoyada en Laravel con HTML y CSS.

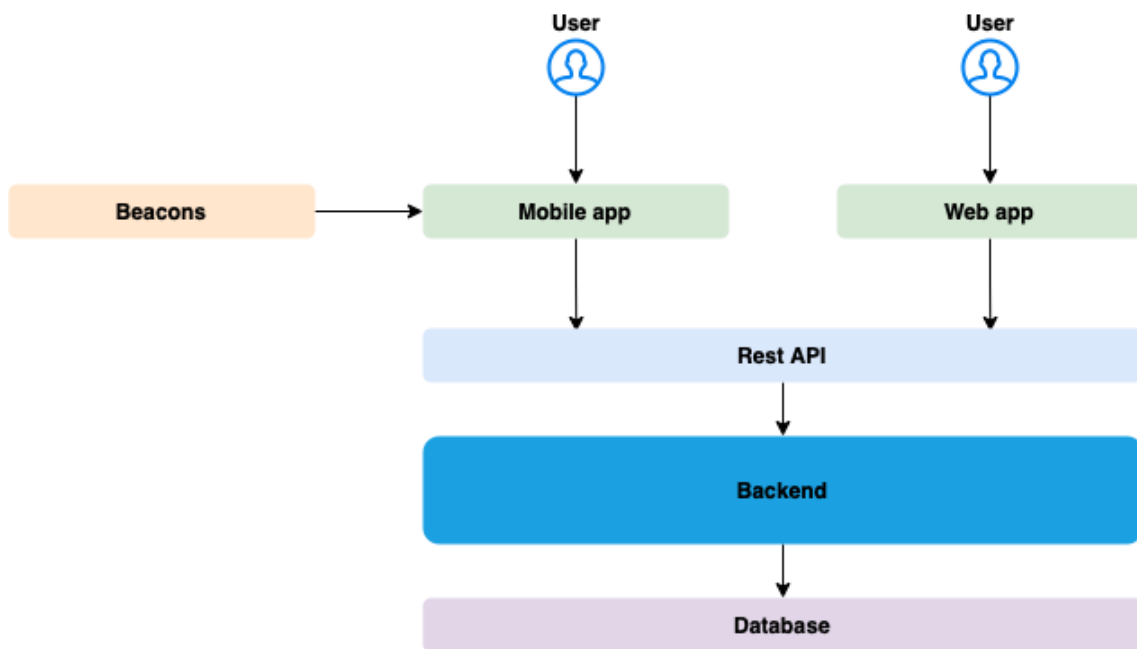


Figura 5.1: Diagrama general de arquitectura.

En la segunda capa, se encuentra el *backend* en Laravel, que se encuentra expuesto mediante una Rest API, encargada de nuestra lógica de negocio.

Por ultima capa, tenemos nuestra comunicación con nuestras estructuras de base de datos, la cual es accesible solamente por la capa anterior.

5.1.1. Diagrama de paquetes

En esta sección se muestra el diagrama de paquetes que muestra cómo se compone el *backend* del proyecto y sus distintas responsabilidades. El diagrama lo visualizamos en la figura 5.2, y sus responsabilidades en la tabla 5.1.

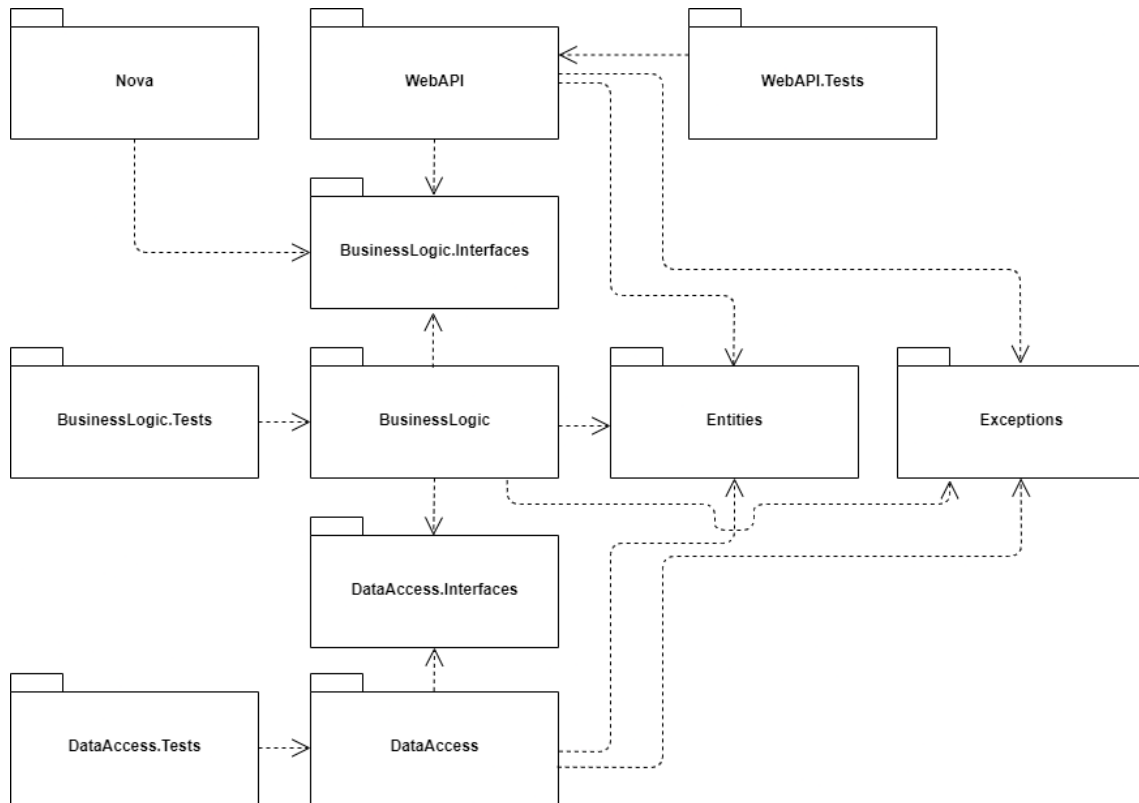


Figura 5.2: Diagrama de paquetes.

Paquete	Responsabilidad
Entities	Se encuentran todas las entidades del sistema.
WebAPI	Se encuentran los <i>endpoints</i> que se disponibilizan al <i>frontend</i> .
Nova	Se encuentran los distintos archivos para generar el <i>frontend web</i> del sistema.
BusinessLogic.Interfaces	Se encuentran las interfaces disponibilizadas para acceder a las funcionalidades del paquete BusinessLogic
BusinessLogic	Se encuentra toda la lógica del sistema.
Exceptions	Se encuentran las excepciones customizadas que se utilizan durante el manejo de errores.
DataAccess.Interfaces	Se encuentran las interfaces disponibilizadas para acceder a las funcionalidades del paquete DataAccess
DataAccess	Se encarga de la comunicación directa con la base de datos para persistir la información.
WebAPI.Tests	Se encuentran las pruebas unitarias del paquete WebAPI
BusinessLogic.Tests	Se encuentran las pruebas unitarias del paquete BusinessLogic
DataAccess.Tests	Se encuentran las pruebas unitarias del paquete DataAccess

Tabla 5.1: Tabla de responsabilidades.

5.1.2. Diagrama de componentes

En este diagrama visualizamos las dependencias entre los distintos componentes del sistema. Se observan los distintos archivos binarios que se generan luego de la compilación del proyecto.

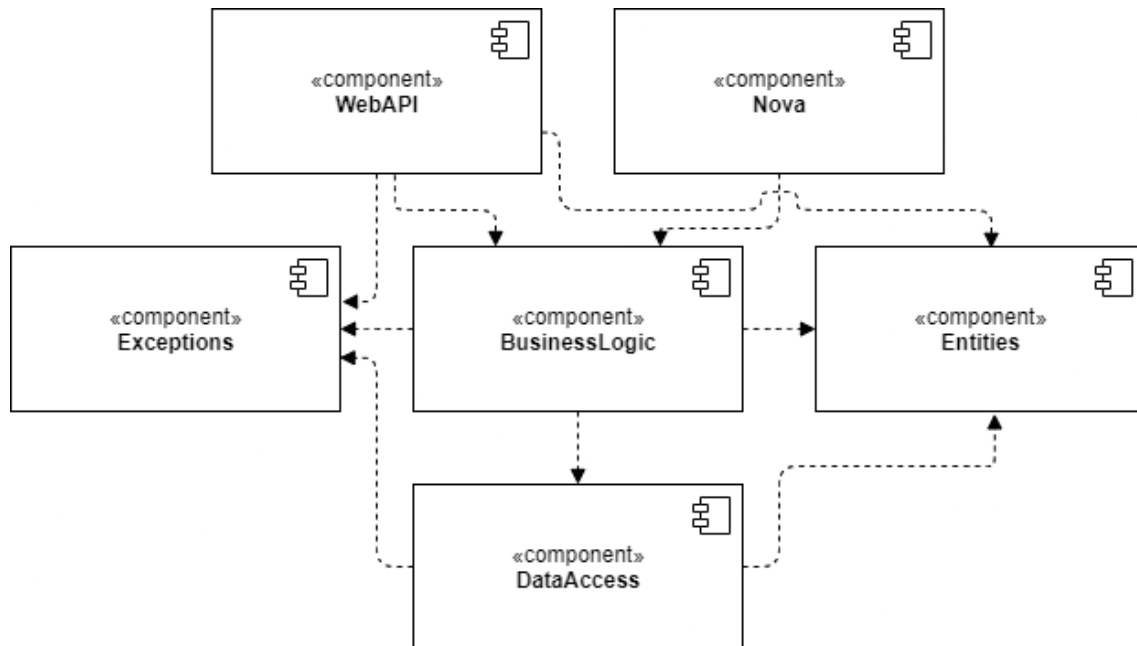


Figura 5.3: Diagrama de componentes.

5.1.3. Diagrama de entrega

En el siguiente diagrama podemos visualizar como se comunican los distintos nodos del sistema a nivel de *hardware* y cuáles archivos de *software* contienen.

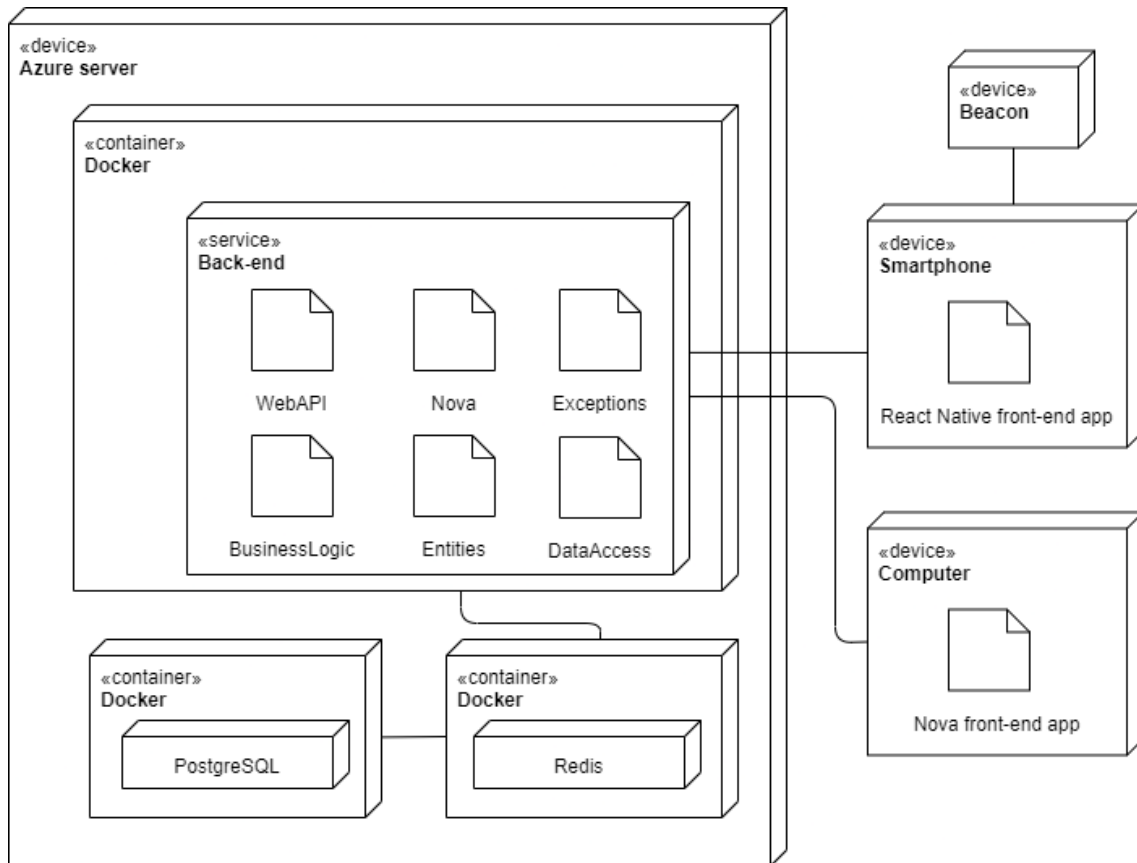


Figura 5.4: Diagrama de entrega.

5.1.4. Diagrama de flujo

En el siguiente se visualiza el flujo de la principal funcionalidad de la aplicación.

Cuando un usuario ingresa al centro comercial (*shopping center*) y es detectado por un *beacon* por la *app*, recibe una lista de descuentos en función de sus preferencias. Si el usuario se continúa desplazando por el centro comercial y la *app* detecta otro *beacon*, se podrían llegar a actualizar los descuentos que se muestran.

Una vez que activa un descuento de su interés, se le generará un código QR para ser utilizado en la tienda que ofrece el mismo.

Cuando el usuario se llega a la tienda para hacer efectivo el descuento, la tienda realiza un *scan* del QR para validar su veracidad y efectivizar el uso del descuento. (para ver los demás flujos de la aplicación, dirigirse al anexo 10.6).

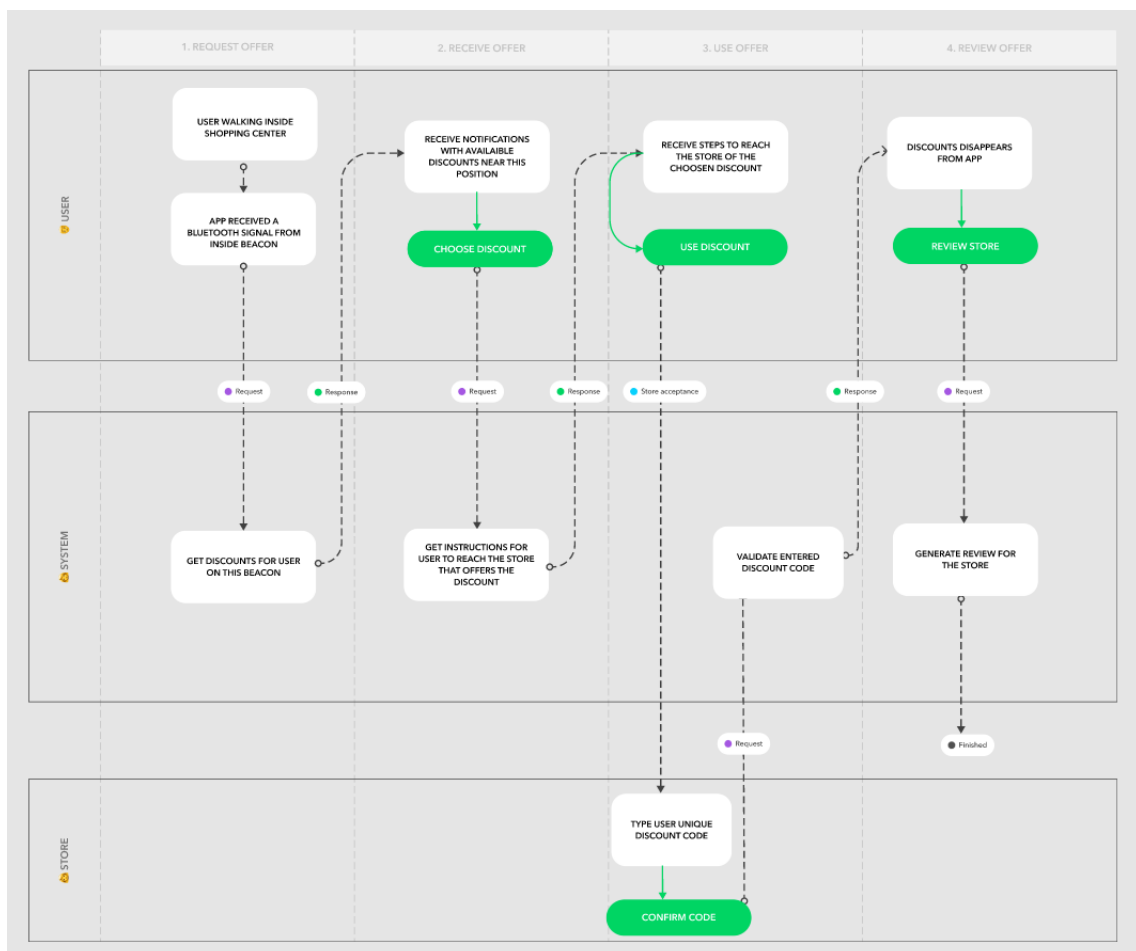


Figura 5.5: Flujo principal de la *app*.

5.1.5. Diagrama de despliegue

Se muestran en este diagrama, los componentes que integran la arquitectura de nuestra solución.

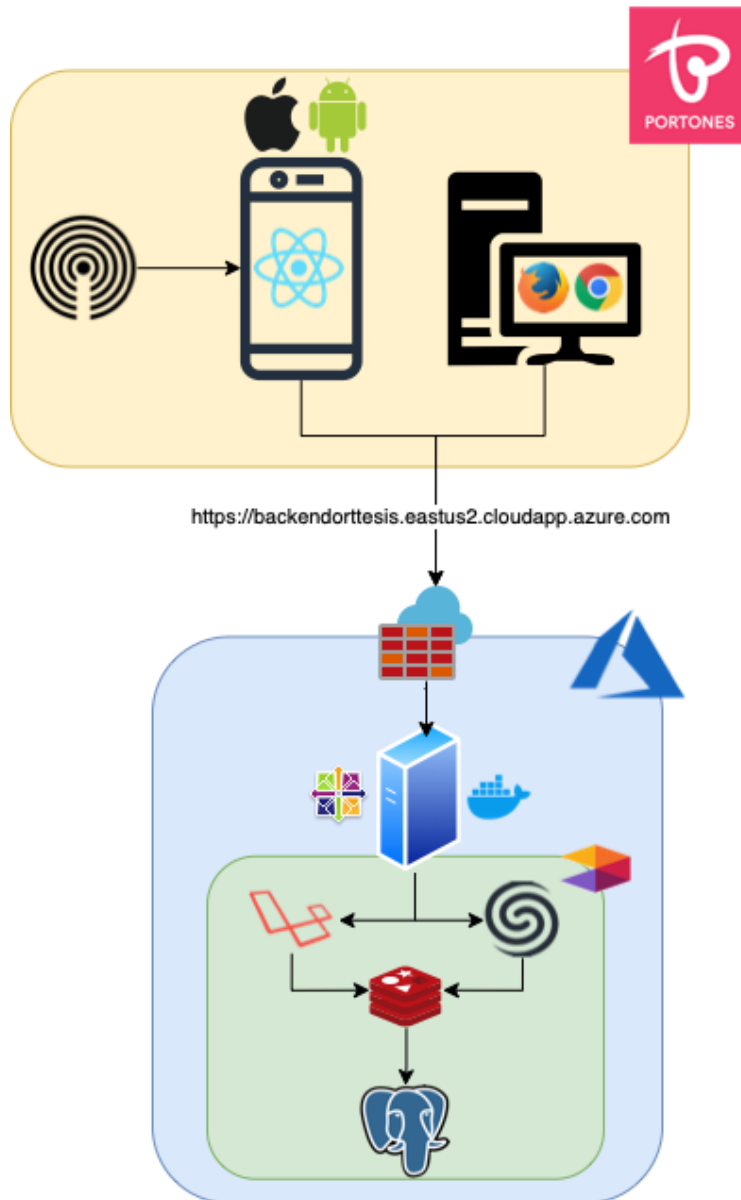


Figura 5.6: Diagrama de despliegue.

Componente	Nombre	Descripción
	iBeacon	Utilizados dentro del shopping para obtener ubicación de los usuarios
	iOS	Sistema operativo de dispositivo móvil en el cual corre nuestra aplicación.
	Android	Sistema operativo de dispositivo móvil en el cual corre nuestra aplicación.
	React Native	Tecnología utilizada por nuestra aplicación móvil.
	Firefox	Navegador web utilizado por nuestra aplicación web.
	Google Chrome	Navegador web utilizado por nuestra aplicación web.
	Portones shopping	Shopping center en donde realizamos la demo.
	Azure firewall	Firewall utilizado sobre nuestro ambiente cloud.

	Microsoft Azure	Proveedor cloud utilizado para hosting de nuestros recursos.
	Docker	Motor utilizado para despliegues de nuestros contenedores.
	CentOS	Sistema operativo utilizado por nuestro servidor con Docker.
	Laradock	Herramienta creada por la comunidad que flexibiliza el uso de Docker con PHP.
	Laravel	Tecnología utilizada por nuestro Backend REST API.
	Laravel Nova	Tecnología utilizada por nuestro portal de administración.
	Redis	Base de datos utilizada para hacer cache.
	Postgresql	Base de datos utilizada para persistir datos.

Figura 5.7: Componentes diagrama de despliegue.

5.1.6. Diagramas de (*Continuous Integration / Continuous Deployment*) (CI/CD)

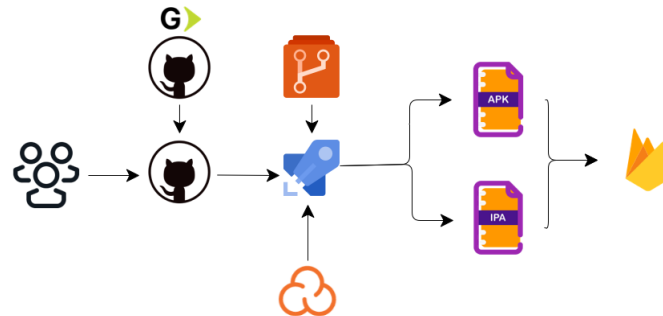


Figura 5.8: Diagrama de CI/CD aplicación móvil.

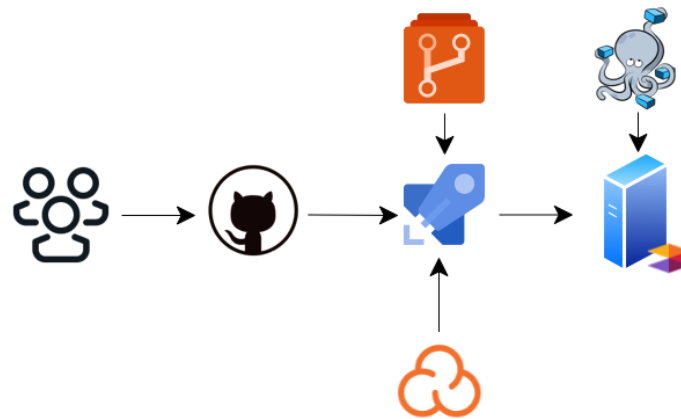


Figura 5.9: Diagrama de CI/CD Laradock.

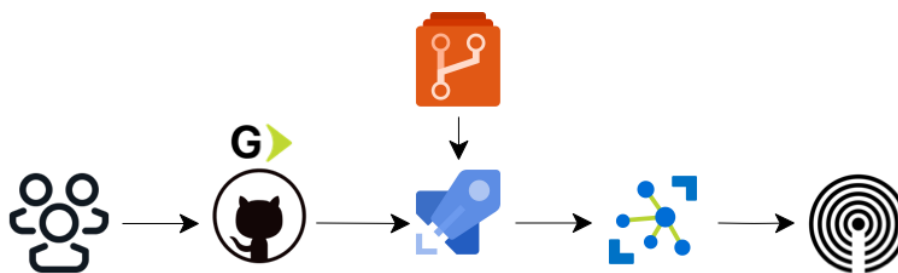


Figura 5.10: Diagrama de CI/CD Beacons.

Componente	Nombre	Descripción
	Equipo desarrollo	Equipo que integra cambios sobre el repositorio centralizado.
	GitHub corporativo	Repositorio utilizado para obtener la librería Geofence.
	GitHub	Repositorio utilizado para alojar el código de la aplicación.
	Azure Repos	Repositorio utilizada para alojar archivos de configuración para el CI/CD.
	Azure Pipelines	Herramienta utilizada para generar nuestros pipelines de CI/CD.
	SonarCloud	Herramienta utilizada para realizar el análisis del código.
	APK	Compilado de nuestra aplicación móvil para dispositivos Android.
	IPA	Compilado de nuestra aplicación móvil para dispositivos iOS.
	Firebase	Herramienta utilizada para distribuir la aplicación móvil a los dispositivos.
	Docker-compose	Herramienta utilizada par ejecutar multiples contenedores.
	Servidor con laradock	Servidor utilizado para desplegar nuestros contenedores sobre Laradock
	IoT Hub	Servicio utilizado para conectarse a nuestros dispositivos IoT (beacons)

Figura 5.11: Componentes diagramas de CI/CD

5.1.7. Diagrama de base de datos

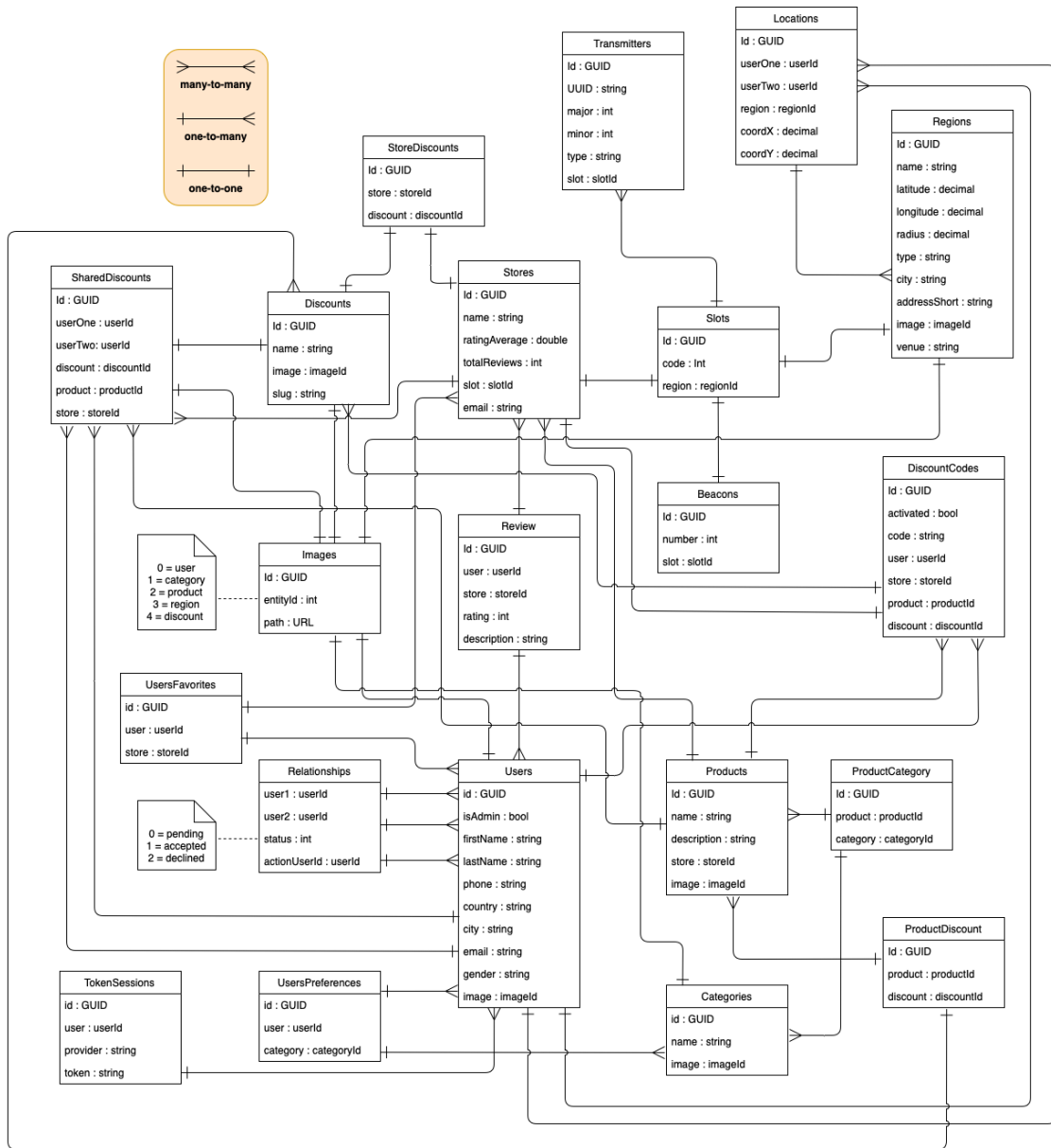


Figura 5.12: Diagrama de base de datos.

5.1.8. Módulos HM-10

Los HM-10 son módulos de *BLE* elegidos para configurarse y ser utilizados como nuestros *beacons* propios. (por información técnica sobre los módulos, ir al anexo 10.11).



Figura 5.13: Módulo HM-10 y pila CR2025.

Cuentan con compatibilidad con muchos dispositivos (desde iPhone 5 y superiores, Samsung Galaxy S4 y superiores), el funcionamiento a desempeñar por el mismo es configurable mediante comandos AT al igual que otros módulos *bluetooth* del mercado.

Fueron los elegidos por el equipo y cliente para utilizarse como *beacons*, en un principio se tuvo en consideración el uso de Estimote [5] por ser dispositivos de gran uso dentro del área, pero por temas de no disponibilidad en el mercado de la región y no querer incurrir en demoras de tener los dispositivos por hacer una compra en el exterior, optamos por la decisión de construir nuestros propios *beacons* a partir de estos módulos.

Son bastante accesibles de conseguir, además por contar con bajo consumo de energía por utilizar *Bluetooth Low Energy* (BLE), se pueden integrar en cualquier solución y son utilizados cuando se necesita comunicación *wireless* en distancias no mayores a 100 metros.

Como se evidencia a continuación, utilizamos unas carcasas provistas por el cliente para poder alojar nuestros módulos HM-10 y soldarles la pila CR2025 para que los mismos tengan su propia fuente de alimentación y no tener que estar constantemente conectados a una fuente de energía continua mediante un convertidor.



Figura 5.14: Carcasa para módulo HM-10 y pila CR2025.

Las mismas fueron identificadas con un código QR en el dorso, el cual utilizaremos para poder identificar de manera rápida a qué *beacon* nos estamos refiriendo sin tener que desmontarlo de la carcasa.



Figura 5.15: Dorso de carcasa armada con módulo HM-10 y pila CR2025.

Obteniendo como resultado, nuestros propios dispositivos *beacons* para utilizar en entornos *indoor*.



Figura 5.16: Frente de carcasa armada con módulo HM-10 y pila CR2025.

5.1.9. Librería Geofence

Esta es la librería provista por el cliente la cual realiza todo el cálculo de localización tanto *indoor* como *outdoor*.

Al ser código propietario del cliente y tener que conservar detalles sobre su implementación, no tuvimos acceso a ningún repositorio para poder visualizar con detalles su implementación. Por lo anterior, nos disponibilizaron un compilado de la aplicación en formato de paquete por poder importar en nuestra solución y nos brindan las funciones detalladas que podemos consumirla.

La librería funciona de manera correcta y la precisión sobre los cálculos para obtener la ubicación en los entornos es muy exacta, ya que tuvimos sesiones con los ingenieros de la misma en donde nos mostraron su funcionamiento, con el fin de comprender mejor su arquitectura y poder sacarnos todas las dudas sobre de qué manera íbamos a utilizar la misma.

Por motivos de confidencialidad con el producto del cliente, no tuvimos la autorización para grabar estas sesiones, pero si tuvimos flexibilidad por parte de los ingenieros para pautar todas las sesiones que nos fuesen necesarias y acceso al canal de Slack para realizar consultas puntuales.

5.2. Atributos de calidad

Por mutuo acuerdo entre el equipo y el cliente acordamos de manejar los atributos de calidad regidos por la norma ISO/IEC 25010[6], los cuales se detallan a continuación:

5.2.1. Adecuación funcional

Característica que representa la capacidad del producto de proporcionar funcionalidades que satisfacen las necesidades declaradas e implícitas cuando el mismo es usado en situaciones específicas.

5.2.2. Eficiencia

Una de las características en la que se quiso tener mayor énfasis fue la *performance*, ya que entendemos como algo vital al tratarse de una aplicación en la cual su *core* se centra en el uso de la geolocalización, la misma debe tener un gran nivel de precisión y que su cálculo sobre la ubicación de los usuarios sea rápida.

Llevándonos a tener que realizar una integración con la librería de Globant y nuestra aplicación de manera que ambas puedan convivir en una única solución y su funcionamiento en sea consistente, exacto, rápido y eficaz, al estar integrando dos tecnologías poco conocidas por el equipo.

Sobre el *frontend*, al optar por el uso de React Native, el funcionamiento de su compilado llega a ser bastante exacto a como si estuviéramos desarrollando totalmente nativo para dispositivos Android o iOS.

Esto se debe a que este *framework* se encuentra optimizado para dispositivos móviles, haciéndolo más rápido y con mayor *performance* por utilizar como principal recurso la *Graphics Processing Unit* (GPU) y no la *Central Processing Unit* (CPU) como otras tecnologías de compilación híbrida que se encuentran en el mercado.

A nivel del *backend* hicimos uso de *caching* con el objetivo de poder devolver de forma rápida la información sin tener que procesar la acción nuevamente. Para esto nos apoyamos en dos tipos de *caching*, estos son *Config caching* y *Routes caching*.

Para que el *caching* fuera eficiente decidimos utilizar Redis, que se destaca en gráficas de *performance benchmarking* y posee información relacionada en la documentación oficial de Laravel.

5.2.3. Compatibilidad

Capacidad de poder intercambiar información o llevar a cabo funciones cuando se comparte un mismo entorno de *hardware* o *software* con otros sistemas.

5.2.4. Usabilidad

Este fue uno de los atributos en el cual decidimos dedicar mayor esfuerzo ya que consideramos que la capacidad sobre el producto para ser entendido, aprendido, usado y resultar atractivo, lo cual es vital para el usuario en cualquier tipo de aplicación.

Además de recibir apoyo por parte de la experta del cliente Verónica Varela quien volcó toda su *expertise* durante las etapas tempranas en el diseño de las interfaces, el equipo hizo fuerte énfasis en las heurísticas de Nielsen, detalladas en el apartado de calidad en el capítulo de gestión de la calidad 10.8.

5.2.5. Fiabilidad/confiabilidad

Este atributo se refiere a la capacidad del sistema o componente para desempeñar las funciones especificadas, cuando es usado bajo ciertas condiciones y en un periodo de tiempo determinado.

Para lograr este cometido, intentamos tener un sistema que se encuentre siempre operativo y accesible cuando se lo requiera, manejando tanto en *frontend* y *backend* respaldos para poder hacer *rollback* en caso de que alguna falla (ya sea tanto de *hardware* como de *software*) deje no disponible alguno de estos componentes de nuestra solución.

5.2.6. Seguridad

Este atributo se centra en la capacidad del sistema para brindar protección de la información y datos de manera que los individuos no autorizados, no puedan modificarlos o consultarlos.

Nuestro sistema de *backend* implementa autenticación mediante *tokens*. Una vez que se inició sesión, se realiza la creación de un nuevo *token* utilizado para realizar las llamadas a nuestra REST API, válido por 1 día.

También se hizo configuración sobre la comunicación hacia nuestro servidor de *backend*, que la misma sea mediante tráfico *Hypertext Transfer Protocol Secure*

(HTTPS) y además mediante la utilización de un certificado *Secure Sockets Layer* (SSL), con el fin de agregar una capa más de seguridad sobre las comunicaciones hacia nuestro servidor.

Las tareas de administración sobre el servidor se realizan mediante conexión *Secure Shell* (SSH) con autenticación por par de llaves pública/privada.

5.2.7. Mantenibilidad

Durante todo el transcurso de la fase de desarrollo, se tuvo siempre en cuenta la características inherentes a la mantenibilidad, con el fin de tener un código limpio al estar trabajando en un proyecto para un cliente, buscando apegarnos en todo momento a las buenas prácticas nombradas en *Clean Code*[7] como:

- Clases no mayores a 200 líneas de código.
- Implementación de manejo de excepciones propias.
- Funciones con nombres descriptivos, con tendencia a tener cero parámetros.
- Nombres magníficos, que revelen su intención, para saber de manera ágil cuál es el cometido/responsabilidad de esa función.
- No tener comentarios en el código, ya que el mismo tiene que ser posible de explicarse sin la necesidad de información extra.
- Métodos pequeños y cohesivos.

En todo momento se trató de respetar el principio de diseño *Open Closed Principle* (OCP), intentando siempre mantener elementos con alta cohesión y el menor acoplamiento posible.

5.2.8. Portabilidad

Un punto importante a tener en cuenta fue la portabilidad, el cual implica la capacidad de que cualquier componente de nuestra solución puede ser transferido de manera efectiva y eficiente de un entorno de *hardware*, *software*, operacional o de utilización a otro.

Sobre el *frontend* al utilizar una tecnología como React Native nos brinda la posibilidad de compilación tanto para dispositivos con sistemas operativos Android como iOS.

Al utilizar Laradock con Docker para nuestro *backend* generado en PHP, logramos independencia de cualquier plataforma de *hosting* y sobre todo del sistema operativo para poder dejar disponible y accesible nuestros servicios, solamente con contar con un equipo con el motor de docker instalado o algún servicio *cloud* que soporte la ejecución de Docker, podemos disponibilizar a nuestros contenedores para desplegar nuestro ambiente.

También al utilizar Laravel con sus funcionalidades de *querybuilder* y *Object-relational mapping* (ORM), logramos independencia de las bases de datos a utilizar, además de las mismas también encontrarse sobre contenedores, podemos intercambiar de manera fácil los conectores sobre qué base de datos queremos emplear en nuestra solución y Laravel se encarga de configurar las entidades y popular las mismas por nosotros.

6. Desarrollo

6.1. *Frontend*

6.1.1. React Native

React Native (de aquí en más RN) es un *framework* y librería Javascript desarrollada originalmente por Facebook Inc. y actualmente mantenido por la comunidad de software libre. Cuenta con millones de desarrolladores alrededor del mundo y ha sido utilizada en el pasado para desarrollar aplicaciones populares como Facebook, Instagram y Airbnb. RN utiliza un esquema declarativo para luego convertirlo en sus respectivos componentes, utilizando el motor JavaScript de la plataforma en que se está ejecutando la aplicación. Los componentes de React pueden consistir de otros componentes o de primitivas. Una de las ventajas de RN son la reutilización de código el cual puede ser ejecutado tanto en Android como en iOS.

6.1.2. Lenguaje seleccionado

Se evaluó la codificación de la aplicación en dos tecnologías, TypeScript y Javascript. Debido a restricciones de tiempo del proyecto, se optó por JavaScript por su similitud con lenguajes previamente utilizados por quienes estarían desarrollando el *frontend*. TypeScript generaría dificultades adicionales como una mayor curva de aprendizaje, las cuales no aportarían ventajas suficientes como para justificar su implantación. Una de sus potenciales ventajas podría haber sido la validación de *props* que son objetos como funciones, *arrays* o cualquier otro tipo de información, enviadas de funciones padre a funciones hijo. Ya que JavaScript es considerado un lenguaje débilmente tipado, se debe contrarrestar la falta de dicha funcionalidad. Se utilizó una librería llamada PropTypes [8] la cual permite definir interfaces para los componentes de forma de que se validen los tipos de *props* enviados a un componente. Esto minimiza la cantidad de errores del sistema por incompatibilidad de tipos.

6.1.3. Migración de *JavaScript Bundler* de App

Tempranamente se detectaron dificultades debido a la ejecución en emuladores, dependencia de librerías al utilizar Expo y al momento de realizar *builds* por lo que se optó por migrar el *bundler* de la aplicación a Metro. Metro permitió acelerar el desarrollo y resolvió diferentes problemáticas relacionadas a compatibilidad de librerías, comportamiento no deseado de la aplicación y mejor desempeño del emulador. En retrospectiva, el cambio fue una decisión acertada y redujo la probabilidad de amenazas al proyecto.

6.1.4. Librería de Geolocalización

Uno de los mayores desafíos técnicos ha sido la integración en el sistema tanto a nivel de *backend* como de *frontend* de la librería provista por el cliente. Particularmente en el *frontend* se debió recurrir a diferentes mecanismos para articular el funcionamiento de la librería con RN. Escrita parcialmente en Swift (manejo iOS) y Kotlin (manejo Android), se debió realizar un análisis previo para definir la integración con la aplicación. Se terminó optando por un *bridge* el cual actúa como nexo para la obtención de los datos geoespaciales y así, ser consumidos por la aplicación escrita en RN y lograr un funcionamiento lo más fluido posible por estar integrando dos tecnologías diferentes.

Otro de los desafíos técnicos fue la definición del formato de los datos de geolocalización para la comunicación entre la librería, el *backend* y el *frontend*. Se realizaron pruebas y se optó por polígonos. Los mismos tienen como ventaja principal la visualización gráfica de los *fences* o vallas en que los usuarios se encuentran.

En cuanto al funcionamiento interno de la librería de geolocalización no es posible profundizar en detalles debido a restricciones de propiedad intelectual y de información confidencial.

6.1.5. Paradigma de programación

Se desarrolló la aplicación utilizando componentes funcionales y *arrow functions*. También llamados *Hooks*, los componentes funcionales fueron introducidos en la versión de React 16.7.0 y proveen una serie de beneficios y optimizaciones frente a componentes de clases. Por otra parte, los componentes funcionales permiten lograr la misma funcionalidad que componentes de clases pero utilizando menor cantidad

de líneas de código.

6.1.6. Repositorio de la aplicación

Para el el repositorio interno de la aplicación se utilizó React Redux, en conjunto con Redux Thunk. Redux permite almacenar en memoria y centralizar los datos que maneja la aplicación, para así poder obtener información desde y hacia diferentes componentes. La alternativa a ese flujo de comunicación entre componentes es utilizar *props*, desde componentes padre a hijos. Redux permite la comunicación, en casos en que los componentes no tienen relación directa -por pertenecer a secciones de la aplicación que no se encuentran directamente relacionadas, lo cual facilita el desarrollo y reduce la cantidad de código requerido para resolver flujos de datos.

Los *thunks* son *middleware* que permiten la comunicación entre la aplicación y Redux. Redux Thunk permite retrasar el lanzamiento (*dispatch*) de acciones, o hacer *dispatch* si alguna condición es alcanzada. Redux Thunk reduce la complejidad en el manejo de efectos secundarios como llamadas asíncronas, las cuales no ocurren instantáneamente.

Para la persistencia en memoria no volátil se utilizó Redux Persist, la cual es una de las alternativas de mayor popularidad y permite mantener el estado de la aplicación al cerrarla. Esto es de relevancia para el almacenamiento de información como *login* y así amenizar la experiencia de los usuarios.

Redux cuenta con alto grado de personalización y diferentes combinaciones para lograr la arquitectura deseada. El patrón para el manejo de datos escogido fue el de *action, reducer*. Separando *slices* por entidades o por pantalla.

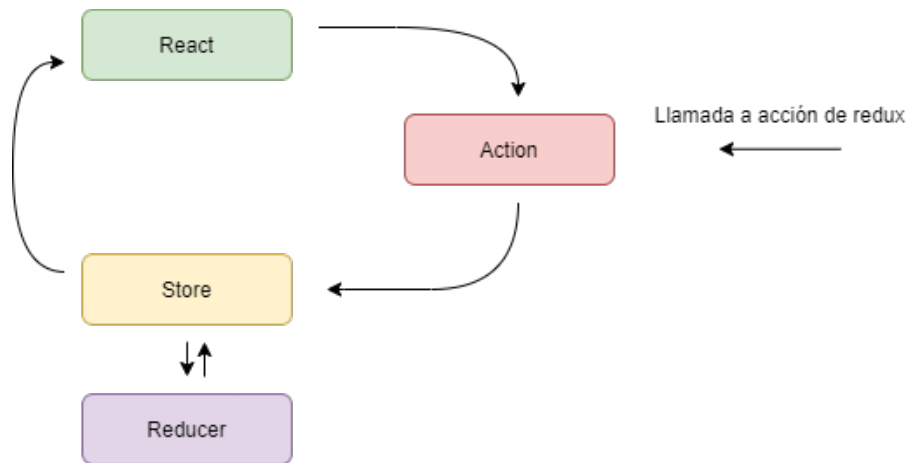


Figura 6.1: Patrón de una sección o *slice* de Redux.

6.1.7. Estados Locales vs React Redux

Uno de los beneficios de utilizar RN es la posibilidad de mantener el estado de elementos de la interfaz tales como efectos, dentro de los componentes. RN facilita un *hook* o función, para lograr ese comportamiento. Esto también se puede lograr por medio de React Redux, en un repositorio centralizado. El criterio para decidir dónde mantener cada estado se basó en recomendaciones de expertos en mantener estados en forma local en la medida de lo posible, para no reducir la *performance* de la aplicación.

6.1.8. React Navigation

React Navigation es una de las librerías de navegación con mayor popularidad en la actualidad. Cuenta con adecuada documentación y permite obtener variados flujos que fueron delineados en Figma por el cliente. La combinación de los diferentes componentes disponibilizados en React Navigation permiten lograr la experiencia de usuario deseada ya que incluye elementos los cuales son estándar en cualquier aplicación *mobile*. Por otra parte, cuenta con alto grado de personalización de efectos y animaciones las cuales enriquecen la percepción del producto por parte del usuario.

La librería cuenta con optimizaciones por defecto que además se pueden complementar con otras por parte del desarrollador. Una de ellas es la posibilidad de pausar la ejecución de pantallas que no están enfocadas por el usuario.

Se utilizaron lineamientos de Material Design para el diseño de la interfaz gráfica de la aplicación. Para la iconografía se utilizó la librería Vector Icons que incluye íconos Material Design desarrollados por Google.

6.1.9. Nuevas librerías y paquetes

Al momento de agregar nuevas librerías a la aplicación se realizó un análisis previo de viabilidad, donde se evaluaron parámetros como popularidad, qué tan actualizada estaba al momento de instalación, posibilidad de resolver la funcionalidad procurada sin necesitar instalación adicional así como también posibles incompatibilidades con el *stack* utilizado. Para asegurar el correcto funcionamiento de las librerías a integrar se solicitaba a los integrantes con ambientes Mac realizar pruebas pertinentes.

6.1.10. Autolinking

Autolink es una funcionalidad (disponible a partir de la versión 0.60 de React Native) la cual simplifica el proceso de instalación de nuevas dependencias, permite al proyecto descubrir y utilizar módulos nativos y *view managers* provistos por librerías. Se realizaron esfuerzos constantes durante el desarrollo para mantener sólo dependencias por medio de *autolink* para mantener la estabilidad de la aplicación.

6.1.11. Logo y Figma

Se optó por delegar la gestión de *marketing*, definición de nombre y diseño de logo de la aplicación al cliente. Esta decisión fue de común acuerdo entre el equipo y el cliente para poder enfocarse en desarrollo y gestión del proyecto y no incurrir en retrasos.

6.2. Backend

6.2.1. Laravel

Laravel es un *framework* de código abierto para PHP que permite el desarrollo rápido. Su principal objetivo es permitir el desarrollo rápido y estructurado permitiendo concentrarnos en la lógica mientras se encarga de los detalles. Laravel provee *features* poderosas así como inyección de dependencias, colas, tareas, *tests* unitarios y de integración, entre otros.

6.2.2. Lenguaje seleccionado

Más allá de que el *framework* sea de los más utilizados y aceptados por la comunidad, el equipo cuenta con gran experiencia en PHP junto a Laravel, lo que permitió tener el *backend* listo antes que las demás partes del sistema, esto trajo oportuni-

des como la de que el cliente la utilizara para otros contextos, permitiendo agregar valor más allá de la propuesta inicial.

6.2.3. Composer

Laravel utiliza Composer para administrar las dependencias, algo que la mayoría de las aplicaciones de PHP realizan hoy en día, lo cual facilita ya que no existe un cambio de contexto para esta administración dentro del lenguaje.

6.2.4. Documentación

La documentación es bastante intuitiva y permite encontrar fácilmente la mayoría de la información que se está buscando. Permite comenzar y profundizar los conocimientos sobre todos los recursos que disponibiliza el *framework*.

6.2.5. Sistema de rutas

Laravel permite trabajar con el concepto de rutas, permitiendo el mapeo de las URLs utilizadas por el cliente a acciones dentro del sistema. Esto nos permitió apoyarnos en REST para el desarrollo de la API.

6.2.6. Blade

Blade es un compilador de *templates*, tiene como principal objetivo reducir la cantidad de código PHP en los distintos archivos HTML y aumentar el reuso de las mismas, para esto disponibiliza una serie de directivas que se utilizan según la necesidad de la página. Blade tiene como beneficio el uso de herencia de secciones, lo cual permite encapsular partes de códigos para el reuso.

6.2.7. Eloquent

Eloquent es el ORM *default* de Laravel, aplica el patrón de diseño ActiveRecord en donde todas las tablas de la base de datos son representadas en el código con una clase de tipo Modelo, que es utilizado para la interacción con la tabla. Los Modelos permiten consultar, leer, escribir y actualizar información de las tablas. La gran ventaja de utilizar Eloquent es la no necesidad de mezclar código SQL con PHP.

6.2.8. QueryBuilder

Asimismo, si se desea escribir código SQL y no utilizar la abstracción que Eloquent nos brinda, es posible utilizar QueryBuilder para construir y ejecutar la con-

sulta en la base de datos.

6.2.9. Artisan

Artisan es una interfaz de línea de comandos que deja a disposición varios comandos para facilitar el desarrollo de la aplicación, también permite crear comandos personalizados.

6.2.10. Laravel Socialite

Socialite proporciona una interfaz fluida y expresiva para la autenticación *OAuth* con Facebook y Google, entre otros. Prácticamente lo único que necesitamos hacer es obtener las credenciales API del proveedor y configurarlas de manera sencilla.

6.2.11. Laravel Nova

Nova es un panel administrativo que se integra dentro del ecosistema de Laravel y está pensando para que sea sencillo de utilizar y modificar. Permite tanto el manejo de las distintas entidades del sistema como generar distintos *dashboards* a partir de la información deseada.

6.2.12. Laradock

Laradock es un entorno de desarrollo para PHP completo. Soporta también una gran variedad de servicios comunes, gran parte preconfigurados para facilitar tener un entorno funcional con PHP con base de datos, *web servers*, balanceadores de carga, manejadores de base de datos, entre otros.

La misma herramienta fue creada por la comunidad de Laravel para poder utilizar docker sobre un entorno de PHP de manera más flexible.

6.2.13. Postman

Se utilizó Postman como herramienta para realizar las distintas pruebas de integración sobre todas las rutas que dejó disponible el *backend*. Postman permite la creación y posibilidad de compartir colecciones entre el equipo de desarrollo y también manejar variables a distintos niveles del ambiente. También permite realizar pruebas y ejecutar acciones según el resultado de las ejecuciones.

6.3. Beacons

Se realizó por parte del equipo el desarrollo sobre los dispositivos HM-10 detallados en el capítulo anterior, para que los mismos funcionen como nuestros *beacons*.

Dicho desarrollo incluye que los dispositivos pasen a realizar de manera permanente un *broadcast* de la señal *bluetooth*, para comunicarse con nuestra aplicación móvil y de esta manera interactuar con ella para poder realizar las acciones que queremos en función de la proximidad sobre ciertos puntos estratégicos, en nuestro caso poder realizar ofrecimiento de los descuentos dentro del *shopping center* cuando pasan cerca de determinadas tiendas.

De la manera que empleamos este desarrollo, podemos posicionar los *beacons* sin la necesidad de estar programándolos para que cumplan su función o tenerlos siempre conectados, ya que mediante la pila que los mismos tienen y contar con la tecnología *BLE*, pueden estar días realizando su función.

Nuestros *beacons* tienen la configuración de los siguientes parámetros para que nos sean de utilidad:

- *UUID*: identificador de fabricante de *hardware*, a partir de este *UUID* los dispositivos móviles deciden si se pueden comunicar con el *beacon*.
- *mayor* y *minor*: par ordenado utilizado para la identificación de nuestros *beacons*.

Lo anterior más el *broadcast* de *bluetooth*, logramos obtener en un analizador de señal lo siguiente, en donde tuvimos nuestra primera interacción con nuestros *beacons* fabricados:

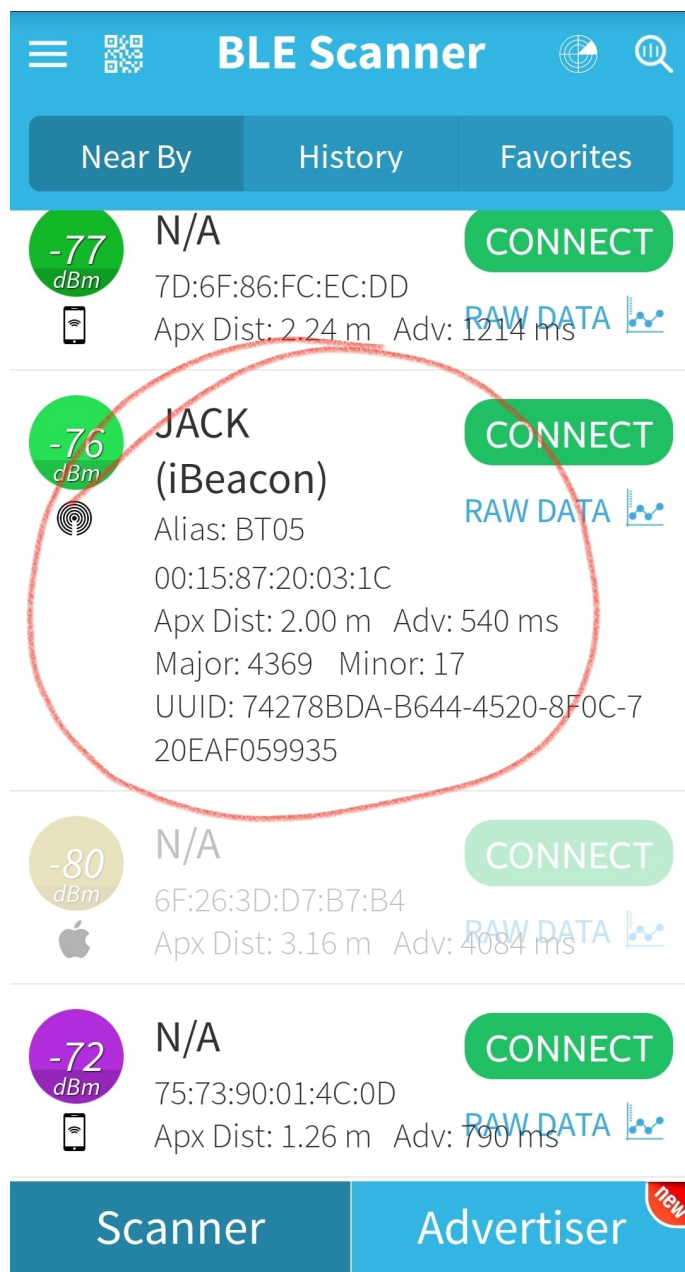


Figura 6.2: *iBeacon* siendo detectado por aplicación de scanneo.

La configuración realizada sobre nuestros módulos HM-10 para que pasen a ser *iBeacon* (utilizados por dispositivos iOS) la realizamos utilizando Arduino IDE mediante la conexión al controlador (por más información sobre el controlador y su conexión, leer anexo 10.11).

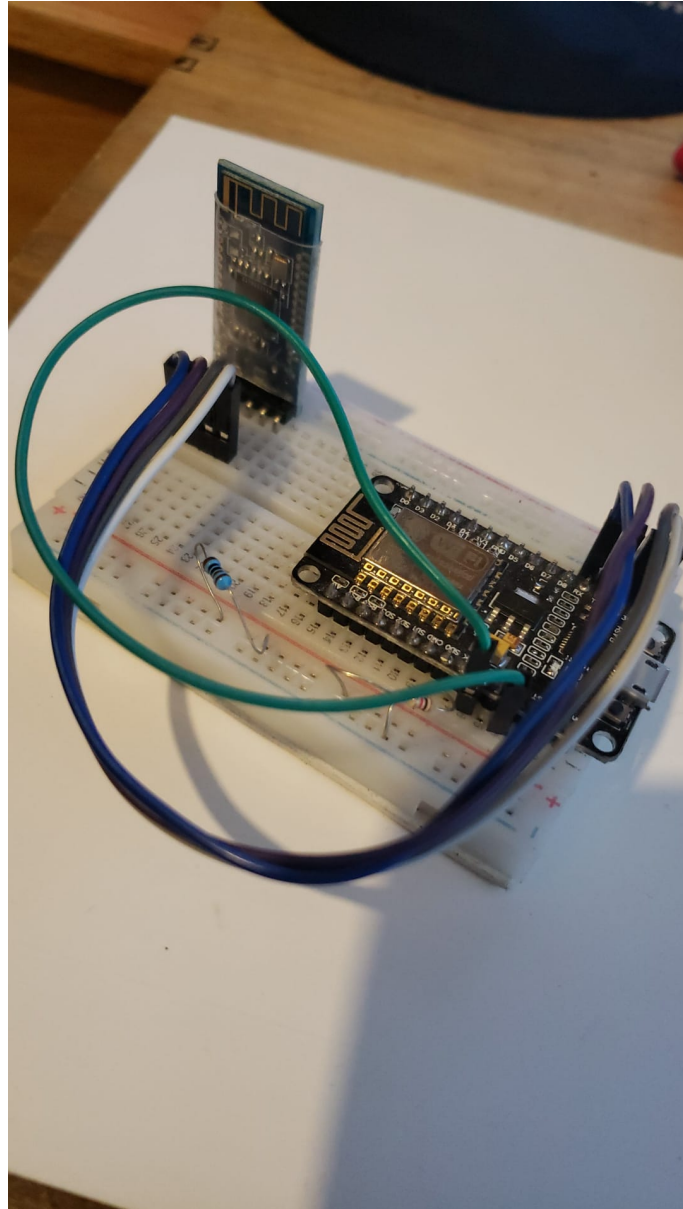


Figura 6.3: Controlador Node MCU CP2012 con puente USB TTL conectado a HM-10.

6.4. DevOps

Con el fin de tener una mayor agilidad para al momento de construir y desplegar nuestras partes de *frontend* y *backend*, optamos por tener los procesos de compilación, *testing* y despliegue de manera automatizada mediante el uso de herramientas de integración continua y despliegue continuo.

Como fue evidenciado en el capítulo anterior con los diagramas de despliegue, al momento de integrar código sobre las ramas estables como *main* y *staging*, se lanzaba el flujo para cada una de nuestras partes, realizándose cada paso de manera automática y parametrizada dependiendo del ambiente en que se trate.

Con la adopción de la metodología DevOps logramos independencia y velocidad al momento de desarrollar y no tener que ocupar tiempo durante esta etapa para tener que hacer los procesos nombrados anteriormente de manera manual, los cuales nos podrían retrasar por introducir errores de manera involuntaria en alguno de los procesos.

Optamos por utilizar Azure DevOps por ya contar con experiencia con esta herramienta y el tiempo a emplear para la configuración de los *pipelines* no insumiría tanto tiempo como sí podría ocurrir en caso de hacerlo con otras alternativas menos conocidas.

7. Gestión de proyecto

7.1. Proceso de desarrollo

A continuación se realizará la presentación sobre el proceso y ciclo de vida utilizados, con las respectivas argumentaciones que nos llevaron a inclinarnos por las siguientes decisiones.

7.1.1. Ciclo de vida

El ciclo de vida es utilizado para definir los pasos que comprenden la planificación, desarrollo y ejecución de producto. Con el fin de determinar el ciclo más adecuado al proyecto, el equipo decidió la realización de un análisis de los factores más influyentes.

Inicialmente, fue realizado un análisis sobre los modelos convencionales, y aunque su utilización busca la misma finalidad, se consideró el enfoque, contexto y habilidades requeridas por parte del equipo para cumplir y aplicarlos correctamente. Los modelos estudiados fueron:

- Cascada.
- Incremental.
- Iterativo.

Finalizado el estudio, se continuaron evaluando las necesidades, requisitos y disponibilidad del cliente, adicionalmente se tuvo en cuenta la capacidad, recursos y conocimientos por parte del equipo.

Sobre las etapas tempranas de los procesos de investigación y planificación se optó, debido a las restricciones temporales, y la existencia de fechas pautadas por la Universidad ORT Uruguay para la conclusión del proyecto emplear una metodología tradicional con el fin de cumplir con los plazos de entrega definidos.

Sobre los RF y RNF del producto contemplado, que, aunque existiese un objetivo y propósito definidos para el producto, en relación a ciertos aspectos, no resulto evidente una profundidad y especificidad de los casos de uso pertenecientes a la aplicación.

En adición a esto, el número de tecnologías para implementar en la solución, más la falta de experiencia en el manejo de algunas de ellas por parte de los integrantes del equipo, y la necesidad de integrarles entre si, determinó que la decisión del equipo fuera la adopción de un ciclo de vida iterativo incremental durante la etapa del desarrollo.

La metodología elegida para este punto nos permitiría presentar diferentes versiones de la aplicación, obteniendo incrementos considerables entre cada una de las versiones, logrando mostrar al cliente los avances obtenidos sobre el análisis, desarrollo, pruebas, problemas resueltos y nuevas funcionalidades.

7.1.2. Metodologías

Metodología tradicional

Sobre esta metodología se trabajó durante el principio y terminación por temas de tiempo y en gran parte por la situación sanitaria. Realizando como punto de partida un *Roadmap* (7.6) con la finalidad de tener a alto nivel todas las actividades a tener que realizar y en qué orden, se optó por realizar un plan de gestión representado en forma de un diagrama GANTT (7.7) en el cual se logran visualizar todas las tareas necesarias a realizar durante el ciclo de vida del proyecto, indicando orden de ejecución y duración de cada una.

Fue por decisión del equipo generar recordatorios en un calendario compartido con el fin de poder realizar el seguimiento sobre el inicio y fin de las tareas establecidas en el diagrama anterior.

Metodología ágil

En las etapas de desarrollo, documentación e implementación la elección fue ir por esta metodología ya que por los tiempos manejados para el proyecto, nos permitiría obtener rápidos incrementos del producto y así mostrar resultados en corto lapso de tiempo. Debido al tipo de producto a obtener, y no contar desde el inicio con todos los flujos a realizar, nos inclinamos por un proyecto de desarrollo apoyado en el *Manifesto for Agile Software Development*[9]

La elección del equipo fue dentro de los diferentes *agile frameworks* enmarca el uso de *scrum*, debido a que los miembros del equipo contaban con experiencia con el uso del *framework*, no teniendo que transitar una curva de aprendizaje extra.

Otro punto tenido en cuenta, fue la posibilidad de involucrar al cliente en el proyecto, ya que es de sumo interés de su parte poder realizar aportes con sus recursos disponibles y *knowhow* técnico para obtener un producto de alta calidad y suma excelencia.

El *framework* seleccionado nos brindó la posibilidad de realizar iteraciones y sesiones posteriores una vez finalizadas las mismas, que nos permitieran realizar mejoras durante el transcurso del proyecto sobre las etapas de desarrollo, documentación e implementación y estar bien preparados para poder responder de manera efectiva y rápida sobre los imprevistos que fueron apareciendo durante el proceso de desarrollo.

Esta elección nos fue de utilidad para poder realizar las iteraciones constantes, con posteriores evaluaciones utilizadas para retroalimentación sobre las buenas acciones realizadas y las no tan bien ejecutadas, llevándonos siempre a realizar acciones correctivas sobre estas últimas y tratar de conservar las buenas acciones para posteriores iteraciones.

Las ceremonias realizadas y artefactos utilizados sobre el *framework* se evidencian en la imagen a continuación; decidimos realizar algunas modificaciones sobre *daily scrum* y *sprint*.

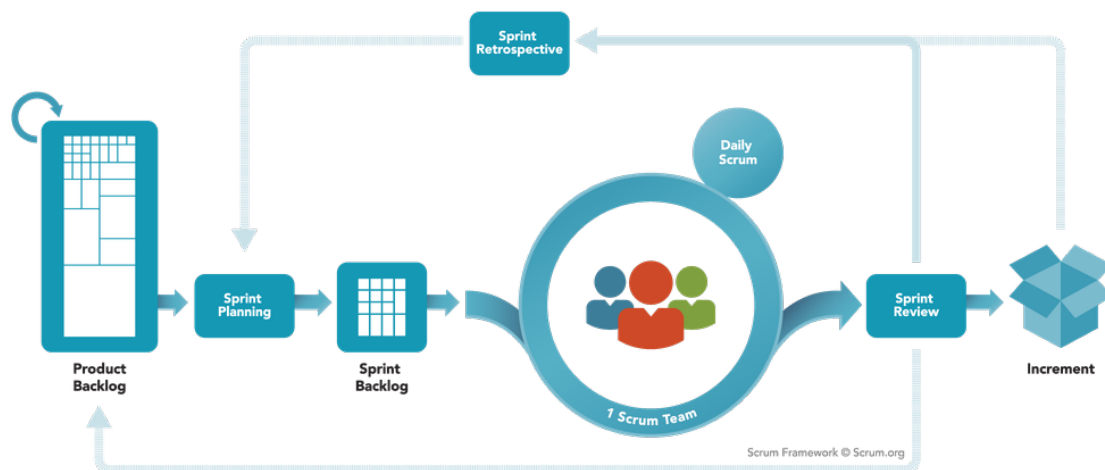


Figura 7.1: Ciclo de Scrum

Product Backlog (PB)

Conformado por los RF y RNF que fueron acordados con el cliente para su inclusión en el producto. El contenido del mismo fue generado a partir de las historias de usuario que intentan evidenciar los deseos y necesidades de los mismos. Todos los objetos dentro del PB se encuentran priorizados, estimados y cumplen con el criterio acordado de *Definition of Ready* (DOR).

El DOR definido por el equipo para que una *User Story* (US) pase a formar parte del *sprint planning* por el equipo fue:

- Fue acordado el alcance con el equipo.
- Se cuenta con toda la información necesaria para su ejecución.
- Fue acordado con el cliente su priorización.
- Su resultado aporta valor al incremento de final de iteración.
- No supera la cantidad de 13 *Story Points*.

Sprint Planning

Ceremonia celebrada durante el inicio de cada iteración, a la misma acudían el *development team* y el *scrum master* con la finalidad de organizar, priorizar y estimar la carga de trabajo correspondiente al *backlog* de la iteración.

Nos apoyamos en *planning poker* para la realización de esta ceremonia, ya que nos pareció enriquecedor ver los diferentes puntos de vista entre los miembros sobre las funcionalidades y poder abundar en detalles, en caso de no coincidir a la hora de la estimación.

Durante estas sesiones de planificación, se hizo fuerte énfasis sobre el criterio acordado de *definition of done* (DOD).

El DOD definido por el equipo para que una US pueda tomarse como completada por el equipo fue:

- Logró cumplir con el criterio de aceptación.
- La funcionalidad desarrollada fue revisada y validada por un ajeno al desarrollo.
- Se integró la *feature branch* a la *branch develop*, siguiendo lo lineamientos de Gitflow.

Sprint Backlog

Conformado por las tareas a realizarse durante una iteración, siempre tratamos de manejarnos con un total de tareas que sumen como máximo 95 SP, ya que esta fue la cantidad de esfuerzo que el equipo podía en su totalidad por iteración.

Scrum Team

Se detalla en la siguiente tabla las roles asignados a cada miembro de proyecto durante su realización.

Rol / Recurso	Miembro
Scrum master	FB
Product owner	FB
Development team	AK,FB,MS,PP

- **Scrum master:** Es el encargado de que el equipo se desempeñe en su más alto nivel posible, eliminando los obstáculos (cualquier cosa que le genere al equipo un bloqueo para desempeñar en tiempo y forma una actividad). Además de realizar el seguimiento para el cumplimiento de los compromisos asumidos y predicar las reglas y valores de *Scrum* con el equipo.

Para este rol, fue elegido el miembro del equipo con mayor experiencia sobre el *framework*, con el fin de poder volcar todo el buen manejo y empleo de los valores a resto de los miembros.

- **Product owner:** Es el encargado de definir el aspecto del producto y que características debe contener. También definiendo como se debe construir y en qué orden, debe expresar de manera clara los requerimientos y priorizarlos en función a las necesidades del proyecto.

Decidimos contar con un miembro del equipo dedicado a este rol ya que su dedicación iba a ser mayor que si el cliente nos designaba un recurso para esto, aunque la decisión sobre modificaciones en los requerimientos deberían de pasar por la confirmación del cliente, decidimos asumir el reto de tener un miembro del equipo en este cargo, que de todos modos mantenía constante comunicación con nuestro contacto en la empresa, por compartir ambos el mismo ambiente laboral.

- **Development team:** Son los encargados de definir como se realiza el trabajo establecido por el *product owner*, el equipo se encuentra estructurado y facultado para organizar y gestionar su propio trabajo de manera autónoma. Colaboran con el *product owner* para comprender y definir mejor los requerimientos y son los encargados de generar valor en función de los incrementos terminados, los cuales forman parte del entregable al final de cada *sprint*.

Para este rol, todos los integrantes participaron como parte del equipo, contando con una buena autogestión y buen entendimiento de problema.

Daily Scrum

Esta ceremonia es una de las de mayor importancia dentro del *framework*, permite mantener una fluida comunicación en el equipo y poder atacar posibles desviaciones a tiempo para llegar con los acuerdos pactados para la iteración.

Las mismas pasaron a ser denominadas por el equipo como "nighthlies" ya que las mismas fueron realizadas diariamente a las 22:00 hs con una duración de no más de 15 minutos, en donde todos los miembros del equipo contábamos con nuestras *dailies* laborales en la mañana. Durante los días sábados realizábamos una *nightly* extendida con el fin de ver los avances logrados y problemas encontrados. En caso de no poder asistir algún miembro, el mismo debía mandar un mensaje por Teams contestando las siguientes preguntas:

- ¿Qué hice ayer?
- ¿Que voy a hacer hoy?
- ¿Tengo algún problema que bloquea alguna de mis tareas?

Los días viernes no correspondientes al final de *sprint*, manteníamos una sesión de sincronización en donde asistían todos los miembros del equipo, el cliente y también el tutor. En estas reuniones nos aportó mucho tener al tutor en ellas, ya que en algunas ocasiones el cliente tuvo demoras con ciertos entregables necesarios para el equipo y contar con el tutor durante estas situaciones, logró evidenciar que cuando teníamos algún atraso con algún requerimiento, no era por un tema del equipo sino meramente del cliente.

Sprint Review

Esta ceremonia se realizó una vez finalizada la iteración, los días viernes sobre las 18:00. En la misma se realizaba un resumen sobre la iteración finalizada con

dashboards generados por el equipo, para luego realizar una demo sobre el incremento obtenido en el producto.

A la misma acudían todos los miembros del equipo, el cliente y también el tutor, para poder evidenciar los avances realizados y cómo era nuestra interacción con el cliente durante esta reunión. También se generó una interacción fructífera entre el cliente y el tutor en las mismas, en donde el cliente manifestó al tutor en varias ocasiones estar muy conforme con el trabajo realizado durante cada iteración.

Increment

Es el avance que se logró para el producto y se obtiene al final de cada iteración, el mismo se integra con los incrementos anteriores y se verifica minuciosamente el consolidado de incrementos obtenidos para garantizar que todos los incrementos en conjunto funcionen sin ningún problema. Es con este consolidado que realizábamos la demo en la *sprint review*.

Sprint Restrospective

Esta ceremonia se realizaba antes de iniciar la nueva iteración, la consideramos como una de las de mayor importancia ya que en ella obteníamos las buenas acciones realizadas, pero sobre todo las no tan bien realizadas.

Eran propuestas por todos los miembros del equipo; acciones a hacer para a tener como un hábito las buenas acciones realizadas y acciones correctivas para intentar no volver a realizar o mejorar las no tan bien realizadas. Luego de una votación nos quedábamos con las tres más votadas a implementar durante la próxima iteración.

Se optó por utilizar *Kanban* para realización de las *sprints retrospectives* además de la gestión de *links* e información relevante al proyecto.

Se utilizó la herramienta Microsoft Planner para la gestión del proyecto y la gestión de las ceremonias, dada la familiaridad con la suite de Microsoft Office 365, aprovechando las licencias provistas por nuestra calidad de estudiantes por parte de la universidad sobre esta plataforma.

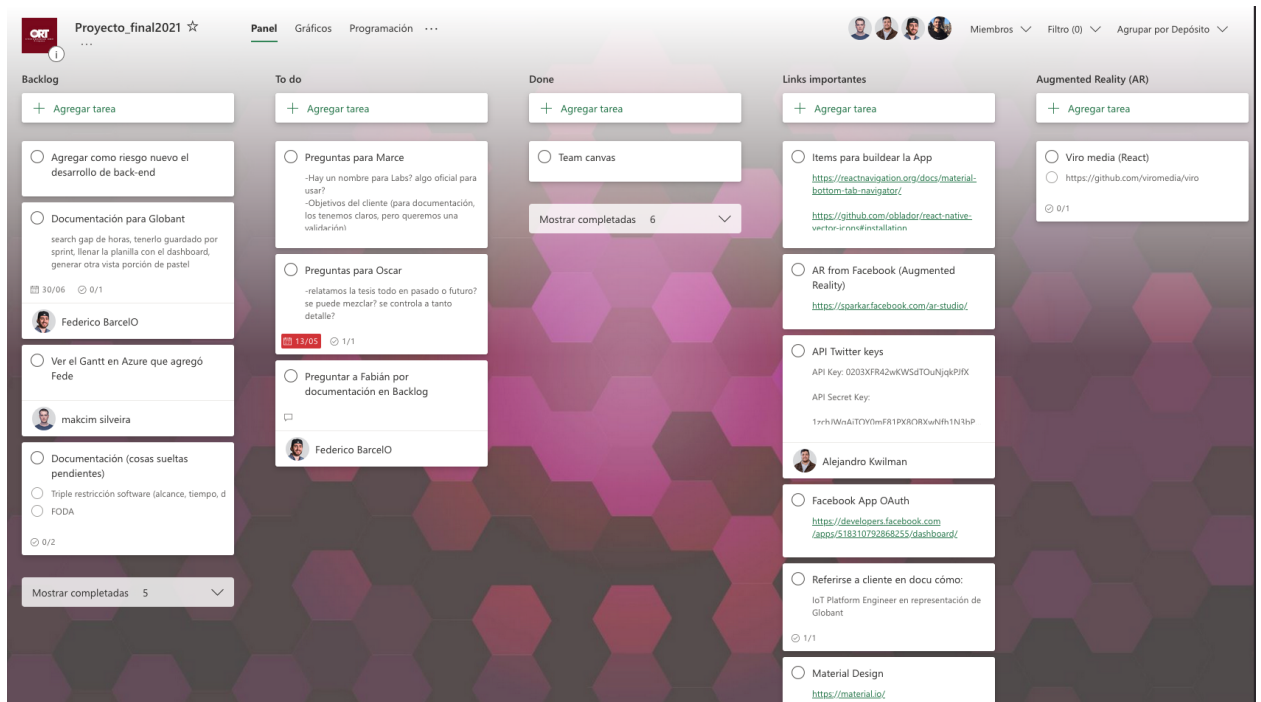


Figura 7.2: Planner para gestión de *links* e información relevante.

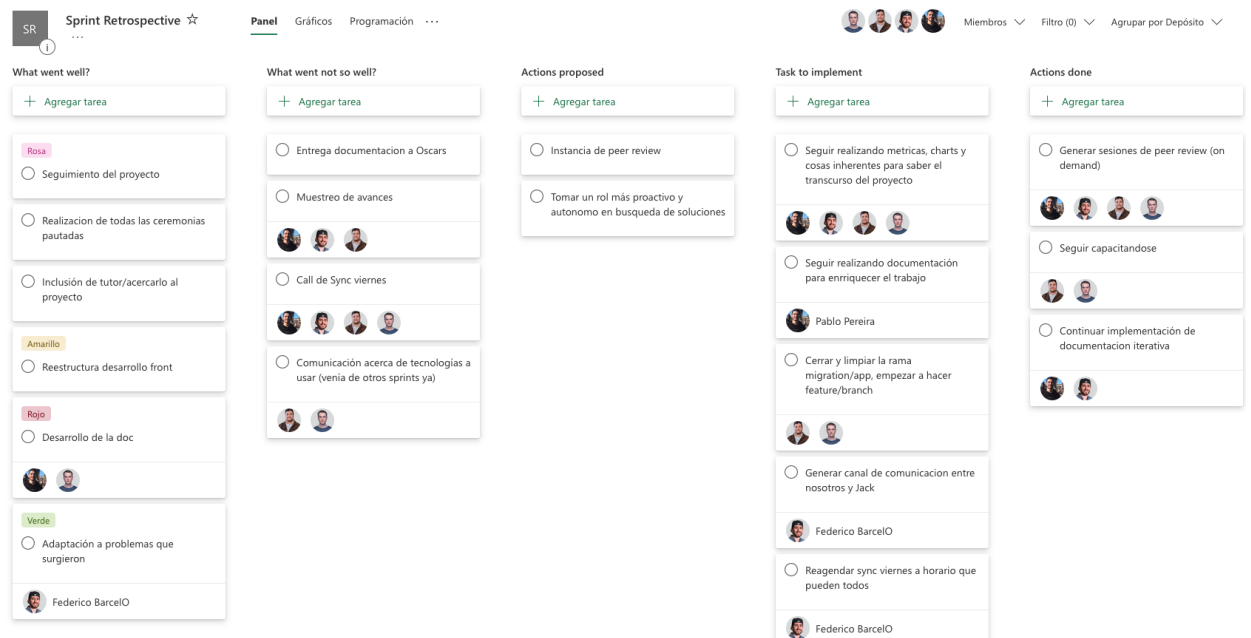


Figura 7.3: Planner *sprint retrospective*

MON 26	TUE 27	WED 28	THU 29	FRI 30	SAT 31	SUN 1 Aug
● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 19:00 Reunión con tutor ● 22:00 Nightly	● 22:00 Nightly Extendida	● 22:00 Nightly
● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 18:30 Sprint review tesis Globant-OR ● 19:00 Reunión con tutor ● 22:00 Nightly	● 19:00 Retrospective ● 20:00 Planning ● 22:00 Nightly Extendida	● 22:00 Nightly
● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 18:30 Sync semanal tesis Globant-OI ● 19:00 Reunión con tutor ● 22:00 Nightly	● 22:00 Nightly Extendida	● 22:00 Nightly
● 17:00 Revisión II ● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 18:30 Sprint review tesis Globant-OR ● 19:00 Reunión con tutor ● 22:00 Nightly	● 19:00 Retrospective ● 20:00 Planning ● 22:00 Nightly Extendida	● 22:00 Nightly
● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 22:00 Nightly	● 18:30 Sync semanal tesis Globant-OI ● 19:00 Reunión con tutor ● 22:00 Nightly	● 22:00 Nightly Extendida	● 22:00 Nightly
● 22:00 Nightly	● 22:00 Nightly	1 Sept ● 22:00 Nightly	2 ● 22:00 Nightly	3 ● 18:30 Sprint review tesis Globant-OR ● 22:00 Nightly	4 ● 19:00 Retrospective ● 20:00 Planning ● 22:00 Nightly Extendida	5 ● 22:00 Nightly

Figura 7.4: Calendario de eventos del mes de Agosto.

7.2. Etapas del proceso

En esta sección se explicará cada una de las etapas involucradas para nuestro proceso de desarrollo. Recordamos que el modelo elegido por el equipo fue un modelo iterativo incremental para la fase comprendida de desarrollo. En este modelo, cada etapa necesita de su anterior etapa para poder funcionar y llegar a lograr tener un incremento del sistema al final de cada iteración, llegando en la última iteración al producto completo.

Como primer paso, se realizó una decisión sobre las tecnologías a utilizar para cada uno de los componentes de la solución propuesta, a partir de una investigación por parte del equipo y posterior validación de la decisión propuesta por parte del cliente. Realizada la elección sobre las tecnologías a utilizar, se realizaron las iteraciones sobre estas fases:

- Diseño y prototipado.
- Desarrollo.
- *Testing*.
- Documentación.

7.2.1. Planificación e Investigación

Para esta primer etapa, se procedió a organizar de qué manera sería ejecutado el trabajo y realizar la investigación sobre el *stack* tecnológico a utilizar. Esta parte tuvo una duración de dos meses.

En lo referido a la planificación, fueron definidas las metodologías a utilizar y marco de gestión, con el fin de poder estar organizados con la ejecución del trabajo, ser competentes a la hora de la ejecución del mismo, obteniendo a partir del marco de gestión varias métricas definidas por el equipo, las cuales nos posibilitarían realizar un seguimiento adecuado del proyecto y análisis posterior al mismo.

Sobre la investigación, todos los miembros del equipo participaron de la misma, se decidió realizar una segmentación de los puntos a investigar por separada, para luego hacer una puesta en común en reuniones pautadas para este fin. Hicimos un fuerte uso de todas fuentes disponibles en internet, ya sea videográficas o documentales. También tuvimos reuniones con expertos por parte del cliente para terminar de

realizar las validaciones una vez elegidas las tecnologías a utilizar; en estas reuniones obtuvimos recomendaciones a tener en consideración sobre el *stack* tecnológico.

En esta etapa también se realizó un análisis de sector con el fin de obtener información de relevancia para el producto y también un *benchmarking* con diferentes aplicaciones que emplean de alguna manera geolocalización.

El proceso anterior nos ayudó a realizar la determinación sobre qué rol iba a desempeñarse cada miembro del equipo con sus respectivas responsabilidades asociadas.

Como fue evidenciado anteriormente, el ciclo de vida elegido fue ir por un híbrido entre una metodología tradicional y una ágil, esto también fue desprendido de la etapa de investigación.

Se dio por concluida esta primer etapa una vez que el equipo pudo determinar de manera clara un plan, que fuera conciso y que no generara ninguna ambigüedad sobre los siguientes puntos:

- Proceso del flujo comunicacional.
- Procesos de gestión y métricas a obtener.
- Proceso de control y gestión de calidad.
- Proceso de gestión de riesgos.

7.2.2. Definición de requerimientos

Para esta segunda etapa obtuvimos como resultado el documento con la especificación de los RF y RNF para nuestro producto. La definición de los mismos se logró a partir de la etapa de investigación previa y de reuniones de manera frecuente con nuestro contacto con el cliente, Marcelo Santamaría y otros expertos.

7.2.3. Diseño y prototipado

Obteniendo el documento como resultado de la etapa anterior en específico sobre los requerimientos funcionales, a partir de la metodología ágil se comenzaron a generar los primeros diseños de la aplicación.

La creación del mismo fue basada en el documento obtenido anteriormente. Para los diseños de interfaz obtenidos el equipo hizo mucho énfasis en generarlos en torno a los usuarios para poder brindar la mejor experiencia a nivel de uso. Seguimos el proceso de *Design thinking* junto a la experta por parte de cliente Verónica Varela, *UX storytelling* y Material Design. Se utilizó la herramienta Figma para generar los prototipos a ser validados con los potenciales usuarios.

En esta etapa, también se definió la arquitectura a alto nivel y fueron definidas las tecnologías a utilizar en cada una de las partes de nuestro sistema:

- *Backend*: Laravel + Laravel Nova.
- *Frontend*: React Native.
- Bases de datos: PostgreSQL + Redis
- Maquetado UI: Figma.
- *Hosting* y DevOps suite: Microsoft Azure + Azure DevOps
- SCM: Overleaf + GitHub.
- *App distribution*: Firebase.

7.2.4. Desarrollo

Culminada la etapa anterior, se pasó a la fase de desarrollo e implementación, tomando como base las especificaciones obtenidas en el punto anterior. Para la estimación y priorización de las tareas que integraban cada iteración se utilizó *planning poker*. El equipo se apegó a las buenas prácticas de *Clean Code* con el fin de conservar un buen mantenimiento y calidad de código.

Por imposición del cliente, se tuvo que mantener un código con alto nivel de cobertura sobre pruebas unitarias y contar con pruebas automatizadas tanto de unidad como de UI, con la condición de asegurar un producto de alta calidad.

7.2.5. Testing

Esta etapa de *testing* fue utilizada para encontrar posibles errores ocurridos durante las iteraciones, teniendo la posibilidad de poder subsanarlos a tiempo, el *testing* se realizó de manera adicional a las pruebas de unidad.

Se realizaron *tests* sobre toda la aplicación, desde funcionalidades, tiempos de espera deseados y hasta la interacción de la misma con los diferentes componentes de la solución. Sobre la UI, fueron empleadas pruebas a nivel de:

- Usabilidad.
- Observación.
- Entrevistas y encuentros.

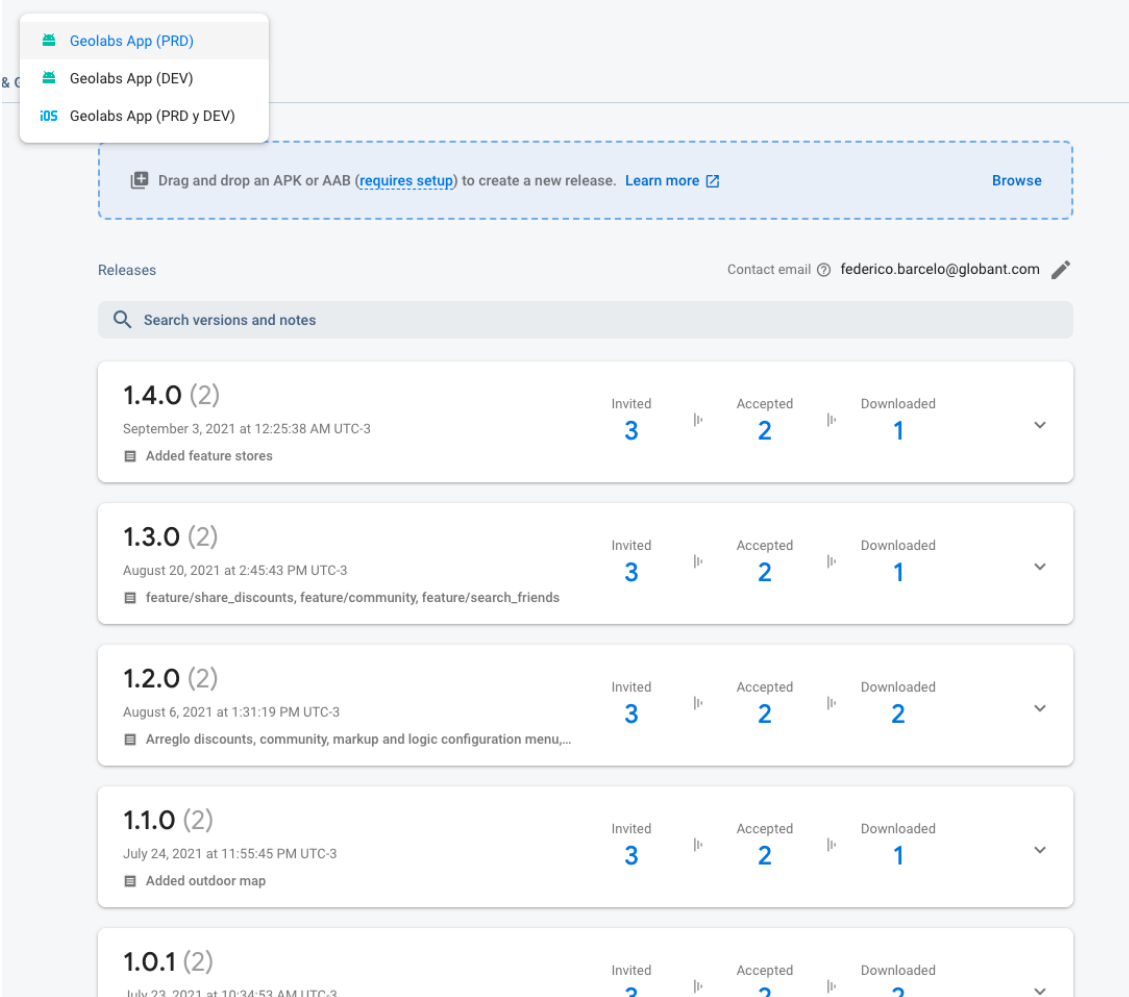
con el fin de corroborar el grado de correctitud sobre el producto y en caso de no ser suficiente, realizar las modificaciones al respecto.

7.2.6. Documentación

Como última etapa, se documentaron las actividades realizadas durante cada iteración. A partir de esto, se obtuvieron las diferentes métricas utilizadas para poder realizar evaluaciones y análisis en relación al resultado obtenido y productividad. Dependiendo de las conclusiones obtenidas por estas métricas, se pudieron realizar modificaciones para obtener mejores resultados en las próximas iteraciones.

7.2.7. Release

Esta fue la etapa utilizada para realizar el despliegue tanto de la aplicación móvil hacia nuestros dispositivos móviles, nuestro *backend* o las bases de datos, el equipo desarrolló *pipelines* para automatizar estas tareas de despliegue, las cuales fueron ejecutadas al final de cada *sprint*, además de tener versionada cada una de las liberaciones que se realizaban para mantener un histórico de las mismas.



The screenshot shows the Firebase console interface for the 'Geolabs App'. At the top, there are three tabs: 'Geolabs App (PRD)', 'Geolabs App (DEV)', and 'Geolabs App (PRD y DEV)'. Below the tabs is a dashed blue box with a prompt: 'Drag and drop an APK or AAB (requires setup) to create a new release. Learn more' and a 'Browse' button. The main section is titled 'Releases' and includes a search bar 'Search versions and notes'. The contact email is 'federico.barcelo@globant.com'. The releases are listed as follows:

Version	Invited	Accepted	Downloaded
1.4.0 (2) September 3, 2021 at 12:25:38 AM UTC-3 Added feature stores	3	2	1
1.3.0 (2) August 20, 2021 at 2:45:43 PM UTC-3 feature/share_discounts, feature/community, feature/search_friends	3	2	1
1.2.0 (2) August 6, 2021 at 1:31:19 PM UTC-3 Arreglo discounts, community, markup and logic configuration menu,...	3	2	2
1.1.0 (2) July 24, 2021 at 11:55:45 PM UTC-3 Added outdoor map	3	2	1
1.0.1 (2) July 23, 2021 at 10:34:53 AM UTC-3	3	2	2

Figura 7.5: Consola de *releases* de la aplicación móvil en Firebase.

7.3. Estrategia de gestión

Con el cometido de lograr asegurar en su totalidad la consumación de los objetivos pautados en el proyecto, se identifican diferentes actividades, las cuales serán detalladas durante el desarrollo de este capítulo.

7.3.1. Roadmap

En la siguiente figura 7.6 se expone de manera gráfica lineal la estrategia general adoptada para el desarrollo de las fases del proyecto.

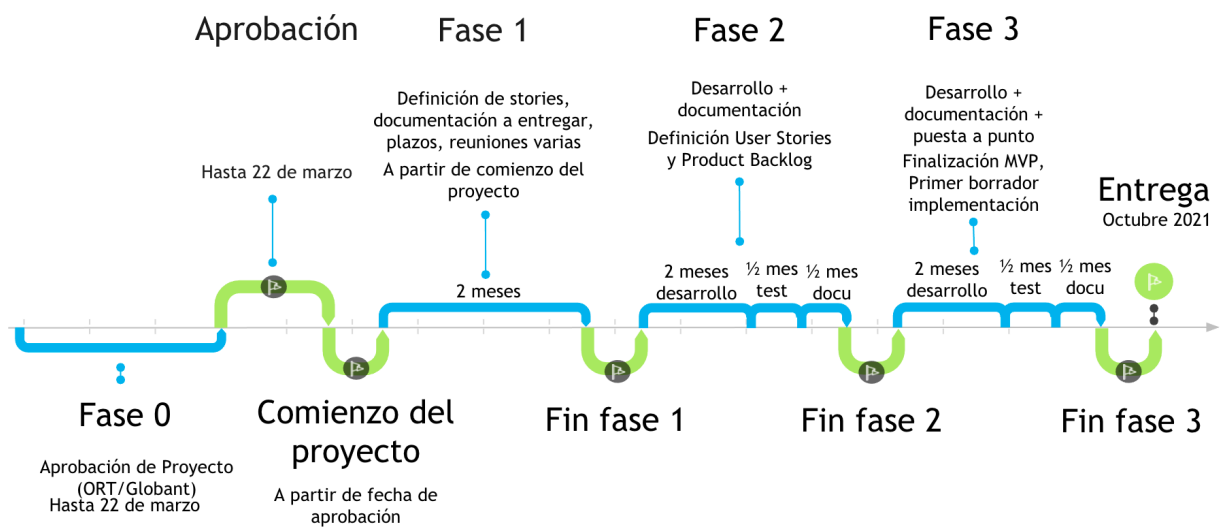


Figura 7.6: Roadmap inicial.

7.3.2. GANTT

En el siguiente diagrama se evidencia la estrategia definida para las etapas de proyecto y se detalla cada una de ellas durante el transcurso del proyecto.

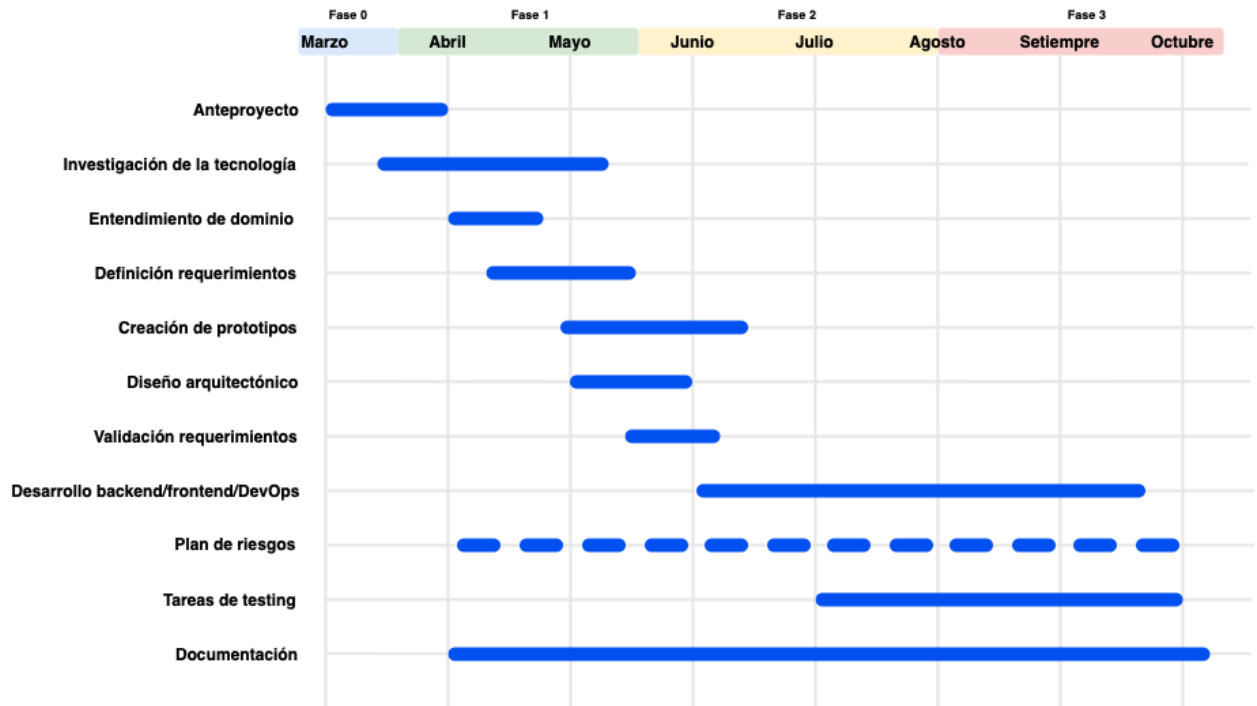


Figura 7.7: Diagrama GANTT.

7.3.3. Planificación de roles y recursos

Con el fin de lograr cubrir todas las necesidades que fueron detectadas durante la fase inicial del proyecto:

- SQA.
- SCM.
- Arquitectura de *software*.
- Gestión y planificación de proyecto.
- Ingeniería de requerimientos.
- Gestión:
 - De riesgos.

- De la comunicación.
- De incidentes.

Se diseñó una matriz con las respectivas responsabilidades de los miembros del equipo, donde cada uno será el responsable sobre el seguimiento y gestión de las mismas. Se ilustra en la siguiente matriz los diferentes roles y sus asignaciones de los recursos disponibles.

Rol / Recurso	<i>FB</i>	<i>MS</i>	<i>PP</i>	<i>AK</i>	<i>MM</i>	<i>VV</i>	<i>FBZ</i>
PO					X		
<i>Team Agilist</i>							X
SM	X						
PM	X						
SCM	X						
<i>Team Leader</i>			X				
<i>Backend developer</i>		X	X	X			
<i>Frontend developer</i>		X	X	X			
DevOps	X						
SQA			X				
QA	X	X	X	X			
UI/UX <i>design</i>	X				X	X	
Analista de requerimientos		X		X			
Arquitecto de <i>software</i>	X		X		X		
Gestión de comunicación				X			
Gestión de riesgos		X					
Gestión de incidentes				X			

7.3.4. Registro de actividades

Utilizamos la extensión Time Log Summary dentro de la plataforma Azure Devops para realizar el registro de horas durante todo el transcurso del proyecto, esto nos sirvió para poder medir los desvíos que se tuvieron, ejecución de las tareas durante las diferentes etapas, cantidad de esfuerzo estimado sobre las tareas y un desglose por el tipo de actividad empleada con su duración de tiempo.

Se generaba una *task* dentro de cada *user story* asignada al *sprint* para poder realizar la carga de horas respectivas.

User	Project	Work Item	Type	Week		2021-04-19							2021-04-26					Totals
				Day of week		19 Mon	20 Tue	21 Wed	22 Thu	23 Fri	24 Sat	25 Sun	26 Mon	27 Tue	28 Wed	29 Thu	30 Fri	
				maksim silveira	Multigance	103 US - Confección del capítulo (10)	Investigación			2:00								
		110 Curso React	Investigación			2:00	1:00				1:00	1:00	2:00	2:00				9:00
		118 Estudio viabilidad de risk poker	Documentación	2:00														2:00
		124 Identificación y seguimiento de riesgos	Documentación		2:00		1:00	1:00	1:00									5:00
		127 Analizar en mayor detalle MapWize	Gestión	3:00	3:00	2:00		1:00										9:00
		128 Analizar en mayor detalle Loopback	Investigación								1:00							1:00
		131 Reunión semanal con Oscar	Investigación								1:00							1:00
		132 Formateo de XLSXs y programación de fórmulas	Gestión					1:00										1:00
		133 Reunión por UI/UX con Vero	Documentación							1:00								1:00
		134 Reunión por integración entre RN y portabilidad con librería Swift	Gestión							1:00								1:00
		140 Nightly Extendida	Documentación							0:30								0:30
		140 Nightly Extendida	Gestión							1:00								1:00
		150 Reunión con Alejandro K.	Gestión								1:00							1:00
		151 Agregando nuevos riesgos	Gestión								1:00							1:00
		152 Puliendo documentación para enviar al tutor	Documentación								1:00							1:00
		159 Sync semanal 30/4	Gestión														1:00	1:00
Totals				5:00	5:00	6:00	2:00	3:00	5:30	1:00	3:00	3:00	2:00	2:00	1:00			38:30

Figura 7.8: Horas correspondientes a Maksim Silveira en *sprint* 1.

Los tipos de actividades posibles a cargar con sus horas correspondientes fueron pautados por el equipo como los siguientes:

Tipo de actividad	Descripción
Desarrollo	Horas ejecutadas en desarrollo y <i>testing</i> .
Documentación	Horas ejecutadas en documentación universitaria o para el cliente.
Gestión	Horas ejecutadas en gestión de los <i>sprints</i> , reuniones con el tutor, cliente y del mismo equipo.
Helpdesk	Horas ejecutadas en brindar soporte a algún miembro del equipo o cliente.
Infraestructura	Horas ejecutadas en despliegue y configuración de servidores, <i>beacons</i> , etc.
Investigación	Horas ejecutadas en investigación/validación sobre tecnologías a utilizar.

7.3.5. Plan de *releases*

La decisión del equipo para el plan de *releases* fue obtener un entregable tanto de *backend* como de *frontend* al finalizar cada uno de los *sprints* pautados, pudiendo tener liberaciones intermedias durante el transcurso de cada uno de los *sprints*, las cuales formarían parte del proceso de incremento del producto, pero no como parte de plan de *release*. Con cada versión obtenida de *release* fue realizada la demo con el cliente y el tutor.



Aplicaciones de prueba



Figura 7.9: App distribution en iPhone con 12 versiones.

7.3.6. Acuerdos y convenciones

Introducción

Al comenzar el proyecto se realizó un *Team Canvas* 10.12 del cual se desprendieron diferentes convenciones para gestionar el trabajo en equipo y fomentar respeto, colaboración mutua y evitar conflictos.

Acuerdos relacionados a documentación

La documentación debe ser revisada por todos los integrantes antes de calificarse como válida para enviar a revisión por el tutor u otro *stakeholder*.

Acuerdos relacionados a repositorios

Para el trabajo con los repositorios, decidió utilizarse GitFlow, y la aprobación de al menos un miembro del equipo para realizar *merges* y *pull requests*.

Acuerdos relacionados a comunicación externa

Para realizar envío de e-mails u otro tipo de comunicación ya sea con el tutor, cliente u otros *stakeholders*; bastan dos respuestas afirmativas en caso de que existan diferencias en el equipo.

7.3.7. Métricas

Durante la duración del proyecto, fueron elegidas las siguientes métricas para realizar seguimiento al trabajo y efectivamente medir el progreso de las tareas, los *sprints* y el proyecto general, y tomar las acciones correspondientes según lo evaluado.

Burndown Charts

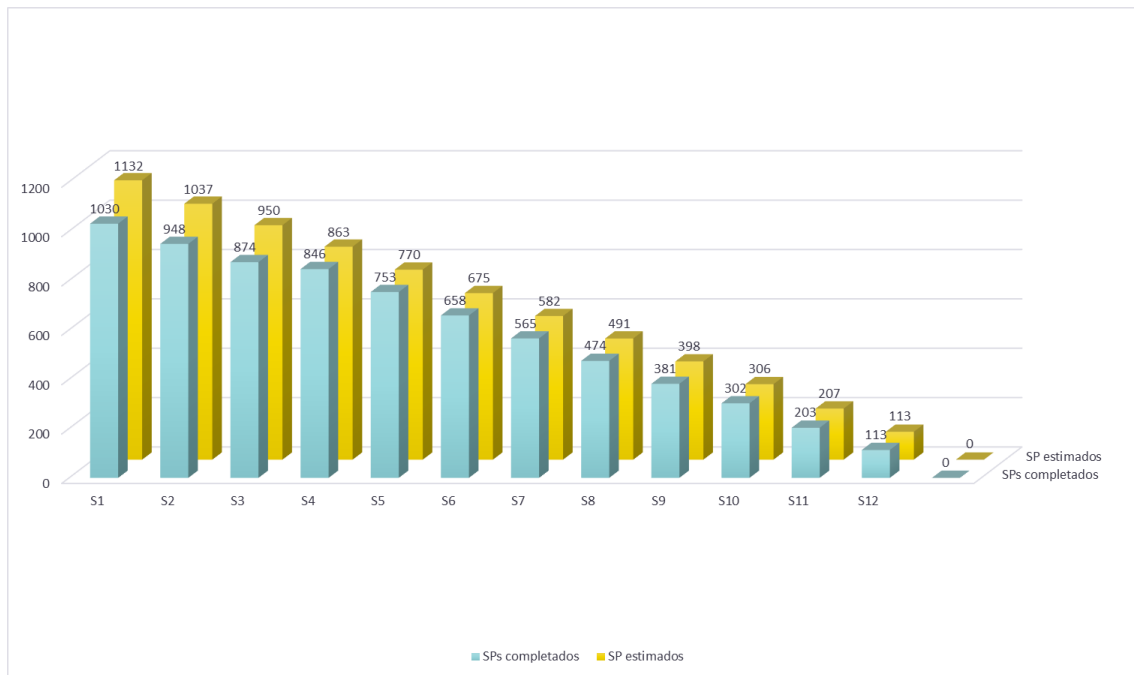


Figura 7.10: Burndown chart.

Velocity Charts

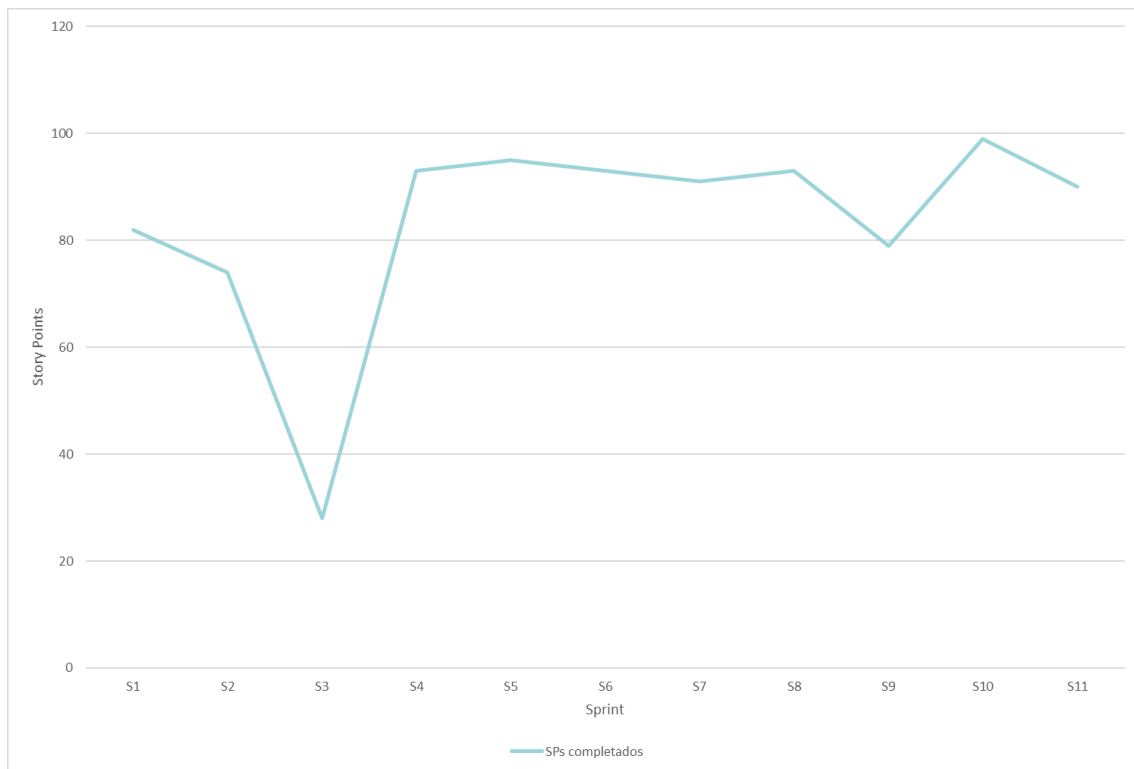


Figura 7.11: Burndown chart.

Estimated SPs vs Real Executed SPs

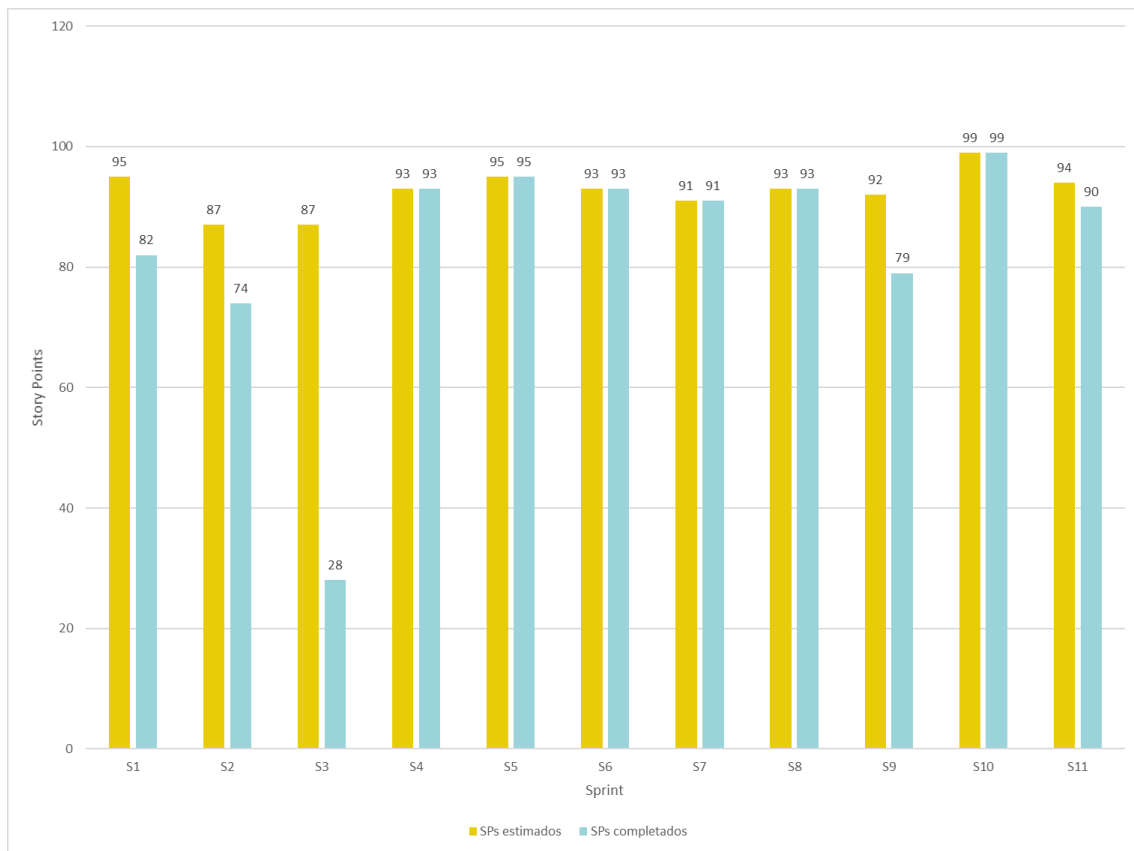


Figura 7.12: *Story points* reales vs. estimados.

Total de horas por tipo de tarea

Con el fin de obtener las horas ejecutadas en el proyecto se tomaron los datos estructurados desde Azure Boards. Luego de procesarlos a través de Microsoft Excel se obtuvo la siguiente información detallada:

Tipo de actividad	Horas ejecutadas
Desarrollo	1105
Documentación	576
Gestión	268
Helpdesk	18
Infraestructura	63
Investigación	356
Total para el proyecto	2386

Tabla 7.1: Tabla de horas ejecutadas por tipo de actividad.

En la siguiente gráfica se pueden ver la cantidad de horas y el porcentaje correspondiente del total de horas ejecutadas del proyecto, discriminadas por tipo de tarea.

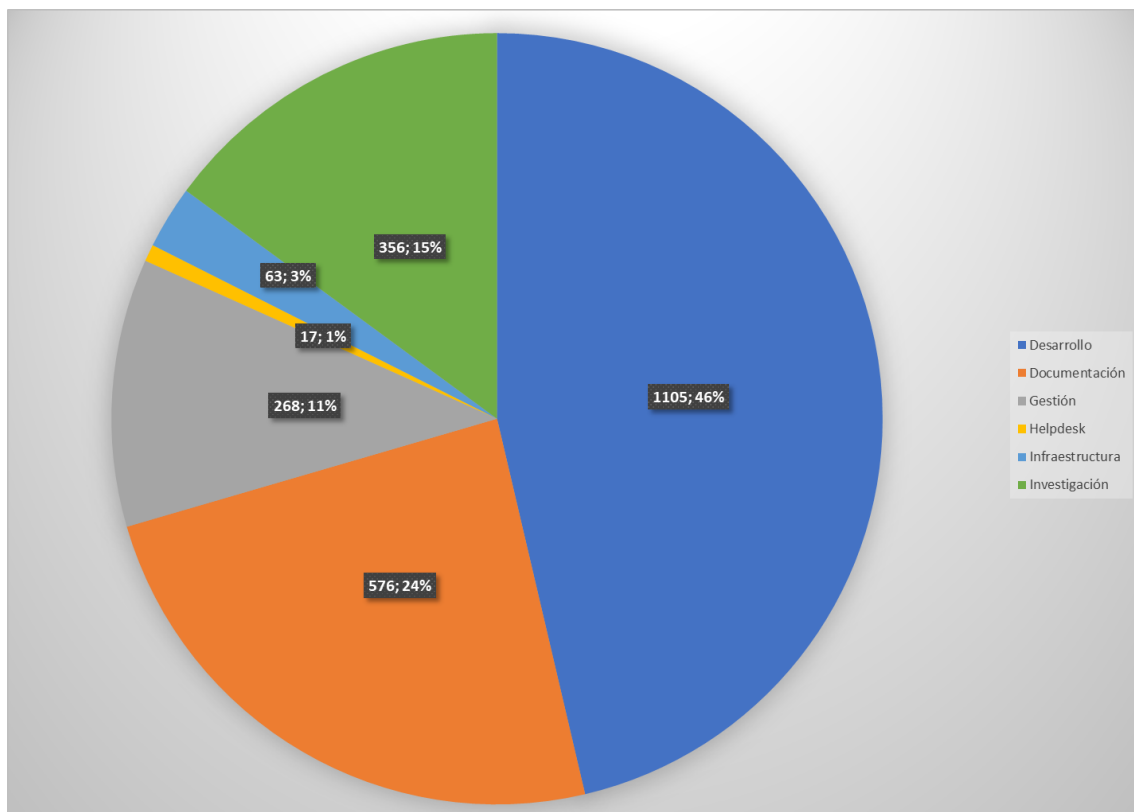


Figura 7.13: Cantidad de horas y porcentaje de horas por tipo de tarea.

7.4. Gestión de riesgos

7.4.1. Plan de riesgos

Introducción

Al comenzar el proyecto nos asesoramos con el Ing. Fabían Bozoglilianian, graduado de Universidad ORT Uruguay y quien forma parte de Globant, procurando orientación en la etapa previa de y anteproyecto. Como principales puntos destacó la gestión de riesgos, así como también diversas herramientas como Team Canvas y Business Canvas para recabar actividades e información.

Tipos de Riesgos

Se identificaron los siguientes tipos de riesgos en el proyecto:

- De datos
- De equipo
- De gestión
- Humanos
- Tecnológicos

7.4.2. Riesgos identificados y mitigación respectiva

Introducción

Los riesgos en un proyecto de software son variados y las amenazas, en particular, pueden impactarlo negativamente durante diferentes etapas del proyecto, alterando el curso del proyecto, al punto de peligrar su continuidad o resultados esperados. Tomando en consideración que el equipo se adentró en un dominio complejo, fue absolutamente necesario contar con una adecuada gestión de los riesgos. Por lo tanto, para poder actuar frente a los mismos en forma óptima, se realizaron diferentes evaluaciones acerca de posibles estrategias y metodologías a implementar. Según PMBOK [10] un riesgo puede no ser necesariamente negativo, sino que puede proveer una oportunidad. También puede que no se materialice y por lo tanto no impacte en el proyecto.

Se mantuvo durante el proyecto una planilla en Microsoft Excel en el repositorio de documentación de Microsoft OneDrive, en la cual se fue iterando para contar con los riesgos actualizados por *sprint*.

Cada riesgo contiene los siguientes atributos:

Atributo	Descripción
ID	Clave de valor incremental compuesta por la primer letra del tipo de riesgos seguida de tres dígitos.
Fecha	Fecha en que fue evaluado o re-evaluado el riesgo.
Probabilidad	Probabilidad de que un riesgo materialice. Valor con escala de 0 a 3. En caso de haber sido mitigado el valor es igual a 0.
Impacto	Valor que va desde -3 a -1 para oportunidades y para riesgos desde 1 a 3.
Magnitud	Se obtiene multiplicando la Probabilidad y el Impacto. Cuanto mayor es el valor, mayor es la amenaza. En caso del valor ser negativo, cuanto menor, mayor es la oportunidad.
Tipo	Categorización del riesgo.
Responsable	<i>Stakeholder</i> quien recibe mayor impacto sobre el riesgo. En su mayoría el equipo es quien fue responsable por los riesgos, si bien individualmente alguno de los integrantes pudiera tener mayor dominio sobre alguno de los riesgos u oportunidades. Para casos en que el riesgo estuviese fuera del alcance del equipo, se definió el <i>stakeholder</i> con el cual se debería intentar resolver o escalar el riesgo.
Acción a tomar	Las acciones a tomar se codificaron por medio de colores, y se mostraron de acuerdo a una escala (por más información dirigirse al anexo 10.7.) <ul style="list-style-type: none"> ▪ Asumir ▪ Monitorear ▪ Prevenir ▪ Mitigar ▪ Mitigado
Plan de acción	Qué acción o serie de pasos tomar para resolverlo o aplicarle la acción a tomar indicada previamente.
Materializado	Indica si el riesgo se concretó.

Tabla 7.2: Atributos de riesgos y sus descripciones.

En caso de que el riesgo haya sido contrarrestado o de haberse disipado, se indicó en color azul y se lo describió como mitigado.

Como se puede ver en la planilla (ver anexo 10.7) se utilizó una codificación de colores para poder visualizar con mayor facilidad qué amenazas al proyecto contrarrestar. Además, los riesgos están discriminados por *sprint*.

Se mantuvo una planilla adicional para los riesgos que debían mitigarse, para tener una visión más clara de los riesgos con mayor gravedad.

En casos particulares, para tratar riesgos de alta magnitud y que evolucionaran con mayor velocidad de la esperada, se citaron reuniones no programadas previamente, con el propósito de no perder perspectiva sobre los mismos. Otro beneficio que proporcionó poder reunirse en forma virtual fue alterar el curso de acción rápidamente y lograr actualizar a los *stakeholders* afectados directamente por el riesgo en cuestión, con la mayor prontitud posible.

Evaluación de Riesgos

Los riesgos podrían detectarse bajo diversas situaciones, por ejemplo al realizar la planificación del sprint o en discusión entre los miembros del equipo durante *nightlies*.

Para evaluar los riesgos se procuró cuantificarlos para realizar una evaluación tanto a nivel individual como a nivel de categoría de riesgo. Esto se realizó en torno a dos componentes, probabilidad e impacto. La combinación de las mismas arrojó un valor de magnitud el cual fue el indicador principal tomado para los riesgos. Se definió una escala de magnitudes y se actuó en consecuencia, basándose en las mismas.

Para dar respuesta a los riesgos, el equipo se ajustó a los lineamientos de PMBOK [10], el cual recomienda que las respuestas a los riesgos deben ser realistas en el contexto del proyecto, se debe dar en acuerdo con los *stakeholders* y se debe definir un responsable de riesgos.

Como fue sugerido por el tutor se categorizaron los riesgos por *sprint*. Esto permitió contar con una vista de los riesgos actuales pero manteniendo el histórico de los riesgos mitigados.

Para el impacto se utilizó una escala, la cual consistió en la multiplicación de

la probabilidad de ocurrencia de un riesgo por el impacto; con ello se obtuvo la magnitud.

Una vez obtenida la magnitud se pudieron visualizar los riesgos gráficamente gracias a un *set* de reglas ingresadas en la hoja de cálculos utilizada para obtener en forma rápida y visual qué riesgos requerían atención inmediata.

Control de Riesgos

Una vez actualizados los riesgos y su magnitud se procedió a la discusión de los mismos con el resto de los integrantes. Dependiendo de la severidad y responsable del mismo se podría llegar a resolver internamente o escalarlo con el *stakeholder* pertinente.

7.4.3. Clasificación y evolución de riesgos respecto a su categoría

Se realizó la clasificación de riesgos en un orden que fuera natural y que permitiera monitorearlos tanto a nivel individual como a alto nivel. Además, esto permitió realizar una presentación de riesgos detectados en las reuniones de presentación de avances (también llamadas *sync*) lo cual complementó la visibilidad general y transparencia de avances para con los *stakeholders* involucrados.

Como se puede apreciar en las gráficas por *sprint*, los mayores períodos de magnitud de riesgos se dieron en torno al *sprint* 3 (Ver 10.101) -donde alcanzó su pico máximo- y retomando intensidad -aunque en menor medida y con tendencia a la baja- sobre los *sprints* 7 y 8. Estos períodos coinciden con los momentos de mayor incertidumbre en el proyecto. En el *sprint* 3, momento en que se estaban definiendo aspectos tecnológicos aún, se tuvo que desviar temporalmente la atención hacia mejorar aspectos señalados por el Revisor, durante la primer revisión académica oficial del proyecto. Otro de los momentos clave en materia de riesgos, se dio como se mencionó anteriormente alrededor del *sprint* 7 y 8 (Ver 10.106), coincidiendo con una acumulación de riesgos, tanto en cantidad, como en la magnitud de los mismos. Sobre mediados del proyecto se puede apreciar la magnitud negativa de los riesgos de tipo producto, en la cual se ve reflejada la oportunidad de expansión del producto a otras áreas de negocio. Desde un comienzo del proyecto se imaginó como una plataforma con diferentes aristas en distintos dominios y negocios, pero no fue hasta ese punto, en que fue detectado como una oportunidad propiamente dicha, e ingresada al sistema de seguimiento de riesgos formalmente.

En cuanto a los tipos de riesgos se puede destacar que en su mayoría fueron desafíos tecnológicos. Sin lugar a dudas los desafíos tecnológicos ocuparon un lugar destacado en la gestión de riesgos. No obstante, con el avance en el desarrollo, estos fueron disipándose y en casos de detectarse nuevos, se lo ha hecho con antelación.

7.4.4. Conclusión y lecciones aprendidas sobre riesgos

La gestión de riesgos resultó fundamental para la buena gestión del proyecto en general, donde cada actividad de apoyo al desarrollo de *software* fue clave en contribución al éxito del mismo. Permitted adelantarse en algunas instancias a situaciones que podrían haberse tornado inmanejables sin una adecuada respuesta en el espacio temporal. Por otra parte, brindó un marco de trabajo estructurado y adicionó solidez

al trabajo del equipo; en diversas instancias el plan de riesgos permitió enfocarse en caminos alternativos al momento de encontrarse con obstáculos que podrían haber provocado retrasos en actividades del ciclo de vida.

7.5. Gestión de configuración

Durante este capítulo se realizarán las explicaciones pertinentes acerca del manejo de los diferentes elementos identificados, los cuales deberían de mantener integridad y versionado durante el transcurso del proyecto.

La definición de herramientas como el establecimiento de normas fueron necesarias para poder realizar esta gestión con calidad y de manera productiva; evitando el retrabajo, el desperdicio de recursos, reduciendo errores, y aumentando la calidad y productividad del proyecto.

Gracias a esto, es que fue posible sostener la integridad del sistema a lo largo del desarrollo del proyecto y nos permitió la obtención de diferentes datos sobre el mismo, haciendo posible la generación de informes gráficos y tablas que nos permitieron realizar seguimiento de nuestro rendimiento y consecuentemente la detección de oportunidades y mejoras, redundando en el aumento de calidad e impactando positivamente en la satisfacción del cliente.

7.5.1. Identificación de elementos de configuración

Fue por decisión del equipo que todo el código fuente, bases de datos y documentos asociados producidos que puedan sufrir modificaciones, deben de ser considerados como elementos de la configuración de *software* y fueron tratados como tales. A continuación se detallan cada uno de estos elementos identificados.

- **Software:**
 - Código fuente *backend*.
 - Código fuente *frontend*.

- **Documentos:**
 - Documentación entrega universitaria.
 - Plan de:
 - SQA.
 - Calidad.
 - Proyecto.
 - Prototipos UI/UX.
 - Análisis de riesgos.

- Documento:
 - ESRE.
 - De arquitectura de *software*.
 - De estrategia de desarrollo.
 - De estrategia de comunicación.
- Minutas:
 - Con tutor.
 - Semanales equipo.
- **Base de datos:**
 - Versiones de:
 - La base de datos.
 - *Set* de datos.
- **Elementos de UI/UX:**
 - Prototipos.
 - Imágenes.

7.5.2. Elección de herramientas

Teniendo en cuenta la gran cantidad de elementos de configuración manejados para el proyecto, nos enfocamos en la utilización de las herramientas más adecuadas para el correcto manejo de cada uno de los elementos mencionados anteriormente.

7.5.3. *Software*

La elección para este apartado fue utilizar una herramienta que utilice un sistema de control de versiones Git.

Organización de repositorio

Por tratarse de un proyecto para Globant, no tuvimos mucha opción al momento de elegir la herramienta para manejar los repositorios, nos fue brindado acceso sobre la organización GlobantUy de GitHub perteneciente a Labs, en donde generamos de manera privada los siguientes repositorios:

- Catalyst-ORT-Front: Repositorio donde se encuentra el código fuente y recursos del *frontend*.
- Catalyst-ORT-API: Repositorio donde se encuentre el código fuente del *backend*.

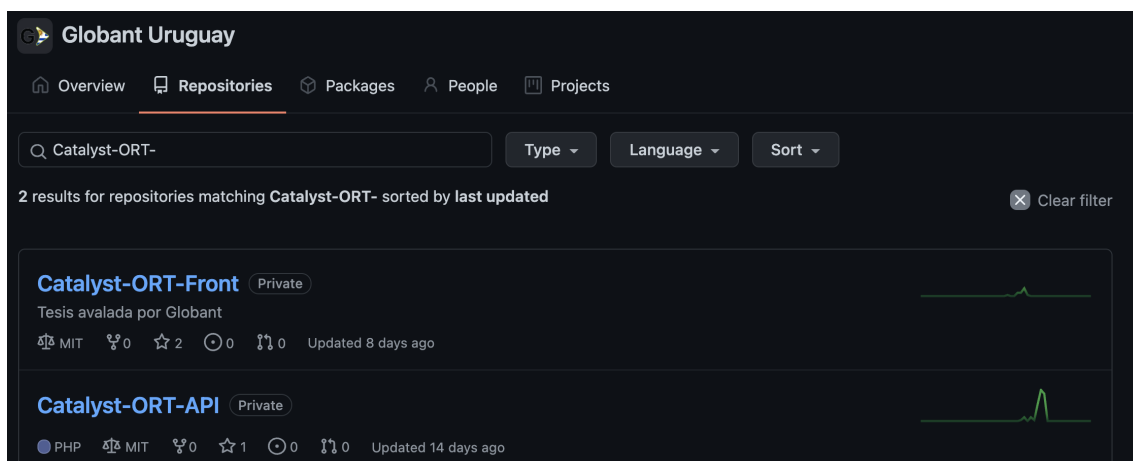


Figura 7.14: Repositorios en GitHub de Labs.

Flujo y gestión de ramas

En cuanto al control y gestión de los flujos de desarrollo el equipo optó por seguir el modelo de ramificación GitFlow.

El objetivo de la aplicación de este modelo es aislar el trabajo completado del que está en desarrollo, facilitando así la colaboración y sincronización, evitando los conflictos.

Esto se logra separando el desarrollo en ramas de manera paralela para funcionalidades específicas (*feature branch*), de tamaño reducido y que pueden ser accedidas por los miembros del equipo de manera simultánea, y a su vez se encuentran aisladas de las ramas principales/estables, evitando el impacto no deseado y la fusión accidental de ramas.

Las ramas definidas por el equipo en las cuales se puede trabajar fueron las siguientes, se utilizaron *tags* con información relevante para identificar de manera rápida ciertos cambios en el código:

- *main*: rama por defecto generada por GitHub, su finalidad es contener el código que se encuentra corriendo en producción, por lo cual tiene que ser código probado y estable. No integramos desarrollo en esta rama hasta tener nuestra versión lista para realizar la demo. No se realizaron modificaciones directamente sobre esta rama.
- *staging*: rama generada por el equipo, la misma es la más cercana a la rama *main*. Desde esta rama se realizó el *merge* a la rama *main*, además por ser la última rama intermedia antes de llegar a *main*, se realizaban en la misma las tareas relacionadas a *testing*. No se realizaron modificaciones directamente sobre esta rama. A partir de esta rama se generaron los *releases* utilizados para realizar las demos los días viernes.
- *develop*: rama generada por el equipo, en la cual se encuentra el último estado del código. Es la rama que más cambios sufrió durante el proyecto, debido a que en la misma se realizaban las integraciones de las ramas *feature*. No se realizaron modificaciones directamente sobre esta rama.
- *feature/x*: rama generada por el equipo, cada vez que una nueva funcionalidad se fue a desarrollar, debía de generarse una rama con el nombre *feature/(nombre de la funcionalidad)* para reconocer de manera rápida de que funcionalidad se trata la rama, la creación de la misma se realizaba a partir de la rama *develop*. Una vez que la funcionalidad se encontraba completada, la rama *feature* pasaba a integrarse en la rama *develop* mediante un *pull request*.
- *hotfix/x*: rama generada por el equipo para arreglar errores de manera rápida detectados en el sistema, a partir del error detectado se generó una rama con el nombre *hotfix/nombre del error* para reconocer de manera rápida a que error se hace referencia, esta rama fuera generada desde la rama *staging* (también puede generarse desde la rama *main*) que fue la utilizada para realizar las

pruebas, una vez solucionado el error se integra el cambio a *staging* y también a la rama inferior de *develop*. En caso de la rama *hotfix* se generara a partir de *main*, al momento de realizar la integración a *main*, el cambio se integra a las ramas inferiores de *staging* y *develop*.

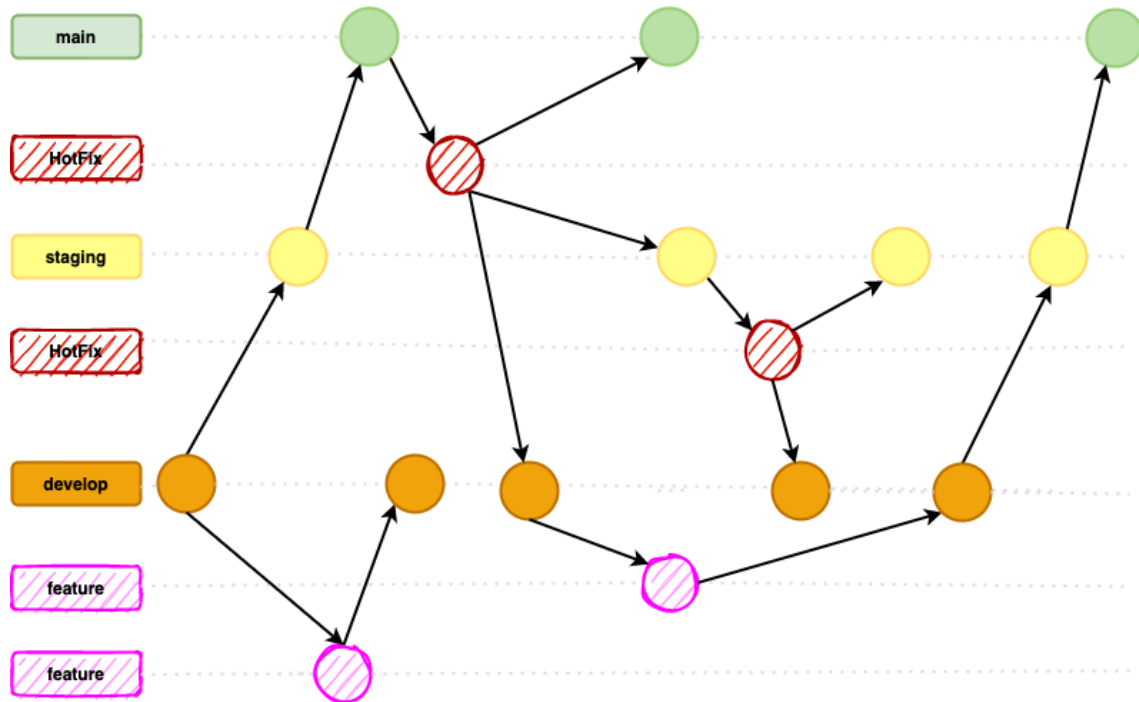


Figura 7.15: *GitFlow* empleado por el equipo.

Readme

Tanto en el repositorio de *frontend* como en *backend* se contó con un *readme* el cual es un archivo de texto enriquecido y contenía diversa información necesaria para los integrantes del proyecto. El mismo fue escrito en su totalidad en idioma inglés, para mantener consistencia con el idioma usado para el código fuente. La finalidad de este archivo es que cualquier persona ajena al proyecto que quiera utilizarlo pueda fácilmente configurar el ambiente y encontrar soluciones a problemas ya reportados.

El mismo contó con la siguiente información:

- Instrucciones acerca de cómo configurar el ambiente de desarrollo
- Instrucciones acerca de posibles soluciones a problemas conocidos
- Información acerca de cómo realizar un despliegue en ambas plataformas de *frontend* utilizadas.

- Dependencias utilizadas para encontrar fácilmente soluciones en caso de ocurrir dificultades o conflictos.

☰ README.md ✎

Dependencies

- [Listed dependencies links used on the React Native project](#)

React Navigation:

[Getting started](#)

[Material Bottom Tab Navigator](#)

Other dependencies:

[React Native Vector Icons](#)

[Material UI Community Icons](#)

[React Native Animatable](#)

[React Native Linear Gradient](#)

[React Native Elements & UI Toolkit](#)

[React Redux](#)

React Native Community Checkbox

- [@react-native-community/checkbox](#)

After installing component Checkbox, you must update pods and rebuild for IOS:

```
pod update
pod install
```

Figura 7.16: Dependencias de aplicación *frontend*.

7.5.4. Documentos

El proceso de generación de la documentación comenzó inicialmente a la par del proyecto, es por esto por lo que, desde las etapas iniciales, ya se contaba con una documentación, actualizada y ordenada, facilitando de esta forma la generación de contenido para las entregas intermedias y el documento académico final.

Documentación general

La documentación de todos los puntos anteriores (con salvedad de la documentación universitaria) se gestionaron mediante OneDrive durante todo el proyecto.

Se generó una estructura de carpetas para manejar una mejor organización y control, la cual se detalla a continuación:

- **Actas reuniones:** Contiene las actas correspondientes a diferentes reuniones de relevancia para el equipo, también contiene algunos vídeos de las mismas.
- **Archivos de interés:** Contiene archivos, guías o *links* a sitios considerados de interés por los miembros del equipo.
- **Documentación Aulas:** Contiene la documentación brindada por parte de la universidad para las diferentes etapas del proyecto, como las documentaciones de formato 302, 303, 304, 306.
- **Documentación Proyecto:** Contiene toda la estructura de subcarpetas de la documentación del proyecto, como pueden ser análisis realizados, diagrama de base de datos, *dashboards* de *sprints*, métricas de proyecto, gestión de calidad, incidentes, riesgos, plan de SQA, ciclo de vida, etc.
- **Entregas formales ORT:** Contiene los archivos que fueron entregados en cada una de las instancias pautadas por la universidad como obligatorias.
- **Fotos personales equipo:** Contiene fotos de los miembros del equipo utilizadas en presentaciones o documentaciones.
- **Presentaciones:** Contiene las presentaciones utilizadas durante las diferentes fases de proyecto y presentaciones de avance.

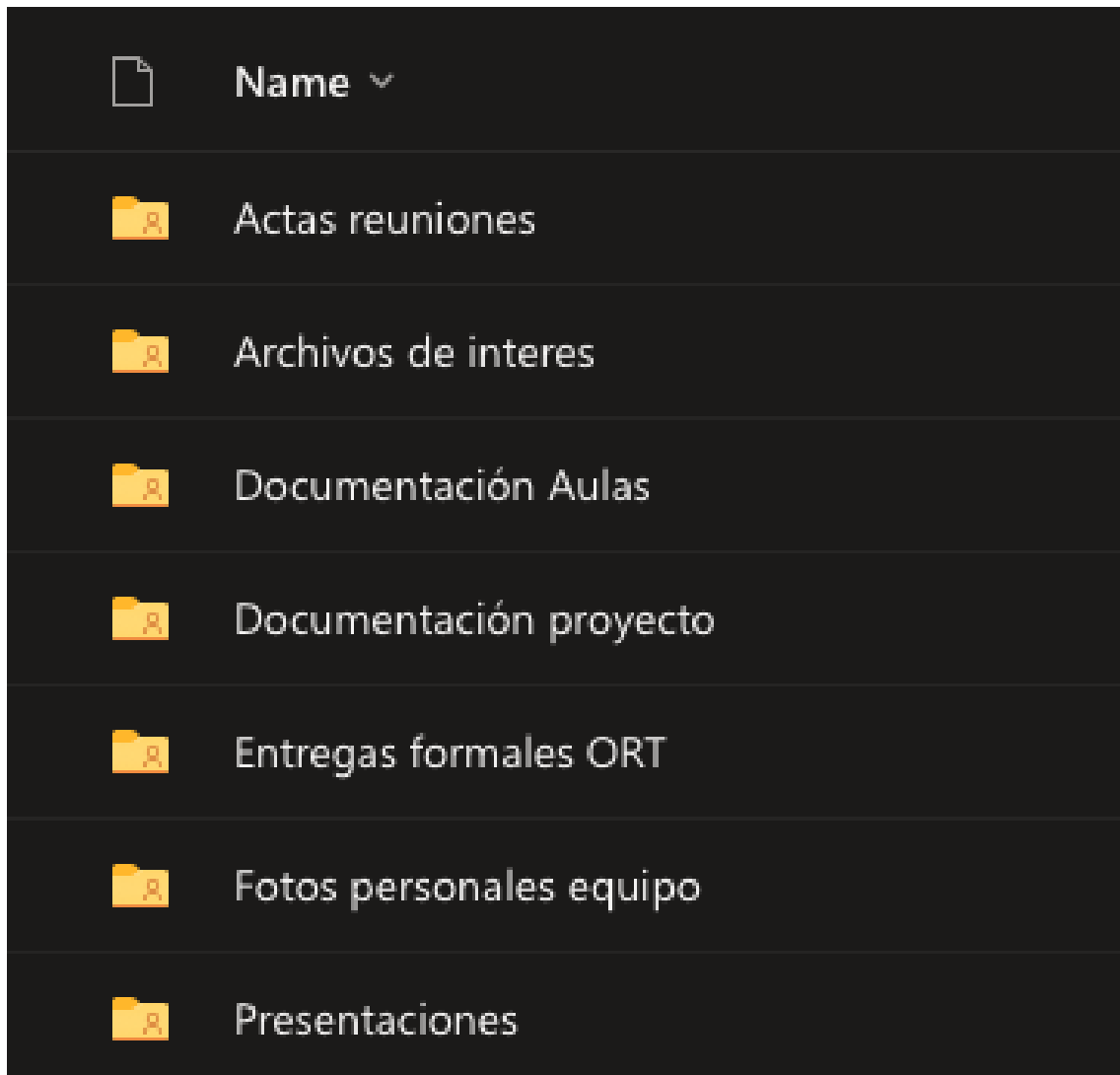


Figura 7.17: Estructura de carpetas definida en OneDrive.

Documentación universitaria

Desde el principio de proyecto utilizamos la plataforma *Overleaf*. Esta herramienta es un editor de formato *online* que emplea el sistema de composición de texto LaTeX.

Permitiendo la colaboración en tiempo real, control de versionado, aplicar formatos personalizados como los requeridos en el documento 302 y poder ir observando a medida que se va generando el documento un *preview* de mismo para ir validándolo.

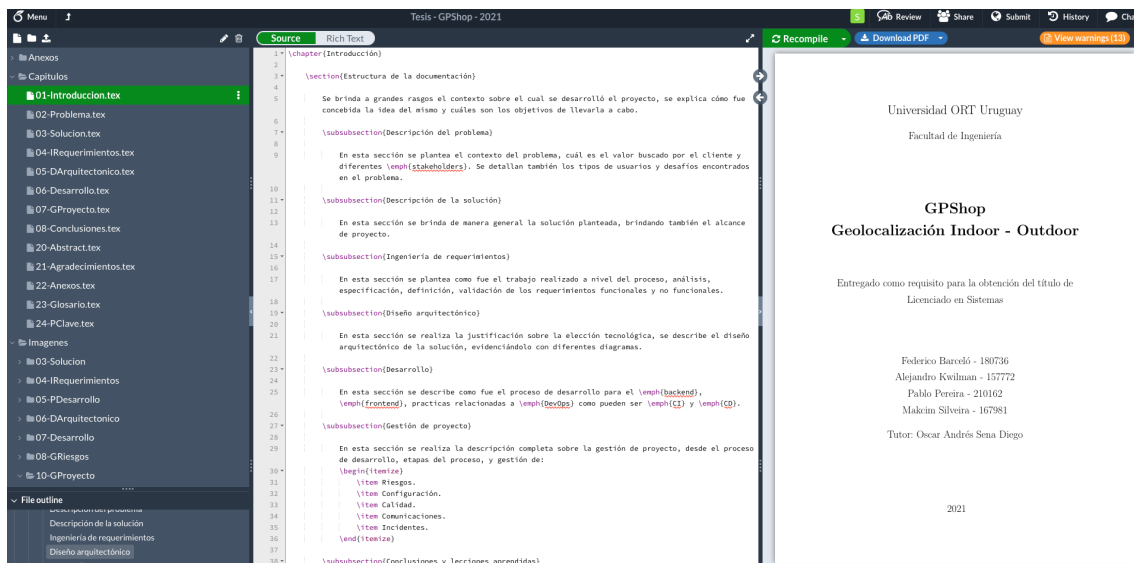


Figura 7.18: Interfaz gráfica de Overleaf.

7.5.5. Base de datos

Se configuró en el servidor en donde se encontraba publicado nuestros servicios de *backend* un *script* para realizar de manera diaria un respaldo de nuestra base de datos relacional.

El *set* de datos de carga se encuentra versionado en el código del *backend* ya que se trabajó con *seed* desde Laravel para popular la base de datos por primera vez y realizar nuevas cargas que fueron siendo necesarias con el avance del proyecto.

7.5.6. Elementos de UI/UX

Para estos elementos se utilizó la herramienta de diseño de interfaces Figma que permite edición en simultáneo y en tiempo real a los miembros del equipo, y cuenta con una serie de herramientas y configuraciones que nos permite completar nuestro trabajo relacionado al diseño de la interfaz sólo con el uso de la misma, al contar con la herramienta por parte del cliente, la misma maneja el versionado de los elementos por ser de modalidad paga.

7.6. Gestión de calidad

En este apartado se tratará la gestión de la calidad, uno de los pilares de la construcción de *software* y uno de los aspectos de mayor visibilidad para clientes y usuarios. La gestión de la calidad tiene diferentes aristas e inclusive para productos similares puede tener diferentes definiciones. La definición de Calidad se compone de diferentes parámetros dependiendo del elemento de configuración que se está tratando, para ilustrar mejor, no aplican los mismos parámetros al código fuente de la aplicación que a la documentación. La mejora en la calidad debe ser constante y siempre existen oportunidades de mejora. Por otra parte, la calidad es un concepto abstracto, por lo que fue el entendimiento del equipo de definir objetivos y métricas para obtener una cuantificación y así poder analizar la calidad desde un punto de vista objetivo, contrastando los resultados con lo esperado por el cliente.

7.6.1. Objetivos para el aseguramiento de la calidad

Por tratarse de una aplicación con elevada complejidad y constante interacción por parte del usuario se debió contar con cuidados adicionales para asegurar la calidad del producto. Por otra parte, el cliente es uno de los líderes de la industria del *software*, lo cual hizo de la calidad un atributo especialmente importante del proceso de ingeniería.

7.6.2. Plan de SQA

Dado que se utilizaron metodologías ágiles se decidió ajustar la gestión de calidad a ese contexto. Para ello se tomaron las recomendaciones del capítulo " *Quality management and agile development*" de *Sommerville*. [11]. En lugar de contar con un elaborado plan de SQA se apoyó en reuniones informales y acoplamiento al ciclo de vida *agile*. [11] A continuación se hace un breve repaso por las principales sugerencias implementadas a lo largo del proyecto.

Se tomó propiedad de los elementos de configuración de forma tal de que si se encontraron defectos, estos fueron subsanados con la mayor prioridad [11].

Los miembros se responsabilizaron al momento de integrar código a ramas estables, de testearlos adecuadamente para no introducir defectos que comprometen la estabilidad de la aplicación. [11]

Para los casos en que existieron incidentes de gravedad alta se procedió a solu-

cionarlos inmediatamente.

Existen otros mecanismos, desarrollados por el equipo y que se fueron dando en forma natural, evolutiva y espontánea. Algunos de ellos se mencionan en otros apartados, como el *checklist* de QA que fue visto como una necesidad de centralizar pequeños cambios y poder hacer seguimiento de los mismos.

7.6.3. Plan de calidad

Actividades para el aseguramiento de la calidad

A continuación se detallarán las diferentes actividades que formaron parte del aseguramiento de calidad del proceso de *software*.

BDD

BDD no reemplaza otras metodologías sino que las potencia y es complementaria a otros *frameworks*. Además de incentivar la colaboración más cercana entre diferentes *stakeholders* [12]. En *frontend* se utilizó Cucumber [13] y para *backend* se utilizó Behat [14]. Para la elección de los *frameworks* de testing se priorizó la facilidad de uso e instalación para reducción de posibles riesgos.

Automatización de pruebas

Uno de los principales objetivos del desarrollo de *software* es la eliminación de tareas repetitivas por parte de los usuarios. Esto también es válido si se aplica en ingeniería de *software*, por lo que se intentó automatizar lo máximo posible actividades que pueden ser realizadas por un actor no humano. Tanto para *backend* como para *frontend* se utilizó SonarQube. Al utilizar la suite de Azure DevOps se contó con integración de SonarQube *out-of-the-box*. Esta herramienta es una de las más utilizadas en la industria y es líder en pruebas automatizadas de código, la cual es otra de las razones por las que se escogió SonarQube en favor de alternativas similares.

A nivel de *backend* se logró cumplir con un 87,7% de cobertura lo cual supera el 60% propuesto por el RNF7. En cuanto a *frontend*, se alcanzó el 45,8% de cobertura, lo cual supera el 40% definido (por más detalles dirigirse a 4.9).

SCM

Otro aspecto que contribuye en materia de calidad al proyecto es SCM, lo cual es imperativo en la práctica de desarrollo de *software* desde hace muchos años. No sólo a nivel de código sino también de documentación. La principal ventaja de contar con un sistema de SCM es contar con reversión de los elementos de configuración, así como también poder tomar cursos alternativos, como en el caso de crear nuevas ramas.

Se utilizó SCM además, para el versionado del documento principal, codificado en LaTeX, con lo cual se cumple con una de las exigencias académicas de la Cátedra de Ingeniería de Software de la universidad.

***Checklist* de QA**

Se realizaron revisiones con el PO para comprobar el funcionamiento y flujos de navegación, detectar comportamientos no deseados y corregirlos antes de las *sprint reviews*.

Checklist QA – detectados 2/09/2021 - Testeado sobre iOS/Dispositivo Físico

Thursday, September 2, 2021 10:52 PM

Integración login backend 📄

Pantalla maps

- Alinear botones
- Espaciado de drawer icon y titulo
- Lógica para cambiar entre mapas indoor/outdoor
- Cambiar entre indoor/outdoor con boton de abajo
- Hacer click en la tienda y mostrarla
- Hacer el click de pending y redireccionar
- [OPCIONAL] Activity indicators en map stack screen

Pantalla discounts

- Espaciado entre tiendas y border bottom

Search de discounts

- Más espaciado e imagenes a la izquierda
- Cambiar titulo en resultado de busqueda
- Linea gris mas oscura
- Agregar empty state cuando no hay resultados (cuando haces enter)

Community

- Bottom border y agrandar pelotas de iconos 📄
- Cambiar opacity en add friend modal
- Cancel y Send más separados, ver Figma
- Search de community, cambiar por las pelotas de nombre
- Cambiar en los resultados el titulo del resultado

Share location

- Hacer

Hacer My discounts

- Hacer shared discounts 📄

Favorites stores

- Espaciado en la lista
- Empty state de busqueda favorites

Store Screen

- Sacar de favoritos endpoint faltante
- Linear gradience sobre imagen

Drawer

- Logout (que cierre la app)
- Sacar avatar
- Poner logo GPShop

Search de users

- Más espaciado e imagenes a la izquierda
- Cambiar titulo en resultado de busqueda
- Linea gris mas oscura
- Agregar empty state cuando no hay resultados (cuando haces enter)
- [OPT] cache de último resultado
- [OPT] cache de últimos 10 resultados

Search de stores

- Más espaciado e imagenes a la izquierda
- Cambiar titulo en resultado de busqueda
- Linea gris mas oscura
- Agregar empty state cuando no hay resultados (cuando haces enter)

Pantalla Inicial

- Logo de GPShop (Figma)
- Conectar con los logins en back-end

Figura 7.19: Checklist de Quality Assurance

Documentación

Como nota adicional, el cliente destacó en múltiples oportunidades el valor de contar con documentación, de la cual este documento forma parte, y será entregado al mismo al finalizar esta etapa del proyecto.

Estándares

Para la documentación se aplicaron los estándares académicos explicitados en los documentos 302, 303, 304 y 306.

Calidad de Código

Un *linter* es una herramienta de análisis de código estático que permite señalar errores, inconsistencias de estilo, malas prácticas, entre otros tipos de elementos de código que pueden generar *bugs* o dificultar la tarea de los desarrolladores a

futuro. En particular se utilizó ESLint. El mismo permitió la detección temprana de prácticas incorrectas que tienden a generar dificultades y degradar la calidad del código en general. ESLint permite utilizar diferentes *plugins*, el principal utilizado fue *eslint-config-airbnb*, desarrollado por Airbnb. De esa manera, se aseguró la inclusión de un conjunto de estándares de la industria como punto de partida. También se utilizaron otros *plugins* como *eslint-plugin-react-native* y *eslint-plugin-react-redux*. Este último fue fundamental para evitar posibles inconsistencias en los ciclos de vida de los componentes funcionales.

Se utilizó Tabnine para complementar las sugerencias de codificación en *frontend*, la cual es una herramienta de autocompletado de código que utiliza AI (Inteligencia Artificial). Además de aumentar la productividad y velocidad de codificación, esta aplicación reduce aún más la probabilidad de errores e introducir defectos en la aplicación.

Si bien se utilizaron diversas herramientas y automatización de algunos de los procesos de calidad, estos se aplicaron en conjunción con las reglas y recomendaciones de *Clean Code* del autor Robert C. Martin [7]. Independientemente de las tecnologías utilizadas, las mismas proveen un marco de trabajo que asegura la celeridad en introducción de nuevo código y que el mismo sea legible y fácil de comprender.

Criterios de interfaz de usuario

Durante el desarrollo del proyecto intentamos cumplir con los lineamientos de usabilidad al igual que la codificación de las funcionalidades, de esta manera tuvimos una relación entre dos aspectos que consideramos como fundamentales para el sistema.

Por recomendación de la experta de *UI/UX* Verónica Varela, se tuvieron que establecer ciertas pautas al momento de empezar a generar las interfaces de usuario, las cuales fueron:

- **Colores:**

Por medio de los colores se pueden transmitir muchos mensajes y terminan jugando un papel muy importante determinando la percepción del sistema.

Se realizó un análisis sobre que transmite cada color y se observaron diferentes interfaces existentes para poder definir la esencia de nuestra aplicación:

- Blanco:
Se asocia con la pureza y luz, también suele tener una connotación positiva.
- Negro:
Este color fue utilizado para el manejo de textos de la aplicación.
- Violeta:
Se asocia con la creatividad, la tecnología y el sector *online*.

■ Iconografía:

Los iconos pueden ser utilizados con varios propósitos, para nuestro producto se decidió de utilizarlos para generar el contenido visual más atractivo, entendible y como una referencia visual rápida para un concepto en específico.

Todos los iconos utilizados fueron de Material Design por restricción del cliente, todo ícono se encuentra asociado de un texto para evacuar cualquier tipo de duda, el logo de la aplicación fue realizado por un equipo interno del cliente.



Figura 7.20: Íconos del *home menu*.

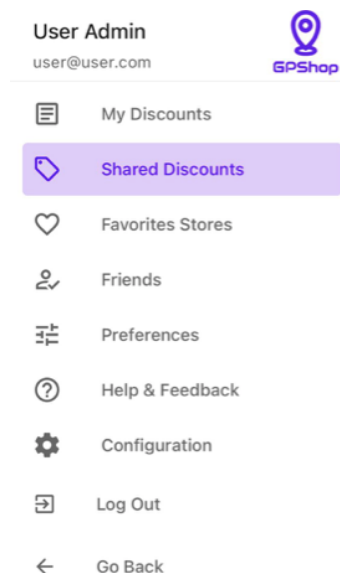


Figura 7.21: Iconos del *hamburger menu*.

- Iteraciones

Los diseños de nuestras pantallas se vieron modificados en diversas oportunidades, debido a detecciones surgidas a partir de análisis sobre heurísticas, pruebas con usuarios, observación, usabilidad y deseabilidad.

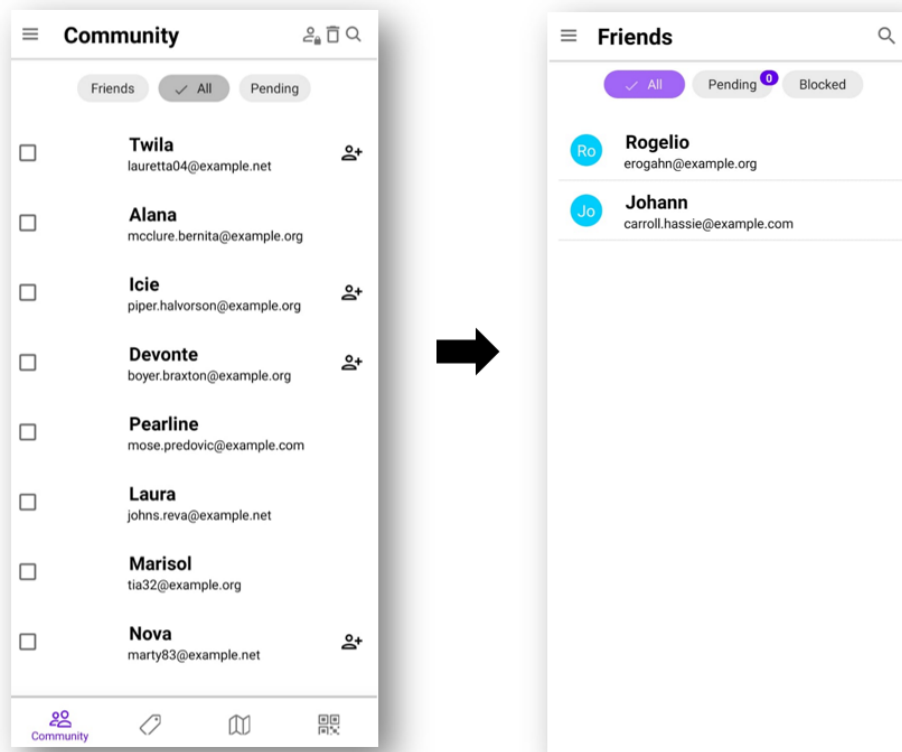


Figura 7.22: Evolución de pantalla de amigos.

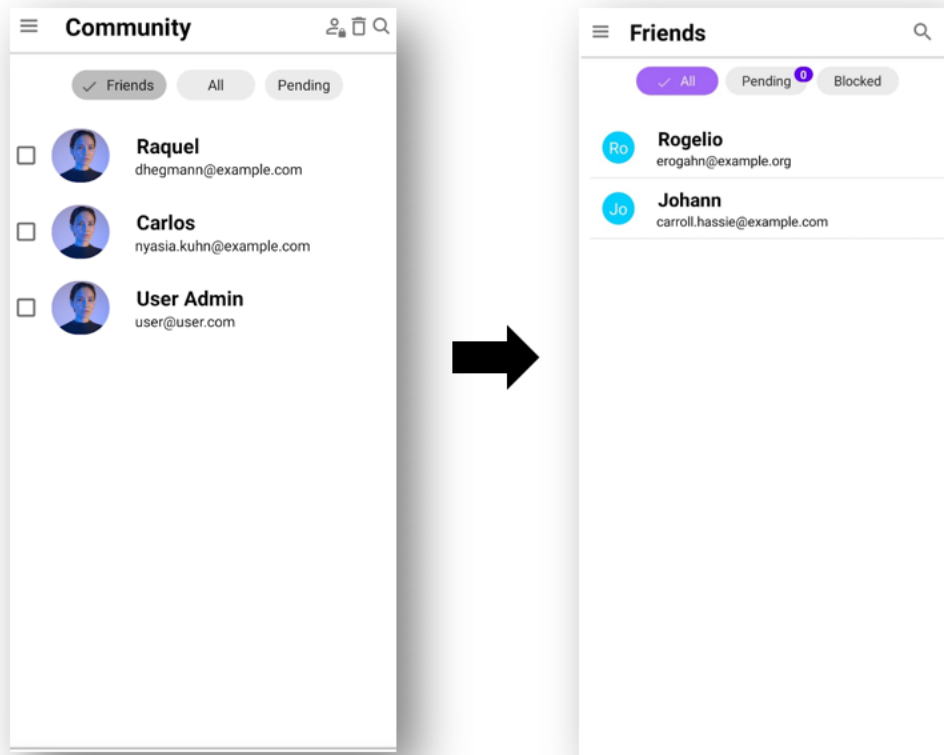


Figura 7.23: Evolución de pantalla de amigos.

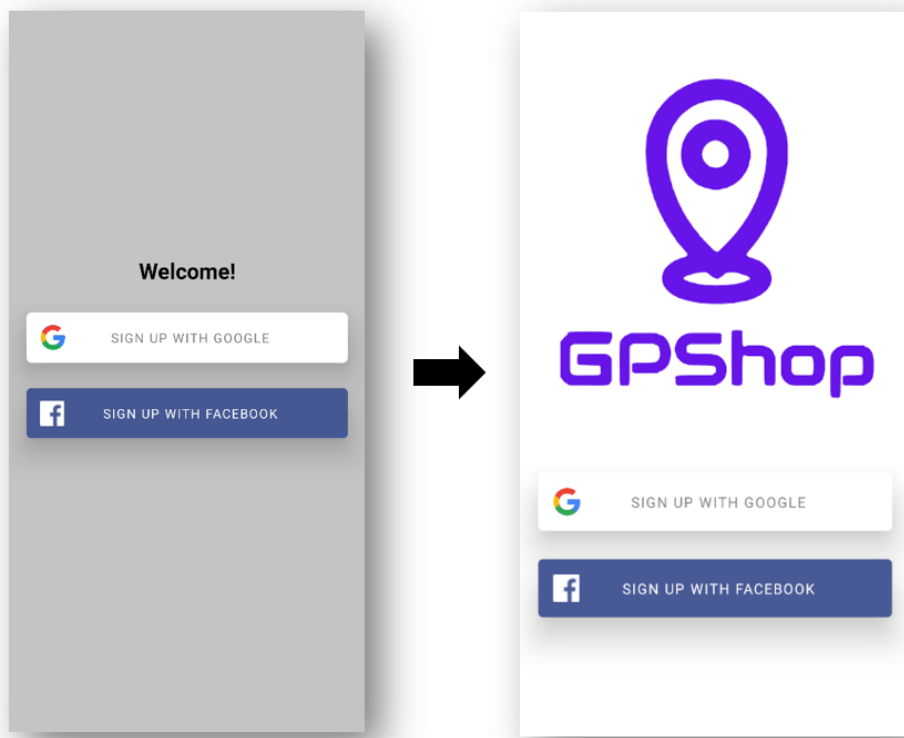


Figura 7.24: Evolución de pantalla de *Login*

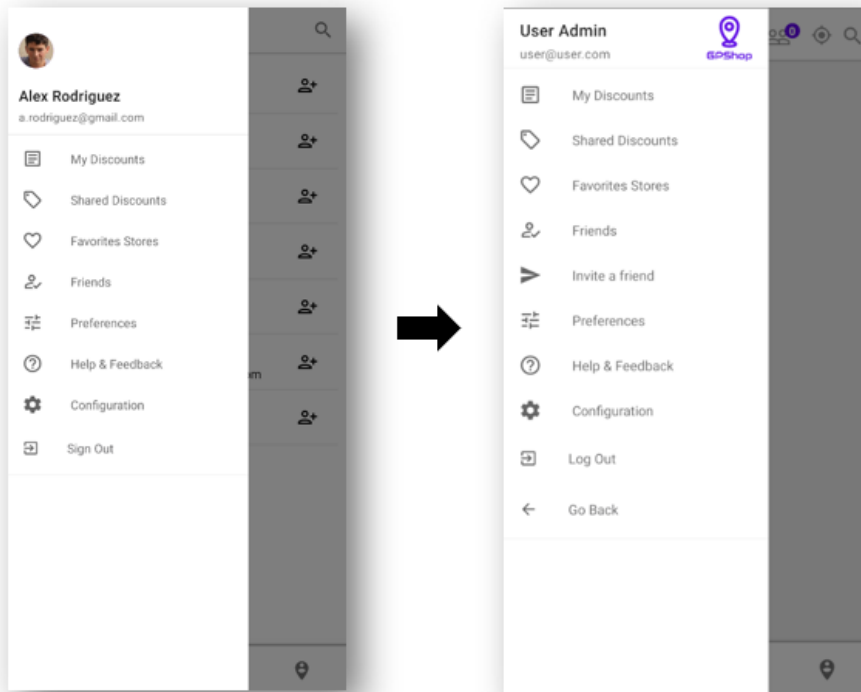


Figura 7.25: Evolución del *hamburger menu*

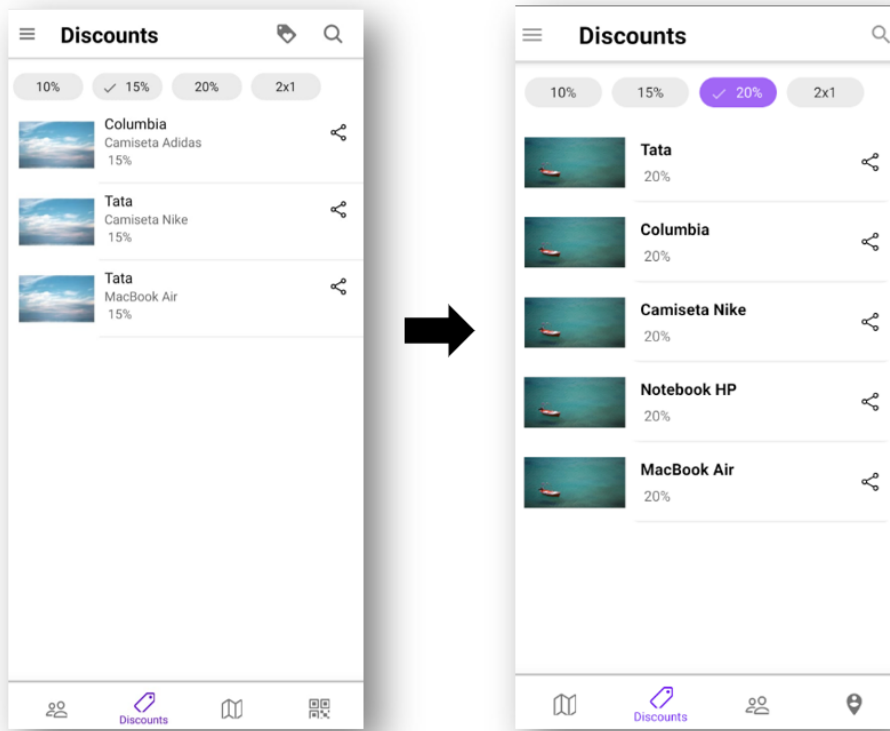


Figura 7.26: Evolución de pantalla de *discounts*

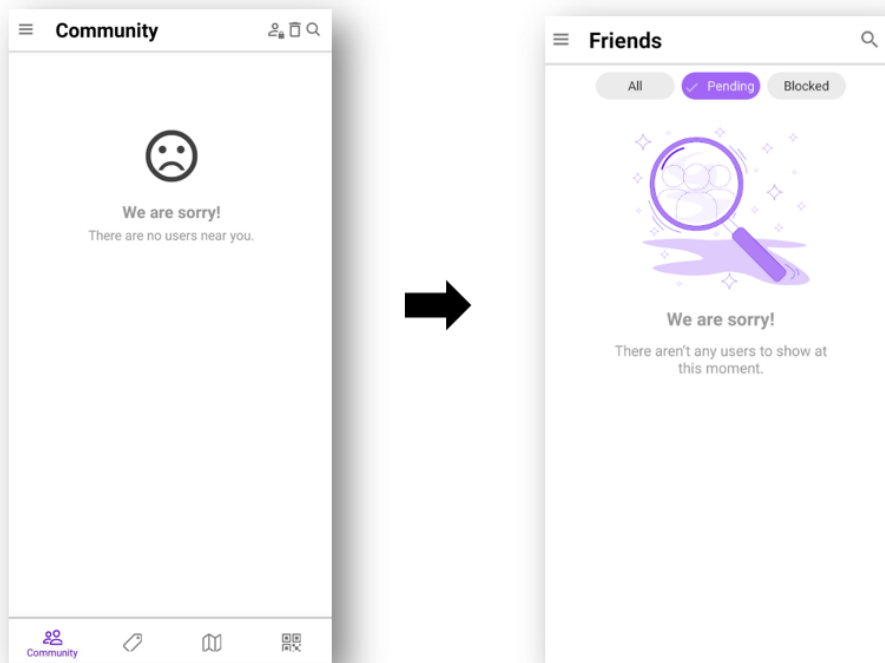


Figura 7.27: Evolución de *empty state* de usuarios

Heurísticas de Nielsen

Las Heurísticas de Nielsen son un conjunto de reglas heurísticas y de buenas prácticas para el desarrollo de la interacción de interfaz humano-computador. En el caso de nuestro sistema, los usuarios interactúan por medio de *frontend*, ya sea a través de la aplicación móvil como por la *webapp*. Existen un conjunto de leyes no escritas, las cuales un usuario final espera implícitamente que sean cumplidas en un sistema. Vale la pena agregar, que muchas ya se cumplían debido al diseño obtenido de Figma, lo cual aportó respaldo al trabajo al haber sido diseñado por un experto UI/UX. Se hace un repaso a continuación de la aplicación de las principales heurísticas aplicadas al *frontend*.

- Visibilidad del estado del sistema: El usuario debe conocer el estado actual del sistema, por ejemplo, cuando se está obteniendo información desde *backend* o la interacción no está disponible brevemente.

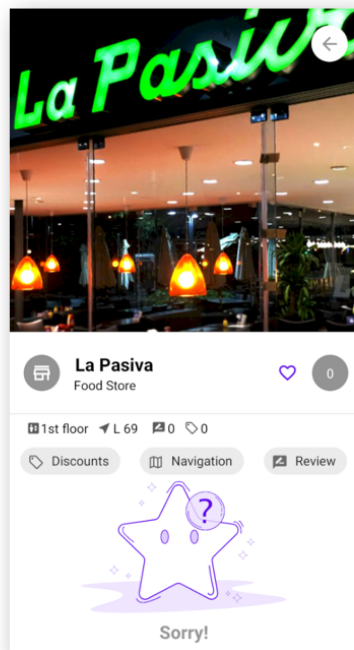


Figura 7.28: *Empty state* de store.

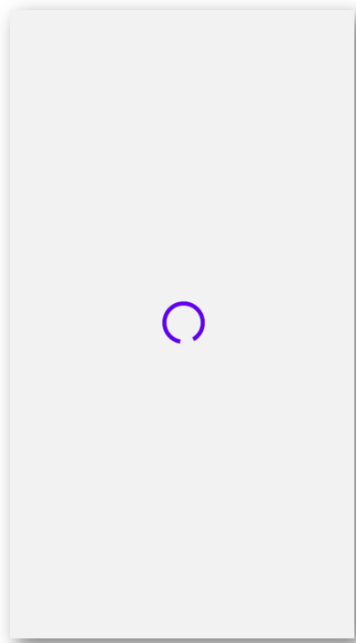


Figura 7.29: *Loading screen*

- Correspondencia entre el sistema y el mundo real: Se respetó el lenguaje y se proveyeron los datos textuales, iconográficos y de otras índoles, presentados en una manera que se adecuaba al contexto y al público objetivo.
- Libertad y control del usuario: Existen atajos para poder retroceder en caso que el usuario desee hacerlo. Estos además, se encuentran en posiciones donde son estándar en cualquier aplicación *web* o *mobile*. En algunos casos se puede realizar la misma acción de diferentes maneras; a su vez, existe personalización a nivel de sistema operativo, ya que algunos gestos tienen dinámicas diferentes en Android y iOS.
- Consistencia y estándares: Se intentó preservar la consistencia a diferentes niveles, desde el lenguaje utilizado a la forma en que se lanzan las animaciones y cómo se muestran los mensajes de errores.
- Prevención de errores: Se tomaron en cuenta posibles errores de los usuarios, y se proveyeron medidas para alertarles antes de cometerlos. En caso de que hayan incurrido conscientemente en errores, se les informa de la situación y se ofrecen alternativas para recuperarse de ellos.
- Reconocimiento en lugar de intentar recordar: Se mantiene información contextual acerca de la posición del usuario en la aplicación debidamente presentada. Para citar casos específicos, esto aplicaría en la búsqueda por ejemplo, recordando los últimos resultados, o informando en qué pantalla el usuario está posicionado.
- Flexibilidad y eficiencia de uso: Existen atajos y diferentes formas de realizar acciones, que están disponibilizadas para los usuarios más experimentados, sin comprometer la usabilidad para otros usuarios.
- Estética y diseño minimalista: Para este punto, se destaca especialmente la implantación de Material Design. Gracias a ello, se alcanza un diseño simple de entender, de buena usabilidad pero que a su vez no está sobrecargada de elementos que puedan obstruir una buena experiencia y navegación.
- Reconocimiento, diagnóstico y recuperación de errores: En caso de ocurrir un error se le informa al usuario indicando el problema en forma clara y amigable.
- Ayuda y documentación: Se priorizó proporcionar información y explicaciones de manera de que no se necesitara documentación adicional para promover la usabilidad en *frontend*. En los casos en que ameritó, se dio información adicional. [15]

Metodologías ágiles y su rol en gestión de la calidad

No sólo es necesario hacerse de herramientas y realizar un control exhaustivo a nivel de código sin antes contar con un proceso que propicie la generación de un producto de calidad. Esto influirá no solo en el producto en sí, sino también en los demás elementos de configuración como documentación, *releases* y repositorio de incidentes.

Uno de los beneficios de optar por un ciclo de vida iterativo en desarrollo fue el de la oportunidad de mejora constante. En específico, se utilizó Scrum, en conjunto con Kanban. Esto permitió continuar optimizando el producto y la aplicación de metodologías ágiles, lo cual generó un circuito en el que se intentó aplicar lecciones aprendidas e integrar mejoras solicitadas tanto en el plano de *software* como en documentación.

Estándares

Para el ámbito de *frontend* se definió en conjunto con el equipo un *set* de reglas pre-acordadas, las cuales fueron integrados al IDE utilizado (Visual Studio Code) para mantener consistencia entre los ambientes de desarrollo. Esto permitió realizar auto-indentación, enforzar reglas del tipo -pero no limitadas a (EOF) *end-of-file-* y mantener un estilo consistente a través del código de *frontend*.

7.6.4. Revisiones

Gracias a la experiencia y conocimiento de FB se contó con un seguimiento y exhaustivo del manejo de los repositorios, donde además de ello, se realizaron pruebas cruzadas de revisión de código al momento de realizar *pull requests*.

Estas instancias además propiciaron la discusión acerca de estándares y oportunidades de mejora. También permitieron optimizar el flujo de trabajo y validar que se estuviesen cumpliendo los estándares definidos con la rigurosidad correspondiente. Cabe destacar, que a medida que se ganó experiencia a nivel de equipo, estas instancias sirvieron para comprobar -a través de la cantidad de *story points* completados por ejemplo- el aumento en la efectividad de la aplicación de las reglas definidas.

Para darle mayor visibilidad y agilizar la integración de nuevo código a las ramas estables se intentó enviar *pull requests* en iteraciones de tiempo más acotadas de manera de que además de reducir la complejidad y posibilidad de conflictos, lograr

un número mayor de revisiones de código.

7.6.5. Validaciones

La validación es un proceso amplio en el que se evalúa el resultado del producto con lo que espera el cliente acerca del mismo y poder saber si alcanza sus expectativas. [11]

Fase previa / Ingeniería de Requerimientos

Se realizaron diferentes entrevistas con potenciales usuarios, los cuales forman parte del cliente. Las validaciones realizadas con ese grupo de usuarios fueron consideradas como factor de éxito ya que permitió validar la usabilidad y utilidad del producto.

Las pruebas realizadas con *wireframes* se consideran exitosas ya que proporcionaron un punto de partida para comenzar la construcción de *software*.

Uno de los aspectos que diferenciaron al proyecto fue el desafío de integrar la librería de geolocalización; existió un elevado grado de complejidad que afectó transversalmente al mismo. Por un lado, estuvo la necesidad de comprensión de las necesidades del cliente, las capacidades de la librería, entender detalles técnicos de la misma, y por otro lado, poder integrar un *stack* tecnológico a la misma para obtener un producto que demostrara el potencial de la propuesta de valor. Si bien se hizo una evaluación previa de tecnologías, al utilizar un lenguaje distinto al utilizado para la librería se necesitó recurrir a *bridges* para lograr que la librería y la aplicación funcionasen en conjunto.

Cuando sucede que la aplicación no permite localizar al usuario o lo hace con dificultad, se despliega una alerta informando de la situación y se ofrece reintentar, para mantener al usuario informado y minimizar el impacto de una experiencia no satisfactoria.

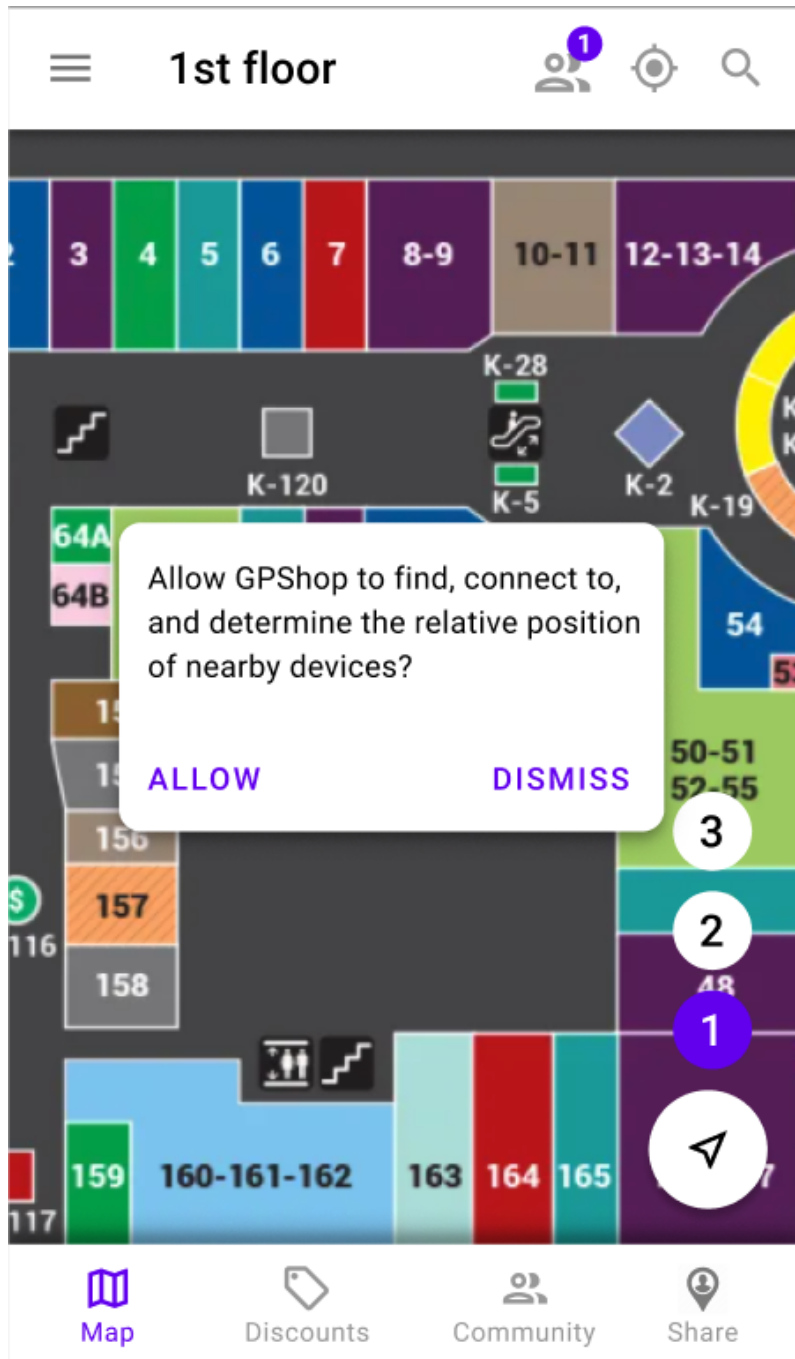


Figura 7.30: Mensaje para solicitud de permisos de GPSShop.

Validación del producto

En el caso de la aplicación se realizaron demos a final de *sprint* de las que participaban *stakeholders* como el cliente, tutor, y donde se exponían las últimas funcionalidades desarrolladas. Estas se comenzaron a realizar luego de comenzar a contar con versiones funcionales de *frontend*, lo que sucedió a partir del *sprint* 4.

El objetivo de dichas presentaciones fue contar con una muestra dinámica del producto funcionando como un todo, donde se generara un espacio para discutir posibles mejoras al mismo tiempo de cerrar funcionalidades para poder continuar avanzando a buen ritmo.

Uno de los elementos centrales en la interacción del sistema con el entorno son los *beacons* y por lo tanto deben ser debidamente testeados para realizar validaciones sobre el sistema como un todo. Como fue mencionado anteriormente, los *beacons* son utilizados para la ubicación del usuario en ambientes interiores. Una falla o una sucesión de las mismas durante la interacción con la aplicación impacta en forma negativa la experiencia de los usuarios reduciendo en forma catastrófica la percepción de calidad.

La validación con *beacons* se llevó a cabo en un entorno cerrado de manera de poder comprobar la efectividad y respuesta de la aplicación en diferentes situaciones con *beacons*. Luego de las pruebas se comprobó que la aplicación cumplía en forma adecuada y se tomaron en cuenta las situaciones que no se consideraron óptimas, para realizar los ajustes pertinentes a nivel de código. La interacción con el *backend* fue exitosa. Se realizaron pruebas con una conexión LTE de aproximadamente *5Mbps/2Mbps*. La aplicación ofreció los descuentos esperados por el sistema contra las tiendas que se asociaron por medio de *beacons* al sistema.

7.6.6. Defectos

Para el registro de defectos se aprovechó la suite de Azure DevOps, donde se ingresaban al momento de encontrarse, para tener centralizados los defectos -señalados en el *board* como *bugs*- en conjunto con las tareas, de manera tal de no perder visibilidad. Se decidió asignarles un riesgo, con una escala de 1 a 5, donde 1 es el menor riesgo, y en caso de tratarse de defectos más complejos se los trató como *carry-over* hacia el siguiente *sprint*.

7.6.7. Verificaciones

Las verificaciones se realizaron a diferentes niveles del producto. Existieron tanto a nivel de aplicación, *backend* y demás elementos de configuración como el código y la documentación.

La estrategia de pruebas se compuso de revisiones cruzadas durante *pull requests*, pruebas unitarias, pruebas de integración, pruebas de campo con *beacons* y evaluaciones específicas de aseguramiento de la calidad con el *product owner*.

7.6.8. Pruebas unitarias

Se escogió realizar pruebas automatizadas para el código de forma tal de contar con un proceso determinístico para testear el código. Por otra parte, esto redujo contratiempos debido a contar con una base segura desde donde partir para expandir funcionalidades.

Otro de los propósitos de contar con una adecuada gestión de la calidad es poder realizar cambios con mayor velocidad, esto es debido a que no se debe lidiar con defectos, los cuales pueden reducir considerablemente la efectividad de los desarrolladores al agregar código.

7.7. Gestión de comunicación

Gestionar de manera eficaz y ordenada la comunicación durante el proyecto fue una pieza clave para el desarrollo del mismo, en razón que durante el transcurso de la realización de la tesis, nos encontramos afectados por la emergencia sanitaria, fue de total acuerdo por parte del equipo evitar la organización de reuniones presenciales (preservando nuestras respectivas burbujas) tanto para instancias internas como externas con terceros involucrados en el desarrollo del proyecto, generándose las nombradas anteriormente de manera virtual.

Apoyándonos en el libro PMBOK [16] con sus tres procesos definidos para esta gestión; es a partir de aquí en donde surge el siguiente plan de comunicación, el cual detalla la interacción con los diferentes interesados.

A continuación, se presentan los distintos canales de comunicación utilizados para comunicarnos con los distintos interesados del proyecto, incluyendo al equipo internamente, cliente, tutor y universidad.

7.7.1. Interesados

Para obtener una buena gestión de comunicación, como primera instancia nos fue necesario identificar los interesados que iban a tener interacción en el proyecto, con esta intención, es que se genera la siguiente tabla:

Actor	Rol
Universidad ORT	Institución académica
Oscar Sena	tutor
Globant	Patrocinadores
Federico Barceló	Integrante del equipo
Alejandro Kwilman	Integrante del equipo
Pablo Pereira	Integrante del equipo
Makcim Silveira	Integrante del equipo

Tabla 7.3: Tabla de interesados

7.7.2. Canales de comunicación

Durante el proyecto se utilizaron los siguientes canales de comunicación entre los interesados:

- Correo electrónico
- Google Meet
- Microsoft Planner
- Microsoft Teams
- Slack
- WhatsApp

En la siguiente tabla se presentan los distintos canales, sus respectivos usos por parte de los interesados y ejemplos de usos para los mismos.

Canal	Uso	Ejemplo de uso
Correo electrónico	Utilizado principalmente para envío de información relevante con la universidad y cliente, además de coordinar las reuniones virtuales.	<ul style="list-style-type: none"> ▪ Envío de información a la Universidad. ▪ Envío de información al cliente. ▪ Envío de citas para reuniones virtuales.
Google meet	Utilizado para la celebración de las reuniones planteadas tanto con el equipo interno como externo.	<ul style="list-style-type: none"> ▪ <i>Dailys.</i> ▪ <i>Weekly sync</i> con cliente. ▪ <i>Planning.</i> ▪ <i>Retro.</i> ▪ <i>Extended session.</i> ▪ Sesiones extras del equipo.
Microsoft Planner	Utilizado para comunicación sobre información de interés tecnológico, guías o manuales.	<ul style="list-style-type: none"> ▪ Guías/manuales de uso. ▪ Links de interés.
Microsoft Teams	Utilizado como canal de comunicación con el tutor.	<ul style="list-style-type: none"> ▪ Weekly sync con el tutor. ▪ Comunicación directa con el tutor.
Slack	Utilizado como canal de comunicación rápida con el cliente.	<ul style="list-style-type: none"> ▪ Comunicación directa con el cliente.
WhatsApp	Utilizado como canal de comunicación informal entre los miembros del equipo de tesis. También utilizado en manera de urgencia para comunicación con el tutor o con el cliente.	<ul style="list-style-type: none"> ▪ Comunicación de carácter urgente con el cliente o el tutor. ▪ Comunicación distendida entre miembros del equipo. ▪ Comunicación sobre pormenores que impacten con agendas ya pautadas.

Tabla 7.4: Canales de comunicación

7.7.3. Plan de comunicación

A continuación podemos observar las distintas interacciones que se dieron entre los interesados, en donde se indica cual es la información que se entrega, quien es el emisor, quien lo recibe, por donde recibe esta información y con qué frecuencia recibe la misma.

Podemos observar que la mayoría de estas interacciones mantuvieron una frecuencia estable ya que se trataban de ceremonias o reuniones ya definidas previamente con los interesados.

¿Qué se comunica?	¿Quién lo comunica?	¿A quién se lo comunica?	¿Por donde se comunica?	¿Cual es la frecuencia con que se comunica?
Consultas sobre temas en particular u cuestiones universitarias.	Miembro del equipo.	tutor.	Reuniones mediante Microsoft teams.	Cuando surge la duda.
Validación sobre US definidas.	SM	cliente.	Reuniones mediante Google Meet.	Cuando se cree necesario.
Cambios sobre riesgos identificados.	PM	Equipo del proyecto.	Reuniones mediante Google Meet.	Bisemanalmente.
Bloqueos que impiden trabajo ya pautado.	Miembro del proyecto.	Equipo del proyecto.	<ul style="list-style-type: none"> ▪ Reuniones mediante Google Meet. ▪ Mensajería instantánea. 	Diariamente.
Avance del trabajo ya pautado.	Miembro del proyecto.	Equipo del proyecto.	<ul style="list-style-type: none"> ▪ Reuniones mediante Google Meet. ▪ Mensajería instantánea. 	Diariamente.
Desviaciones en el trabajo ya pautado.	PM	Equipo del proyecto.	Reuniones mediante Google Meet.	Semanalmente.
Presentación de avances sobre entregables del proyecto.	Equipo del proyecto	<ul style="list-style-type: none"> ▪ tutor. ▪ cliente. 	Reuniones mediante Google Meet.	Semanalmente.
Trabajo a realizar	Miembro de proyecto	Equipo del proyecto.	Reuniones mediante Google Meet.	Diariamente.
Plan de SQA	Responsable de SQA.	Equipo de proyecto.	Reuniones mediante Google Meet.	Cuando este definido el plan de SQA.
Plan de SCM	Responsable de SCM.	Equipo de proyecto.	Reuniones mediante Google Meet.	Cuando este definido el plan de SQA.

Tabla 7.5: Plan de comunicación

7.8. Gestión de incidentes

Para gestión de incidentes se mantuvo una planilla durante el proyecto, en la cual los usuarios podían reportar incidentes y los cuales se priorizaban desde gravedad baja a gravedad alta. Una vez resueltos los incidentes se los calificaba como Resuelto. (dirigirse a anexo 10.126 por más detalles).

8. Conclusiones y lecciones aprendidas

8.1. Estado actual

A pesar del desafío tecnológico que significó llevar a cabo un proyecto de este porte, el sistema desarrollado logró cumplir con los requerimientos pactados con el cliente y objetivos establecidos al comienzo del proyecto. Se alcanzó un MVP que comprueba la viabilidad en el uso de la geolocalización (sobre todo interna) en un contexto específico (centros comerciales). Es de destacar, además, que no sólo se cumplieron todos los objetivos sino que ya se está utilizando parte del producto para otro contexto.

El mismo consiste en la utilización del *backend* para asistir a miembros del cliente a localizar diferentes puntos de interés dentro de sus oficinas. Esto, por otra parte, muestra que se puede aplicar la geolocalización a otros ámbitos y del éxito que significa que el cliente concuerde con la visión del equipo y haya podido visualizar el potencial del producto, incluso antes de haber llegado a la instancia de entrega académica.

8.2. Opinión del cliente

El concepto transmitido por parte del cliente para al equipo fue muy positivo y alentador, dejando en claro que se logró realizar un buen entendimiento y análisis del producto a obtener, superando todas las expectativas pautadas sobre todos los ámbitos.

Resaltando la constante comunicación por parte del equipo y disponibilidad cuando se necesitaba de algún miembro del mismo para solucionar ciertos problemas que fueron surgiendo durante la etapa de desarrollo.

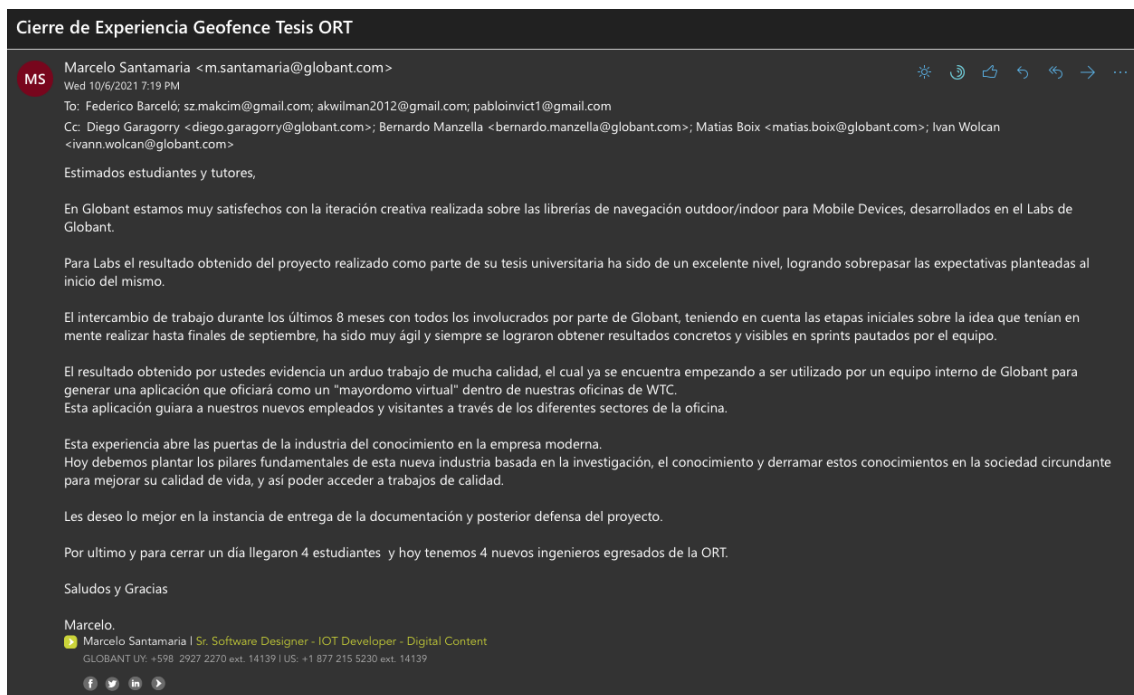


Figura 8.1: Correo por parte de Marcelo Santamaria, *IoT Platform Engineer*.

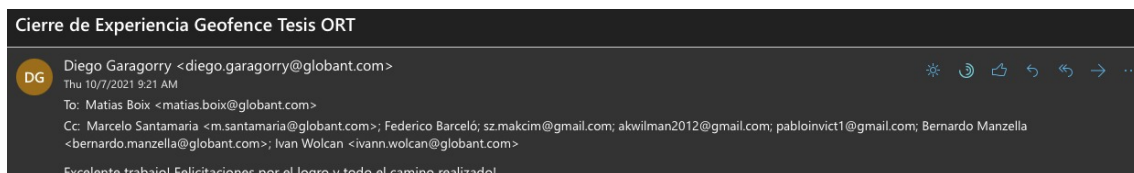


Figura 8.2: Respuesta por parte de Diego Garagorry, *TDC Uruguay & Chile*.

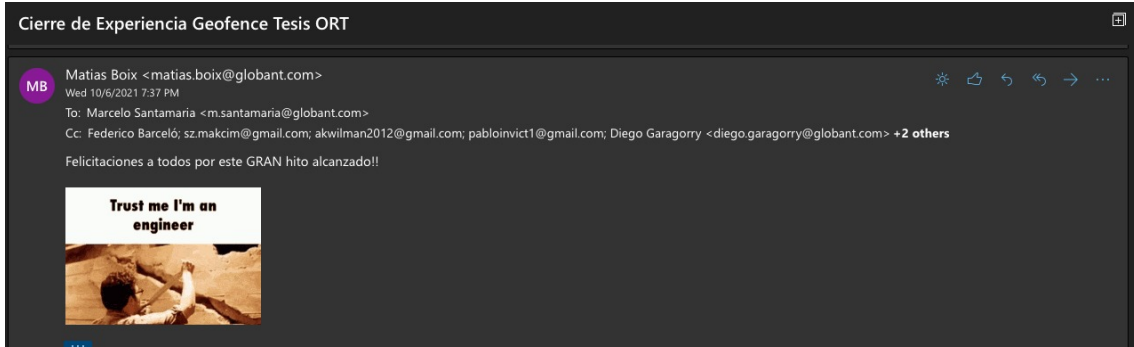


Figura 8.3: Respuesta por parte de Matías Boix, *Tech Partner Data and AI*.

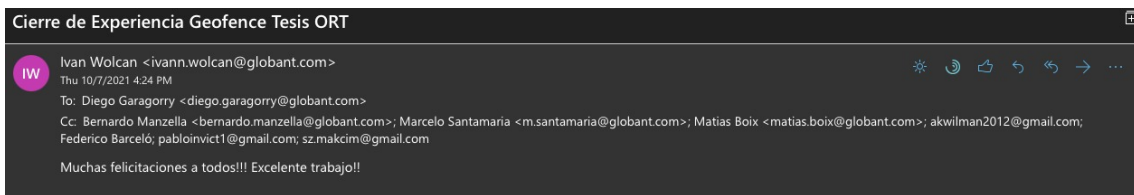


Figura 8.4: Respuesta por parte de Iván Wolcan, *Subject Matter Expert*.

8.3. Conclusiones generales

8.3.1. Objetivos del equipo

A nivel de equipo existió una evolución tanto en lo personal como en el ámbito colectivo. Por otro lado, el proyecto propició un escenario donde se pudieron aplicar los variados conocimientos adquiridos a lo largo de la carrera académica y laboral, combinándolos con lo transmitido por nuestro tutor y el cliente, los cuales fueron instrumentales en el camino al éxito.

El conjunto de tecnologías utilizado fue otro de los obstáculos que debieron de ser sorteados. Esto fue logrado por medio de diferentes maneras pero siempre con el objeto de respetar los estándares establecidos, lo esperado por el cliente y apoyándose en las metodologías escogidas las cuales fueron fundamentales para no desviarse de los objetivos y el *roadmap* pactado.

8.3.2. Objetivos del cliente

Se logro un producto base que utilice como *core* la librería GeoFence, la cual no se esta utilizando en ningún desarrollo interno ni para clientes. Pasando a tener un producto el cual va a poder ser utilizado por otros desarrollos sin importar su contexto de uso.

8.3.3. Objetivos del proyecto

Se logro realizar un proyecto que integre varias tecnologías y en practica sobre un contexto real los procesos de gestión de proyecto e ingeniería de software. Logrando así un producto que respete los atributos de calidad acordados.

8.3.4. Objetivos del producto

Por medio de un uso innovador de la geolocalización se logró obtener un producto atractivo del punto de vista de negocio, logrado en combinación con el *stack* tecnológico requerido y a nivel de hardware logrado por medio de *beacons*.

8.3.5. Académicos

Según lo pautado por la universidad, se logró generar realizar una tesis que englobe muchos aspectos adquiridos durante el proceso de formación estudiantil en conjunto con una empresa de plaza, logrando obtener un trabajo final de tesis exitoso por cumplir con lo pautado con el cliente y las normativas universidad vigentes.

8.4. Lecciones aprendidas

Los aprendizajes que obtuvimos durante el proyecto fueron variados y de diferentes índoles, desde aspectos tecnológicos, trabajo en equipo, manera de comunicarnos con los diferentes actores interesados, gestión de un proyecto real y sobre todo el llevar a cabo el desarrollo de un producto para un cliente durante el contexto sanitario de los últimos dos años.

Sobre lo tecnológico tuvimos un gran aprendizaje. Antes de comenzar con el proyecto no contábamos con gran experiencia sobre implementaciones que utilizaran herramientas de geolocalización, como tampoco integraciones con estructuras de *backend* y *frontend*. Teniendo también aprendizaje sobre el uso de nuevas herramientas y *frameworks* propios de las diferentes tecnologías.

Se logró aprender sobre los aspectos anteriores para poder realizar un producto de calidad y que supere las expectativas esperadas por el cliente, siguiendo las mejores prácticas obtenidas en las diferentes materias de desarrollo durante la carrera en la universidad.

A nivel de gestión, nos fue posible trabajar sobre un proyecto en un contexto real para un cliente (con las exigencias y cuestiones que esto implica), logrando poner en práctica todas las etapas referidas a la gestión de proyectos e Ingeniería de Software.

En cuanto al trabajo en equipo, los integrantes del equipo ya se conocían entre sí por haber compartido trabajos con anterioridad en otras asignaturas, por lo cual mediante una buena distribución de las tareas a realizar en función de los roles, logramos que el trabajo fluyera correctamente, la comunicación entre los miembros del equipo y el cliente fuera ágil y el ambiente del equipo fuera bueno.

Se espera por parte del equipo poder volcar toda la experiencia y conocimientos adquiridos durante el transcurso del proyecto en nuevos proyectos ya sea en ámbitos laborales o personales.

8.5. Próximos pasos

Desde el principio del proyecto, la idea era poder obtener una plataforma base que pudiera ser utilizada sin importancia del contexto en donde quisiera emplearse una solución que utilice la geolocalización.

Al día de hoy ya se encuentra desarrollando una aplicación *inhouse* por parte de Globant para uso en las oficinas de WTC, además se están llevando a cabo estudios de viabilidad con el fin de realizar ofrecimientos a clientes con soluciones que integren la plataforma base obtenida con el proyecto.

Por parte del cliente se está teniendo en consideración la contratación de los miembros del equipo que no son *Globers* para que puedan aportar con el *know-how* obtenido durante el proyecto a futuros productos que integren la plataforma base.

9. Referencias bibliográficas

- [1] R. S. Pressman, *Software engineering: a practitioner's approach*, 3rd ed. New York, USA: McGraw-Hill, Inc., Jan. 1992.
- [2] G. D. Gil, “Herramienta para implementar LEL y escenarios (TILS),” Tesis, Universidad Nacional de La Plata, Buenos Aires, Argentina, 2002. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/4057>
- [3] Microsoft, “What is azure devops?” Sep. 2021. [Online]. Available: <https://docs.microsoft.com/en-us/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops&tabs=agile-process>
- [4] Microsoft, “Agile process work item types and workflow,” Dec. 2018. [Online]. Available: <https://docs.microsoft.com/en-us/azure/devops/boards/work-items/guidance/agile-process-workflow?view=azure-devops>
- [5] Estimote, “Estimote, home.” [Online]. Available: <https://estimote.com/>
- [6] ISO/IEC, “Systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models.” [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>
- [7] R. C. Martin, *Clean code: a handbook of agile software craftsmanship*. Stoughton, Massachusetts: Pearson, Aug. 2008.
- [8] R. Community, “Typechecking with proptypes.” [Online]. Available: <https://reactjs.org/docs/typechecking-with-proptypes.html>
- [9] Kent Beck et al., “Manifesto for agile software development.” [Online]. Available: <https://agilemanifesto.org/>
- [10] Project Management Institute, Ed., *Guide to the Project Management Body of Knowledge (PMBOK Guide) and the Standard for Project Management*, 7th ed. Pennsylvania, USA: Project Management Institute, Inc, Jan. 2021.
- [11] I. Sommerville, *Software Engineering*, 10th ed. Essex, England: Pearson Education, 2016.

- [12] J. F. Smart, *BDD in Action*. Shelter Island, NY, USA: Charles Scribner's Sons, 2014.
- [13] SmartBear, "What is cucumber?" [Online]. Available: <https://cucumber.io/docs/guides/overview/>
- [14] B. Konstantin Kudryashov, "behat documentation." [Online]. Available: <https://docs.behat.org/en/latest/guides.html>
- [15] J. Nielsen, "10 usability heuristics for user interface design," on Apr. 24, 1994; Updated Nov. 15, 2020. [Online]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [16] Project Management Institute, Ed., *A guide to the project management body of knowledge (PMBOK guide)*, 5th ed. Pennsylvania, USA: Project Management Institute, Inc, Jan. 2013.
- [17] Google, "Firebase app distribution," accessed on: 14/08/2021. [Online]. Available: <https://firebase.google.com/docs/app-distribution>
- [18] arduino.cc, "Arduino ide." [Online]. Available: <https://www.arduino.cc/en/software>
- [19] Microsoft, "Backlogs, portfolios, and agile project management," Jan. 2021. [Online]. Available: <https://docs.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops?view=azure-devops>
- [20] K. A. from Southern New Hampshire University, "What is back end development?" 2021. [Online]. Available: <https://www.kenzie.academy/blog/front-end-vs-back-end-whats-the-difference/>
- [21] A. F. Osborn, *Principles and Procedures of Creative Problem Solving*. New York, USA: Charles Scribner's Sons, 1953.
- [22] M. U. B. Aftab, *Building Bluetooth Low Energy Systems*. Birmingham, UK: Packt Publishing, May 2017.
- [23] P. Christensson, "Ble definition." [Online]. Available: <https://techterms.com/definition/ble>
- [24] Docker, "Use containers to build, share and run your applications." [Online]. Available: <https://www.docker.com/resources/what-container>

- [25] STEN PITTEP, “What is continuous deployment?” [Online]. Available: <https://www.atlassian.com/continuous-delivery/continuous-deployment>
- [26] S. PITTEP, “What is continuous integration?” [Online]. Available: <https://www.atlassian.com/continuous-delivery/continuous-integration>
- [27] Microsoft, “What is devops?” [Online]. Available: <https://azure.microsoft.com/en-us/overview/what-is-devops/>
- [28] E. Team, “What is firebase?” [Online]. Available: <https://www.educative.io/edpresso/what-is-firebase>
- [29] D. Morales, “What’s the difference between frontend and backend?” [Online]. Available: <https://medium.com/geekculture/whats-the-difference-between-frontend-and-backend-47ac556c863d>
- [30] Scrum Manager BOK, “Epic.” [Online]. Available: <https://www.scrummanager.net/bok/index.php?title=Epic>
- [31] Expo, “Introduction to expo.” [Online]. Available: <https://docs.expo.dev/>
- [32] DataCentric, “Cómo distinguir entre geolocalización y georeferenciación.” [Online]. Available: <https://www.datacentric.es/blog/geomarketing/diferencia-entre-geolocalizacion-y-georeferenciacion/>
- [33] DataCentric, “Cómo distinguir entre geolocalización y georeferenciación.” [Online]. Available: <https://www.datacentric.es/blog/geomarketing/diferencia-entre-geolocalizacion-y-georeferenciacion/>
- [34] Happn, “Encuentra con quien te has cruzado.” [Online]. Available: <https://www.happn.com/es/>
- [35] C. University, “Meaning of heuristic in english.” [Online]. Available: <https://dictionary.cambridge.org/dictionary/english/heuristic>
- [36] N. Campana, “What does an iot engineer do?” [Online]. Available: <https://www.freelancermap.com/blog/what-does-an-iot-engineer-do/>
- [37] R. S. . Medium, “Metro bundler.” [Online]. Available: <https://rishabh0297.medium.com/role-of-metro-bundler-in-react-native-24d178c7117e>
- [38] D. RADIGAN, “¿qué es kanban?” [Online]. Available: <https://www.atlassian.com/es/agile/kanban>

- [39] Google, “Kotlin faq.” [Online]. Available: <https://kotlinlang.org/docs/faq.html>
- [40] Google, “Guidelines.” [Online]. Available: <https://material.io/design/guidelines-overview>
- [41] Mapwize, “Indoor mapping —& wayfinding for smart buildings.” [Online]. Available: <https://www.mapwize.io/>
- [42] Microsoft, “¿qué es azure?” [Online]. Available: <https://azure.microsoft.com/es-es/overview/what-is-azure/>
- [43] Microsoft, “When to use microsoft project, planner, to do, or the tasks app in teams.” [Online]. Available: <https://support.microsoft.com/en-us/office/when-to-use-microsoft-project-planner-to-do-or-the-tasks-app-in-teams-8f950d32-d5f4-40db-a8>
- [44] E. Ries, “Minimum viable product: a guide.” [Online]. Available: <http://www.startuplessonslearned.com/2009/08/minimum-viable-product-guide.html>
- [45] Disney, “Deja que my disney experience sea tu guía.” [Online]. Available: <https://disneyworld.disney.go.com/es-us/plan/my-disney-experience/>
- [46] R. C. Martin, “Design principles and design patterns, principles of object oriented class design, the open closed principle.” [Online]. Available: https://fi.ort.edu.uy/innovaportal/file/2032/1/design_principles.pdf
- [47] Overleaf, “Who we are.” [Online]. Available: <https://es.overleaf.com/about>
- [48] Scrum.org, “What is a product owner?” [Online]. Available: <https://www.scrum.org/resources/what-is-a-product-owner>
- [49] Niantic, “¡os damos la bienvenida, entrenadores!” [Online]. Available: <https://pokemongolive.com/es/>
- [50] TechSlang, “What is a proof of concept (poc)?” [Online]. Available: <https://www.techslang.com/definition/what-is-a-proof-of-concept/>
- [51] QRACorp, “What is a functional requirement?” [Online]. Available: <https://qracorp.com/functional-vs-non-functional-requirements/>
- [52] QRACorp., “What is a non-functional requirement?” [Online]. Available: <https://qracorp.com/functional-vs-non-functional-requirements/>
- [53] K. Schwaber and J. Sutherland, “The definitive guide to scrum: The rules of the game.” [Online]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>

- [54] SonarQube, “Sonarqube documentation.” [Online]. Available: <https://docs.sonarqube.org/latest/>
- [55] R. Mourya, “Sonarqube — sonarscanner : a step by step guide.” [Online]. Available: <https://medium.com/swlh/sonarqube-sonarscanner-setup-1a633b654828>
- [56] Atlassian, “¿qué es el control de versiones?” [Online]. Available: <https://www.atlassian.com/es/git/tutorials/what-is-version-control>
- [57] C. University, “stakeholder.” [Online]. Available: <https://dictionary.cambridge.org/es/diccionario/ingles/stakeholder>
- [58] S. Rodriguez, “Meet swarm: 5 things you need to know about foursquare’s new app.” [Online]. Available: <https://www.latimes.com/business/technology/la-fi-tn-foursquare-swarm-app-5-things-20140515-story.html>
- [59] Apple, “Swift.” [Online]. Available: <https://developer.apple.com/swift/>
- [60] Tinder, “¿qué es tinder?” [Online]. Available: <https://www.help.tinder.com/hc/es-es/articles/115004647686--Que-es-Tinder->
- [61] Google, “Cómo funciona waze.” [Online]. Available: <https://support.google.com/waze/answer/6078702?hl=es>

Glosario

AK: Alejandro Kwilman, estudiante de Universidad ORT Uruguay.

App Distribution: Proceso por el cual se distribuye la aplicación para que los *testers* puedan realizar pruebas sobre ella y realizar verificaciones.[17]

Arduino IDE: Entorno de desarrollo de Arduino.[18]

Azure Boards: Es la herramienta de *Azure* para controlar los procesos y el flujo de trabajo, cuenta con tableros e interfaces para el control de los *Sprints* y demás ceremonias inherentes a los marcos de trabajo ágil. Es una de las cinco herramientas de *Azure DevOps*. [3]

Azure DevOps: Es una plataforma que cuenta con cinco herramientas desde las cuales se puede implementar el desarrollo de los distintos proyectos de software, bajo la metodología *DevOps*. [19]

Backend: Parte del desarrollo que hace que el sistema sea interactivo. Es la parte abocada al manejo de la lógica de negocio, también conocida como *server-side*. La principal diferencia con el *frontend* es la ausencia de interacción directa con el usuario. [20]

BDD: Conjunto de prácticas de ingeniería que permiten crear *software* de mayor valor, enfocándose en las reglas de negocio. Se apoya en TDD (*Test Driven Development*). También se lo señala como un enfoque "Out-in" debido a que se hace mayor foco en la parte visible para el usuario al escribir pruebas. [12]

Brainstorming: Es una herramienta de trabajo grupal que facilita el surgimiento de nuevas ideas sobre un tema o problema determinado. La lluvia de ideas es una técnica de grupo para generar ideas originales en un ambiente relajado.[21]

Beacons: Es una baliza *Bluetooth*: de corto alcance única por cada dispositivo, que envía señales de forma que las aplicaciones de los *Smartphones*: o *Tablets*: reconocen estas señales y dichas señales son capaces de lanzar notificaciones sobre dichos dispositivos móviles.[22]

Behat: *Framework* utilizado para realizar BDD sobre PHP. [13]

BLE: Bluetooth Low Energy, variacion del estandar Bluetooth para bajo consumo de energía.[23]

Containers: Un *Container* es una unidad estándar de software que empaqueta el código y todas sus dependencias para que la aplicación se ejecute de forma rápida y confiable de un entorno informático a otro. Una imagen de contenedor es un paquete de software ligero, independiente y ejecutable que incluye todo lo necesario para ejecutar una aplicación: código, tiempo de ejecución, herramientas del sistema, bibliotecas del sistema y configuraciones.[24]

Continuous deployment (CD): La implementación continua (CD) es un proceso de lanzamiento de software que utiliza pruebas automatizadas para validar si los cambios en la base del código son correctos y estables para la implementación autónoma inmediata en un entorno de producción.[25]

Continuous integration (CI): La integración continua (CI) es la práctica de automatizar la integración de cambios de código de múltiples contribuyentes en un solo proyecto de software.[26]

Cucumber: Framework utilizado para realizar BDD sobre diferentes lenguajes, para el proyecto en específico, JavaScript. [13]

DevOps: DevOps (acrónimo en inglés de *development* -desarrollo- y *operations* -operaciones-). Es la unión de personas, procesos y tecnología para proporcionar valor a los clientes de forma continua.[27]

FBZ: Fabian Bozoglilianian, *Team Agilist* en Globant.

FB: Federico Barceló, estudiante de Universidad ORT Uruguay.

Firestore: Firestore es un backend-as-a-Service (Baas). Proporciona a los desarrolladores una variedad de herramientas y servicios para ayudarlos a desarrollar aplicaciones de calidad, hacer crecer su base de usuarios y obtener ganancias. Está construido sobre la infraestructura de Google.[28]

Frontend: Parte visible para el usuario final; visualiza e interactúa con el mismo para obtener respuesta del sistema. [29]

Epic: Épica, en Scrum: una historia de usuario que por su gran tamaño, el equipo descompone en historias con un tamaño más adecuado para ser gestionada con los principios y técnicas ágiles.[30]

Expo ”Expo es un *framework* y plataforma para aplicaciones universales *React*. Es un conjunto de herramientas y servicios construidos alrededor de *React Native* y plataformas nativas que ayuda a construir, desplegar y rápidamente iterar en iOS, Android y *webapps* desde el mismo *codebase* JavaScript/TypeScript.” [31]

Feature: Característica, prestación.

Geolocalización: Geolocalización se define como la identificación de la ubicación de un dispositivo por ejemplo un radar, teléfono móvil o cualquier aparato tecnológico conectado a internet. Está relacionada con los sistemas de detección de posición, pero añade datos como información de la zona, calles, locales.[32]

Georeferenciación: Proceso por el cual se dota de un sistema de referencia de coordenadas terreno a una imagen digital que originariamente se encuentra en coordenadas píxel.[33]

Glober: Empleado de Globant.

Happn: Aplicación móvil para conocer a la gente con la que se cruza por la calle y que funciona registrando la ubicación de todos los usuarios que la tengan activa y entran en rango de proximidad[34]

Heurística: Método por el cual el usuario descubre y aprende por sí mismo el funcionamiento, en lugar de ser educado explícitamente. [35]

Internet of Things (IoT) Platform Engineer: un ingeniero de IoT desarrolla servicios innovadores que ayudan a los usuarios a recibir, controlar y gestionar la información. También están a cargo de desarrollar y administrar una amplia gama de dispositivos, plataformas, software, hardware y sistemas de IoT a través de una combinación de datos, tecnología e investigación. [36]

Javascript Bundler: Encargado de convertir diversas entradas como archivos y opciones, transformándolos en un solo archivo JavaScript [37].

Kanban: Kanban es un marco de trabajo muy popular a la hora de implementar un desarrollo de software ágil y de DevOps. Requiere una comunicación en tiempo real sobre la capacidad y una total transparencia del trabajo. Los elementos de trabajo se representan visualmente en un tablero de kanban, lo que permite a los miembros del equipo ver el estado de cada uno en cualquier momento.[38]

Kotlin: ”Kotlin es un lenguaje *open source* estáticamente tipado que apunta

la JVM, Android, JavaScript y *Native* Es desarrollado por JetBrains.” [39]

Material Design: ”Material es un sistema adaptable de pautas, componentes y herramientas que soportan las mejores prácticas de diseño de interfaz de usuario. Simplifica la colaboración entre diseñadores y desarrolladores, y ayuda al equipo a rápidamente construir productos atractivos.” [40]

MS: Makcim Silveira, estudiante de Universidad ORT Uruguay.

Mapwize: Mapwize es una plataforma mapeo de interiores para la creación de servicios digitales dentro de edificios y eventos.[41]

MM: Marcelo Santamaria, *IoT Platform Engineer* en Globant.

Microsoft Azure: La plataforma Azure está compuesta por más de 200 productos y servicios en la nube diseñados para ayudarle a dar vida a nuevas soluciones que permitan resolver las dificultades actuales y crear el futuro.[42]

Microsoft Planner: Planner es una aplicación móvil y web para gestionar del trabajo que permite crear planes, asignar tareas, chatear, y ver gráficos asociados al progreso del equipo. [43]

Minimum Viable Product (MVP): El producto mínimo viable es la versión de un nuevo producto que permite a un equipo recopilar la máxima cantidad de aprendizaje validado sobre los clientes con el menor esfuerzo.[44]

My Disney Experience: Es una aplicación móvil que permite ver y actualizar tus planes, hacer reservaciones para restaurantes, realizar búsquedas en los mapas de los 4 Parques Temáticos, conocer los tiempos de espera de las atracciones en tiempo real y usar las funciones de la MagicBand. [45]

Open Closed Principle: Principio de diseño de programación orientada a objetos.[46]

Overleaf: Overleaf es un editor colaborativo en línea y en tiempo real para artículos, tesis, informes técnicos y otros documentos escritos en el lenguaje de marcado LaTeX. [47]

PO: *Product Owner*, rol dentro del equipo de *Scrum*, responsable por la gestión de *Product Backlog*. [48]

PP: Pablo Pereira, estudiante de Universidad ORT Uruguay.

Pokémon Go: Pokémon GO es un videojuego de realidad aumentada inspirado en el video juego homónimo original y en la serie de TV, basado en la localización desarrollado por Niantic para dispositivos iOS y Android. [49]

Proof of concept (POC): Un PoC es evidencia de que una idea de negocio funciona. Por lo general, es un documento que presenta la viabilidad de una idea, así como los resultados de prueba verificables del concepto, diseño o plan en cuestión. [50]

Requerimientos funcionales (RF): Los requisitos funcionales definen el comportamiento básico del sistema. Esencialmente, son lo que el sistema hace o no debe hacer, y pueden pensarse en términos de cómo responde el sistema a las entradas. Los requisitos funcionales generalmente definen comportamientos si / entonces e incluyen cálculos, entrada de datos y procesos comerciales. [51]

Requerimientos no funcionales (RNF): Mientras que los requisitos funcionales definen lo que el sistema hace o no debe hacer, los requisitos no funcionales especifican cómo debe hacerlo el sistema. Los requisitos no funcionales no afectan la funcionalidad básica del sistema. Incluso si no se cumplen los requisitos no funcionales, el sistema seguirá cumpliendo su propósito básico. [52]

PropTypes: Librería utilizada para *typechecking*, la cual hace detecta errores de tipos de propiedades (*props*). [8]

RRSS: Redes Sociales.

Scrum: *Framework* ligero que ayuda a las personas, los equipos y las organizaciones a generar valor a través de soluciones adaptativas para problemas complejos. [53]

SonarQube: Herramienta *open-source* automática de revisión de código, utilizada para detectar *bugs*, vulnerabilidades y *code smells* en código. Puede integrar *workflows* para realizar inspección de código continua. [54] [55]

Source Code Management (SCM): Es la práctica de rastrear y gestionar los cambios en el código de software. Los sistemas de control de versiones son herramientas de software que ayudan a los equipos de software a gestionar los cambios en el código fuente a lo largo del tiempo. [56]

Stakeholders: Una persona como un empleado, cliente o ciudadano que está

involucrado con una organización, sociedad, etc. y, por lo tanto, tiene responsabilidades hacia ella y un interés en su éxito. [57]

Swarm: Red social basada en geolocalización desarrollada por Foursquare. [58]

Swift: "Swift es un poderoso e intuitivo lenguaje de programación para iOS, iPadOS, macOS, tvOS, y watchOS. Escribir código Swift es interactivo y divertido, la sintáxis es concreta pero expresiva, e incluye características modernas que los desarrolladores aman. El código Swift es seguro por defecto, aunque produce software que es ejecutado muy rápidamente." [59]

Tinder: Aplicación de citas basada en geolocalización y la mecánica *Swipe Right* para seleccionar gustos. [60]

VV: Verónica Varela, Visual Designer en Globant.[3]

Waze: Aplicación de tráfico y navegación basada en la comunidad que se creó como una herramienta de navegación social para coches de particulares. [61]

10. Anexos

10.1. Anexo 1 - Resúmenes *sprints*

10.1.1. *Sprint 1*

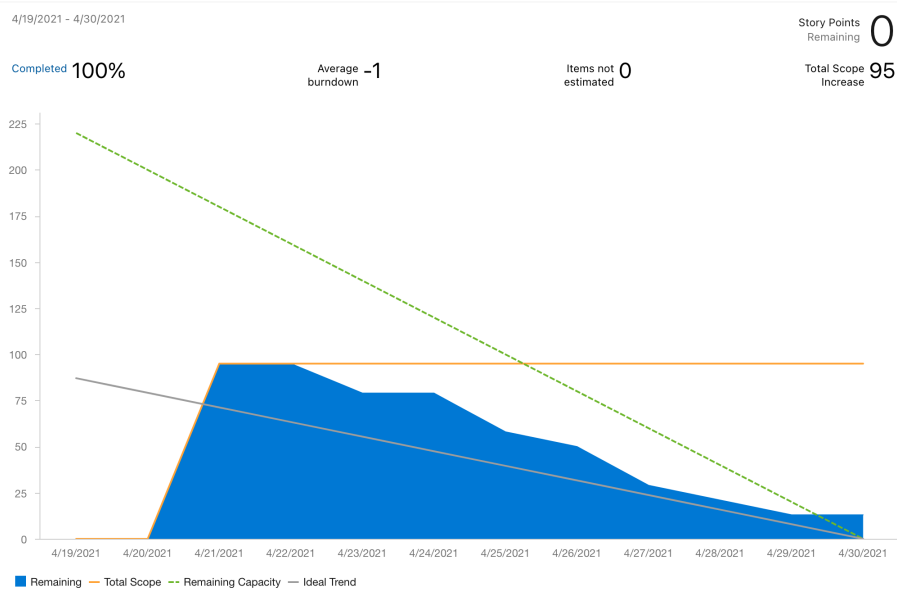


Figura 10.1: *Sprint 1 - Burndown chart*

Sprint 1

Fecha de inicio: 19/04/2021
 Fecha de fin: 30/04/2021



Figura 10.2: *Sprint 1 - Dashboard*

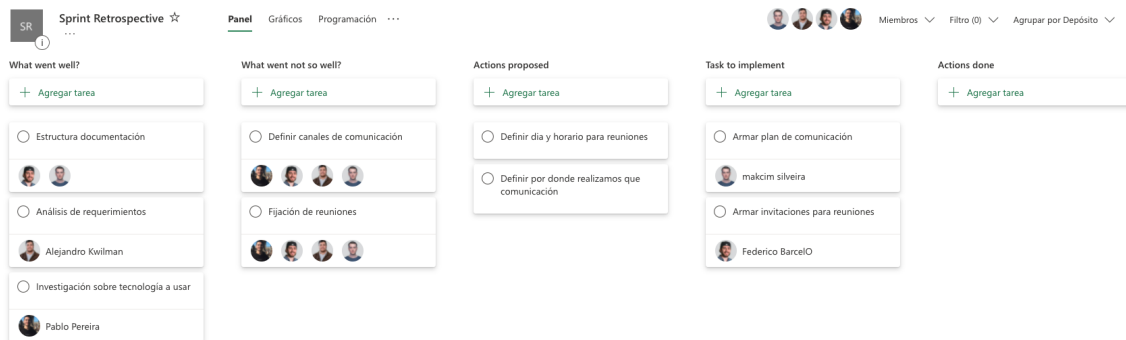


Figura 10.3: *Sprint 1 - Retrospective*

10.1.2. *Sprint 2*

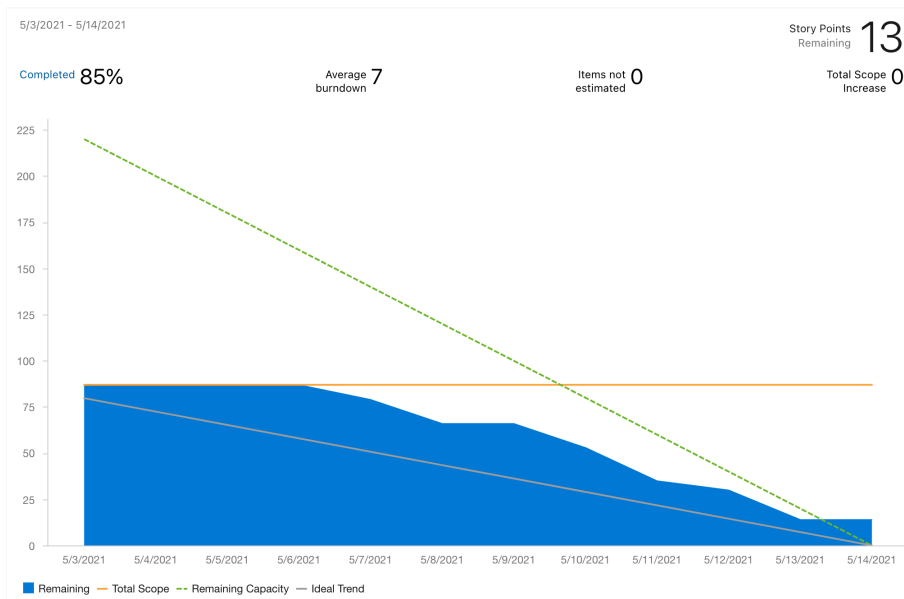


Figura 10.4: *Sprint 2 - Burndown chart*

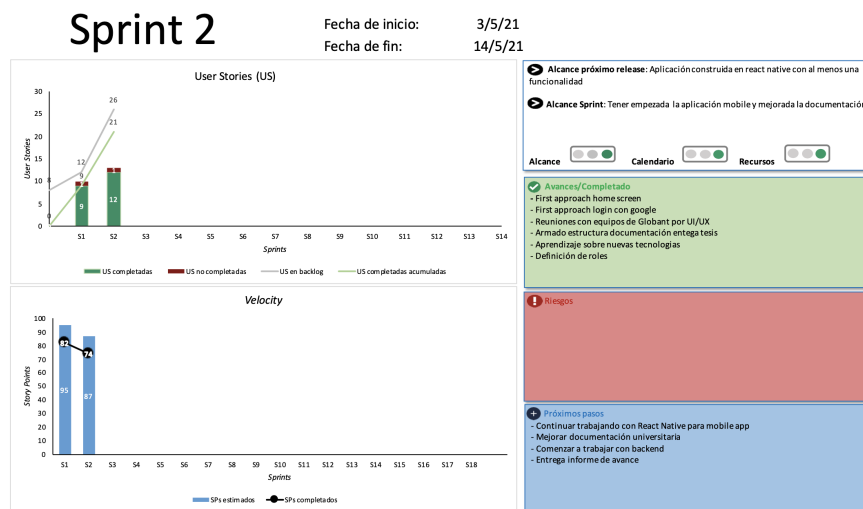


Figura 10.5: *Sprint 2 - Dashboard*

SR Sprint Retrospective ☆


Panel Gráficos Programación ...

Miembros Filtro (0) Agrupar por Depósito


What went well?

+ Agregar tarea


Prototipos iniciales pantallas app mobile



Integración con expertos del cliente

 Alejandro Kwilman





Definición de tecnologías a utilizar

 Pablo Pereira

What went not so well?

+ Agregar tarea

Forma de que mostramos avances de final de Sprint

Actions proposed



+ Agregar tarea

Generar estructura para mostrar avances

Task to implement

+ Agregar tarea


Armar Dashboard de resumen para sprints

Actions done

+ Agregar tarea

Armar invitaciones para reuniones

 Federico BarcelO

Armar plan de comunicación


 makim silveira

Figura 10.6: *Sprint 2 - Retrospective*

10.1.3. *Sprint 3*

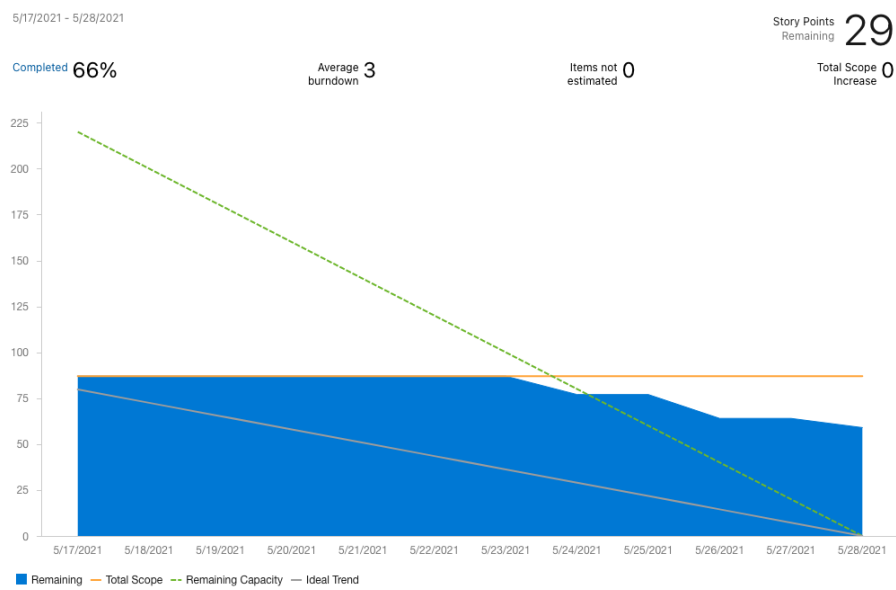


Figura 10.7: *Sprint 3 - Burndown chart*

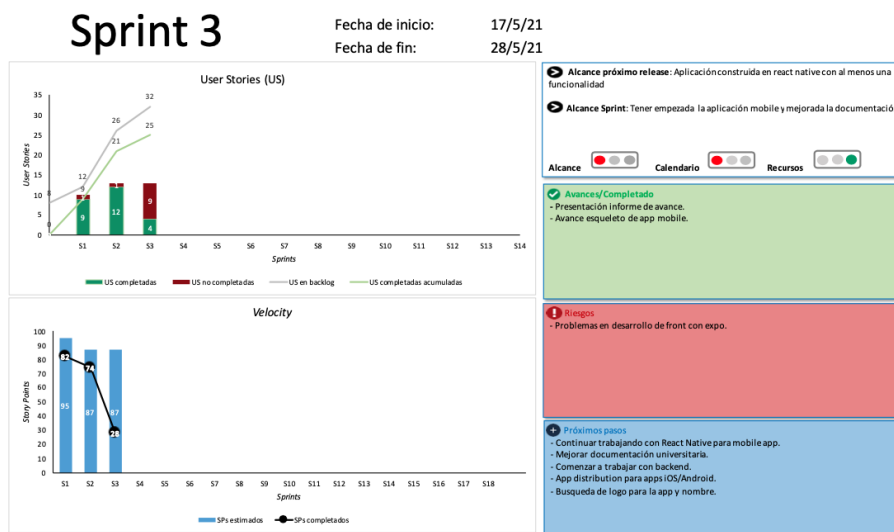


Figura 10.8: *Sprint 3 - Dashboard*

SR Sprint Retrospective ☆

Panel Gráficos Programación ...

Miembros Filtro (0) Agrupar por Depósito

What went well?

+ Agregar tarea

- Buen trato con tutor/cliente
- Sinergia con tutor (integrarlo)
- Forma de presentación de avances
- Gestión del proyecto (métricas, charts)
- Documentación en general
- Asignación de roles

What went not sot well?

+ Agregar tarea

- Leer, pienso, entiendo y luego hago
- Hacer las cosas
- Dependencia de otros para solucionar problemas de código
- Trabajo en react-native
- Organización demo
- Alguna presentación con tutor

Actions proposed

+ Agregar tarea

- Instancia de peer review
- Tomar un rol más proactivo y autónomo en búsqueda de soluciones
- Seguir realizando documentación para enriquecer el trabajo
- Seguir realizando métricas, charts y cosas inherentes para saber el transcurso del proyecto

Task to implement

+ Agregar tarea

- Generar sesiones de peer review (on demand)
- Seguir capacitandose
- Continuar implementación de documentación iterativa

Actions done

+ Agregar tarea

Figura 10.9: *Sprint 3 - Retrospective*

10.1.4. *Sprint 4*

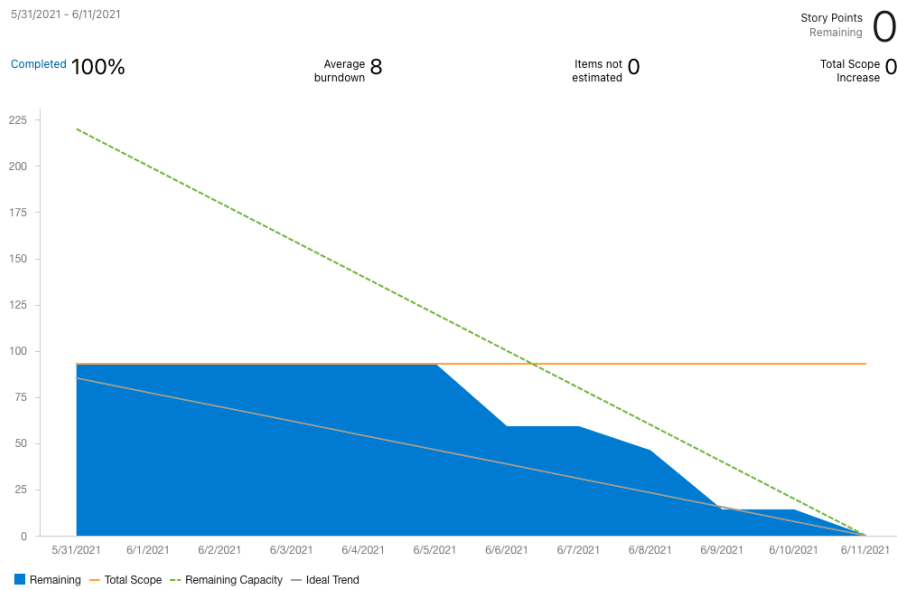


Figura 10.10: *Sprint 4 - Burndown chart*

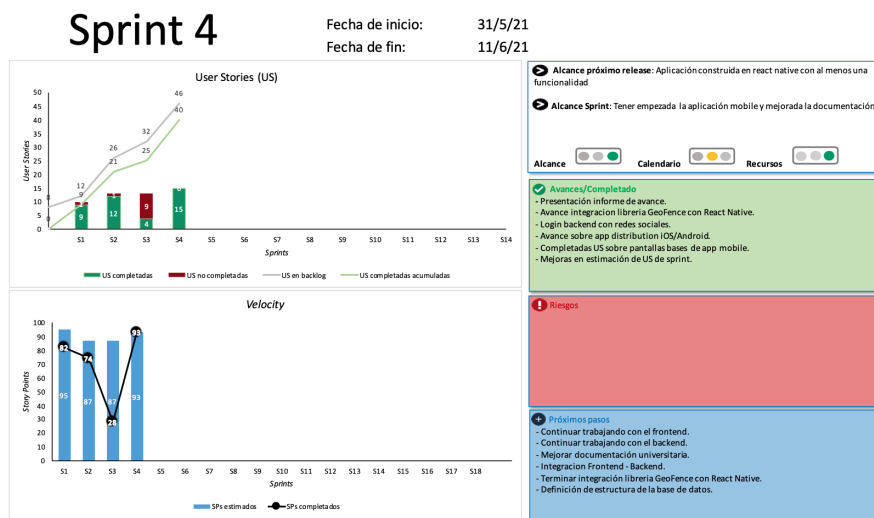


Figura 10.11: *Sprint 4 - Dashboard*

SR Sprint Retrospective ☆

Panel Gráficos Programación ...

Miembros Filtro (0) Agrupar por Depósito

What went well?

+ Agregar tarea

Rosa

- Seguimiento del proyecto
- Realización de todas las ceremonias pautadas
- Inclusión de tutor/acercarlo al proyecto

Amarillo

- Reestructura desarrollo front

Rojos

- Desarrollo de la doc

Verde

- Adaptación a problemas que surgieron

Federico BarcelO

What went not so well?

+ Agregar tarea

- Entrega documentación a Oscars
- Muestreo de avances
- Call de Sync viernes
- Comunicación acerca de tecnologías a usar (venia de otros sprints ya)

Actions proposed

+ Agregar tarea

- Instancia de peer review
- Tomar un rol más proactivo y autonomo en búsqueda de soluciones

Task to implement

+ Agregar tarea

- Seguir realizando metricas, charts y cosas inherentes para saber el transcurso del proyecto
- Seguir realizando documentación para enriquecer el trabajo
- Cerrar y limpiar la rama migration/app, empezar a hacer feature/branch
- Generar canal de comunicacion entre nosotros y Jack
- Reagendar sync viernes a horario que pueden todos

Actions done

+ Agregar tarea

- Generar sesiones de peer review (on demand)
- Seguir capacitandose
- Continuar implementación de documentacion iterativa

Figura 10.12: *Sprint 4 - Retrospective*

10.1.5. *Sprint 5*

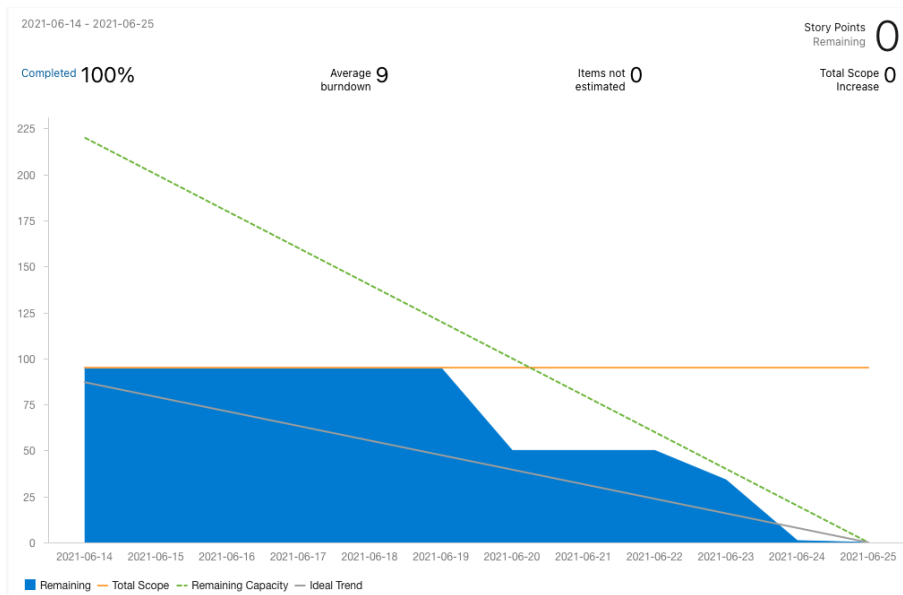


Figura 10.13: *Sprint 5 - Burndown chart*

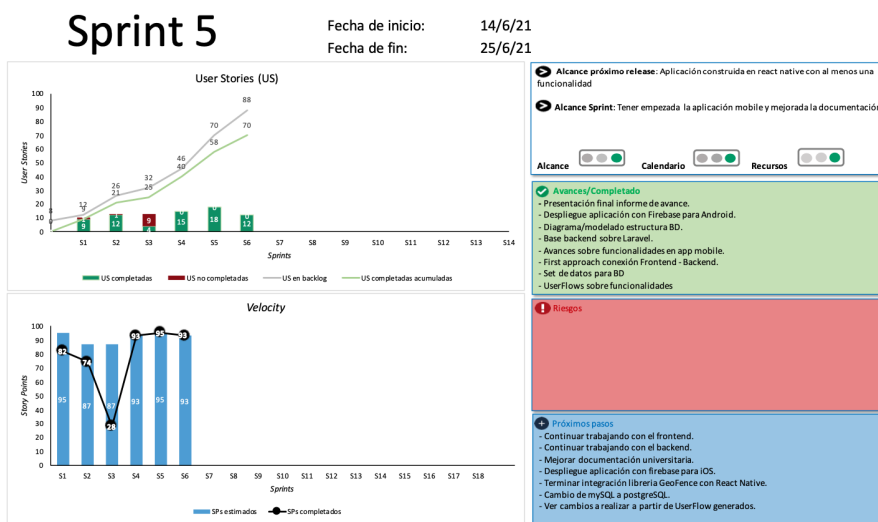


Figura 10.14: *Sprint 5 - Dashboard*

SR Sprint Retrospective ☆

Panel Gráficos Programación ...

Miembros Filtro (0) Agrupar por Depósito

What went well?

+ Agregar tarea

- Implementación de back-end
- Implementación de Redux/states
- Colección de postman
- Alcance sprint
- Firebase
- Segunda presentacion

What went not so well?

+ Agregar tarea

- Llegar con lo estimado
- Conseguir lo que debe Jack
- Autonomia

Actions proposed

+ Agregar tarea

- Utilizar mas el canal de WPP para dudas
- Presionar mas a Jack por temas pendientes
- Cambiar horario destinado a las horas a tesis
- Sacar distracciones y emplear horas de manera concentradas

Task to implement

+ Agregar tarea

- Mostrar al cliente avances significativos y notificarlo brevemente sobre lo que necesita cumplir para el proyecto en plazo máximo de 3 dias
- Implementar nuevos horarios y técnicas de concentración
- Tomar un rol más proactivo y autónomo en búsqueda de soluciones
- Seguir realizando metricas, charts y cosas inherentes para saber el transcurso del proyecto

Actions done

+ Agregar tarea

- Realizar mejoras en presentación de doc. de ORT
- Definición de estándares de codificación
- Instancia de peer review
- Cerrar y limpiar la rama migration/app, empezar a hacer feature/branch
- Generar sesiones de peer review (on demand)
- Seguir realizando documentación para enriquecer el trabajo

Figura 10.15: *Sprint 5 - Retrospective*

10.1.6. *Sprint 6*

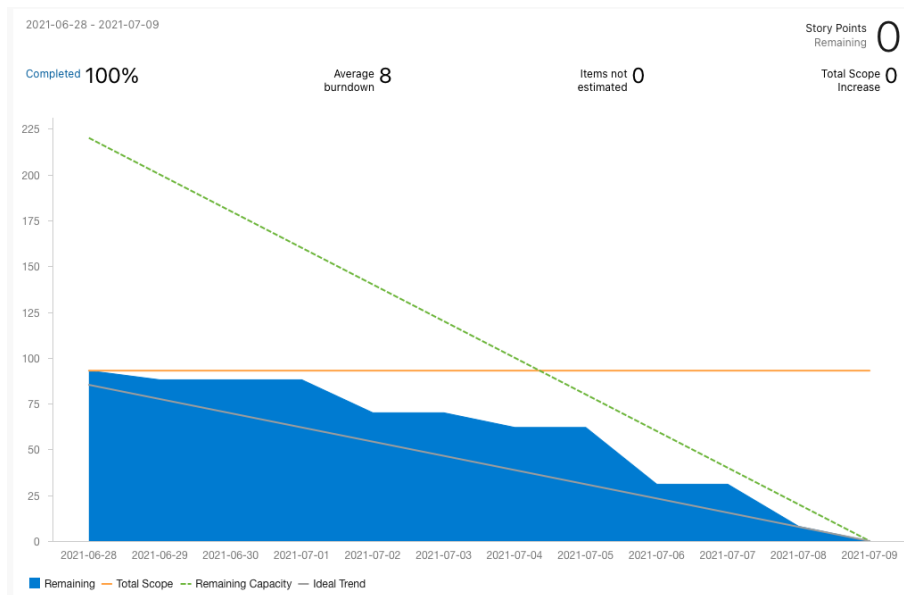


Figura 10.16: *Sprint 6 - Burndown chart*

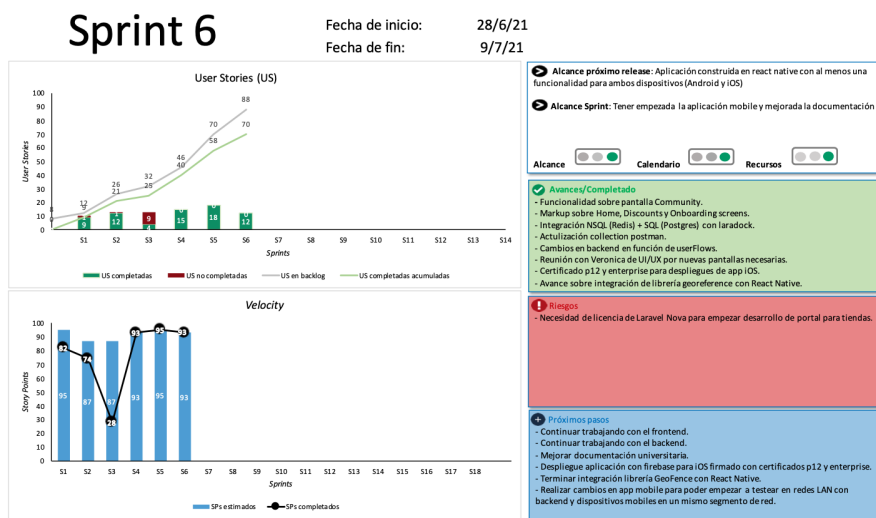


Figura 10.17: *Sprint 6 - Dashboard*

SR Sprint Retrospective ☆

Panel Gráficos Programación ...

Miembros Filtro (0) Agrupar por Depósito

What went well?

+ Agregar tarea

- Organización durante el Sprint
- Reuniones con Jack y Oscar
- Implementación de BD's requeridas por el cliente
- Reunión con Oscar
- Reuniones con Vero por nuevos temas de UI/UX
- Avance en pendientes de Jack

What went not so well?

+ Agregar tarea

- Autonomía en resolución de dificultades

Actions proposed

+ Agregar tarea

- Pedir ayuda externa para la solución de problemas X
- Hacerle seguimiento a Vero por nuevas pantallas
- Presionar mas a Jack por temas pendientes
- Cambiar horario destinado a las horas a tesis
- Sacar distracciones y emplear horas de manera concentradas

Task to implement

+ Agregar tarea

- Pedir ayuda a algún conocido que sepa del tema
- Ping cada 2/3 días por temas pendientes a Jack y Vero
- Mostrar al cliente avances significativos y notificarlo brevemente sobre lo que necesita cumplir para el proyecto en plazo máximo de 3 días
- Implementar nuevos horarios y técnicas de concentración
- Tomar un rol más proactivo y autónomo en búsqueda de soluciones
- Seguir realizando metricas, charts y cosas inherentes para saber el transcurso del proyecto

Actions done

+ Agregar tarea

- Realizar mejoras en presentación de doc. de ORT
- Definición de estándares de codificación
- Instancia de peer review
- Cerrar y limpiar la rama migration/app, empezar a hacer feature/branch
- Generar sesiones de peer review (on demand)
- Seguir realizando documentación para enriquecer el trabajo

Pablo Pereira

Figura 10.18: *Sprint 6 - Retrospective*

10.1.7. *Sprint 7*

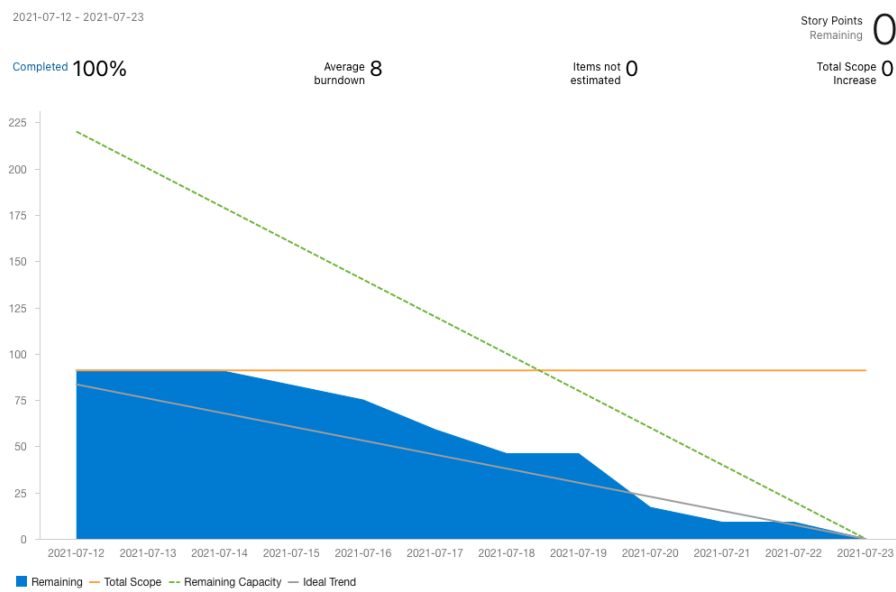


Figura 10.19: *Sprint 7 - Burndown chart*

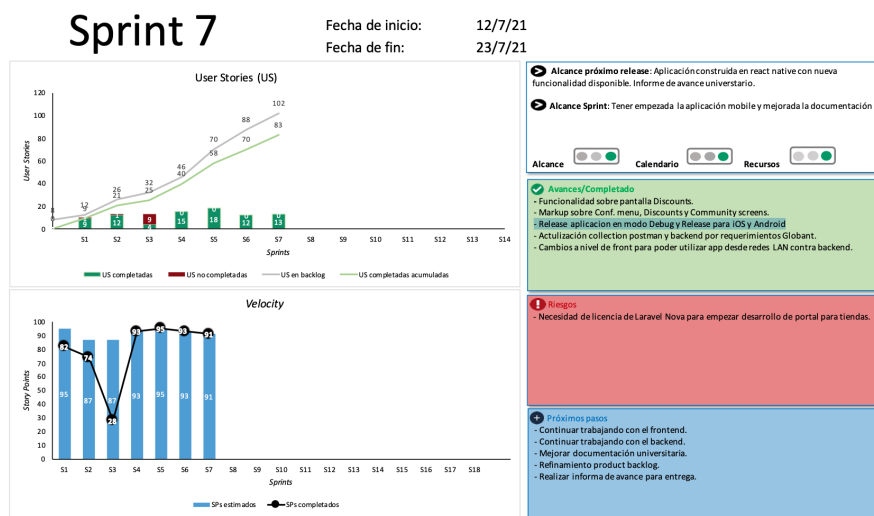


Figura 10.20: *Sprint 7 - Dashboard*





SR Sprint Retrospective ☆

Panel Gráficos Programación ...

Miembros Filtro (0) Agrupar por Depósito



What went well?

+ Agregar tarea

- Cambios en figma
- 
- Reunion de sync con cliente y tutor
- Investigación de integración de RN maps
-  Pablo Pereira
- Cumplimiento con alcance
-  Alejandro Kwilman
- Cambios solicitados por cliente para apertura de la API
- Despliegue y builds Android IOS
- 

What went not so well?

+ Agregar tarea

- Falta de preparación para demo con el cliente
- 
- Calidad de código
-  makcim silveira




Actions proposed

+ Agregar tarea

- Tener practicado/pensado que vamos mostrar en demo
- Seguir realizando validaciones sobre Figma
- Pedir ayuda externa para la solución de problemas X

Task to implement

+ Agregar tarea

- Interacion nueva sobre proyecto de Figma
- 
- Hacer video de feature para la demo
- 
- Seguir realizando metricas, charts y cosas inherentes para saber el transcurso del proyecto
- 

Actions done

+ Agregar tarea






- Tomar un rol más proactivo y autónomo en búsqueda de soluciones
- 
- Implementar nuevos horarios y técnicas de concentración
-  Alejandro Kwilman
- Mostrar al cliente avances significativos y notificarlo brevemente sobre lo que necesita cumplir para el proyecto en plazo máximo de 3 días
- 
- Ping cada 2/3 días por temas pendientes a Jack y Vero
- 
- Realizar mejoras en presentación de doc. de ORT
- 
- Definición de estándares de codificación

Figura 10.21: *Sprint 7 - Retrospective*

10.1.8. *Sprint 8*

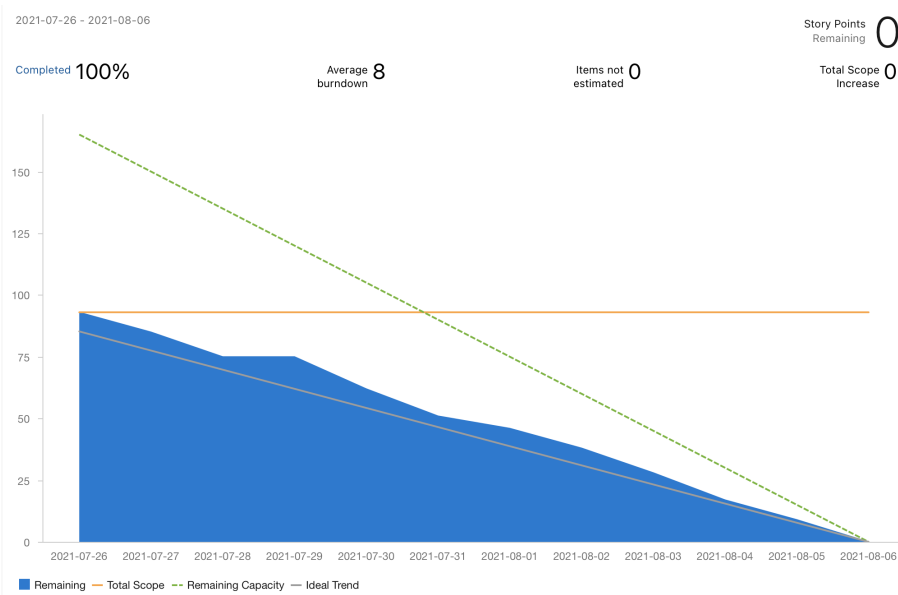


Figura 10.22: *Sprint 8 - Burndown chart*

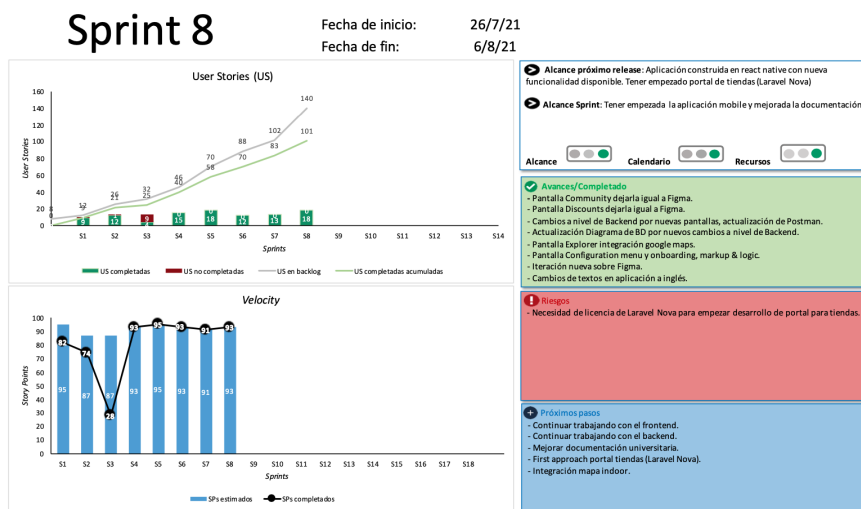


Figura 10.23: *Sprint 8 - Dashboard*

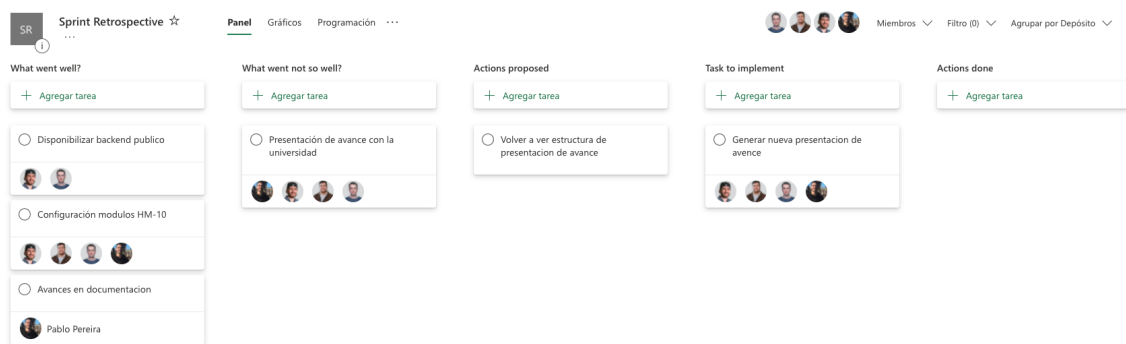


Figura 10.24: *Sprint 8 - Retrospective*

10.1.9. *Sprint 9*

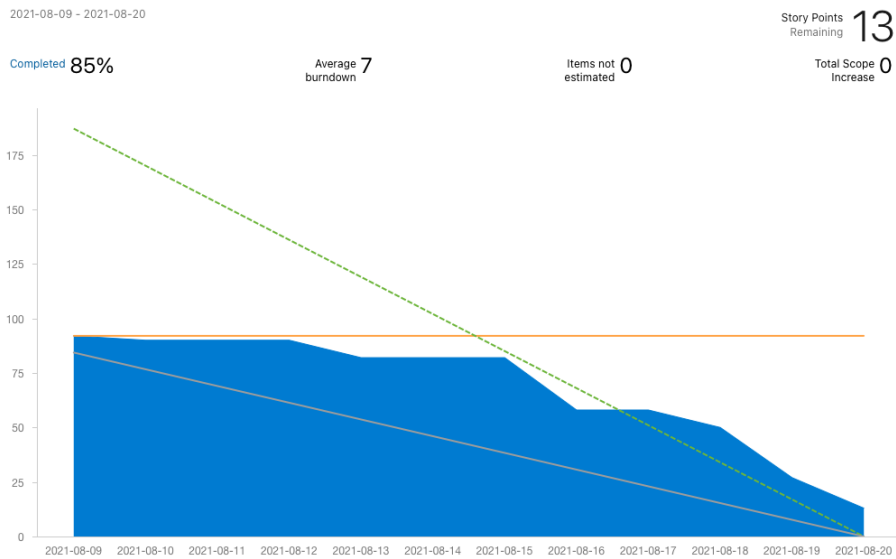


Figura 10.25: *Sprint 9 - Burndown chart*

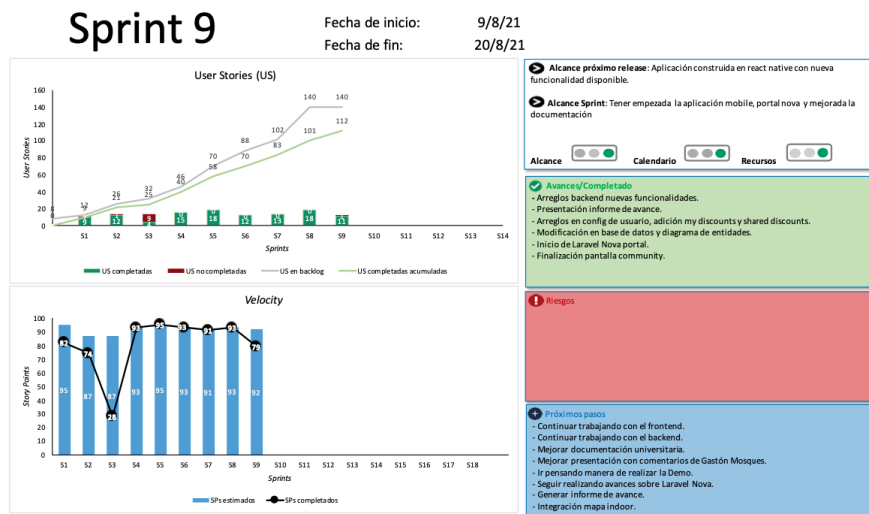


Figura 10.26: *Sprint 9 - Dashboard*

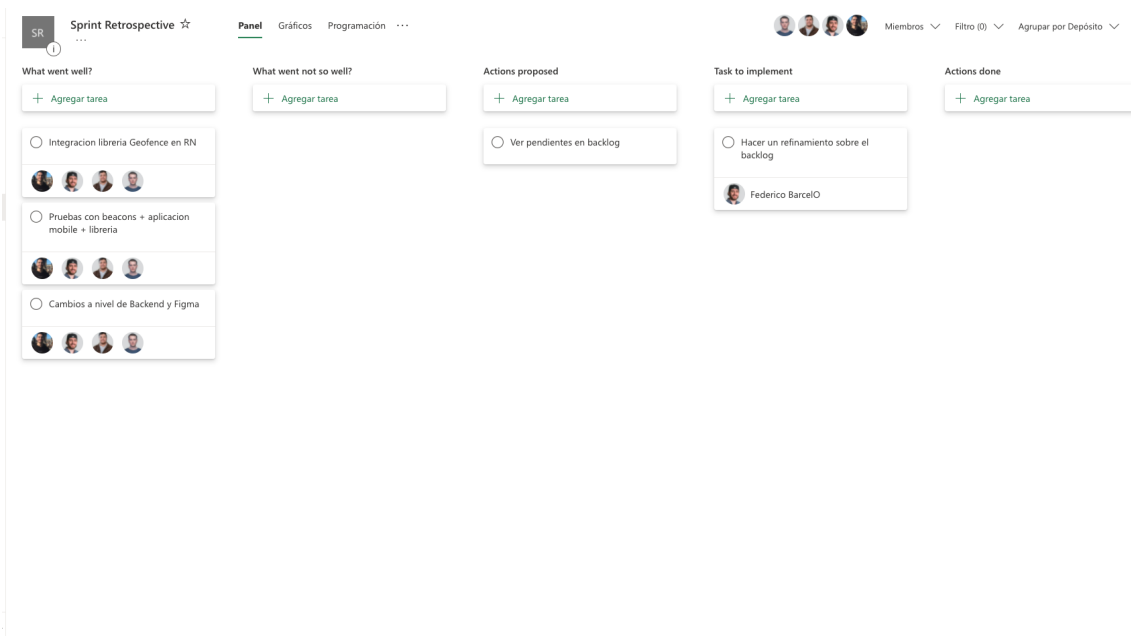


Figura 10.27: *Sprint 9 - Retrospective*

10.1.10. *Sprint 10*

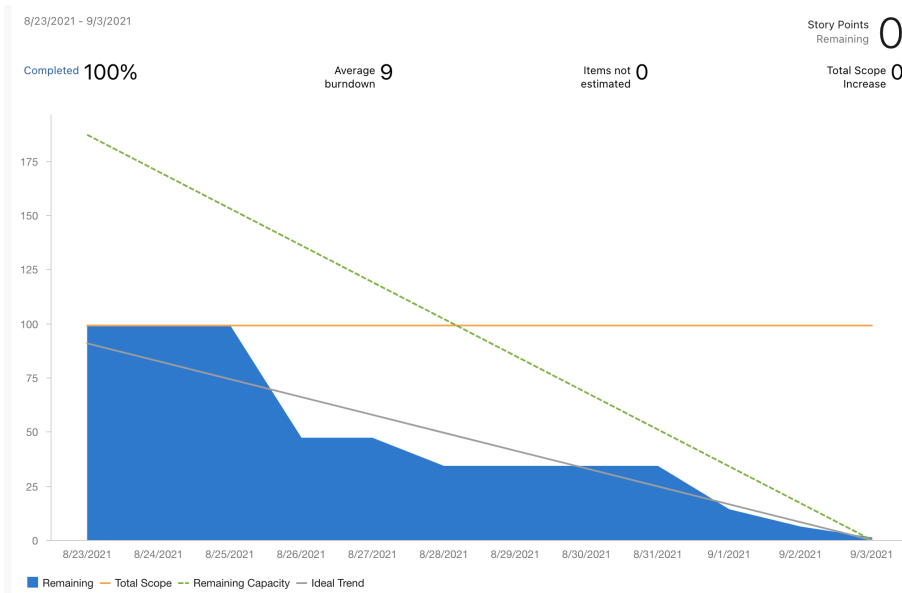


Figura 10.28: *Sprint 10 - Burndown chart*

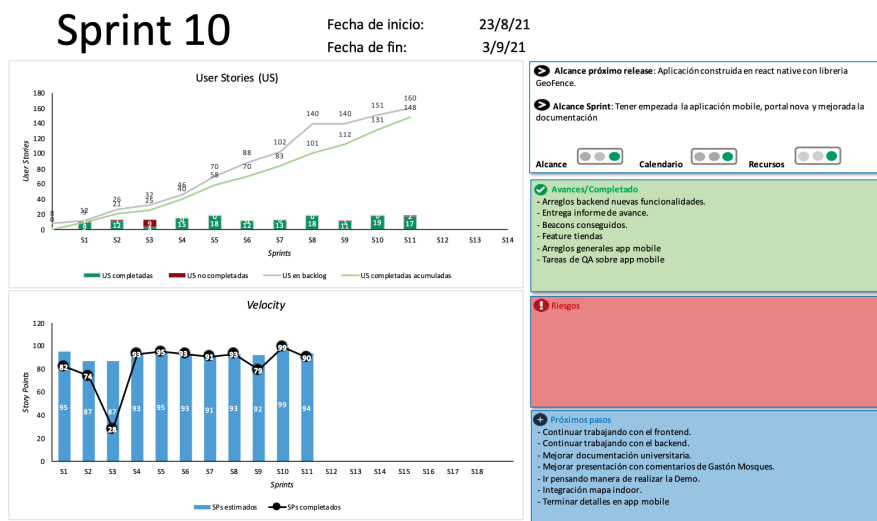


Figura 10.29: *Sprint 10 - Dashboard*

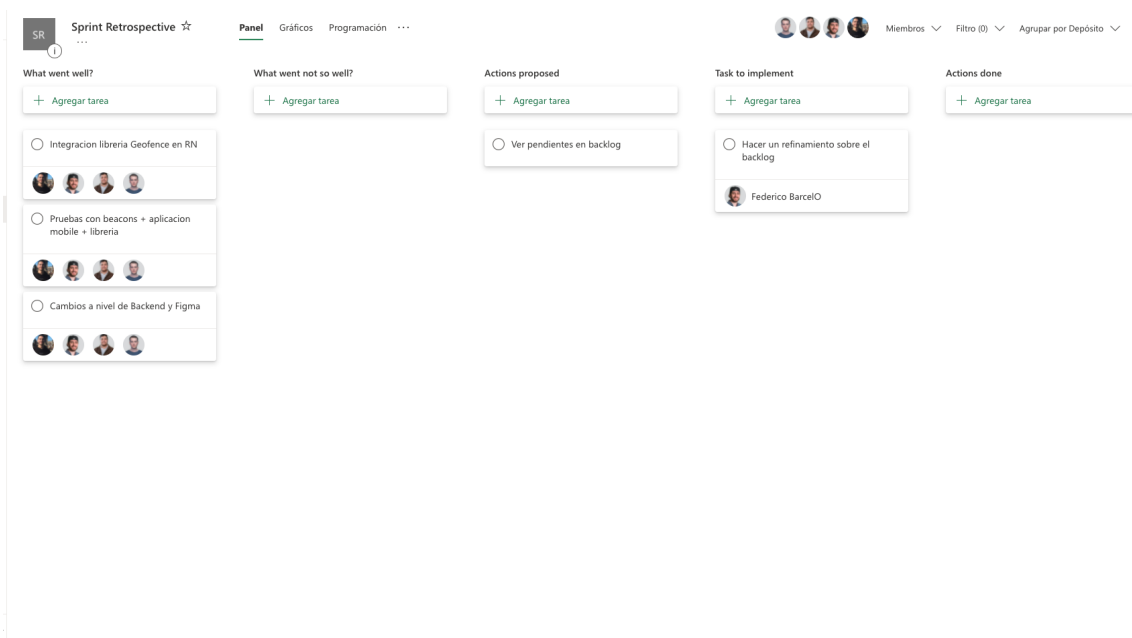


Figura 10.30: *Sprint 10 - Retrospective*

10.1.11. *Sprint 11*

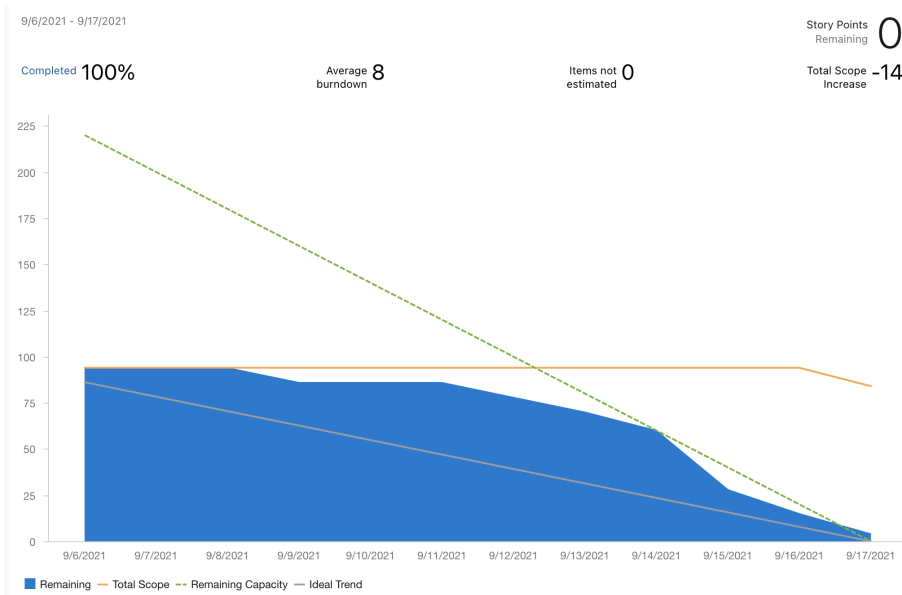


Figura 10.31: *Sprint 11 - Burndown chart*

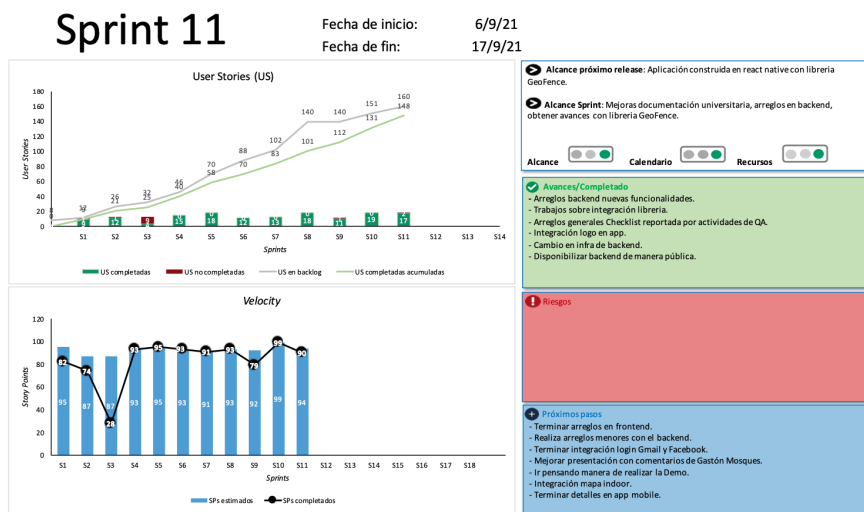


Figura 10.32: *Sprint 11 - Dashboard*



SR Sprint Retrospective ☆

Panel Gráficos Programación ...

Miembros Filtro (0) Agrupar por Depósito










What went well?

+ Agregar tarea

- Pendientes aplicación mobile
- Cambio backend
-  
- Demos
- Informe de negocios
- QA
- Integración librería

What went not so well?

+ Agregar tarea

- Revisión presentación y demo
-   
- Documentación entrega
-   
- Riesgos
- Rosa
- Oscar todavía debe la revisión
-   







Actions proposed

+ Agregar tarea

- Realizar roadmap de la demo
- Segmentar en mas tareas la presentación
- Segmentar en mas tareas la documentación
- Deadline para revisión por parte de Oscar

Task to implement

+ Agregar tarea

- Generar backlog con pending de presentación
-  Federico BarcelO
- Generar backlog con pending de documentación
-  Federico BarcelO
- Poner deadlines por temas pendientes
-  
- Seguir realizando metricas, charts y cosas inherentes para saber el transcurso del proyecto
-  

Actions done

+ Agregar tarea











- Hacer video de feature para la demo
-  
- Implementar nuevos horarios y técnicas de concentración
-  Alejandro Kwilman
- Realizar mejoras en presentación de doc. de ORT
-  
- Cerrar y limpiar la rama migration/app, empezar a hacer feature/branch
-  
- Seguir capacitandose
-  
- Reagendar sync viernes a horario que pueden todos
-  Federico BarcelO

Figura 10.33: *Sprint 11 - Retrospective*

10.2. Anexo 2 - User persona



Figura 10.34: User persona - Andrés



Figura 10.35: User persona - Ana



Figura 10.36: User persona - Olivia

10.3. Anexo 3 - Prototipo

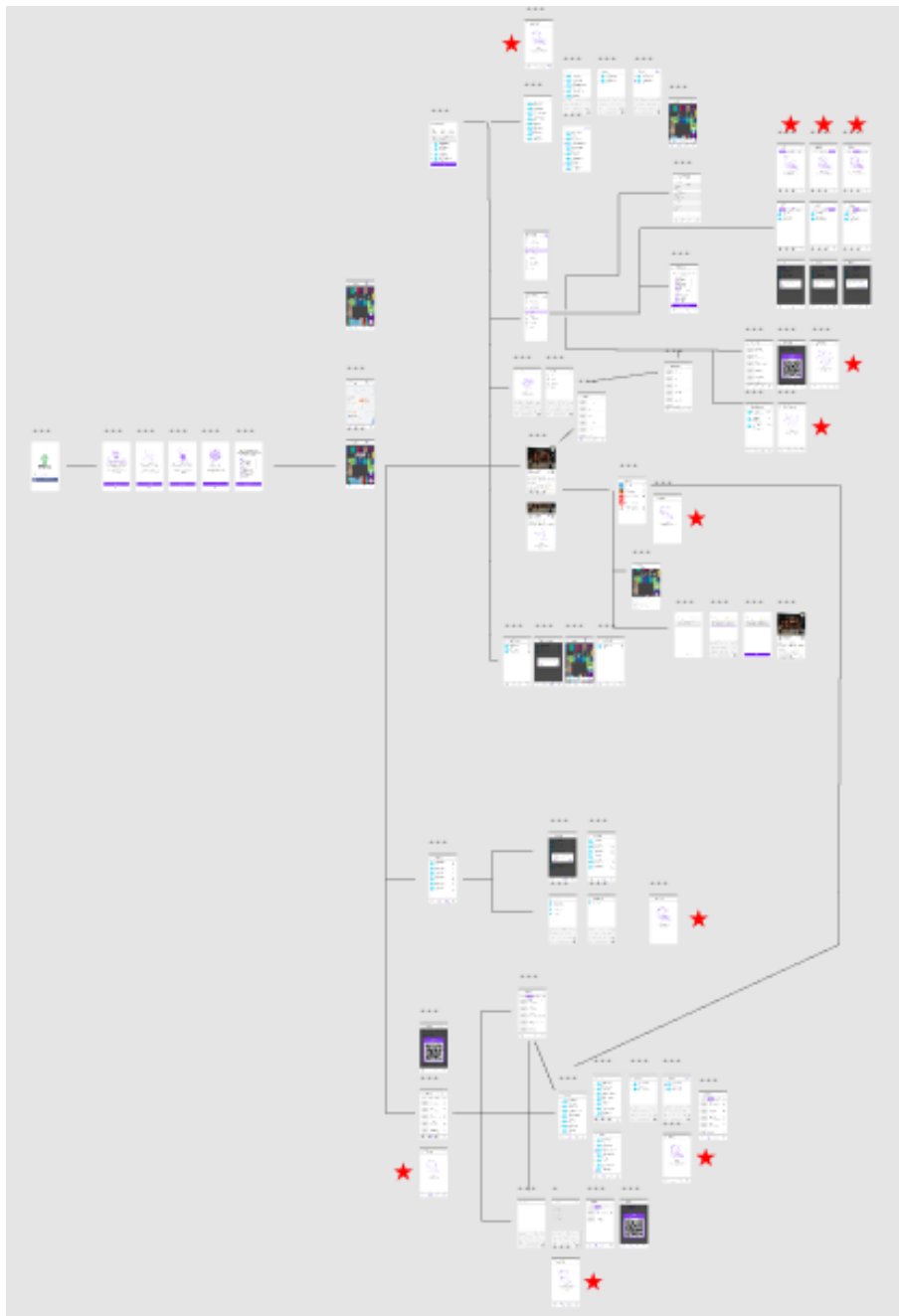


Figura 10.37: Vista general de flujos de prototipado en Figma.

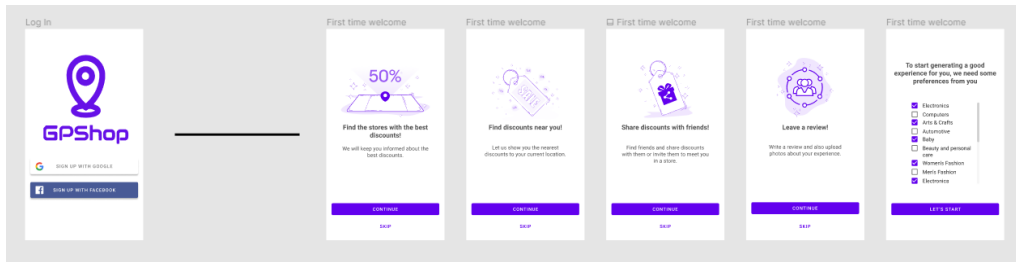


Figura 10.38: Flujo de *onboarding*.

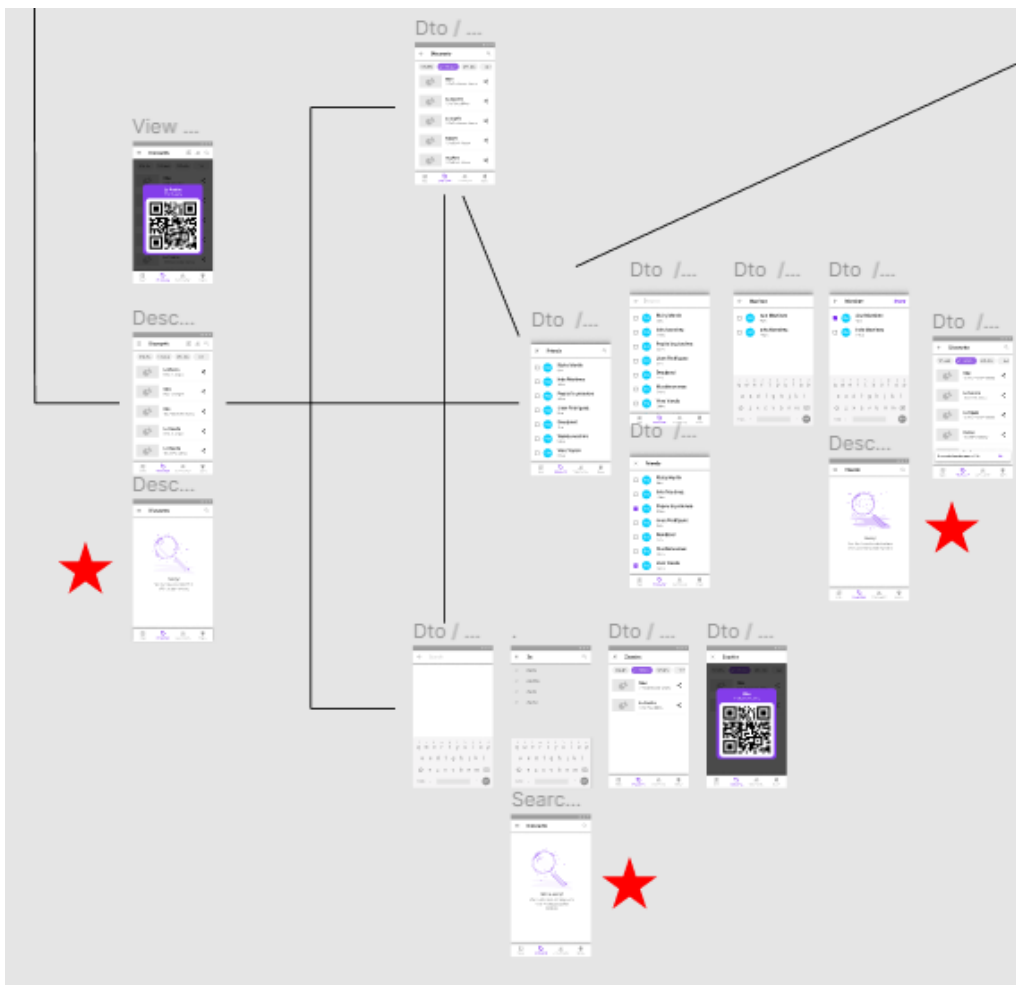


Figura 10.39: Flujo de pantalla *discounts*.

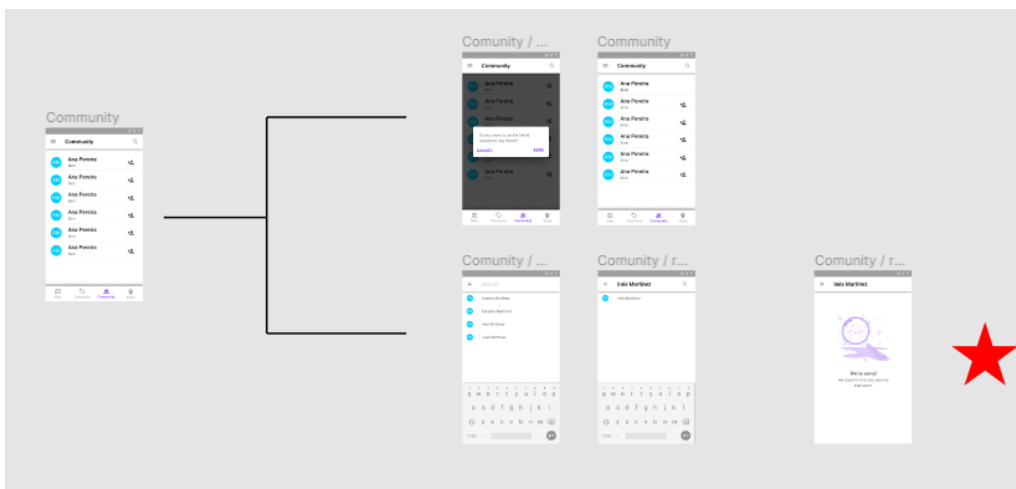


Figura 10.40: Flujo de pantalla *community*.

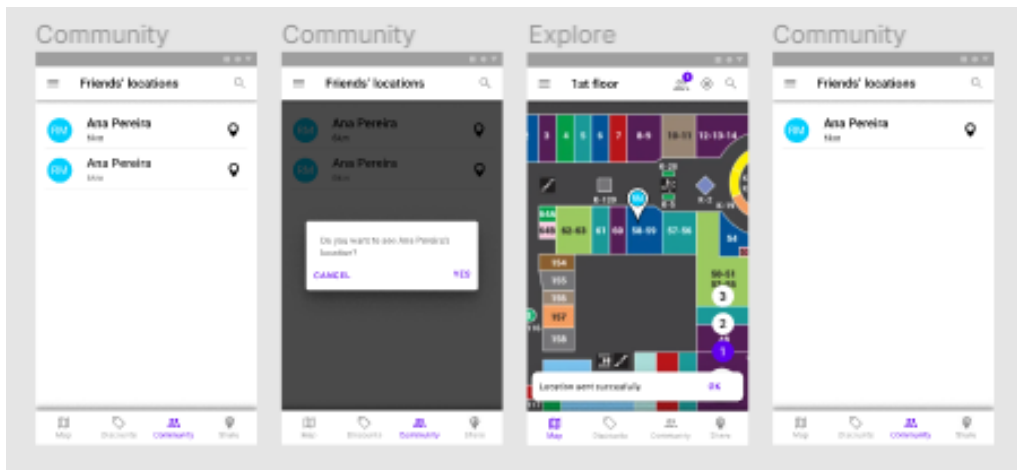


Figura 10.41: Flujo de compartir ubicación.

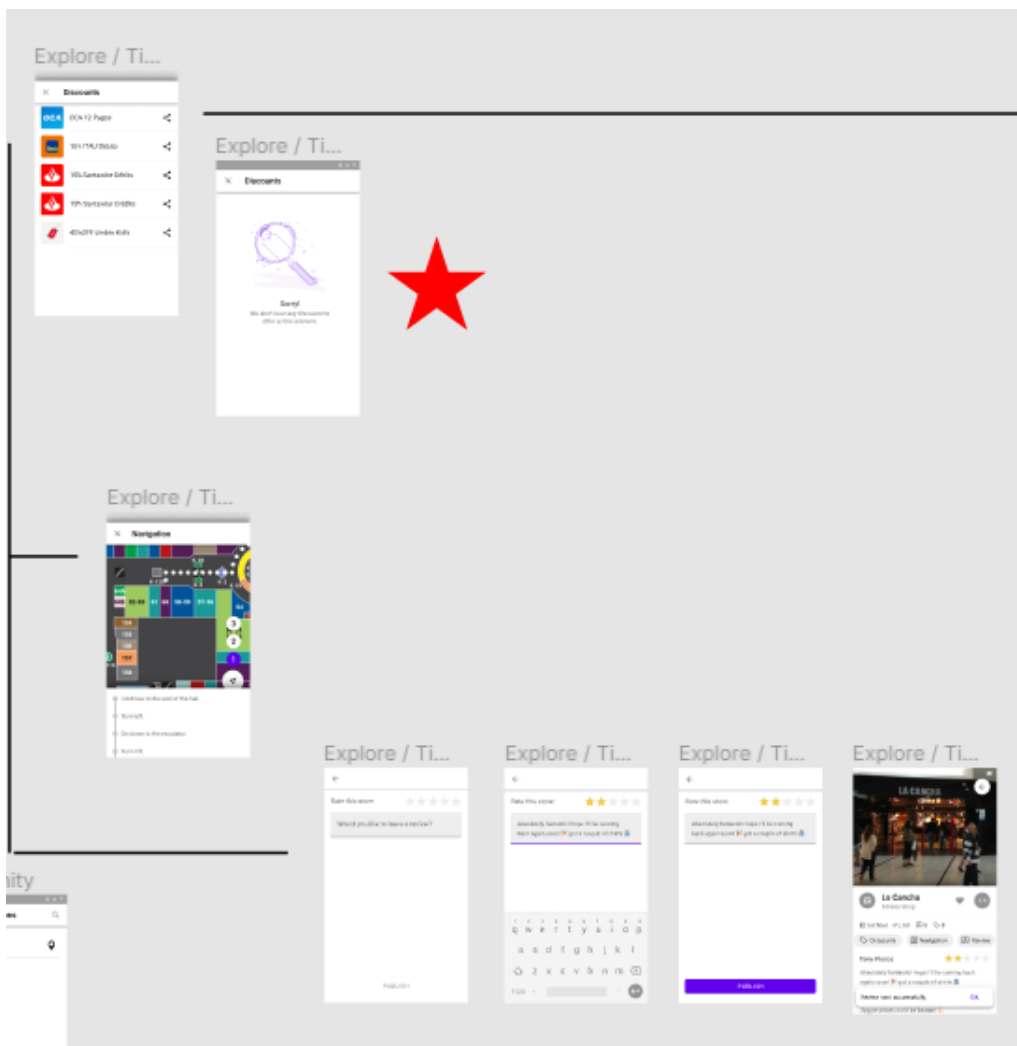


Figura 10.42: Flujo de reseña de tienda.

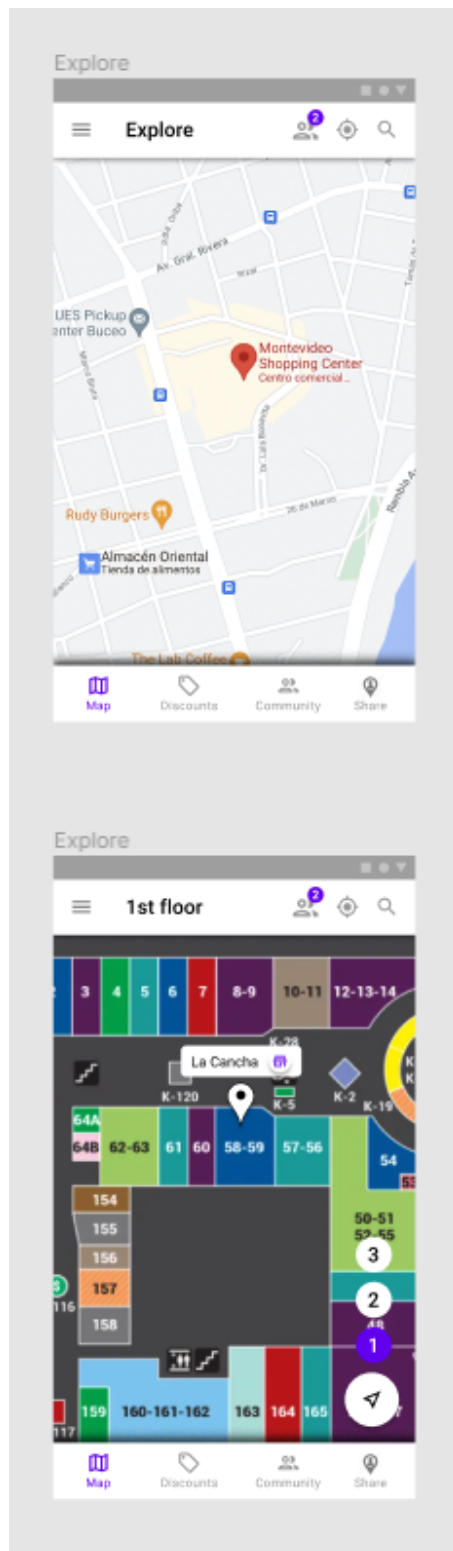


Figura 10.43: Flujo de exploración de mapa interno y externo.

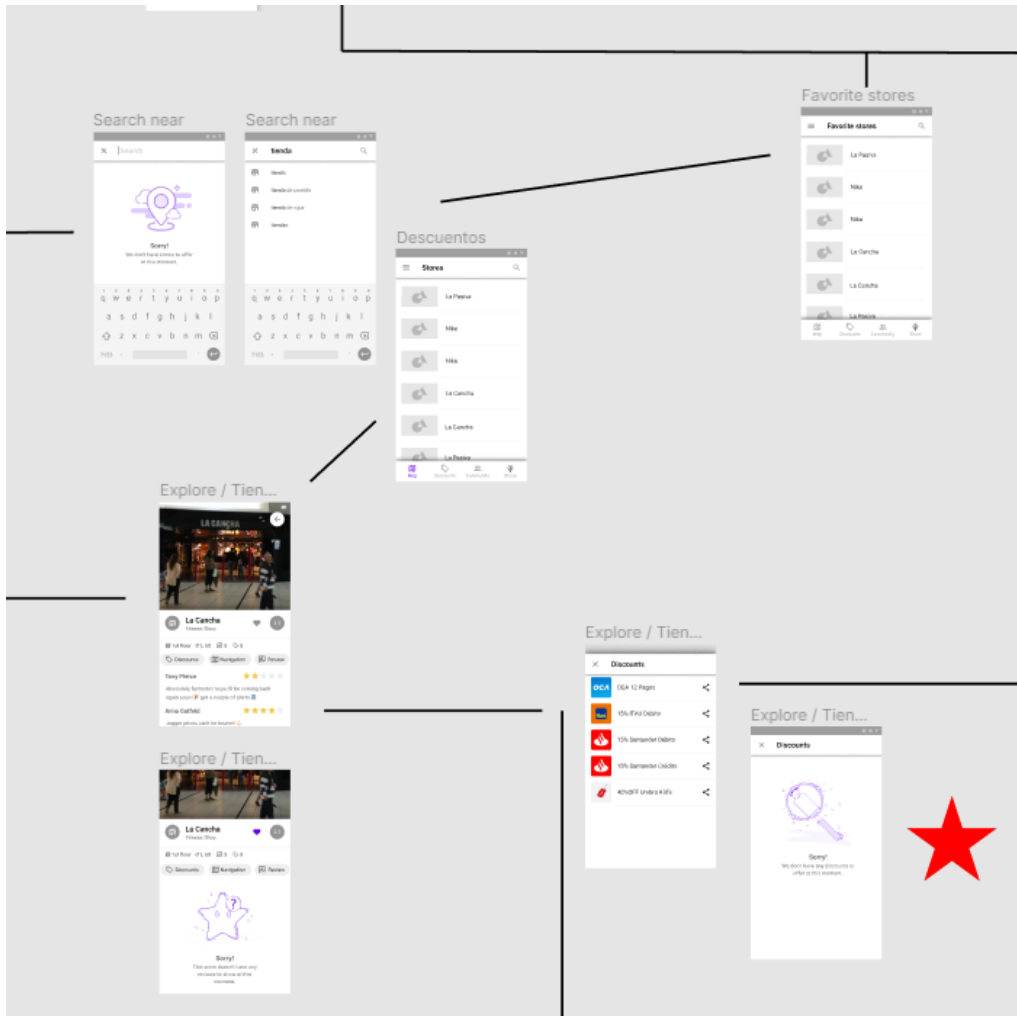


Figura 10.44: Flujo de tiendas, favoritas, búsqueda y favoritas.

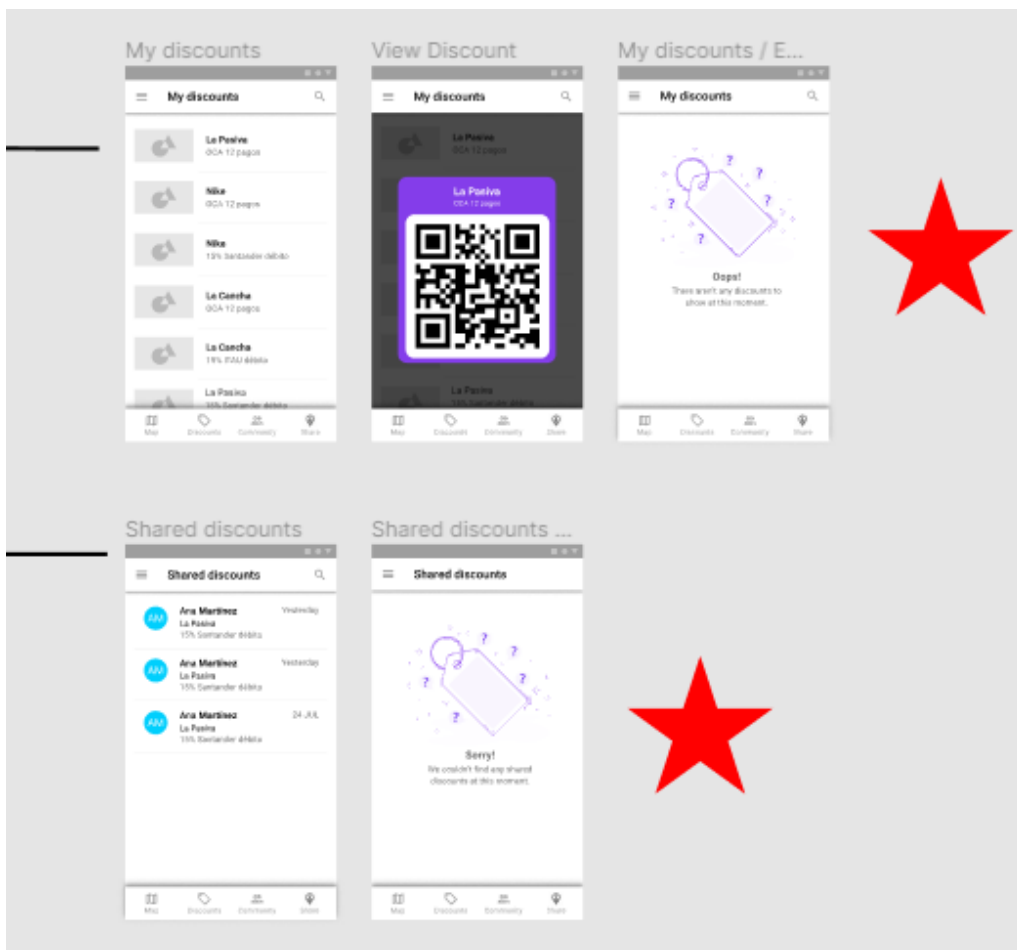


Figura 10.45: Flujo de descuentos.

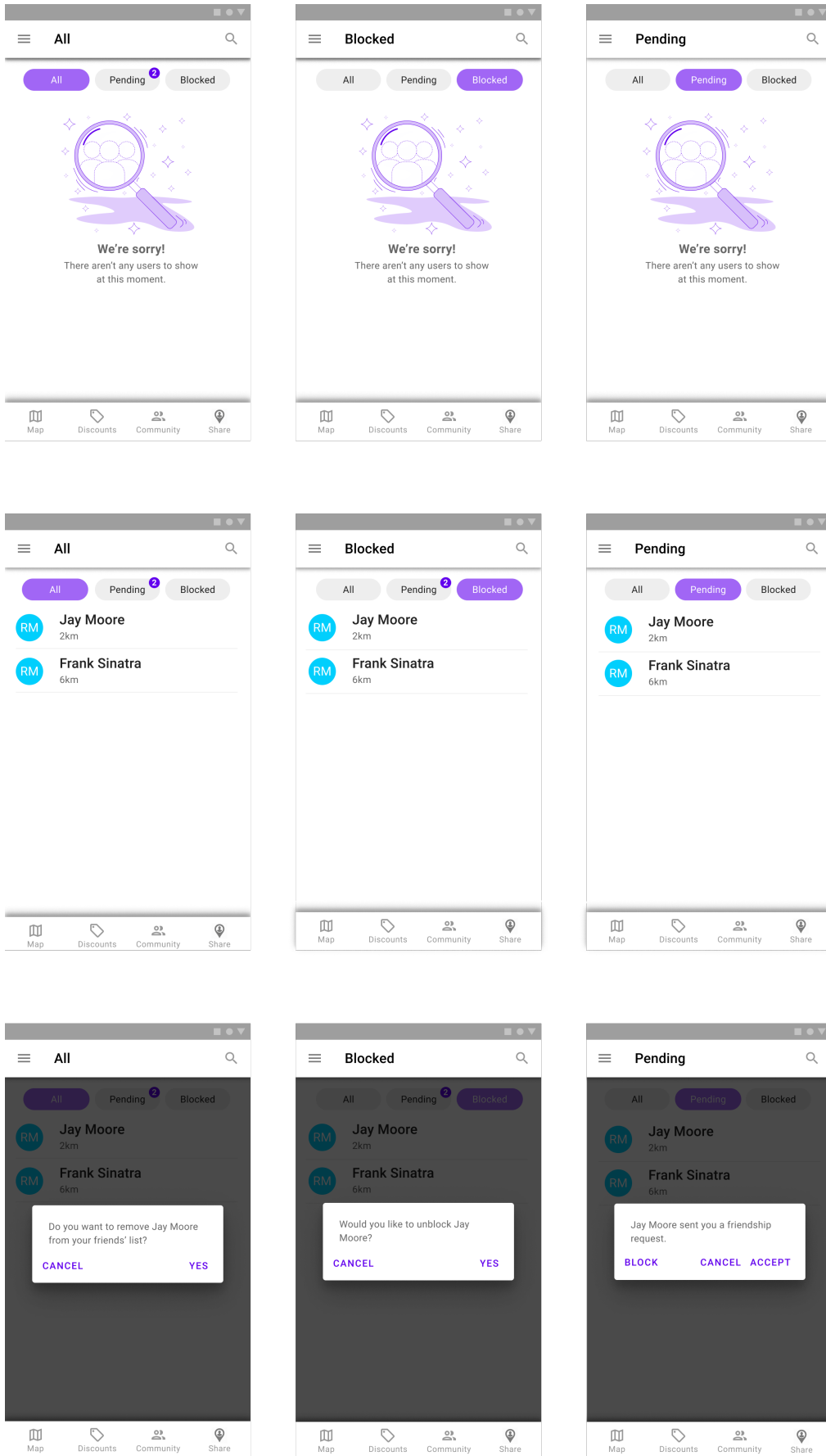


Figura 10.46: Flujo de pantallas en *friends*.

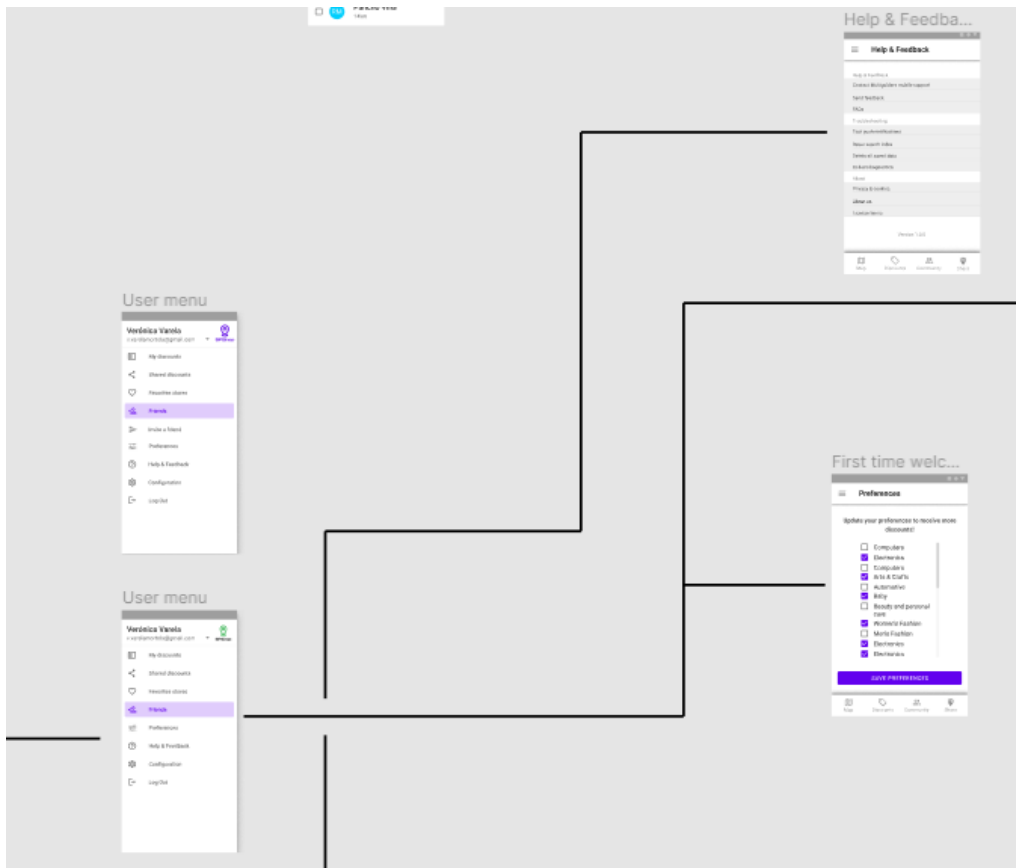


Figura 10.47: Flujos de *Help & Feedback* y *preferences*.

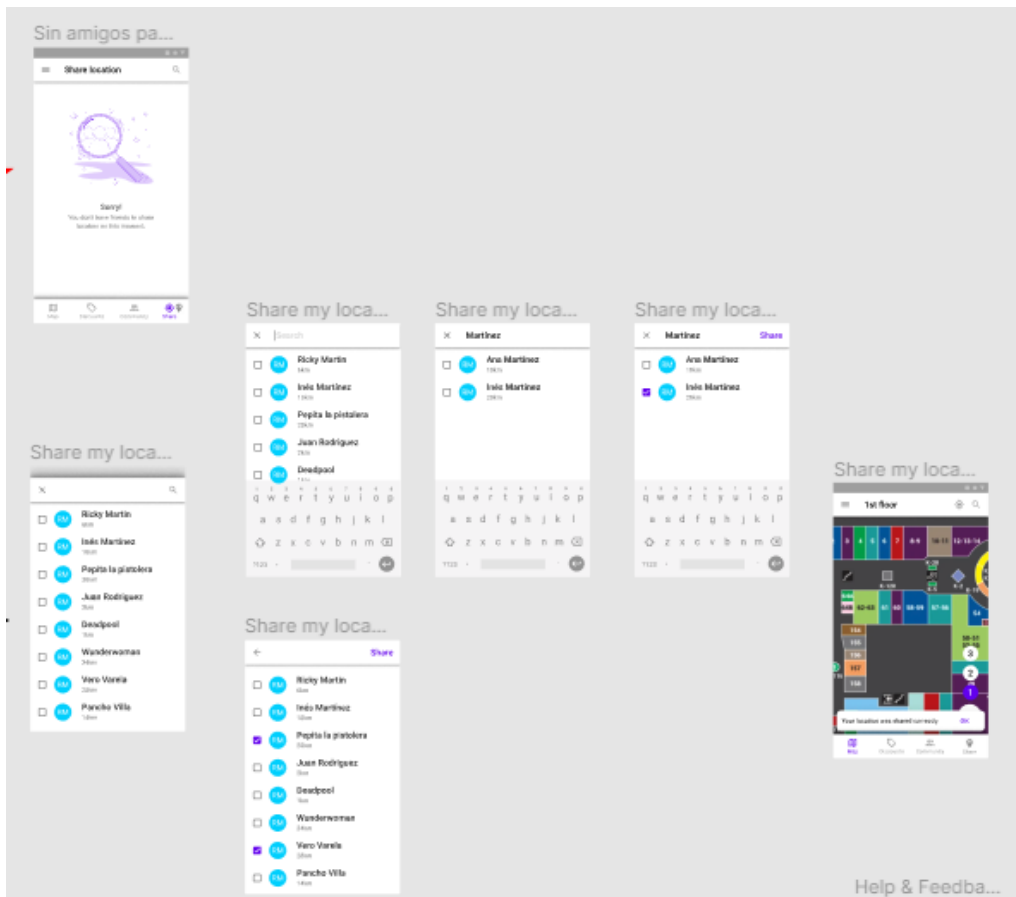


Figura 10.48: Flujo de *share my location*.

10.4. Anexo 4 - ESRE

10.4.1. Datos del proyecto

Proyecto	Plataforma base geolocalización		
Cliente:	Globant	País:	Uruguay

10.4.2. Propósito del documento

En este documento se brindan detalles sobre los puntos acordados de los requerimientos necesarios para la implementación de la plataforma en formato de *user stories* para los RF y RNF.

10.4.3. Requerimientos funcionales

RF1 - Explorar mapa

Descripción: Como usuario de *app mobile* quiero ser capaz de poder navegar en los mapas libremente.

Criterios de aceptación: Se tiene que poder visualizar el mapa tanto *outdoor* como *indoor* dependiendo la ubicación de usuario y se debe poder realizar una navegación de forma libre.

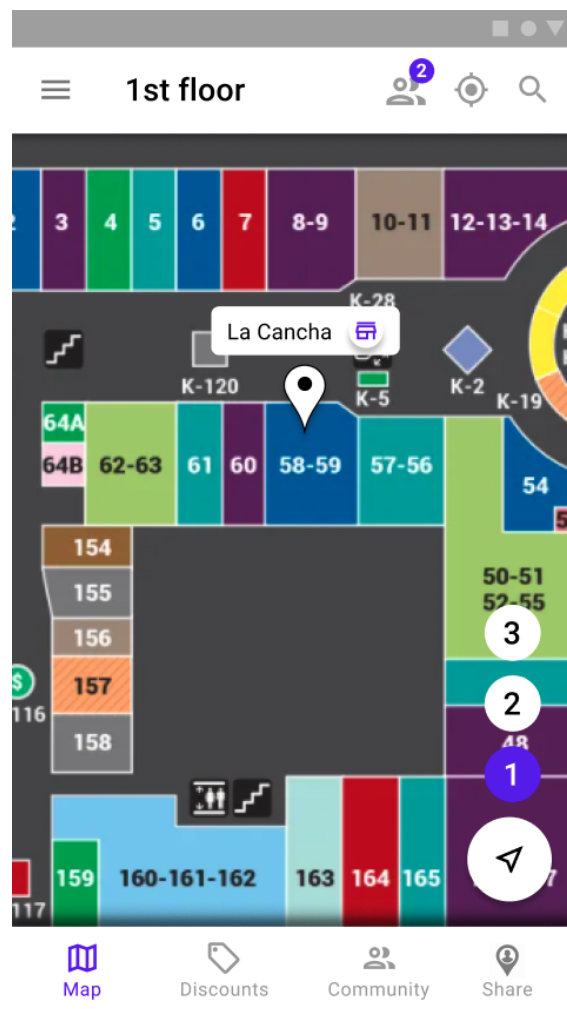


Figura 10.49: RF1 - Pantalla mapa *indoor*.



Figura 10.50: RF1 - Pantalla mapa *outdoor*.

RF2 - Buscador tiendas

Descripción: Como usuario *app mobile* quiero ser capaz de poder buscar tiendas por nombre dentro del mapa.

Datos solicitados:

- Tipo o nombre de tienda.

Criterios de aceptación: Se debe poder obtener una lista de tiendas a partir del texto buscado. En caso de no obtener ningún resultado, se debe mostrar mensaje acorde a no resultados encontrados.

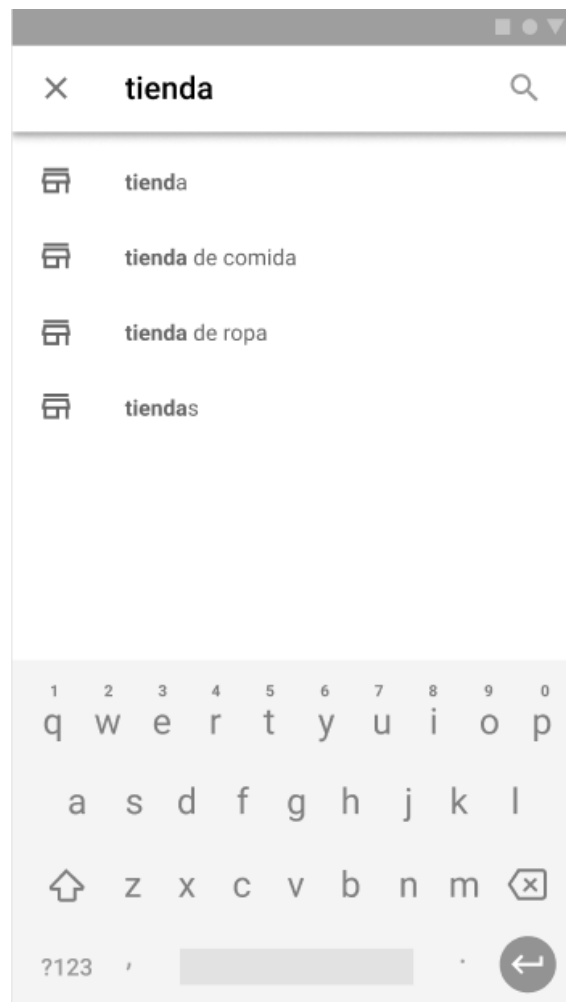


Figura 10.51: RF2 - Pantalla búsqueda de tienda encontrada.

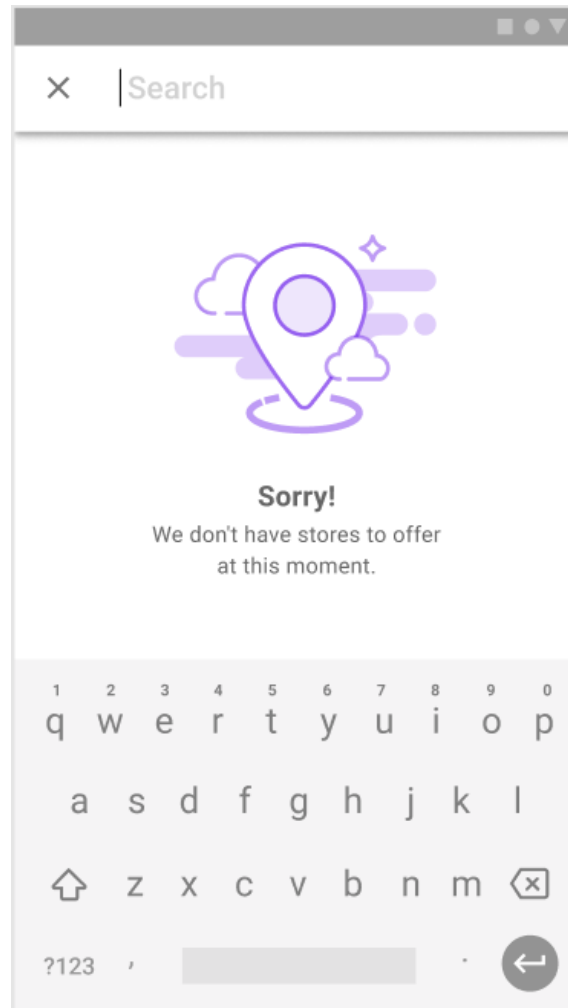


Figura 10.52: RF2 - Pantalla búsqueda de tienda no encontrada.

RF3 - Hacer reseñas

Descripción: Como usuario *app mobile* quiero ser capaz de poder dejar reseñas a una tienda.

Datos solicitados:

- Cantidad de estrellas.
- Texto de reseña.

Criterios de aceptación: Se debe poder ingresar un texto y cantidad de estrellas sobre la reseña. En caso de cumplir con los dos puntos anteriores y publicar la reseña, se tiene que desplegar un mensaje de publicación exitosa.

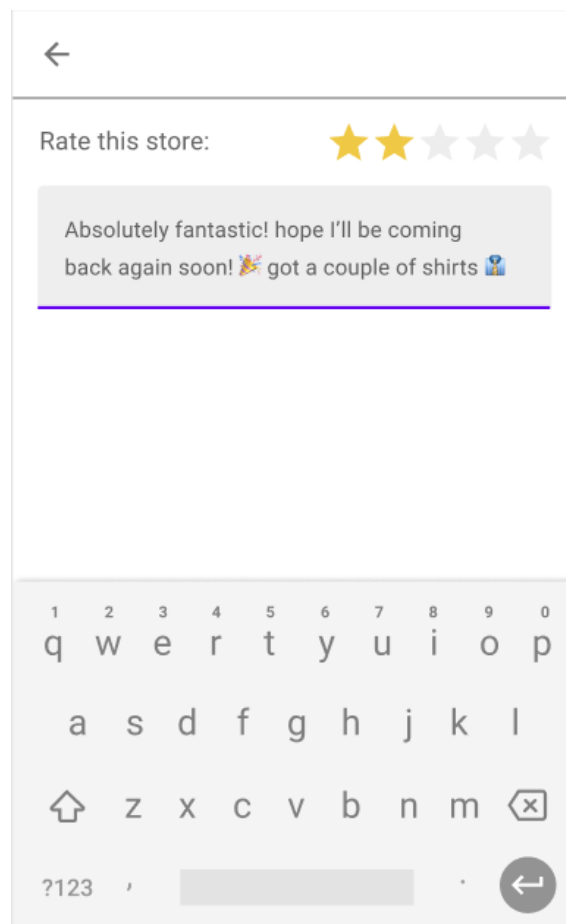


Figura 10.53: RF3 - Pantalla de realizar reseña.

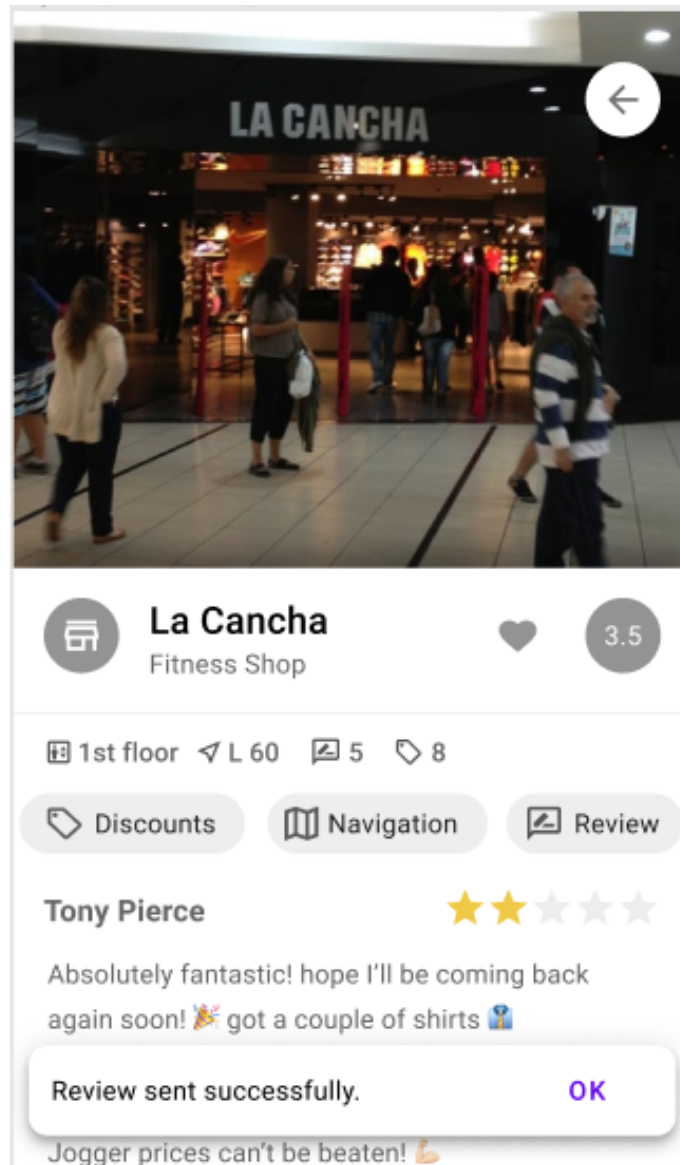


Figura 10.54: RF3 - Pantalla confirmación reseña realizada.

RF4 - Agregar a favoritos

Descripción: Como usuario *app mobile* quiero ser capaz de poder agregar una tienda a mis tiendas favoritas.

Criterios de aceptación: Se debe poder seleccionar en la descripción de la tienda como favorita. La misma tiene que pasar a estar en la lista de tiendas favoritas para el usuario.

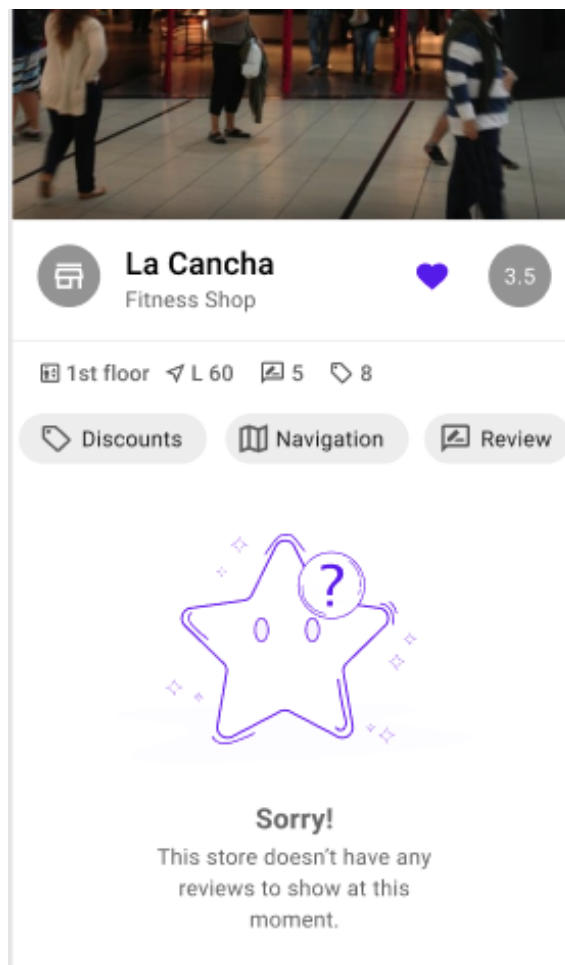


Figura 10.55: RF4 - Pantalla detalle de tienda marcada como favorita.

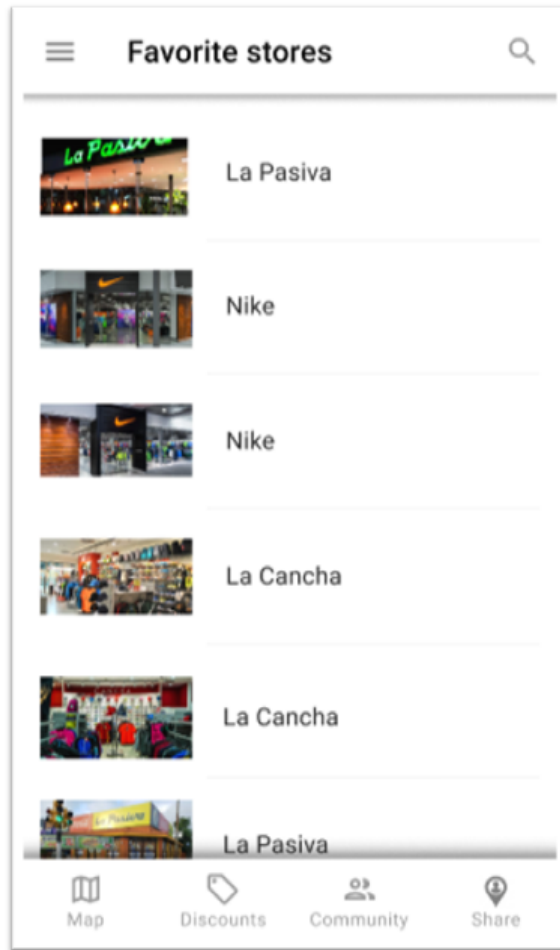


Figura 10.56: RF4 - Pantalla de tiendas favoritas.

RF5 - Información de tienda

Descripción: Como usuario *app mobile* quiero ser capaz de poder obtener la información relevante de una tienda.

Criterios de aceptación: Se debe poder seleccionar una tienda y obtener los detalles generales correspondientes a la misma, además de sus descuentos ofrecidos, promedio, ubicación, manera de llegar hasta la misma y las reseñas obtenidas por parte de los usuarios.

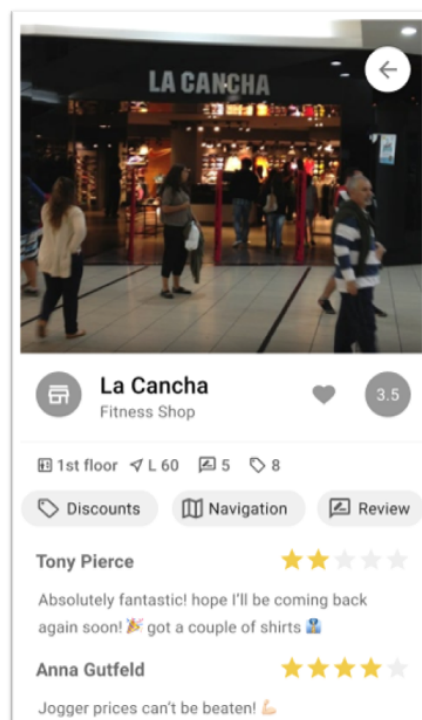


Figura 10.57: RF5 - Pantalla detalle de tienda.

RF6 - Descuentos de tienda

Descripción: Como usuario *app mobile* quiero ser capaz de poder obtener los descuentos ofrecidos por una tienda.

Criterios de aceptación: Se debe poder seleccionar una tienda y obtener todos los descuentos correspondientes a la misma. En caso de no contar con descuentos para ofrecer, se debe mostrar un mensaje acorde a no descuentos disponibles.

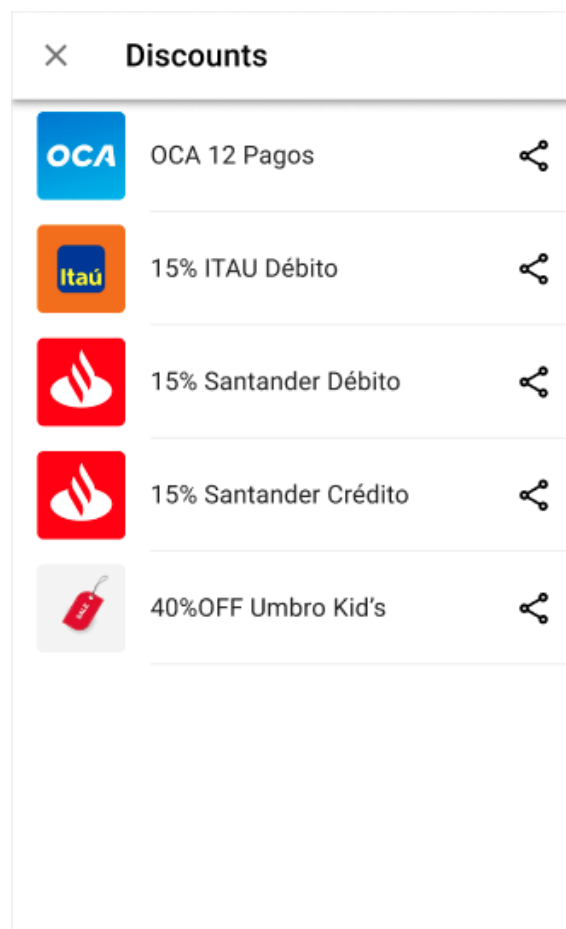


Figura 10.58: RF6 - Pantalla de descuentos de la tienda.

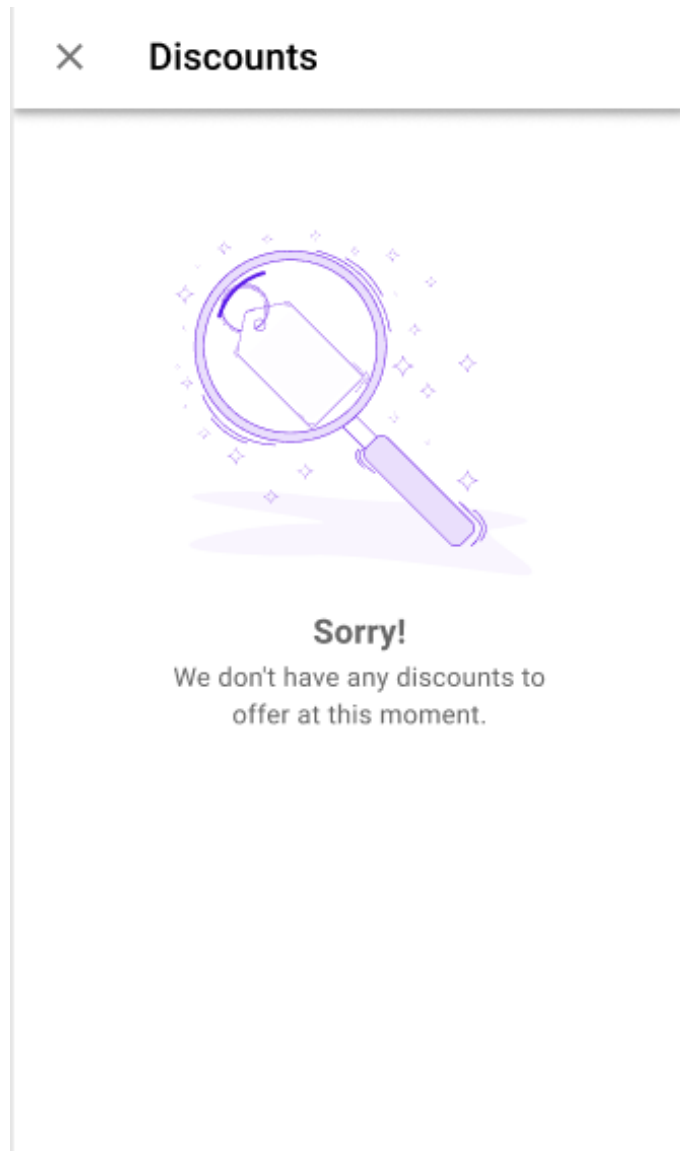


Figura 10.59: RF6 - Pantalla de descuentos no encontrados de la tienda.

RF7 - Compartir ubicación

Descripción: Como usuario *app mobile* quiero ser capaz de poder compartir mi ubicación actual con algún amigo.

Criterios de aceptación: Se debe poder elegir amigos con los cuales poder compartir mi ubicación actual. En caso de compartir la ubicación con algún amigo, se tiene que desplegar un mensaje de ubicación compartida exitosamente.

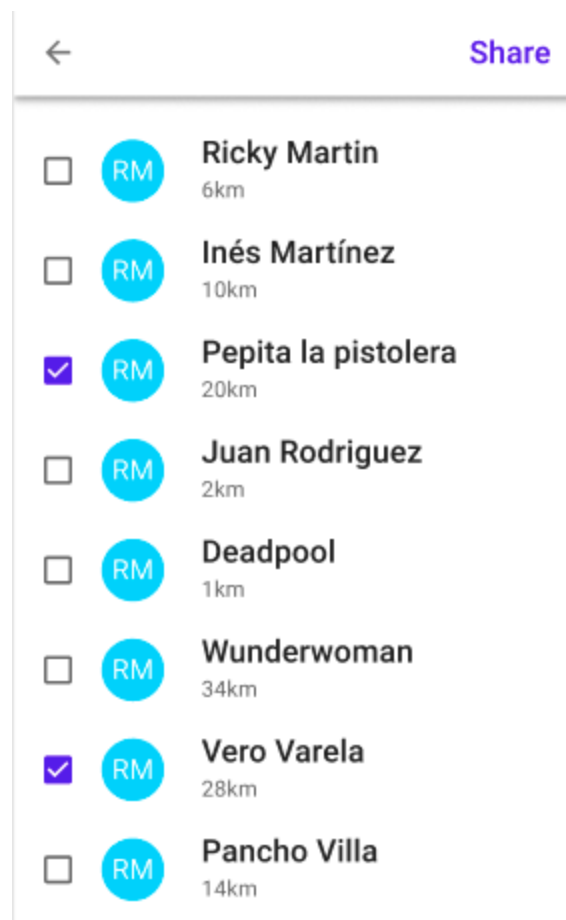


Figura 10.60: RF7 - Pantalla de lista de amigos a compartir ubicación.



Figura 10.61: RF7 - Pantalla confirmación ubicación compartida.

RF8 - Visualizar descuentos con filtros

Descripción: Como usuario *app mobile* quiero ser capaz de poder visualizar descuentos filtrados.

Criterios de aceptación: Se debe obtener una lista de descuentos a partir del texto buscado. En caso de no obtener ningún resultado, se debe mostrar mensaje acorde a no resultados encontrados.

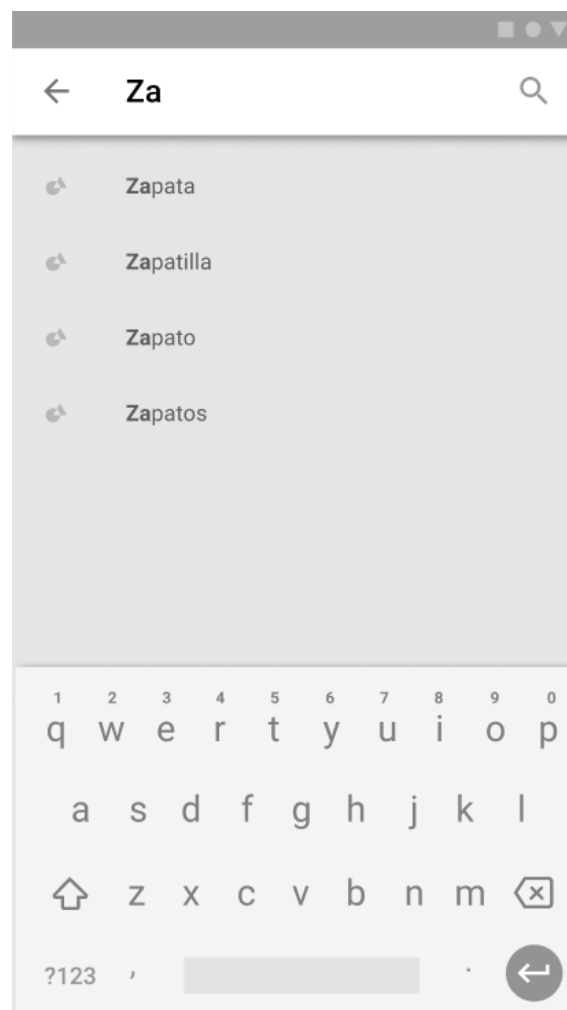


Figura 10.62: RF8 - Pantalla búsqueda de descuentos encontrada.

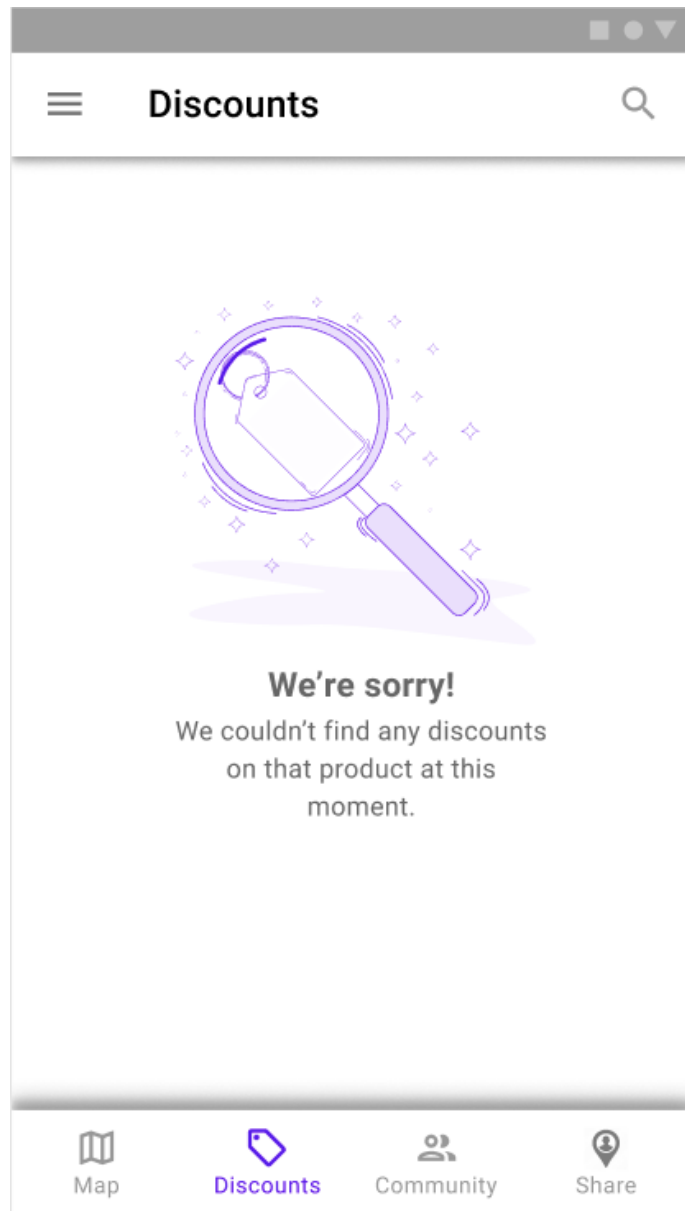


Figura 10.63: RF8 - Pantalla búsqueda de descuentos no encontrada.

RF9 - Buscar descuentos

Descripción: Como usuario *app mobile* quiero ser capaz de buscar descuentos libremente.

Criterios de aceptación: Se debe obtener una lista de descuentos en función a las preferencias de usuario en la opción de descuentos. En caso de no contar con descuentos para el usuario, se debe mostrar un mensaje acorde a no descuentos disponibles.

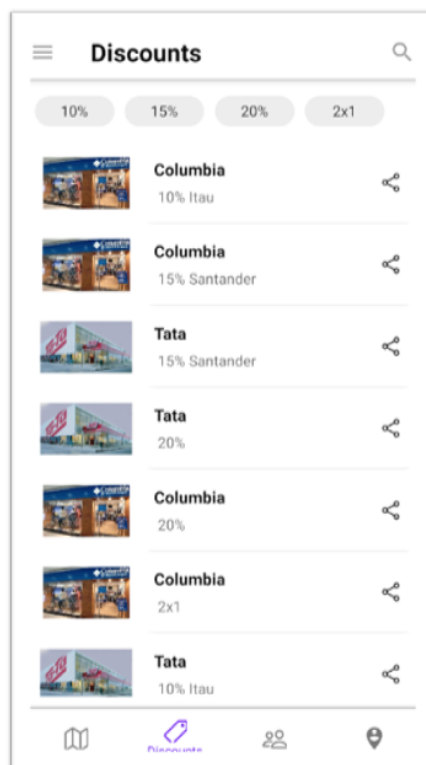


Figura 10.64: RF9 - Pantalla de descuentos en función a preferencias.

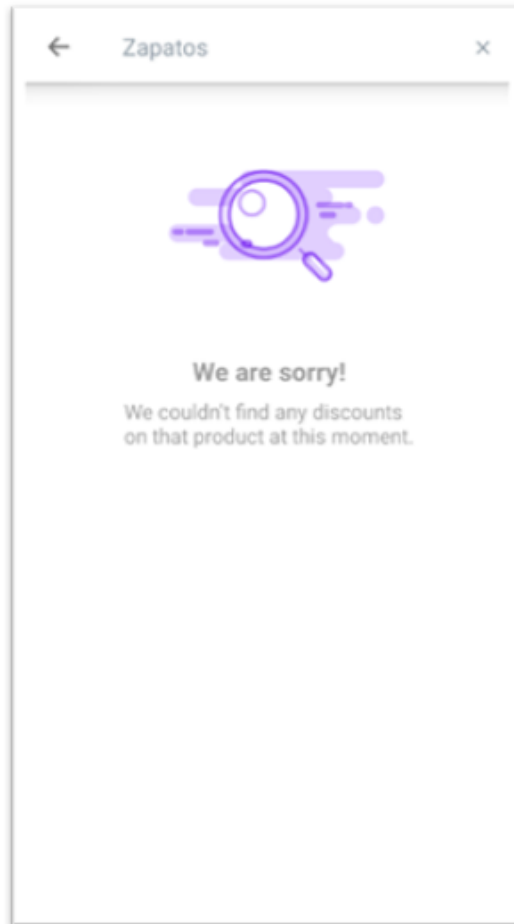


Figura 10.65: RF9 - Pantalla de descuentos no encontrados.

RF10 - Visualizar descuentos en tiempo real

Descripción: Como usuario *app mobile* quiero ser capaz de visualizar los descuentos en tiempo real en el mapa.

Criterios de aceptación: Se debe poder visualizar en el mapa, los descuentos activos en función de las preferencias de usuario en las tiendas ubicadas en el mapa. En caso de no encontrar descuentos para el usuario, se mostrará el mapa sin descuentos.



Figura 10.66: RF10 - Pantalla de descuentos en tiempo real en el mapa.

RF11 - Descuentos activos

Descripción: Como usuario *app mobile* quiero ser capaz de visualizar los descuentos que tengo activados.

Criterios de aceptación: Se deben poder visualizar todos los descuentos que fueron activados por el usuario. En caso de no contar con descuentos activos, se debe mostrar un mensaje acorde a descuentos activos no disponibles.

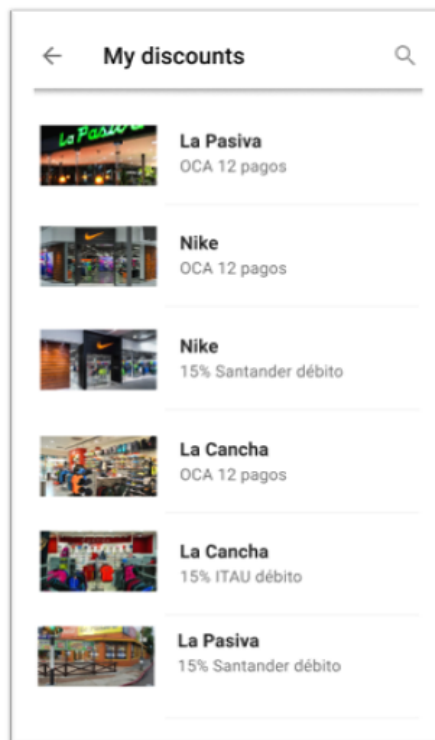


Figura 10.67: RF11 - Pantalla de descuentos activos.

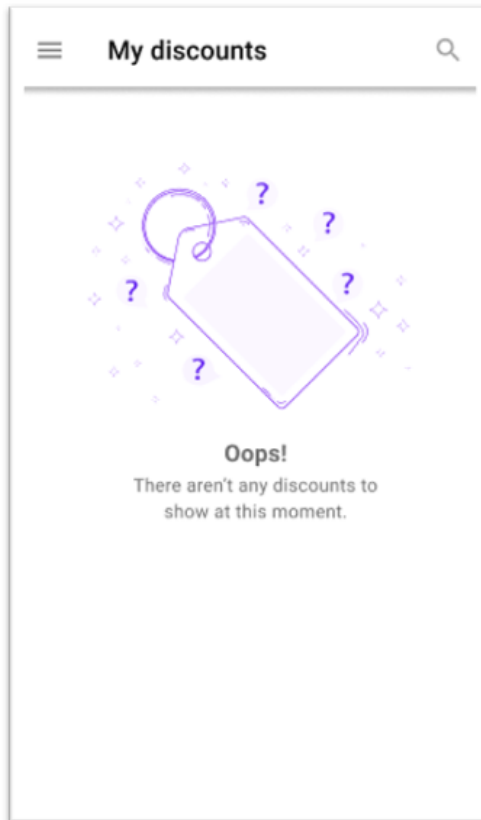


Figura 10.68: RF11 - Pantalla de descuentos activos no encontrados.

RF12 - Indicaciones para llegar a tienda que ofrece descuento

Descripción: Como usuario *app mobile* quiero ser capaz de tener indicaciones para poder llegar a la tienda que ofrece el descuento de mi interés.

Criterios de aceptación: Se debe poder obtener indicaciones de pasos a realizar desde el menú con los detalles de la tienda para poder llegar a la misma.

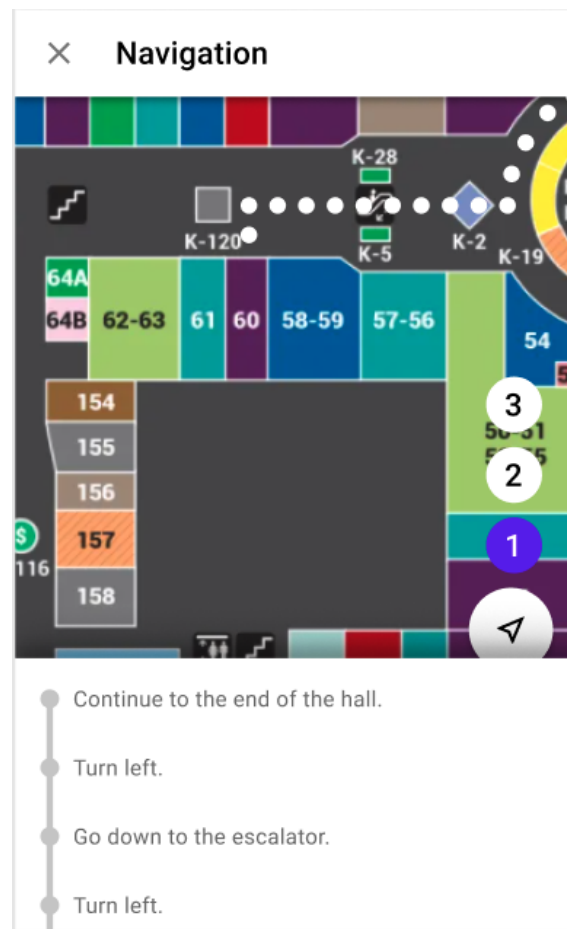


Figura 10.69: RF12 - Pantalla de indicaciones para llegar a tienda.

RF13 - Visualizar amigos

Descripción: Como usuario *app mobile* quiero ser capaz de visualizar mi lista de amigos.

Criterios de aceptación: Se debe poder obtener una lista de amigos para el usuario. En caso de no contar con amigos, se debe mostrar un mensaje acorde a no amigos disponibles.

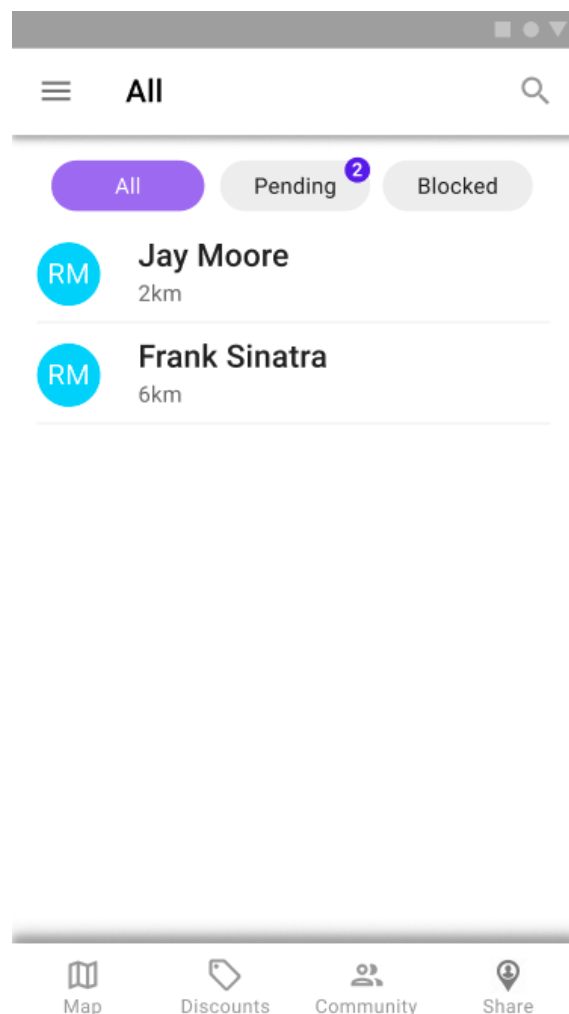


Figura 10.70: RF13 - Pantalla de amigos.

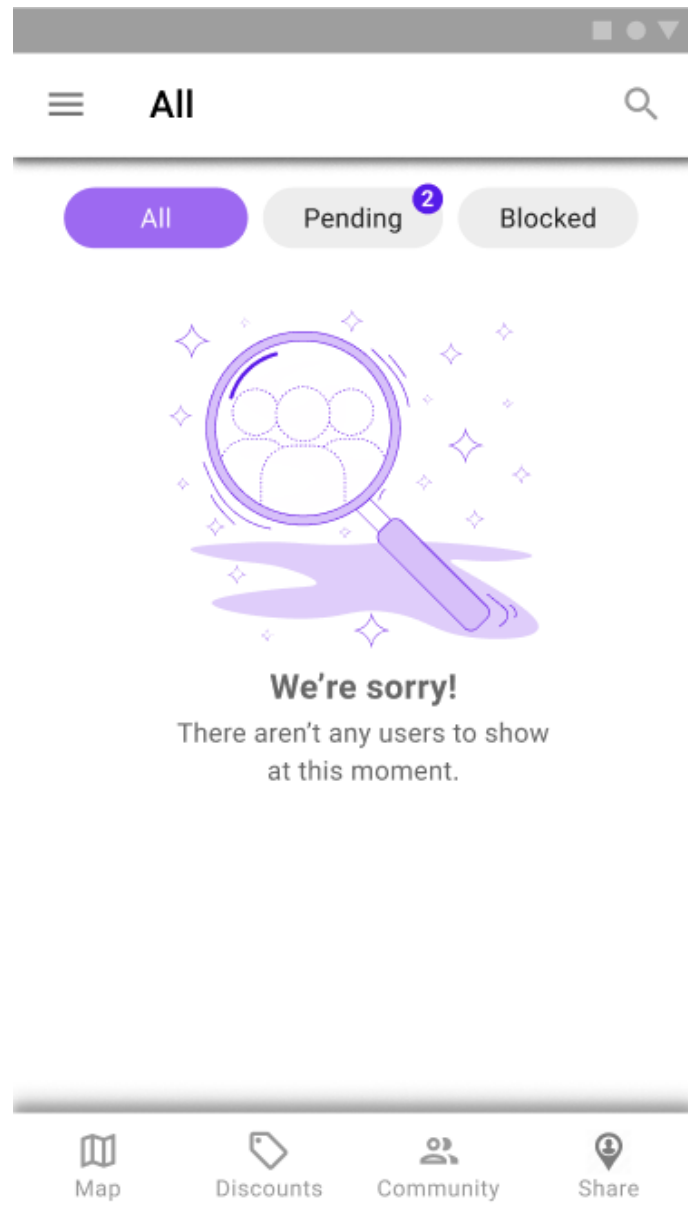


Figura 10.71: RF13 - Pantalla de amigos no encontrados.

RF14 - Agregar amigos

Descripción: Como usuario *app mobile* quiero ser capaz de agregar un usuario como mi amigo.

Datos solicitados:

- Usuario a querer agregar.

Criterios de aceptación: Se debe poder obtener una lista de todos los usuarios de la aplicación, seleccionar uno y enviarle una solicitud de amistad.

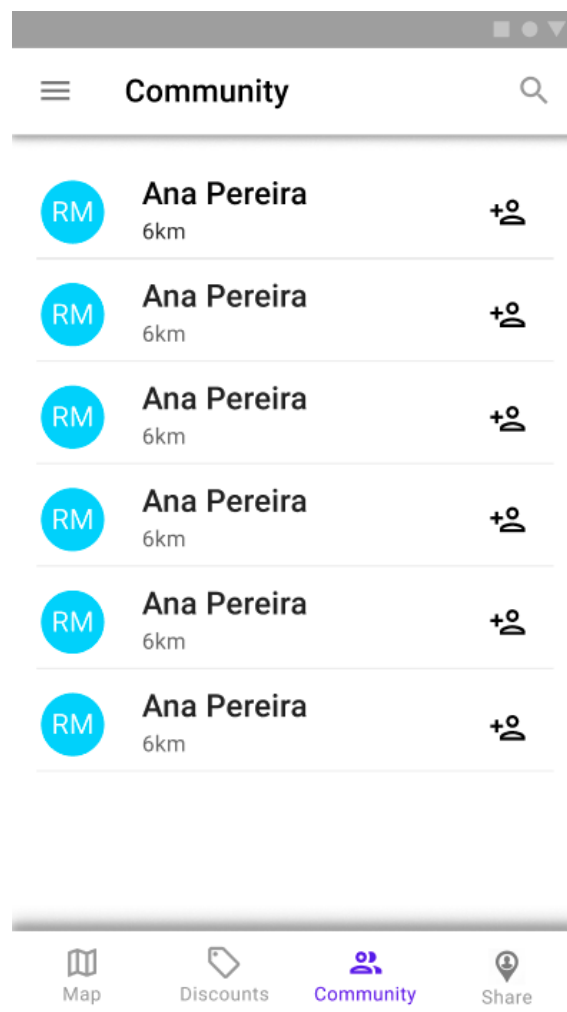


Figura 10.72: RF14 - Pantalla con lista de usuarios de la aplicación.

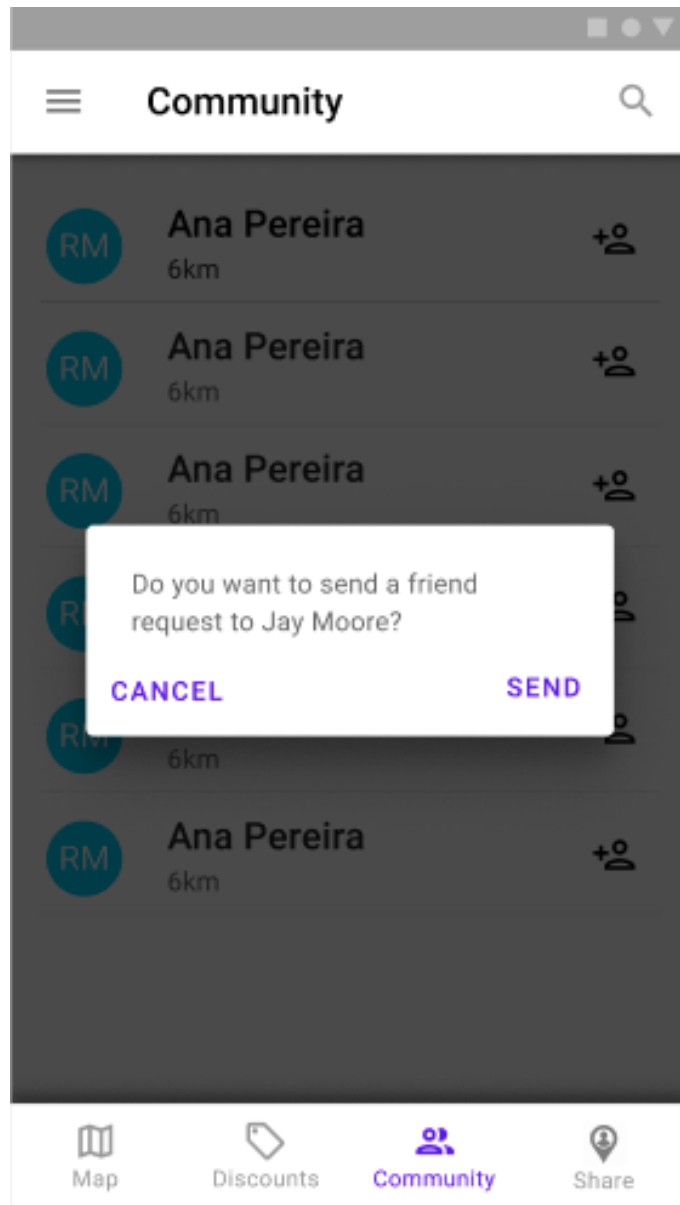


Figura 10.73: RF14 - Pantalla de envío solicitud de amistad.

RF15 - Buscar usuario

Descripción: Como usuario *app mobile* quiero ser capaz de buscar a otro usuario de la aplicación.

Datos solicitados:

- Nombre de usuario.

Criterios de aceptación: Se debe poder obtener una lista de usuarios en función al texto ingresado en el menú de *community*. En caso de no obtener ningún resultado, se debe mostrar mensaje acorde a no usuarios encontrados.

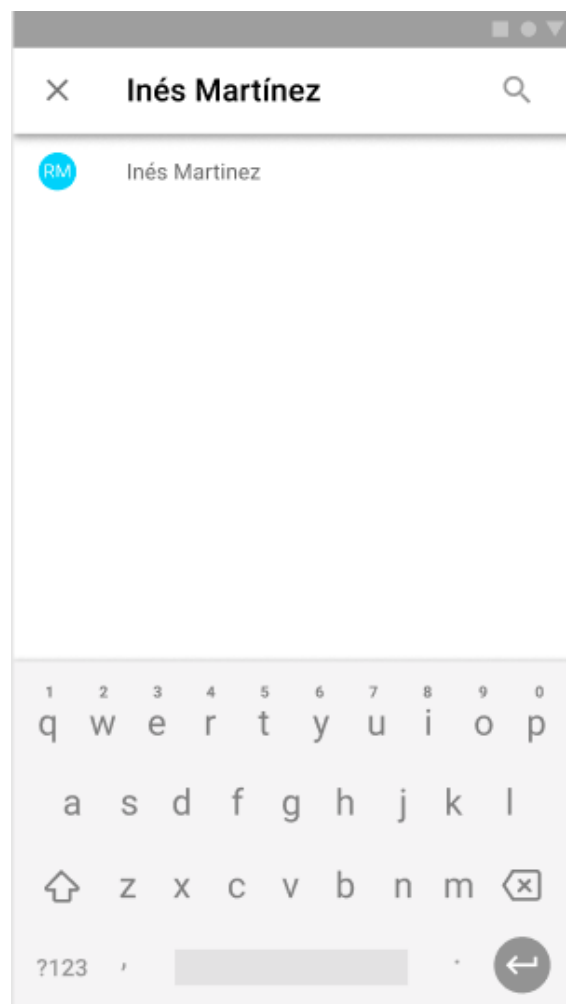


Figura 10.74: RF15 - Pantalla búsqueda de usuarios encontrados.

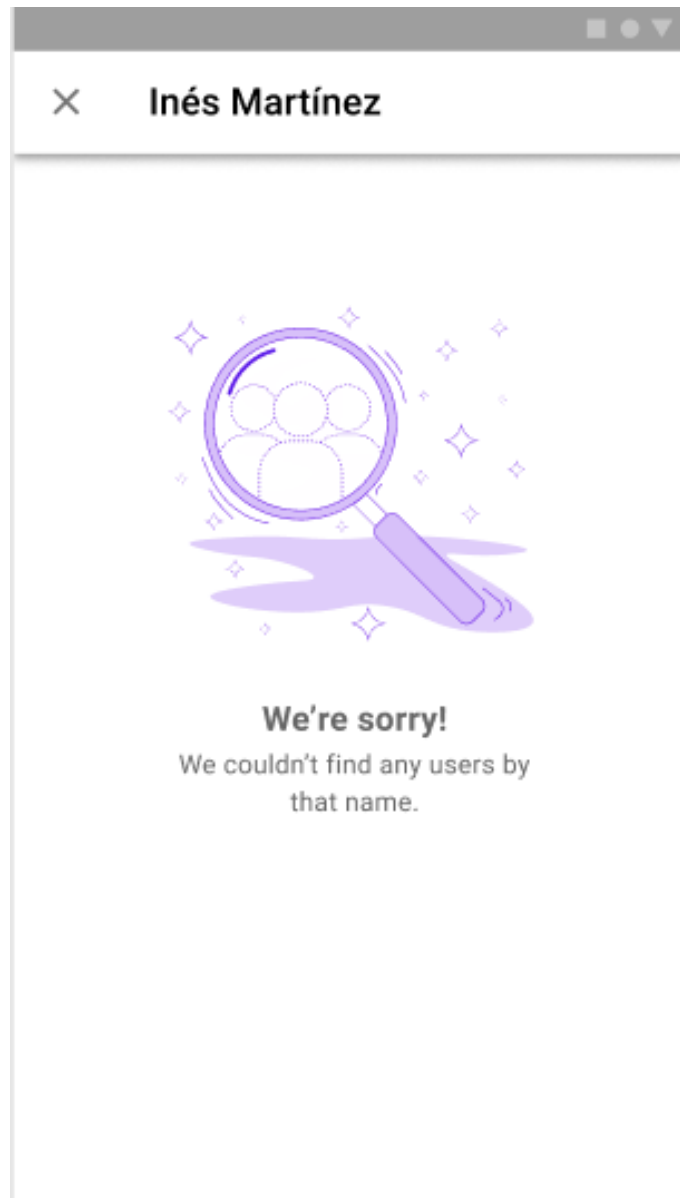


Figura 10.75: RF15 - Pantalla búsqueda de usuarios no encontrados.

RF16 - Invitar a usar la *app mobile*

Descripción: Como usuario *app mobile* quiero ser capaz de poder invitar a un amigo no usuario de la aplicación a que la use.

Datos solicitados:

- Nombre de usuario.

Criterios de aceptación: Se debe poder enviar una invitación a un amigo que no utiliza la *app mobile*, para que pueda descargársela.

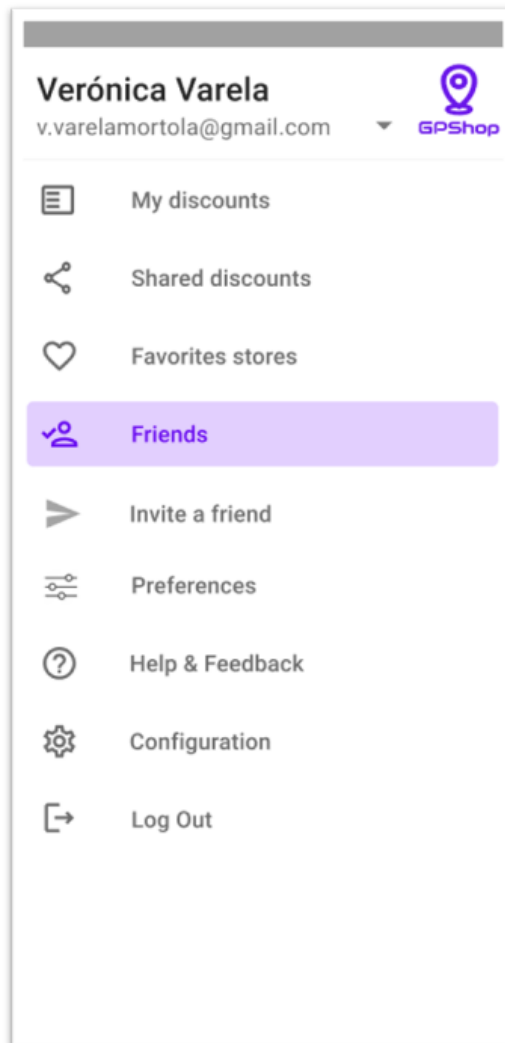


Figura 10.76: RF16 - Opción en *drawer*.

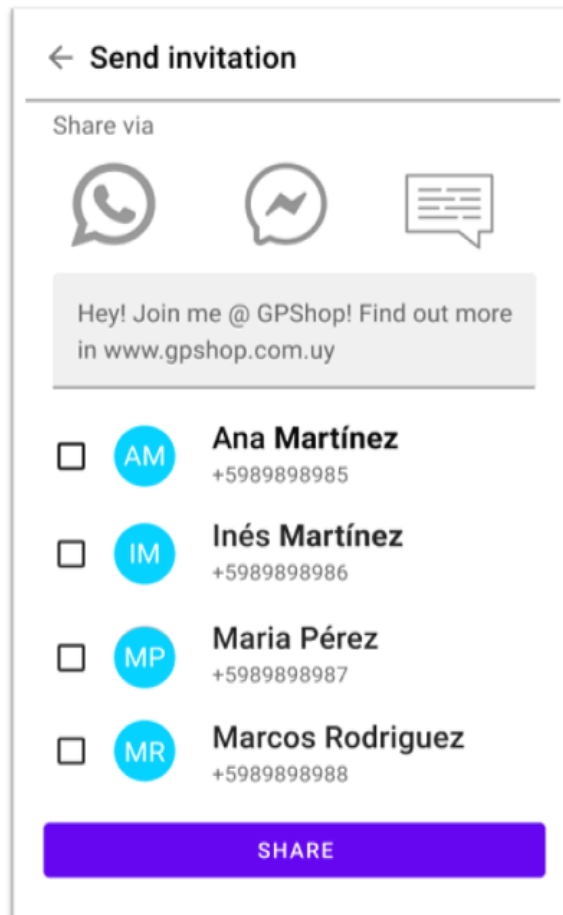


Figura 10.77: RF16 - Pantalla de invitación a la aplicación.

RF17 - Generación de QR para descuento

Descripción: Como usuario *app mobile* quiero ser capaz de poder generar un código QR para poder utilizar el descuento en la tienda que lo ofrece.

Criterios de aceptación: Se debe poder activar un QR para un descuento deseado, el mismo debe quedar en los descuentos activos del usuario. Se deberá de desplegar el QR una vez el mismo fue activado o consultado nuevamente.

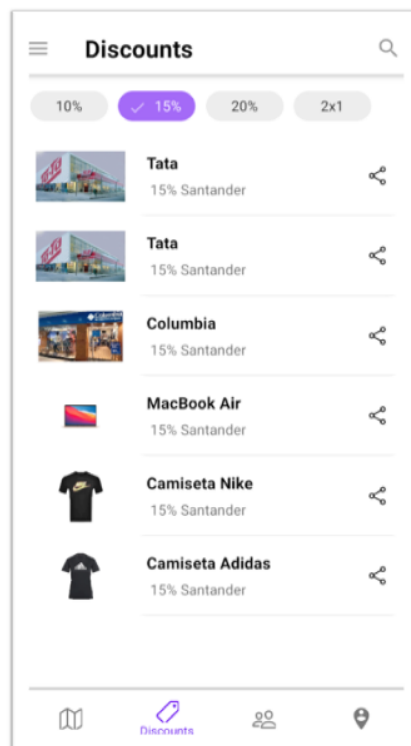


Figura 10.78: RF17 - Pantalla de descuentos filtrados.

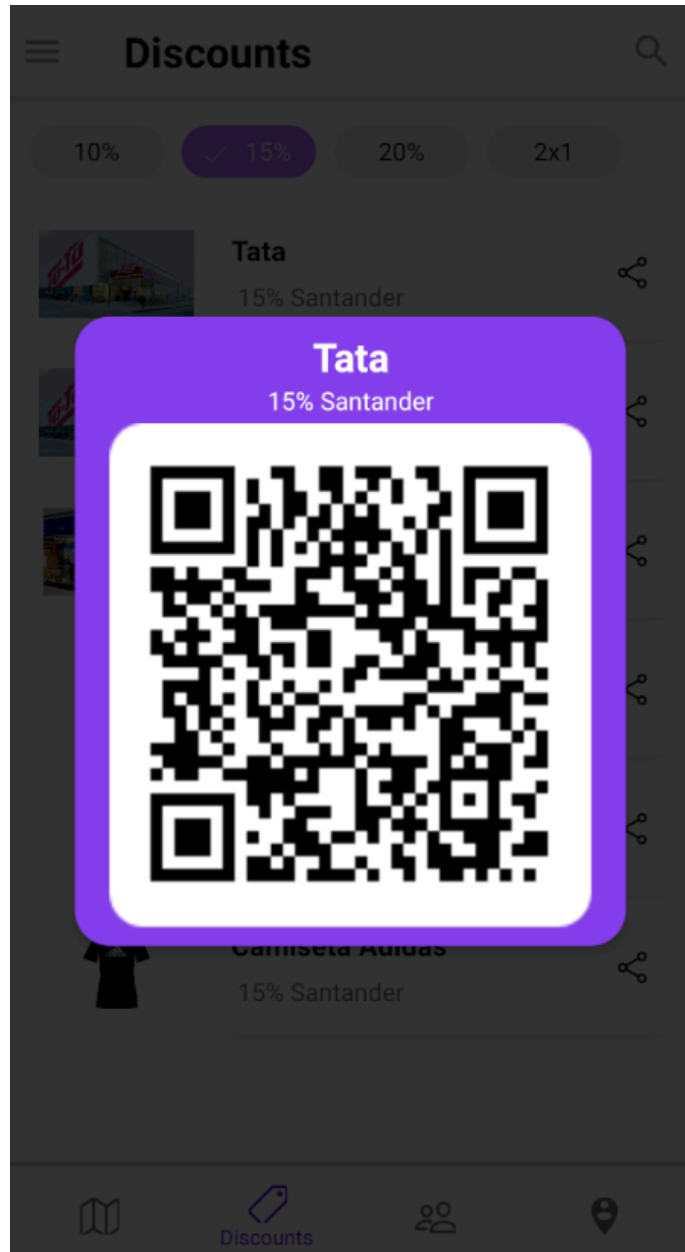


Figura 10.79: RF15 - Pantalla de QR generado para el descuento seleccionado.

RF18 - *Login*

Descripción: Como usuario *app mobile* o usuario *web* tienda quiero ser capaz de poder iniciar sesión en el sistema para poder utilizar la *app mobile*.

Datos solicitados:

- Tipo de cuenta.

Criterios de aceptación: Se debe poder realizar *login* en la aplicación mediante los servicios de autenticación de las redes sociales Gmail y Facebook.

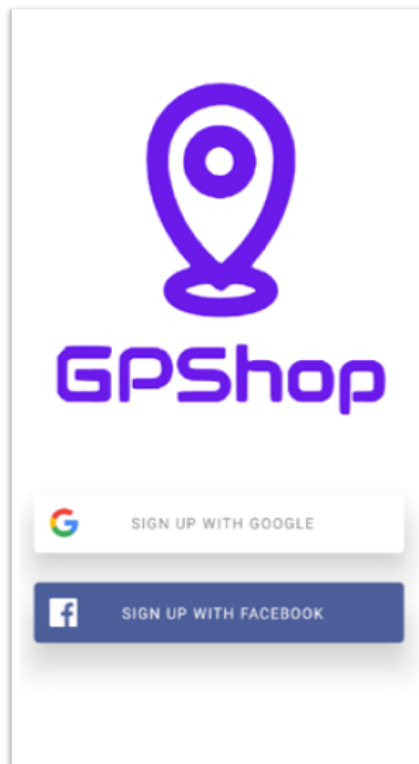


Figura 10.80: RF18 - Pantalla de *login* de la *app mobile*.

RF19 - Configuración de usuario

Descripción: Como usuario *app mobile* quiero ser capaz de poder realizar cambios en la configuración de mi perfil.

Criterios de aceptación: Se deben poder visualizar las opciones correspondientes el menú de usuario, como descuentos activos, descuentos compartidos, tiendas favoritas, amigos, preferencias, *help & feedback*, configuración y *logout*.

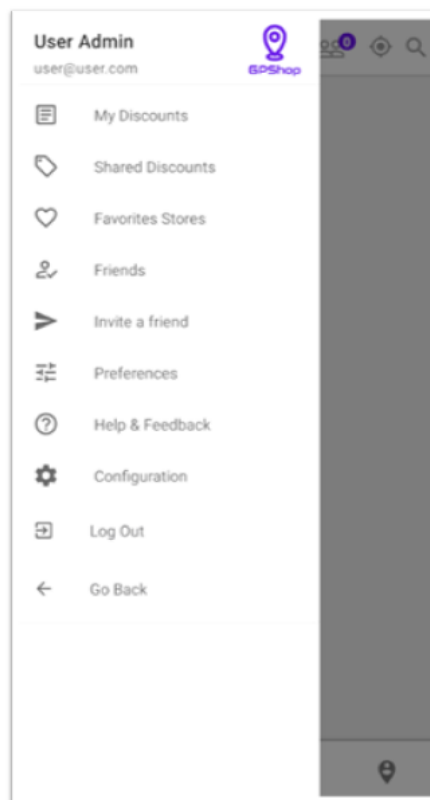


Figura 10.81: RF19 - Pantalla de configuración de usuario.

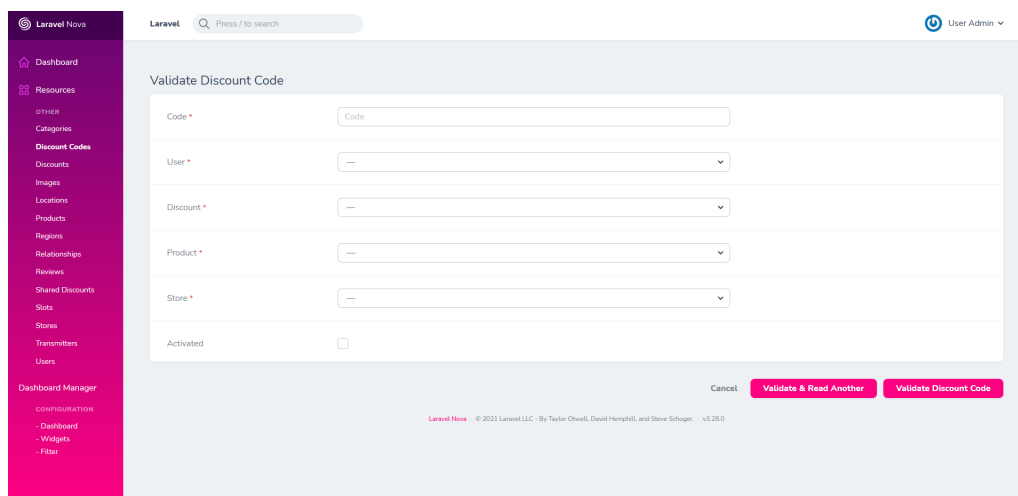
RF20 - Validación de QR activo

Descripción: Como usuario web tienda quiero ser capaz de poder realizar validaciones sobre los QR para validar el descuento.

Datos solicitados:

- Código QR.

Criterios de aceptación: Se debe poder realizar la lectura de un código QR por parte de la tienda para poder validar la veracidad de mismo y luego poder hacer efectivo el descuento.



The screenshot shows a web application interface for validating a discount code. The page title is "Validate Discount Code". The form contains the following fields:

- Code ***: A text input field with a placeholder "Code".
- User ***: A dropdown menu with a minus sign.
- Discount ***: A dropdown menu with a minus sign.
- Product ***: A dropdown menu with a minus sign.
- Store ***: A dropdown menu with a minus sign.
- Activated**: A checkbox.

At the bottom right of the form, there are three buttons: "Cancel", "Validate & Read Another", and "Validate Discount Code". The footer of the page includes the text "Laravel Nova © 2021 Laravel LLC - By Taylor Otwell, David Hemphill, and Steve Schoger v3.28.0".

Figura 10.82: RF20 - Pantalla para lectura de código QR.

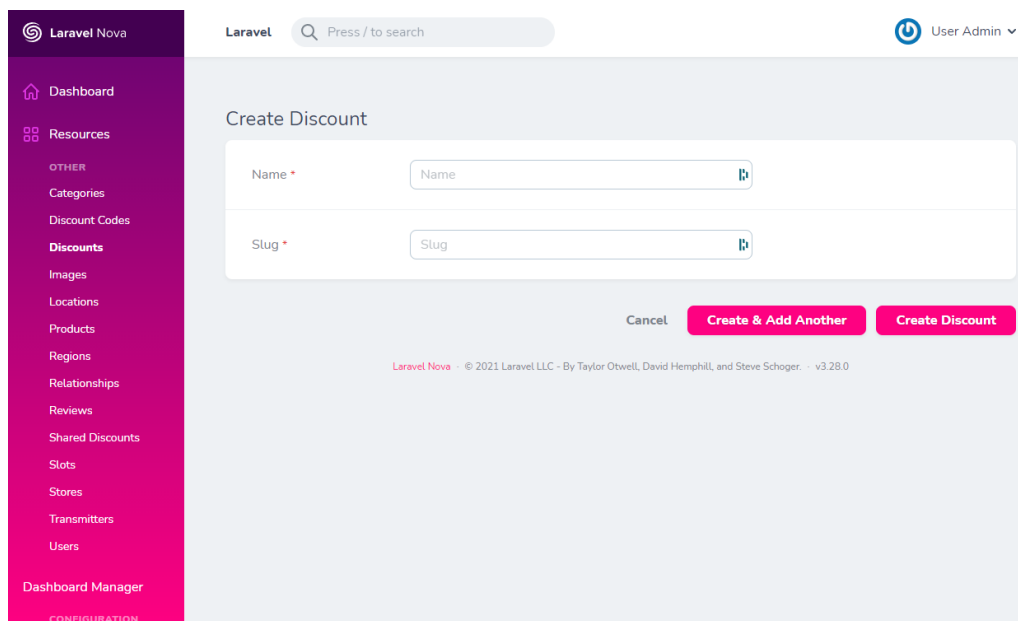
RF21 - Realizar Alta/baja/modificación promoción

Descripción: Como usuario *web* tienda quiero ser capaz de poder realizar alta, baja o modificación de una promoción.

Datos solicitados:

- Nombre de usuario.

Criterios de aceptación: Se debe poder realizar las tareas de administración básicas sobre las promociones por parte de las tiendas.



The screenshot shows the 'Create Discount' form in the Laravel Nova admin interface. The form has two input fields: 'Name' and 'Slug'. Below the form are three buttons: 'Cancel', 'Create & Add Another', and 'Create Discount'. The sidebar on the left contains a navigation menu with items like 'Dashboard', 'Resources', 'Discounts', and 'Users'. The top of the page shows the 'Laravel Nova' logo, a search bar, and the user 'User Admin'.

Figura 10.83: RF21 - Pantalla para alta de promoción.

The screenshot displays the Laravel Nova interface for managing discounts. On the left is a sidebar with a navigation menu including Dashboard, Resources, and various entity types like Categories, Discount Codes, Discounts, Images, Locations, Products, Regions, Relationships, Reviews, Shared Discounts, Slots, Stores, Transmitters, and Users. The main content area is titled 'Discounts' and features a search bar and a 'Create Discount' button. Below this is a table with columns for ID, Name, and Slug. The table contains four entries, each with a checkbox, an ID number, a name, a slug, and a set of action icons (eye, edit, delete). The footer of the interface includes the text 'Laravel Nova - © 2021 Laravel LLC - By Taylor Otwell, David Hemphill, and Steve Schoger. - v3.28.0'.

ID	NAME	SLUG	
4	2x1	2x1	<input type="checkbox"/>
3	20%	20%	<input type="checkbox"/>
2	15% Santander	15%	<input type="checkbox"/>
1	10% Itau	10%	<input type="checkbox"/>

Figura 10.84: RF21 - Pantalla para modificación o baja de promoción.

RF22 - AR

Descripción: Como usuario quiero ser capaz de poder realizar visualizaciones mediante realidad aumentada de ciertos productos.

Criterios de aceptación: Se debe poder visualizar mediante el dispositivo móvil ciertas imágenes en modalidad de realidad aumentada para ciertos productos.

RF23 - RRSS

Descripción: Como usuario quiero ser capaz de poder realizar posteos sobre los descuentos en diferentes redes sociales.

Datos solicitados:

- Red social.
- Texto del mensaje.
- Descuento.

Criterios de aceptación: Se debe poder realizar publicaciones en redes sociales sobre descuentos deseados.

10.4.4. Requerimientos no funcionales

RNF1 - Integración librería Geofence

Descripción: Se debe realizar la integración de la librería Geofence sobre la aplicación para manejo de cálculos de ubicación sobre ambientes tanto *indoor* como *outdoor*.

RNF2 - *App mobile* para iOS y Android

Descripción: Se debe disponibilizar la *app mobile* para los dispositivos móviles con sistemas operativos iOS y Android.

RNF3 - API REST sobre Laravel

Descripción: Se debe disponibilizar el *backend* mediante una API REST sobre Laravel.

RNF4 - PostgreSQL y Redis

Descripción: Se debe contar con un esquema de base de datos que soporte PostgreSQL para esquema relacional y Redis para *caching*.

RNF5 - Idioma de codificación

Descripción: Se debe realizar el desarrollo en idioma inglés, con el fin de seguir un estándar y no generar ambigüedad con mezclas de idiomas.

RNF6 - Extensibilidad

Descripción: Se debe realizar el desarrollo de tal manera que sea cerrado a modificación y abierto a extensión. Evadiendo el retrabajo en todo ámbito.

RNF7 - Pruebas de código y cobertura

Descripción: Se debe contar con pruebas automatizadas tanto para *backend* como *frontend*.

RNF8 - Flujos de acciones

Descripción: Se debe contar con flujos cortos para las funcionalidades del sistema.

RNF9 - Seguridad

Descripción: Se debe contar con seguridad entre la comunicación de los componentes de *frontend* y *backend*, mediante autenticación por token más encriptación sobre datos sensibles almacenados en la base de datos.

RNF10 - *Backend* agnóstico

Descripción: Se debe contar con un *backend* genérico que pueda ser utilizado fácilmente en un nuevo contexto cuando se quiera implementar una nueva solución.

RNF11 - Utilizar Material Design

Descripción: Se deben de utilizar los recursos disponibles de *Material Design* para la construcción de las interfaces.

10.5. Anexo 5 - Análisis de negocios

10.5.1. Introducción

En el documento presentado a continuación, se desarrollará el análisis de negocios para la situación actual correspondiente al mercado de la aplicación presentada en este proyecto (GPSshop).

Este análisis negocios consta de dos análisis de entorno que evalúan el microentorno (análisis de 5 fuerzas competitivas de Porter) y el macroentorno (análisis PESTEL). Estos análisis fueron elegidos por que se complementan entre sí, para lograr una amplia visión del entorno y el mercado donde se planea implementar GPSshop, en este caso Uruguay.

Finalmente, las conclusiones resultado de dichos análisis y la conclusión.

10.5.2. Sobre de las 5 fuerzas competitivas de Porter



Figura 10.85: Diagrama de Las 5 fuerzas competitivas de Porter

Sobre el análisis

Este análisis es un modelo estratégico confeccionado en 1979 por el Ingeniero y Profesor de la Harvard Business School, Michael Porter.

El modelo establece un marco para analizar el nivel de competencia dentro de una industria determinada, con el fin de desarrollar una estrategia de negocio, concluyendo con la respectiva estructuración de las 5 fuerzas competitivas, cuya acción conjunta define la intensidad de la competencia y la rivalidad dentro de una industria, y por consecuencia, el atractivo de dicha industria analizada en relación con las oportunidades y al potencial, tanto de inversión como de rentabilidad.

10.5.3. Las Fuerzas

Poder de negociación de los clientes

Existen 3 factores fundamentales, (aunque no los únicos) que determinan el poder de negociación de los clientes:

- Cantidad y capacidad de organización de los clientes. Si se organizan colectivamente, pueden obtener un poder de negociación mayor y pujar por mejores precios.
- Cantidad de proveedores y productos sustitutos. En otras palabras, las alternativas de los clientes a la adquisición del servicio o que tan “único” es el servicio.
- Posibilidad de integrar hacia atrás.

Rivalidad entre competidores

La rentabilidad de un sector es definida por la rivalidad de sus competidores: como regla general, un sector es más rentable cuantos menos competidores existan para el mismo y de la misma forma a la inversa.

La rivalidad se presenta cuando uno o más competidores son presionados, o ven la oportunidad de mejorar su posición en el mercado.

Los movimientos competitivos de una empresa surten efectos visibles en sus competidores, y estas buscarán contrarrestar dichos efectos, por lo que se dice que las empresas competidoras son mutuamente dependientes.

Los factores que determinan la intensidad de la rivalidad competitiva pueden cambiar y son dinámicos, algunos ejemplos son:

- Gran número de competidores o igualmente equilibrados.
- Crecimiento lento del sector industrial.
- Costos fijos o elevados de almacenamiento.
- Falta de diferenciación o costos cambiantes.
- Incrementos importantes de capacidad.
- Diversidad de competidores.
- Intereses estratégicos elevados.
- Fuertes barreras de salida.

Amenaza de nuevos entrantes

La amenaza de ingreso a un sector industrial depende mayormente en las barreras de entrada presentes en dicho sector.

Porter identifica 6 principales barreras:

- Economías de escala.
- Diferenciación del producto.
- Costos cambiantes.
- Acceso a los canales de distribución.
- Desventajas en costo independientes de las economías de escala.
- Políticas gubernamentales.

Poder de negociación de los proveedores

De manera similar a los clientes, los proveedores cuentan con 3 factores determinantes de su poder de negociación:

- Cantidad y capacidad de organización de los proveedores. Si se organizan colectivamente, pueden obtener un poder de negociación mayor y fijar los precios.
- Que tan clave es el insumo en la cadena de producción, su diferenciación o si se requieren costos altos para cambiar de proveedor.
- Que tan probable es la amenaza de integrar hacia adelante.

Amenaza de productos sustitutos

Este factor pondera la posibilidad de que el producto o servicio, sea reemplazado por otro de similares características, que satisfaga la necesidad de una manera diferente.

Por ejemplo: si nuestra necesidad es "saciar nuestra sed", podemos consumir: agua mineral, café, té, refrescos o jugos.

Si bien estos productos son diferentes, el objetivo que buscan conseguir es el mismo: "saciar la sed". Adicionalmente, cada uno de los nombrados productos, puede variar su éxito a la hora de cumplir con el objetivo de "saciar nuestra sed".

Algunos factores que pueden impactar en esta categoría son:

- La propensión del cliente a utilizar productos sustitutos.
- Los precios relativos y habilidad del producto para satisfacer la necesidad del cliente.
- Costo de sustitución y disponibilidad de sustitutos.
- Cantidad de proveedores y nivel percibido de diferenciación.

10.5.4. Sobre el análisis PESTEL

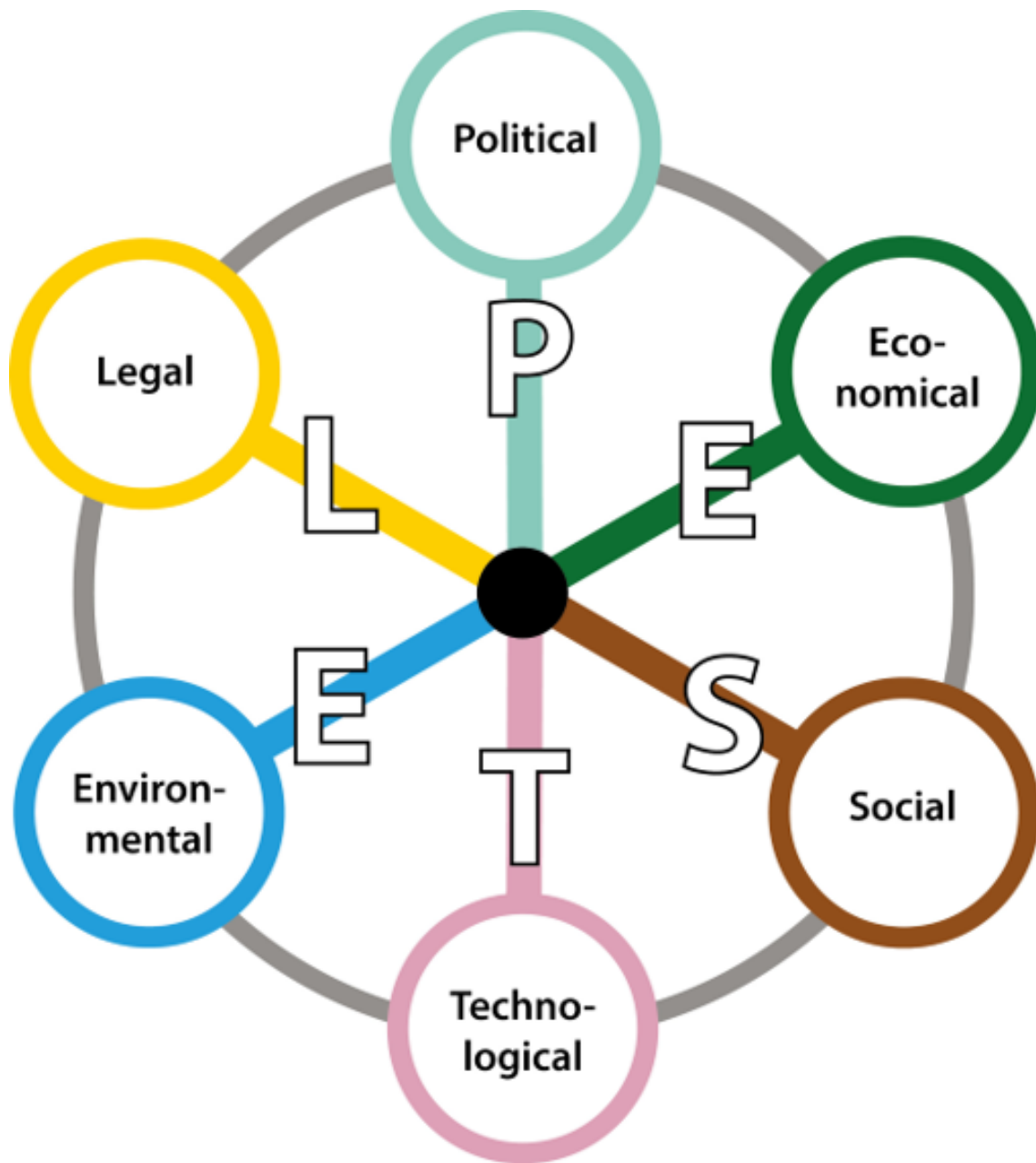


Figura 10.86: Diagrama Análisis PESTEL

Sobre el análisis

Esta herramienta de análisis estratégico es utilizada con el fin de realizar un análisis para comprender la situación del macro entorno logrando de esta forma obtener conocimiento de como nos pudiese afectar y permitiéndonos anticipar como reaccionar y desenvolvemos antes estos cambios; completando de esta forma el alcance del análisis presentado anteriormente.

Las variables que son consideradas por este análisis son las siguientes:

- Políticas.
- Económicas.
- Socioculturales.
- Tecnológicas.
- Ecológicas.
- Legales.

Política

La variable política hace referencia a:

- Las políticas de los diferentes niveles de gobierno
- Política fiscal
- Tratados comerciales
- Subvenciones para el sector

Económica

La variable económica se refiere a:

- Niveles de renta.
- inflación.
- Niveles de desempleo.
- Tipos de cambio.
- Factores macroeconómicos de cada país.

Socioculturales

La variable sociocultural está centrada en:

- Nivel educativo de la población.
- La forma en la que nos comunicamos y relacionamos.
- Nivel de ingresos de la población.
- Diversidad de la población.
- Estructura familiar y de edad de la población.

Tecnológica

La variable tecnológica enfoca su análisis en:

- Las nuevas formas de producción y/o distribución.
- Los niveles de inversión en I+D.
- Nuevas tecnologías o paradigmas tecnológicos que puedan provocar algún tipo de disrupción.

Ecológica

La variable ecológica hace énfasis en:

- El impacto en la contaminación.
- Conciencia social ecológica.
- Leyes de protección medioambientales.

Legal

La variable legal tiene su foco en:

- Privacidad de los usuarios y tratamiento de datos.
- Sectores protegidos o regulados.
- Leyes de protección medioambientales.
- Leyes antimonopolio.

10.5.5. Análisis de las 5 fuerzas competitivas de Porter

Clientes

Definimos a nuestros clientes como: Todos aquellos lugares, en espacios abiertos o cerrados que contengan más de un local comercial en su superficie.

Algunos ejemplos incluyen, pero no se limitan a:

- Shopping Centers y malls.
- Grandes superficies.
- Galerías y Strip-Malls.

Si bien podría decirse que en este sector los clientes están organizados y agrupados, se está volviendo un mercado crecientemente más competitivo dada la gran cantidad de nuevos proyectos e inversiones.

En cuanto a la cantidad de proveedores y sustitutos, si bien en el mercado internacional existen empresas y aplicaciones que proveen servicios similares, no existen de momento proveedores de este servicio a nivel local.

Debido a la complejidad y al grado de expertise e inversión en I + D y a la distancia que representa del modelo de negocios actual de los clientes, las posibilidades de integración hacia atrás son remotas.

Proveedores

A la fecha, existen multitud de fabricantes de Beacons BLE con diversos precios y calidades y con alta capacidad de producción. Por este motivo podríamos catalogar a este insumo como un commodity.

Si bien contar con este insumo es fundamental para la continuidad del servicio, su consideración como un commodity nos da la pauta de que un eventual cambio de proveedores no generaría mayores interrupciones en el funcionamiento del negocio.

De manera similar al análisis de los clientes, la complejidad y el grado de expertise e inversión en I + D y a la distancia que representa del modelo de negocios actual de los proveedores, la integración hacia adelante es muy poco probable.

Sustitutos

Como posibles sustitutos del servicio, podemos encontrar diversas alternativas que incluso muchas veces coexisten en paralelo para alcanzar a diferentes sectores de la población.

Según el público objetivo del negocio podemos encontrar como sustitutos los siguientes:

- Vouchers de descuentos, panfletos y librillos de ofertas.
- Publicidad en internet (mailing, RRSS, anuncios en la web).
- Anuncios en la vía pública, radio y televisión.
- Tarjetas de fidelización y códigos promocionales.

Nuevos entrantes

Si bien nos encontramos ante un potencial nuevo mercado, donde el acceso a los materiales es relativamente abundante, como fue mencionado anteriormente, es necesaria una considerable inversión en I + D, y en recursos con la expertise adecuada al nivel necesario para el despliegue y puesta en marcha exitoso del equipamiento y el sistema, también es de suma importancia el efecto de red, que podría considerarse la ventaja competitiva clave del sector.

Rivalidad en el sector

Actualmente es un mercado que a nivel local se encuentra virgen, y con pocos posibles competidores directos e indirectos.

La mayor barrera de entrada se encuentra en los ya mencionados expertise e inversión en I + D, y en el posible efecto de red del primer competidor que ingrese al mercado.

10.5.6. Conclusiones análisis 5 fuerzas

Clientes:

Podemos concluir que, si bien los clientes se encuentran agrupados y organizados, la competencia del sector tiende al alza, y dada la poca, por no decir nula capacidad de integrar hacia atrás o la nula competencia del sector, el poder de los clientes es moderadamente bajo.

Proveedores:

En cuanto a proveedores, podemos concluir que estamos en un mercado de compradores.

Los proveedores son muchos y no están organizados, y si bien el producto es crucial para el desarrollo de la actividad comercial, estamos frente a un bien que podría tomarse como un commodity, porque su diferenciación es escasa y el costo de cambio es prácticamente nulo.

Por último la probabilidad de integración hacia adelante es extremadamente baja, dado que es necesario una inversión considerable en I + D, y es un mercado demasiado alejado de las actividades clave del negocio de los proveedores.

Sustitutos:

Para los sustitutos, podemos observar que no existe un sustituto que cumpla de manera alternativa con todas las funcionalidades proporcionadas por el producto, y que la combinación de varios de ellos puede acercarse, aunque dejando de lado la característica de geolocalización y a un costo considerablemente superior al que es posible establecer con el proyecto.

Por lo tanto podemos concluir que la amenaza es moderadamente baja.

Nuevos entrantes:

En cuanto a nuevos entrantes, el mercado es nuevo y lleno de oportunidades, las barreras de entrada son bajas y puede resultar un mercado muy atractivo, si bien requiere una considerable inversión en I + D, esa es una ventaja competitiva que, aunque importante, es eventualmente saldable, se requiere una buena estrategia para llegar primero y establecer barreras de entrada, el efecto de red puede ser una barrera de entrada muy importante a corto y mediano plazo.

Por todo esto podemos concluir que la amenaza es moderada.

Rivalidad del sector:

Como es un mercado en eclosión, no existen proveedores del servicio a nivel local.

Las barreras de entrada son relativamente bajas, y es fundamental lograr la mayor expansión posible para conseguir el efecto de red lo más antes posible.

Se puede considerar la rivalidad como nula, pero con mucho potencial si no se levantan las barreras de entrada adecuadas.

10.5.7. Análisis PESTEL

Política:

- Uruguay cuenta con un interesante abanico de medidas para impulsar las inversiones, que incluye, pero no se limita a zonas francas, agencias dedicadas a la promoción y asesoramiento para inversiones y leyes especiales de promoción de inversiones.
- La postura comercial del Uruguay tiende a la apertura de la inversión y el comercio de manera multilateral y está orientado no solo a la región sino al mundo.
- Uruguay en este momento está rankeado en el puesto 11 para Latinoamérica y caribe del índice Doing Business, sin embargo, existe un gran esfuerzo y premura por mejorar las condiciones.
- Recientemente (08/09/2021), se confirmó la negociación de un TLC con China quien es el principal destino de las exportaciones uruguayas desde 2013.

Económica:

- En el apartado económico, como fue mencionado anteriormente, Uruguay busca liberalizar su economía y apuesta por la apertura comercial multilateral apuntando a la región y al Mercosur, así como al resto del mundo.
- En los últimos años se ha incrementado el número de zonas francas y se ha agilizado los trámites para la realización de inversiones, generando las condiciones para el germinado de la industria de servicios internacionales, TICs e I + D.
- Si bien la economía mundial se encuentra desacelerada dada la coyuntura sanitaria actual, Uruguay ha estado al frente como uno de los países que mejor ha enfrentado la emergencia sanitaria, con tasas de vacunación elevadas, y con bajo nivel de contagio y hospitalización, lo que según el BID augura una rápida recuperación económica.

Sociocultural:

- Si bien la economía viene cumpliendo 17 años de crecimiento ininterrumpido hasta 2019, hasta 2014, la economía crecía a impulso de todos los componentes de la demanda. Desde entonces, el consumo siguió creciendo, pero a menor ritmo, solo sostenido por el salario mientras el empleo caía. Esta situación se ve agravada por la coyuntura sanitaria actual, aunque el desempleo parece disminuir, producto del comienzo de la reactivación económica causada por la reapertura de los sectores comerciales afectados por la crisis sanitaria, y las flexibilizaciones de contingencia implementadas por el Ministerio de Trabajo y Seguridad Social.
- En cuanto al sistema educativo y a la educación de la población, Uruguay se encuentra por encima del promedio de la región, con tasas de alfabetización del 98 %.
- La media de edad de la población es de 36 años.

Tecnológica:

- Uruguay cuenta con un mercado avanzado de telecomunicaciones, con una excelente infraestructura y tasa más alta de penetración de banda ancha en Latinoamérica.
- Fruto del alto grado educativo mencionado anteriormente se cuenta con talento STEM comprometido, competitivo y altamente calificado.
- Exportaciones TIC 100 % exoneradas de Impuesto a la Renta.
- Uruguay es el mayor exportador de software per cápita de Latinoamérica.
- Sector declarado de interés desde 1999.

Ecológica:

- En el año 2018 el 38 % de la generación de energía eléctrica fue eólica (siendo el segundo país del mundo en porcentaje de esta energía en su matriz eléctrica), el 7 % de biomasa, algo más del 3 % fotovoltaica, la térmica poco menos del 3 % y el resto -casi la mitad- hidroeléctrica.
- En la actualidad Uruguay es uno de los países líderes en el mundo en producción de energía eólica, junto con Dinamarca, Irlanda y Alemania, con más de un tercio de su electricidad proveniente de parques eólicos.

- Uruguay repite este año como el país latinoamericano con una mejor transición energética hacia las fuentes renovables, ocupando un destacado 13^o puesto en el índice global elaborado por el Foro Económico Mundial y la consultora Accenture, publicado ayer y liderado por Suecia.

Legal:

- Ley de promoción de inversiones: Todos los proyectos de inversión bajo este régimen y los promovidos por el Poder Ejecutivo, tienen la posibilidad de computar como parte del pago de IRAE entre el 30% y el 100% del monto invertido.
- Ley de zonas francas: Los beneficios otorgados a las empresas establecido incluyen la exención del 100% de los impuestos nacionales, así como los dividendos distribuidos entre accionistas extranjeros también están exentos de los impuestos en el país.
- Beneficios específicos al sector de TI: Exenciones fiscales para exportación de programas informáticos y servicios conexos, exención para los productos informáticos, exención para los servicios informáticos, incentivo vinculado al talento.
- Ley de empleo juvenil.

10.5.8. Conclusiones

Durante los últimos dos años, la movilidad de la sociedad se ha visto afectada por la situación sanitaria que afecta a nivel mundial, llevando a tener que reinventarse a varios segmentos del mercado, con el fin de poder seguir realizando sus actividades en la llamada nueva normalidad y como consecuencia de esto, la necesidad de tener nuevas herramientas que se logren adaptar al nuevo estilo de vida instaurado por la crisis sanitaria.

Adicionalmente, las grandes superficies se enfrentan a un cambio generacional, donde la tendencia se ubica en favorablemente hacia el sector del e – commerce. Si bien en el 2019 Uruguay fue el país con más ventas de retail per cápita de Latinoamérica, el sector del e – commerce está en continuo crecimiento, apalancado en la situación sanitaria.

Esta aplicación pretende satisfacer la necesidad de ser ese canal mediante el uso de la geolocalización en exteriores e interiores para hacer llegar a los usuarios ofertas de su interés basados en la localización del usuario, así como también otros servicios también basados en la localización, como ubicación de las tiendas dentro del centro comercial.

De esta forma también se pretende ser el medio por el cual el sector retail incorpore esta pieza para proporcionar esa experiencia diferente a sus clientes.

Teniendo esto en cuenta y de acuerdo con lo analizado anteriormente se puede concluir que el macro – entorno presenta una oportunidad favorable, con un mercado en el principio de una acelerada recuperación, en un entorno político, económico y legal que demuestran una postura de apoyo al sector, y un mercado virgen con un gran potencial.

El desarrollo de este mercado debe tener un despliegue ágil y levantar barreras de entrada rápidamente. La estrategia debe ser apuntar a la diferenciación, con un nivel de servicio que mantenga al cliente satisfecho y a su vez apoyarlo en la implementación de la nueva tecnología poniendo al servicio consultores especializados que le asesoren para conseguir éxito en la utilización de la plataforma.

10.5.9. Fuentes

- Oportunidades de inversión Sector Retail
- Magic Quadrant for Indoor Location Services, Global
- Clasificación de las economías, DOING BUSINESS
- Examen de las Políticas Comerciales
- Plan estratégico de política exterior 2020-2025
- Lacalle Pou recibió carta de China y anunció que está negociando un TLC
- BID destaca recuperación comercial de Uruguay tras golpe de covid-19
- Covid-19 Data Uruguay - WHO
- Visualizador de casos coronavirus COVID-19 en Uruguay
- Repensando Uruguay: Fortalezas, debilidades y lecciones aprendidas tras la pandemia
- Instituto Nacional de Estadística, Uruguay
- Información sobre medidas del MTSS relacionadas al Covid-19
- El sistema educativo uruguayo: Estudio de diagnóstico y propuesta de políticas públicas para el sector

10.6. Anexo 6 - Flujos principales aplicación

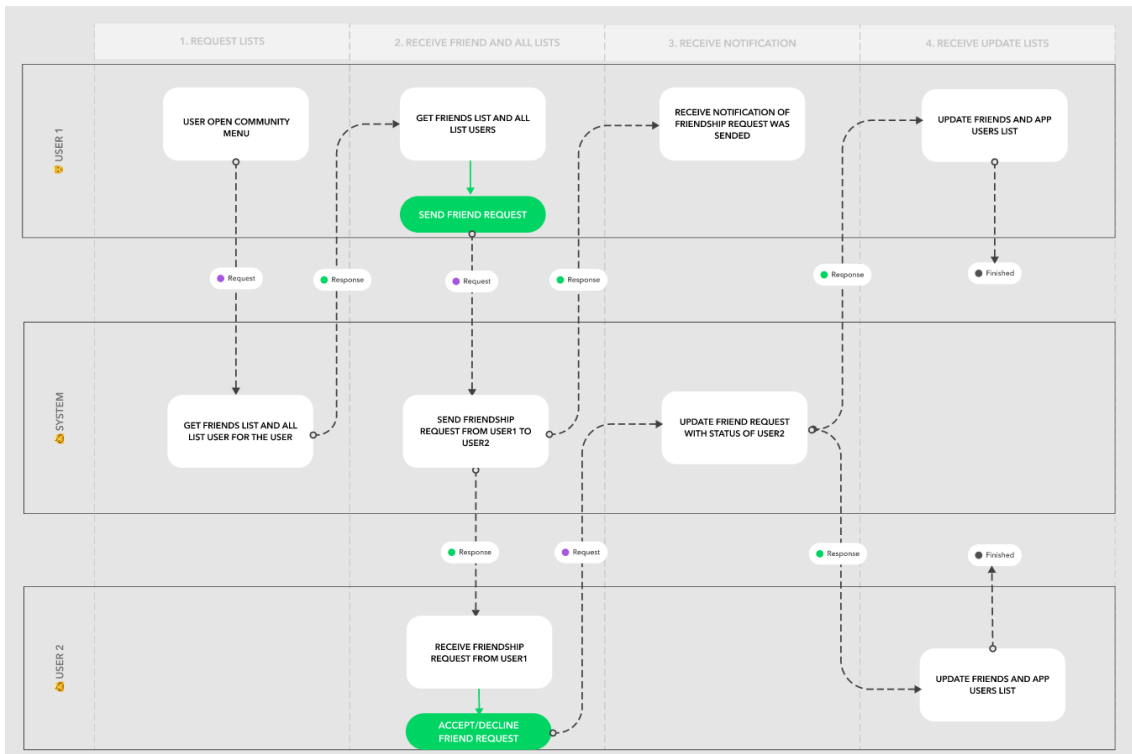


Figura 10.87: Flujo community add friend.

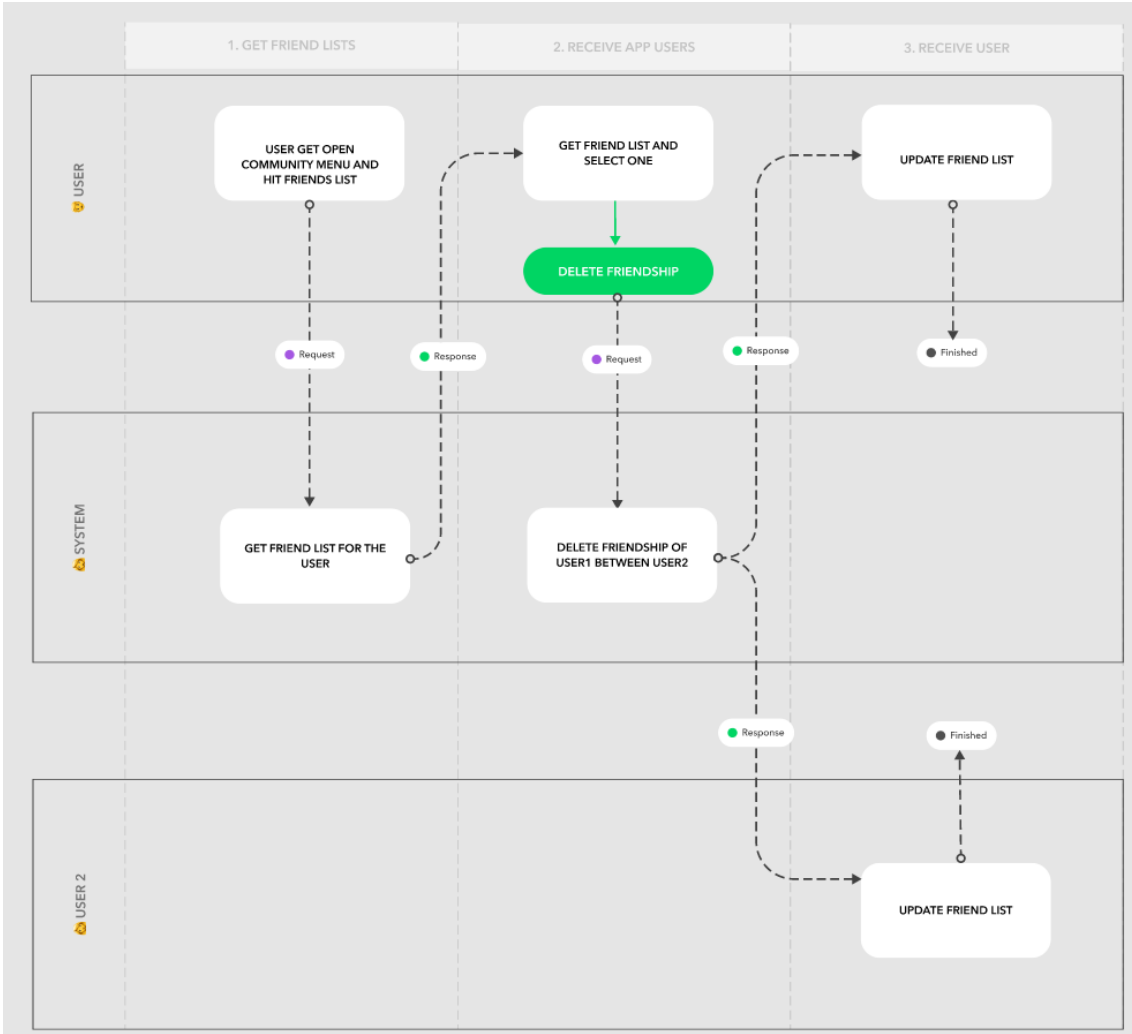


Figura 10.88: Flujo community delete friend.

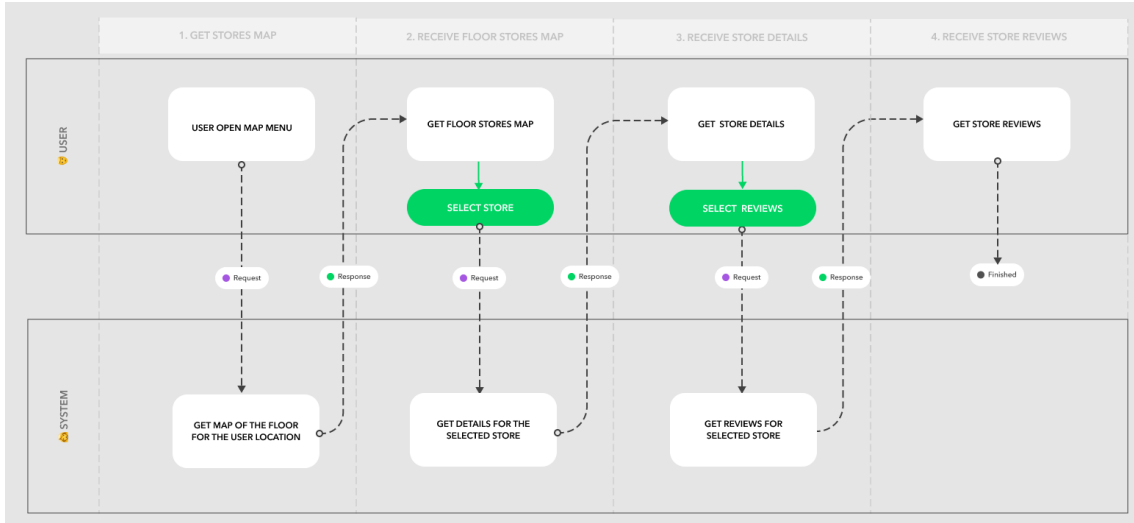


Figura 10.89: Flujo store get reviews.

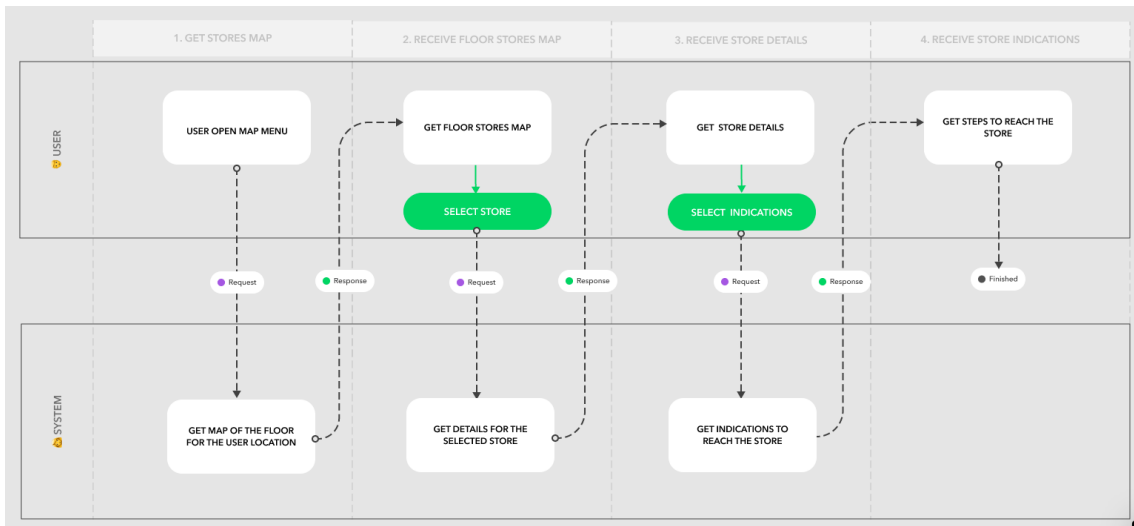


Figura 10.90: Flujo reach store location.

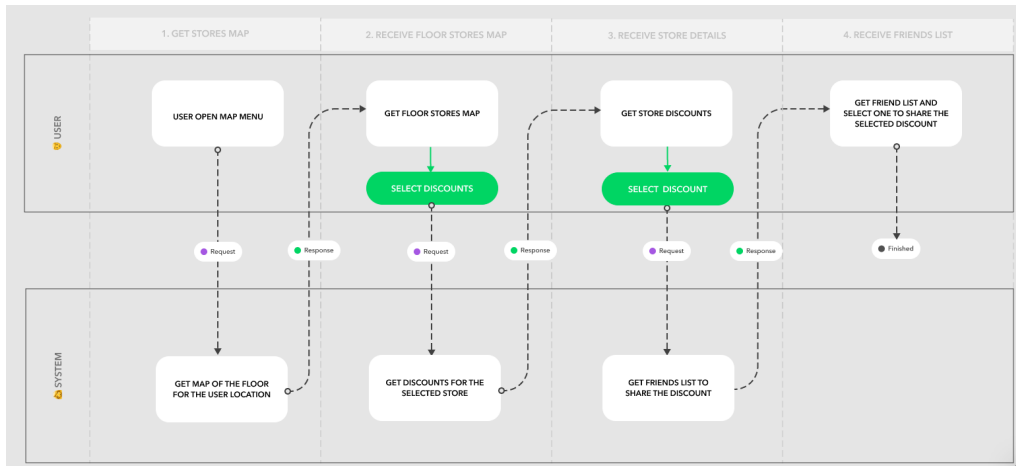


Figura 10.91: Flujo share store discount.

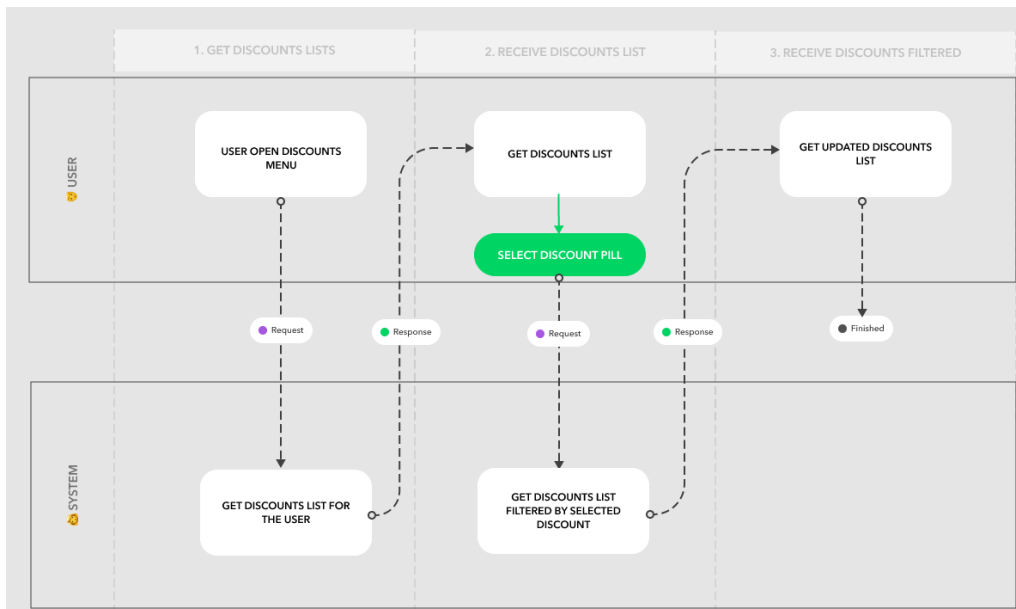


Figura 10.92: Flujo filtered discounts.

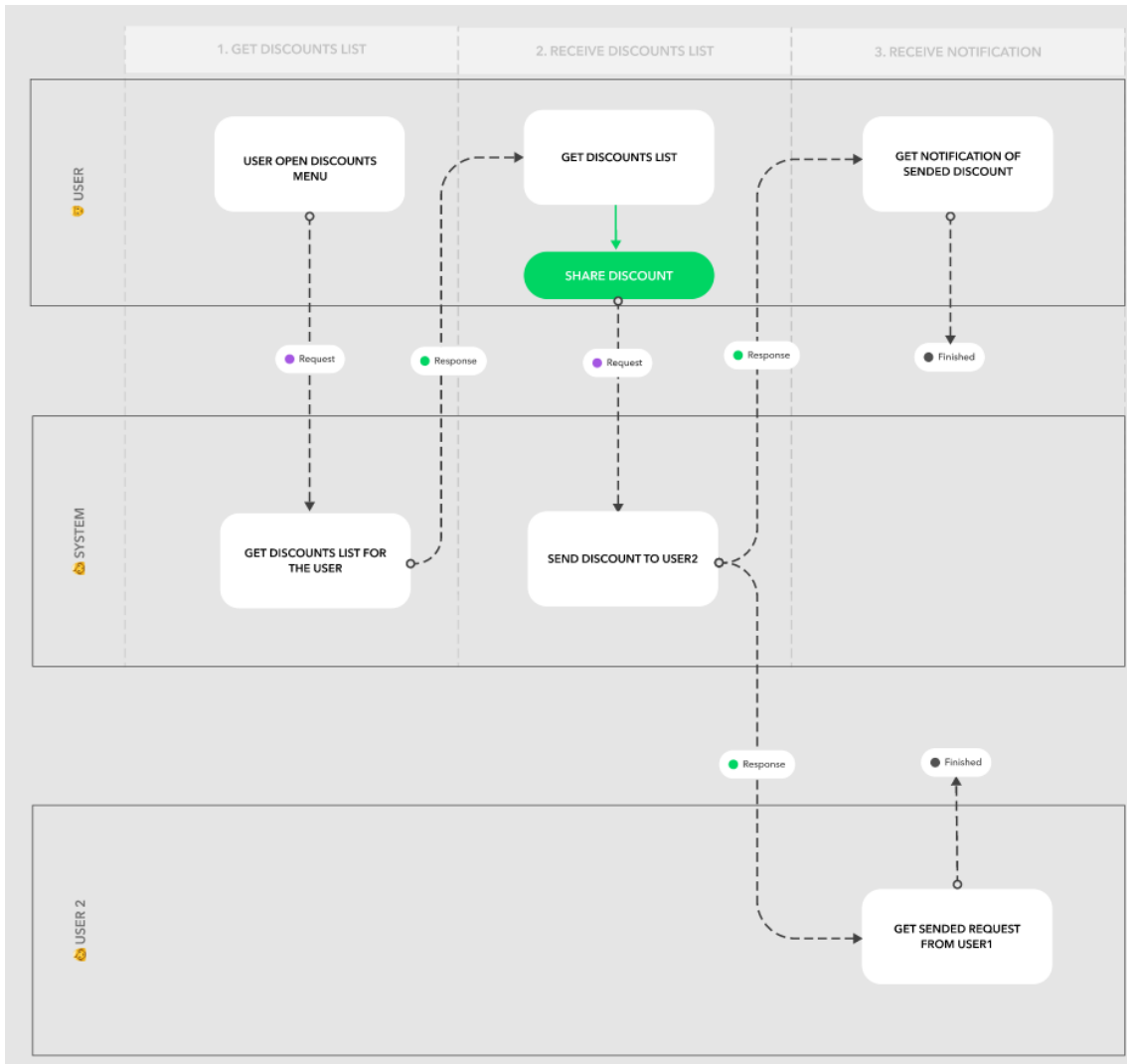


Figura 10.93: Flujo share discount.

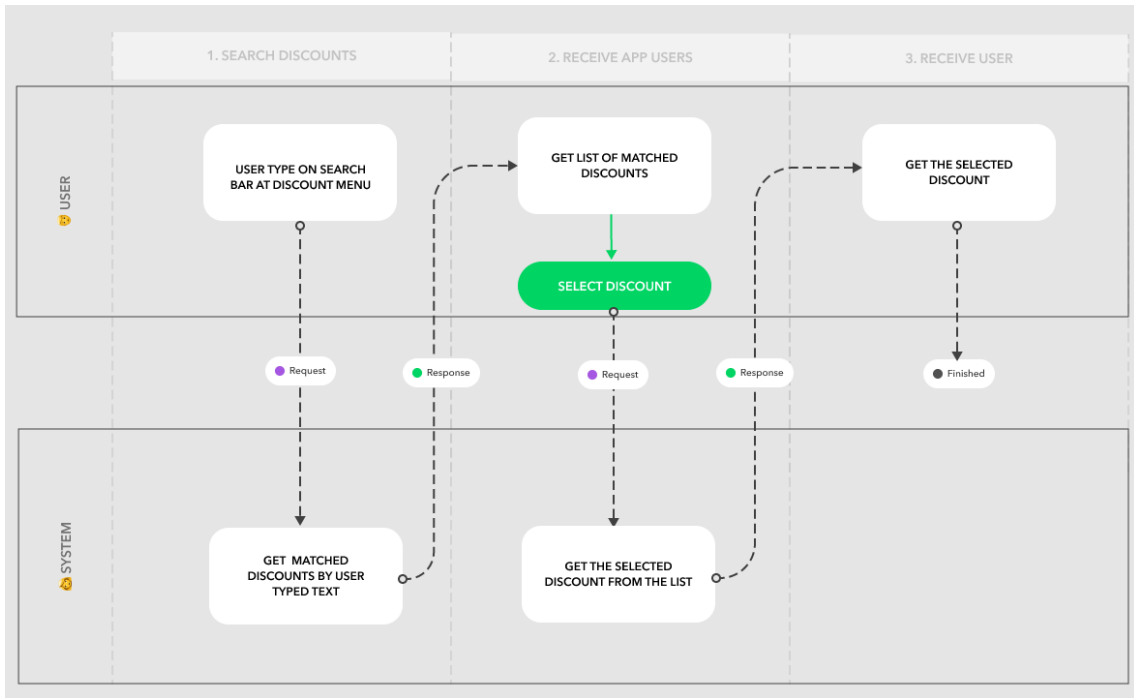


Figura 10.94: Flujo filtered typed discounts.

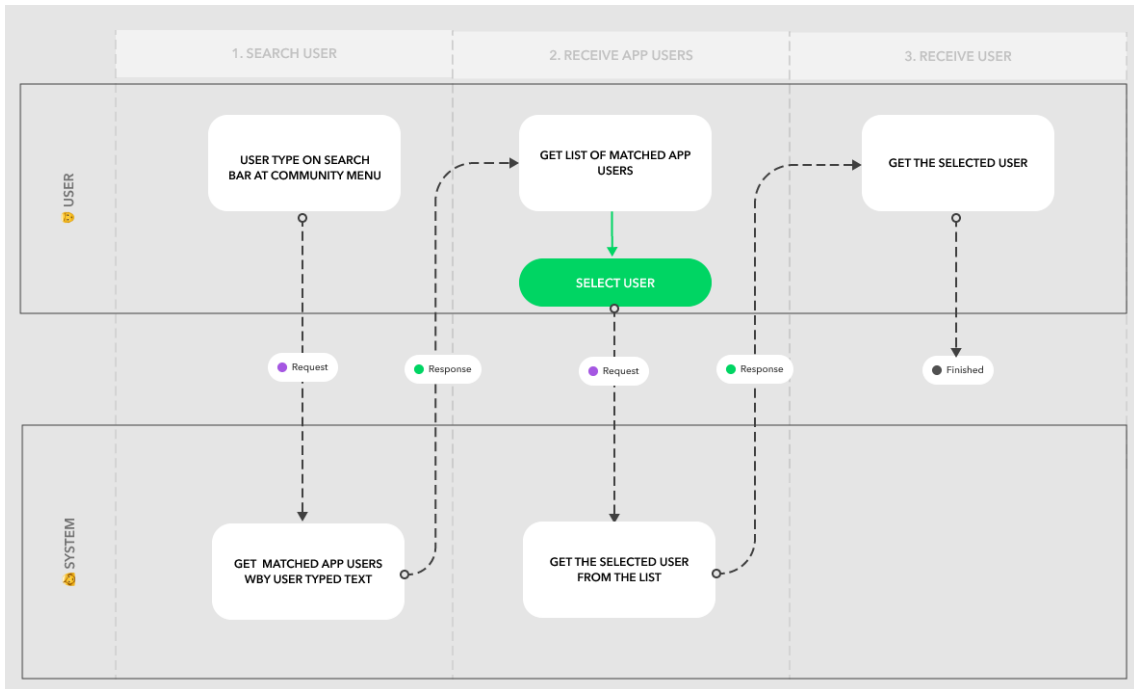


Figura 10.95: Flujo user search.

10.7. Anexo 7 - Gestión de Riesgos

10.7.1. Escala de Riesgos para Amenazas.

		IMPACTO →		
		1	2	3
PROBABILIDAD ↓	1	ASUMIR - 1 -	MONITOREAR - 2 -	PREVENIR - 3 -
	2	MONITOREAR - 2 -	PREVENIR - 4 -	MITIGAR - 6 -
	3	PREVENIR - 3 -	MITIGAR - 6 -	MITIGAR - 9 -

Figura 10.96: Acciones a tomar ante riesgos.

10.7.2. Escala de Riesgos para Oportunidades.

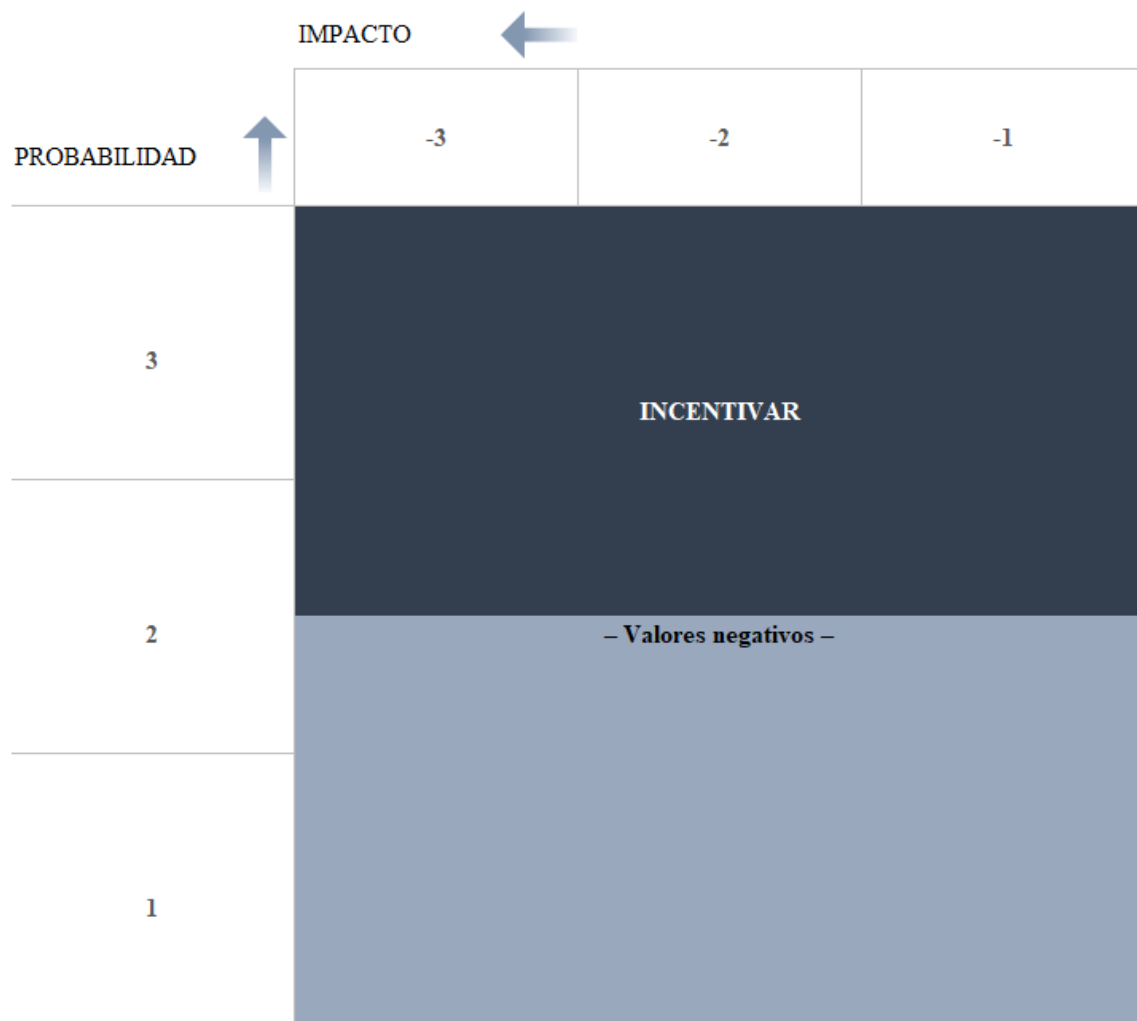


Figura 10.97: Acciones a tomar ante oportunidades.

10.7.3. Riesgos evaluados en *Sprint 0*.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
E001	17/3/2021	1	3	3	Equipo	Equipo	Prevenir	En caso de ocurrir se puede ajustar la carga de trabajo para el sprint en que se encuentre para subsanar la situación.		
H001	18/3/2021	1	3	3	Humano	Equipo	Prevenir	Cumplir con lo estipulado en Scrum, realizar dailies para mantener visibilidad de lo pactado.		
H002	18/3/2021	1	3	3	Humano	Equipo	Prevenir	Otro integrante puede tomar el rol.		
H003	18/3/2021	1	3	3	Humano	Equipo	Prevenir	Resolver entre el equipo, ya sea en alguna de las dailies o en sprint retrospective.		
G003	19/3/2021	1	1	1	Gestión	Equipo	Asumir			
P002	25/3/2021	2	3	6	Producto	Equipo	Mitigar	Utilización de Scrum para lograr avance continuo.		
G004	30/3/2021	1	2	2	Gestión	Equipo	Monitorear		Se intentó avanzar por otras líneas hasta contar con el acceso.	
G005	30/3/2021	1	2	2	Gestión	Equipo	Monitorear	Intentar recibir feedback seguido de forma que el tutor pueda detectar posibles elementos que no estén de acuerdo a la exigencia de correctores.		

Figura 10.98: Riesgos en Sprint 0.

10.7.4. Riesgos evaluados en *Sprint 1*.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
T002	5/4/2021	2	3	6	Tecnología	Equipo	Mitigar	Reunión con experto de negocio para discutir alternativas.	Podría solicitarse préstamo de equipos a Globant, comprar por cuenta de los integrantes.	
T006	5/4/2021	2	3	6	Gestión	Equipo	Mitigar	Acordar reuniones semanales fijas con el experto de negocio.		
G001	14/4/2021	1	3	3	Gestión	Equipo	Prevenir	Contactar con UX-UI designer para obtener una manera de romper épicas de manera más ágil.		
G002	14/4/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
T001	16/4/2021	1	3	3	Tecnología	Experto de negocio	Prevenir	Contactar a experto de librería.	Al evacuar dudas con el experto de la librería se aclaró que el riesgo de no poder integrar es menor al inicialmente creído, ya que React Native cuenta con opciones para integrar librerías nativas.	
G004	16/4/2021	1	0	0	Gestión	Equipo	Mitigado		Se obtuvo acceso a la librería.	
G006	18/4/2021	1	0	0	Gestión	Equipo	Mitigado			
T002	23/4/2021	1	2	2	Tecnología	Equipo	Monitorear	En principio no se deberán adquirir equipos pero se mantiene en monitoreo.	Luego de una reunión con el experto de negocio y un experto de la librería; se discutieron alternativas de integración con Swift y React Native, lo cual evitaría tener que programar en Swift (y necesitar equipamiento Apple).	

Figura 10.99: Riesgos en Sprint 1.

10.7.5. Riesgos evaluados en *Sprint 2*.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
T003	8/5/2021	3	2	6	Tecnología	Equipo	Mitigar	Complementar con cursos de React Native; delegar la mayor parte de desarrollo a Pablo Pereira ya que tiene mayor expertise en desarrollo de aplicaciones web.		
T001	8/5/2021	1	0	0	Tecnología	Equipo	Mitigado			
G001	8/5/2021	1	0	0	Gestión	Equipo	Mitigado			
G002	8/5/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
H001	8/5/2021	1	2	2	Humano	Equipo	Monitorear			
H002	10/5/2021	1	0	0	Humano	Equipo	Mitigado			
H003	10/5/2021	1	2	2	Humano	Equipo	Monitorear			
P002	10/5/2021	2	3	6	Producto	Equipo	Mitigar	Se continúa trabajando en mejorar la aplicación de Scrum		
E001	10/5/2021	1	2	2	Equipo	Equipo	Monitorear			
G003	10/5/2021	1	2	2	Gestión	Equipo	Monitorear			
G005	10/5/2021	1	2	2	Gestión	Equipo	Monitorear			
T003	10/5/2021	1	3	3	Tecnología	Equipo	Prevenir			

Figura 10.100: Riesgos en Sprint 2.

10.7.6. Riesgos evaluados en *Sprint 3*, parte I.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
T006	20/5/2021	3	3	9	Tecnología	Equipo	Mitigar	Reducir dependencia de herramientas prescindibles de React Native (como Expo).		
E001	27/5/2021	1	2	2	Tecnología	Equipo	Monitorear			
G001	28/5/2021	0	1	0	Gestión	Equipo	Mitigado			
G002	28/5/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
G004	29/5/2021	0	0	0	Gestión	Equipo	Mitigado			
G005	29/5/2021	1	2	2	Gestión	Equipo	Monitorear			
G006	29/5/2021	0	1	0	Gestión	Equipo	Mitigado			
H001	29/5/2021	1	2	2	Humano	Equipo	Monitorear	Mantener ceremonias de Scrum y continuar en contacto frecuente por medios más informales (como Whatsapp) para alineamiento interno del equipo.		
H002	29/5/2021	1	2	2	Humano	Equipo	Monitorear			
H003	29/5/2021	1	2	2	Humano	Equipo	Monitorear	Mantener contacto frecuente, facilitar espacios de tiempo para discutir problemas que puedan surgir, respetar las ceremonias de Retrospective (Scrum).		
P001	29/5/2021	1	2	2	Producto	Equipo	Monitorear	Aún no se comenzó a almacenar datos; ya se cuenta con una User Story para desarrollar este aspecto.		
P002	29/5/2021	1	2	2	Producto	Equipo	Monitorear	Continuar trabajando, recibiendo feedback del cliente, del tutor, y próximamente (Lunes 31 de Mayo) feedback en la reunión con el revisor de ORT.		
T001	29/5/2021	0	1	0	Tecnología	Equipo	Mitigado			
T002	29/5/2021	1	2	2	Tecnología	Equipo	Monitorear	Se mantiene el monitoreo ya que aún no se logró la integración total entre front-end, back-end y librería.		
T003	29/5/2021	1	2	2	Tecnología	Equipo	Monitorear	Se continúa en capacitación en la medida en que es posible para poder adelantarse a desafíos técnicos que puedan surgir.		
T004	29/5/2021	0	1	0	Tecnología	Equipo	Mitigado	Se optó por resolver esta funcionalidad en el back-end, lo que permite reducir dependencia de librerías de terceros.		
T006	29/5/2021	0	1	0	Tecnología	Equipo	Mitigado	Se completó la migración a React Native CLI, lo cual permite mayor flexibilidad y reduce la dependencia de Expo.	A modo de ejemplo, algunas librerías utilizadas no son compatibles con Expo, lo cual agrega un conjunto de problemas que deben ser resueltos frecuentemente. Al no depender de Expo, se puede tener mayor libertad y reducir la cantidad de retrabajo en diferentes situaciones (como el agregado de librerías de terceros).	
P003	29/5/2021	1	-1	-1	Producto	Equipo	Incentivar	Asistir a instancia de presentación del proyecto que pueda generarse por parte de Globant para realizar difusión del producto.		
T007	30/5/2021	2	3	6	Tecnología	Equipo	Mitigar	Comenzar a testear en sprints tempranos para evitar mayor cantidad de dificultades sobre el fin del proyecto.		
G007	30/5/2021	3	3	9	Gestión	Equipo	Mitigar			Si

Figura 10.101: Riesgos en *Sprint 3*, parte I.

10.7.7. Riesgos evaluados en *Sprint 3*, parte II.

T002	29/5/2021	1	2	2	Tecnología	Equipo	Monitorear	Se mantiene el monitoreo ya que aún no se logró la integración total entre front-end, back-end y librería.		
T003	29/5/2021	1	2	2	Tecnología	Equipo	Monitorear	Se continúa en capacitación en la medida en que es posible para poder adelantarse a desafíos técnicos que puedan surgir.		
T004	29/5/2021	0	1	0	Tecnología	Equipo	Mitigado	Se optó por resolver esta funcionalidad en el back-end, lo que permite reducir dependencia de librerías de terceros.		
T006	29/5/2021	0	1	0	Tecnología	Equipo	Mitigado	Se completó la migración a React Native CLI, lo cual permite mayor flexibilidad y reduce la dependencia de Expo.	A modo de ejemplo, algunas librerías utilizadas no son compatibles con Expo, lo cual agrega un conjunto de problemas que deben ser resueltos frecuentemente. Al no depender de Expo, se puede tener mayor libertad y reducir la cantidad de retrabajo en diferentes situaciones (como el agregado de librerías de terceros).	
P003	29/5/2021	1	-1	-1	Producto	Equipo	Incentivar	Asistir a instancia de presentación del proyecto que pueda generarse por parte de Globant para realizar difusión del producto.		
T007	30/5/2021	2	3	6	Tecnología	Equipo	Mitigar	Comenzar a testear en sprints tempranos para evitar mayor cantidad de dificultades sobre el fin del proyecto.		
G007	30/5/2021	3	3	9	Gestión	Equipo	Mitigar			Sí

Figura 10.102: Riesgos en Sprint 3, parte II.

10.7.8. Riesgos evaluados en *Sprint 4*.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
G002	11/6/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
G008	11/6/2021	1	2	2	Gestión	Equipo	Monitorear	Se cuentan con herramientas y metodologías adecuadas. Se debe monitorear para que el riesgo no se materialice.		
T009	11/6/2021	1	3	3	Tecnología	Equipo	Prevenir			Si

Figura 10.103: Riesgos en Sprint 4.

10.7.9. Riesgos evaluados en *Sprint 5*.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
H001	14/6/2021	1	2	2	Humano	Equipo	Monitorear			
H002	14/6/2021	1	2	2	Humano	Equipo	Monitorear			
H003	14/6/2021	1	3	3	Humano	Equipo	Prevenir	Segregar mejor las tareas del sprint para reducir dependencias innecesarias/hacer mejor aprovechamiento del tiempo.		
P001	14/6/2021	1	0	0	Producto	Equipo	Mitigado	Ya se cuenta con la arquitectura de datos así como también back-end. Se han realizado tests en forma temprana los cuales resultaron satisfactorios.		
P002	14/6/2021	1	3	3	Producto	Equipo	Prevenir	Redoblar esfuerzos en front-end que a la fecha no se encuentra tan avanzado como el back-end.		
T002	14/6/2021	1	0	0	Tecnología	Equipo	Mitigado			
T003	14/6/2021	1	3	3	Tecnología	Equipo	Prevenir	Continuar capacitación en paralelo a desarrollo para reducir retrasos.		
G002	14/6/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
G008	14/6/2021	1	2	2	Gestión	Equipo	Monitorear			
T005	14/6/2021	3	3	9	Tecnología	Equipo/Cliente	Mitigar	Se incluye al Cliente para plantear dificultades encontradas y proveer solución antes de que derive en un riesgo materializado.		
T008	14/6/2021	1	0	0	Tecnología	Equipo	Mitigado	La migración de librerías fue mínima en dificultades y ya se cuenta con una alternativa robusta para persistencia en memoria de los datos obtenidos de la API.		
T007	14/6/2021	1	0	0	Tecnología	Equipo	Mitigado	Ya se realizó el despliegue (en Firebase) a dispositivos de todos los integrantes del equipo.		
P003	14/6/2021	1	-1	-1	Producto	Experto de negocio	Incentivar			
T010	14/6/2021	1	2	2	Tecnología	Equipo	Monitorear			
P004	14/6/2021	3	-3	-9	Producto	Equipo	Incentivar	Cultivar relación con el cliente y proponer nuevos escenarios para expandir áreas de interés.		

Figura 10.104: Riesgos en Sprint 5.

10.7.10. Riesgos evaluados en *Sprint 6*.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
G002	28/6/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
T005	28/6/2021	2	3	6	Tecnología	Equipo/Cliente	Mitigar	Se incluye al Cliente para plantear dificultades encontradas y proveer solución antes de que derive en un riesgo materializado.		
T010	28/6/2021	1	2	2	Tecnología	Equipo	Monitorear			
T011	28/6/2021	1	3	3	Tecnología	Equipo	Prevenir	Ver plan de mitigación		
G009	28/6/2021	1	2	2	Gestión	Equipo	Monitorear			
G010	28/6/2021	1	3	3	Gestión	Equipo/Cliente	Prevenir	Buscar alternativas para llegar a la información necesaria. Discutirlo con otros <i>stakeholders</i>		
P004	28/6/2021	3	-3	-9	Producto	Equipo	Incentivar	Cultivar relación con el cliente y proponer nuevos escenarios para expandir áreas de interés.		

Figura 10.105: Riesgos en Sprint 6.

10.7.11. Riesgos evaluados en *Sprint 7*.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
G002	12/7/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
H004	12/7/2021	2	2	4	Tecnología	Equipo	Prevenir	Reforzar y resaltar importancia de consistencia en el código	Discordancia en estándares de codificación	Si
T005	12/7/2021	2	3	6	Tecnología	Equipo/Ciente	Mitigar	Se incluye al Cliente para plantear dificultades encontradas y proveer solución antes de que derive en un riesgo materializado.		
T010	12/7/2021	2	2	4	Tecnología	Equipo	Prevenir	En caso de no obtener licenciamiento por parte del Cliente, se procede a asumir los costos por parte del equipo.		Si
T011	12/7/2021	1	3	3	Tecnología	Equipo	Prevenir	Consultar con experto de negocio		
T012	12/7/2021	1	3	3	Tecnología	Equipo	Prevenir			
T013	12/7/2021	1	3	3	Tecnología	Equipo	Prevenir	Realizar cambios en back-end para que no se retrase desarrollo en front-end		
T014	12/7/2021	3	3	9	Tecnología	Equipo	Mitigar	Ver plan de mitigación		
G009	12/7/2021	1	2	2	Gestión	Equipo	Monitorear			
G010	12/7/2021	1	3	3	Gestión	Equipo/Ciente	Prevenir	Buscar alternativas para llegar a la información necesaria. Discutirlo con otros <i>stakeholders</i>		
P004	12/7/2021	3	-3	-9	Producto	Equipo	Incentivar	Cultivar relación con el cliente y proponer nuevos escenarios para expandir áreas de interés.		

Figura 10.106: Riesgos en *Sprint 7*.

10.7.12. Riesgos evaluados en *Sprint 8*.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
T010	26/7/2021	2	2	4	Tecnología	Equipo	Prevenir	En caso de no obtener licenciamiento por parte del Cliente, se procede a asumir los costos por parte del equipo.		Si
T011	26/7/2021	1	3	3	Tecnología	Equipo	Prevenir	Consultar con experto de negocio		
T012	26/7/2021	1	3	3	Tecnología	Equipo	Prevenir			
T013	26/7/2021	1	3	3	Tecnología	Equipo	Prevenir	Realizar cambios en back-end para que no se retrase desarrollo en front-end		
T014	26/7/2021	3	3	9	Tecnología	Equipo	Mitigar	Ver plan de mitigación		
G002	26/7/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
E002	26/7/2021	2	1	2	Equipo	Scrum Master	Monitorear	Por el momento no afectaría la carga de trabajo, se reduce la carga para el sprint del miembro del equipo y se negocia el alcance en dos sprints en caso de necesitarse.		
H004	26/7/2021	2	2	4	Tecnología	Equipo	Prevenir	Reforzar y resaltar importancia de consistencia en el código	Discordancia en estándares de codificación	
G009	26/7/2021	1	2	2	Gestión	Equipo	Monitorear			
G010	26/7/2021	1	3	3	Gestión	Equipo/Cliente	Prevenir	Buscar alternativas para llegar a la información necesaria. Discútilo con otros <i>stakeholders</i>		
P004	26/7/2021	3	-3	-9	Producto	Equipo	Incentivar	Cultivar relación con el cliente y proponer nuevos escenarios para expandir áreas de interés.		

Figura 10.107: Riesgos en Sprint 8.

10.7.13. Riesgos evaluados en *Sprint 9*.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
G002	5/8/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
H004	5/8/2021	1	2	2	Tecnología	Equipo	Monitorear	Reforzar y resaltar importancia de consistencia en el código	Discordancia en estándares de codificación	
G009	5/8/2021	1	2	2	Gestión	Equipo	Monitorear			
G010	5/8/2021	1	3	3	Gestión	Equipo/Cliente	Prevenir	Buscar alternativas para llegar a la información necesaria. Discutirlo con otros <i>stakeholders</i>		
T011	5/8/2021	1	3	3	Tecnología	Equipo	Prevenir	Consultar con experto de negocio		
T012	5/8/2021	1	3	3	Tecnología	Equipo	Prevenir			
T013	5/8/2021	1	3	3	Tecnología	Equipo	Prevenir	Realizar cambios en back-end para que no se retrase desarrollo en front-end		
T014	5/8/2021	3	3	9	Tecnología	Equipo	Mitigar	Ver plan de mitigación		
P004	5/8/2021	3	-3	-9	Producto	Equipo	Incentivar	Cultivar relación con el cliente y proponer nuevos escenarios para expandir áreas de interés.		Si

Figura 10.108: Riesgos en Sprint 9.

10.7.14. Riesgos evaluados en *Sprint* 10.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
G002	25/8/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
H004	25/8/2021	1	2	2	Tecnología	Equipo	Monitorear	Reforzar y resaltar importancia de consistencia en el código	Discordancia en estándares de codificación	
G009	25/8/2021	1	2	2	Gestión	Equipo	Monitorear			
G010	25/8/2021	1	3	3	Gestión	Equipo/Cliente	Prevenir	Buscar alternativas para llegar a la información necesaria. Discutirlo con otros <i>stakeholders</i>		
T011	25/8/2021	1	3	3	Tecnología	Equipo	Prevenir	Consultar con experto de negocio		
T012	25/8/2021	1	3	3	Tecnología	Equipo	Prevenir			
T013	25/8/2021	1	3	3	Tecnología	Equipo	Prevenir	Realizar cambios en back-end para que no se retrase desarrollo en front-end		
T014	25/8/2021	3	3	9	Tecnología	Equipo	Mitigar	Ver plan de mitigación		
P004	25/8/2021	3	-3	-9	Producto	Equipo	Incentivar	Cultivar relación con el cliente y proponer nuevos escenarios para expandir áreas de interés.		Si

Figura 10.109: Riesgos en *Sprint* 10.

10.7.15. Riesgos evaluados en *Sprint* 11.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
T011	10/9/2021	1	3	3	Tecnología	Equipo	Prevenir	Consultar con experto de negocio		
T012	10/9/2021	1	3	3	Tecnología	Equipo	Prevenir			
T013	10/9/2021	1	3	3	Tecnología	Equipo	Prevenir	Realizar cambios en back-end para que no se retrase desarrollo en front-end		
G002	10/9/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
H004	10/9/2021	1	2	2	Tecnología	Equipo	Monitorear	Reforzar y resaltar importancia de consistencia en el código	Discordancia en estándares de codificación	
G009	10/9/2021	1	2	2	Gestión	Equipo	Monitorear			
G010	10/9/2021	1	3	3	Gestión	Equipo/Cliente	Prevenir	Buscar alternativas para llegar a la información necesaria. Discutirlo con otros <i>stakeholders</i>		
P004	10/9/2021	3	-3	-9	Producto	Equipo	Incentivar	Cultivar relación con el cliente y proponer nuevos escenarios para expandir áreas de interés.		Si

Figura 10.110: Riesgos en *Sprint* 11.

10.7.16. Riesgos evaluados en *Sprint* 12.

ID	Fecha	P	I	M	Tipo	Responsable	Acción a tomar	Plan de acción	Observaciones	Materializado
T011	22/9/2021	0	3	0	Tecnología	Equipo	Mitigado	Consultar con experto de negocio		
G002	22/9/2021	1	2	2	Gestión	Equipo	Monitorear	Aumentar la carga horaria o replanificar el próximo sprint		
G009	22/9/2021	1	2	2	Gestión	Equipo	Monitorear			
G010	22/9/2021	1	3	3	Gestión	Equipo/Ciente	Prevenir	Buscar alternativas para llegar a la información necesaria. Discutirlo con otros <i>stakeholders</i>		
P004	22/9/2021	3	-3	-9	Producto	Equipo	Incentivar	Cultivar relación con el cliente y proponer nuevos escenarios para expandir áreas de interés.		Si

Figura 10.111: Riesgos en *Sprint* 12.

10.7.17. Plan de mitigación de riesgos.

ID	Descripción	Plan de acción	Responsable
P002	No lograr un MVP del producto.	Aplicar Scrum, iterar en cada Sprint mejorando las prácticas propuestas en el framework.	Producto
T003	Falta de conocimiento en desarrollo de front-end y de React Native.	Complementar con cursos de React Native; delegar la mayor parte de desarrollo a Pablo Pereira ya que tiene mayor expertise en desarrollo de aplicaciones web.	Tecnología
T005	Se incluye al Cliente para plantear dificultades encontradas y proveer solución antes de que derive en un riesgo materializado.	Se propone hacer un cambio de tecnología desde LoopBack hacia PHP/Laravel, lo cual tiene por cometido acelerar el desarrollo y poder tener mayor control sobre el back-end. Facilitar la aplicación de Open-Closed Principle.	Tecnología
T006	Reducir dependencia de herramientas prescindibles de React Native (como Expo).	Migrar herramientas/librerías de building desde Expo a una alternativa nativa de React Native.	Tecnología
G006	Falta de monitoreo por parte de experto del negocio.	Mantener reuniones por lo menos semanalmente para estar alineados entre cliente y equipo.	Gestión
G007	Retrabajo por presentación de avances para revisores de ORT	Aumentar la carga horaria en el siguiente sprint para contrarrestar el impacto que tuvo dedicar horas a preparación de documentación y presentación ante revisores de ORT.	Gestión
T007	Despliegue de la aplicación en dispositivos físicos.	Comenzar a testear en sprints tempranos para evitar mayor cantidad de dificultades sobre el fin del proyecto.	Tecnología
T009	Migración en librerías utilizadas para React Redux. Existen diferentes alternativas y se debe definir el uso de librerías auxiliares como Thunk, así como también la arquitectura de Redux utilizada.	Estudiar alternativas con el equipo y hacer uso de los cursos de React Native y documentación disponible en la web.	Tecnología
T014	Pruebas con beacons en escenario real.	Para descartar dificultades adicionales, se cuenta con back-end portado a la nube de forma de realizar una prueba lo más independiente posible. Testear con beacons en ambiente interno con aplicación móvil. En caso de no contar con comportamiento y precisión esperados realizar pruebas en etapa temprana para ajustar esos parámetros.	Tecnología

Figura 10.112: Plan de mitigación de riesgos.

10.7.18. Distribución de riesgos por sprint.

ID	Descripción	Plan de acción	Responsable
P002	No lograr un MVP del producto.	Aplicar Scrum, iterar en cada Sprint mejorando las prácticas propuestas en el framework.	Producto
T003	Falta de conocimiento en desarrollo de front-end y de React Native.	Complementar con cursos de React Native; delegar la mayor parte de desarrollo a Pablo Pereira ya que tiene mayor expertise en desarrollo de aplicaciones web.	Tecnología
T005	Se incluye al Cliente para plantear dificultades encontradas y proveer solución antes de que derive en un riesgo materializado.	Se propone hacer un cambio de tecnología desde LoopBack hacia PHP/Laravel, lo cual tiene por cometido acelerar el desarrollo y poder tener mayor control sobre el back-end. Facilitar la aplicación de Open-Closed Principle.	Tecnología
T006	Reducir dependencia de herramientas prescindibles de React Native (como Expo).	Migrar herramientas/librerías de building desde Expo a una alternativa nativa de React Native.	Tecnología
G006	Falta de monitoreo por parte de experto del negocio.	Mantener reuniones por lo menos semanalmente para estar alineados entre cliente y equipo.	Gestión
G007	Retrabajo por presentación de avances para revisores de ORT	Aumentar la carga horaria en el siguiente sprint para contrarrestar el impacto que tuvo dedicar horas a preparación de documentación y presentación ante revisores de ORT.	Gestión
T007	Despliegue de la aplicación en dispositivos físicos.	Comenzar a testear en sprints tempranos para evitar mayor cantidad de dificultades sobre el fin del proyecto.	Tecnología
T009	Migración en librerías utilizadas para React Redux. Existen diferentes alternativas y se debe definir el uso de librerías auxiliares como Thunk, así como también la arquitectura de Redux utilizada.	Estudiar alternativas con el equipo y hacer uso de los cursos de React Native y documentación disponible en la web.	Tecnología
T014	Pruebas con beacons en escenario real.	Para descartar dificultades adicionales, se cuenta con back-end portado a la nube de forma de realizar una prueba lo más independiente posible. Testear con beacons en ambiente interno con aplicación móvil. En caso de no contar con comportamiento y precisión esperados realizar pruebas en etapa temprana para ajustar esos parámetros.	Tecnología

Figura 10.113: Plan de mitigación de riesgos.

10.7.19. Magnitud por sprint y tipo de riesgos.

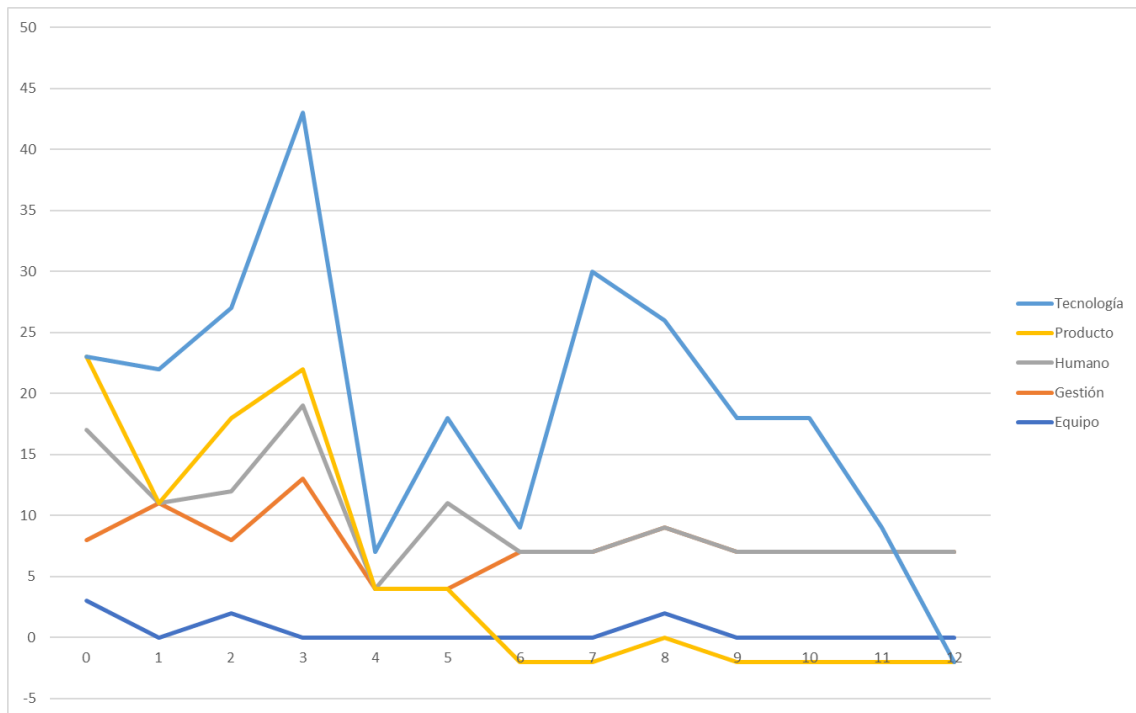


Figura 10.114: Magnitud por sprint y tipo de riesgos.

10.7.20. Magnitud por sprint y tipo de riesgos.

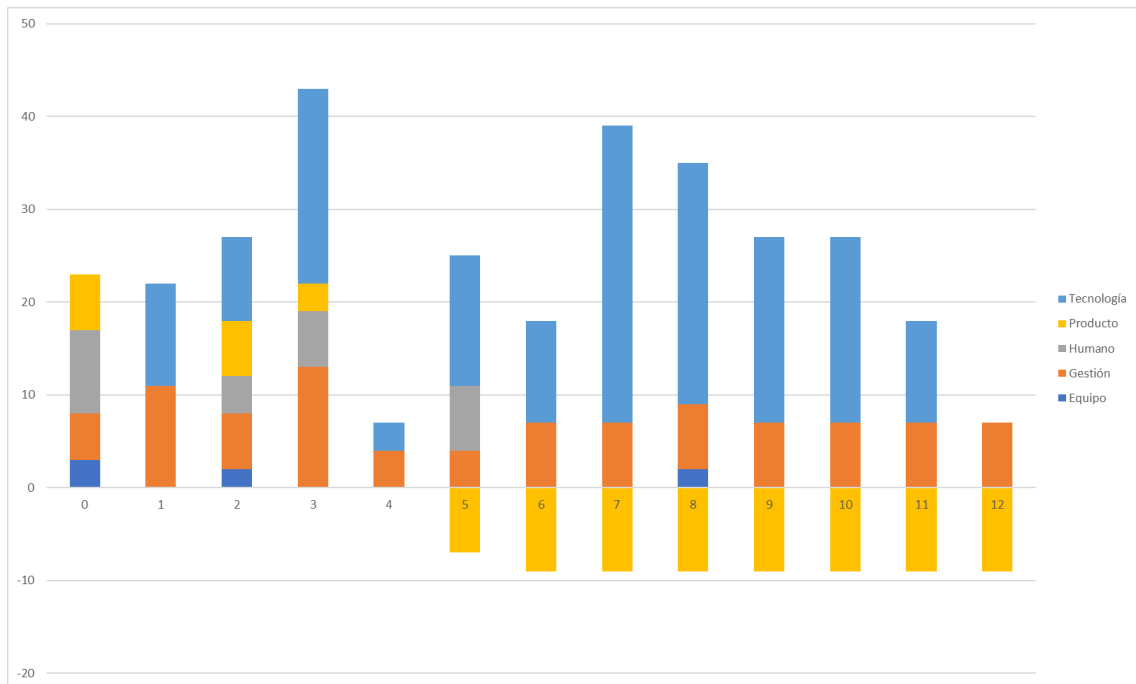


Figura 10.115: Magnitud por sprint y tipo de riesgos.

10.7.21. Mayor magnitud acumulada de tipo de riesgos en sprints.

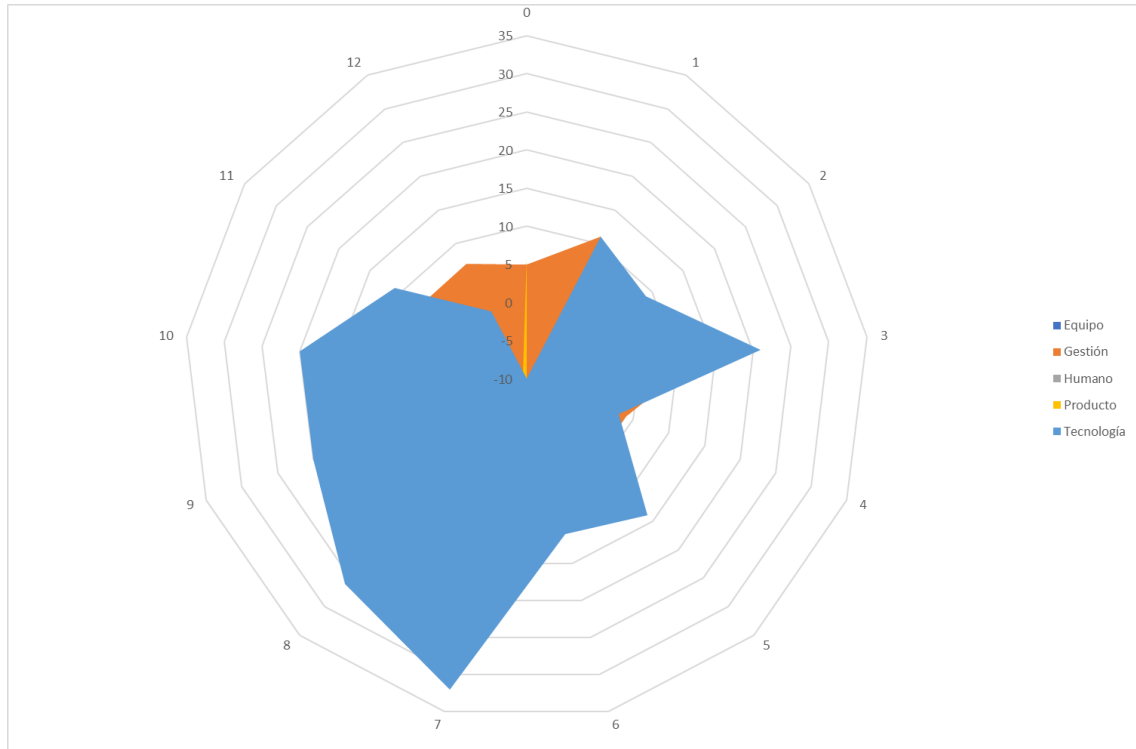


Figura 10.116: Mayor magnitud acumulada de tipo de riesgos en sprints.

10.8. Anexo 8 - Gestión de Calidad

10.8.1. Cobertura de pruebas automatizadas en SonarQube - *Frontend*

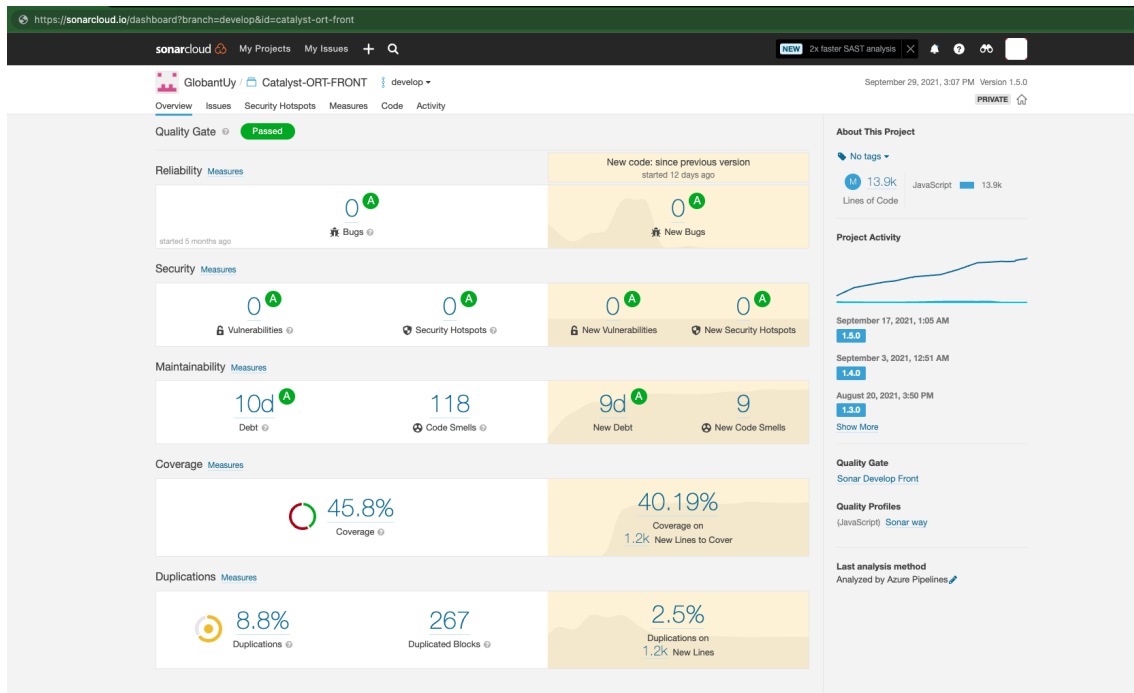


Figura 10.117: Dashboard de SonarQube - *Frontend*.

10.8.2. Reglas de calidad definidas para pruebas de SonarQube - *Frontend*

The screenshot shows the SonarQube interface for configuring quality gates. The left sidebar lists quality gates: 'Sonar default', 'Sonar Develop Backend' (marked as DEFAULT), 'Sonar Develop Front' (selected), and 'Sonar way' (marked as BUILT-IN). The main area displays the configuration for 'Sonar Develop Front'. It includes a 'Create' button and action buttons: 'Rename', 'Copy', 'Set as Default', and 'Delete'. The configuration is divided into two sections: 'Conditions on New Code' and 'Conditions on Overall Code'. Each section contains a table of metrics with their operators and values, along with 'Edit' and 'Delete' icons.

Conditions on New Code
Conditions on New Code apply to all branches and to Pull Requests.

Metric	Operator	Value	Edit	Delete
Maintainability Rating	is worse than	A		
Reliability Rating	is worse than	A		
Security Rating	is worse than	A		

Conditions on Overall Code
Conditions on Overall Code apply to long-lived branches only.

Metric	Operator	Value	Edit	Delete
Coverage	is less than	41.0%		
Duplicated Lines (%)	is greater than	12.0%		
Security Hotspots Reviewed	is less than	100%		
Security Review Rating	is worse than	A		

Figura 10.118: *Gate rules* en SonarQube - *Frontend*.

10.8.3. Cobertura de pruebas automatizadas en SonarQube - Backend

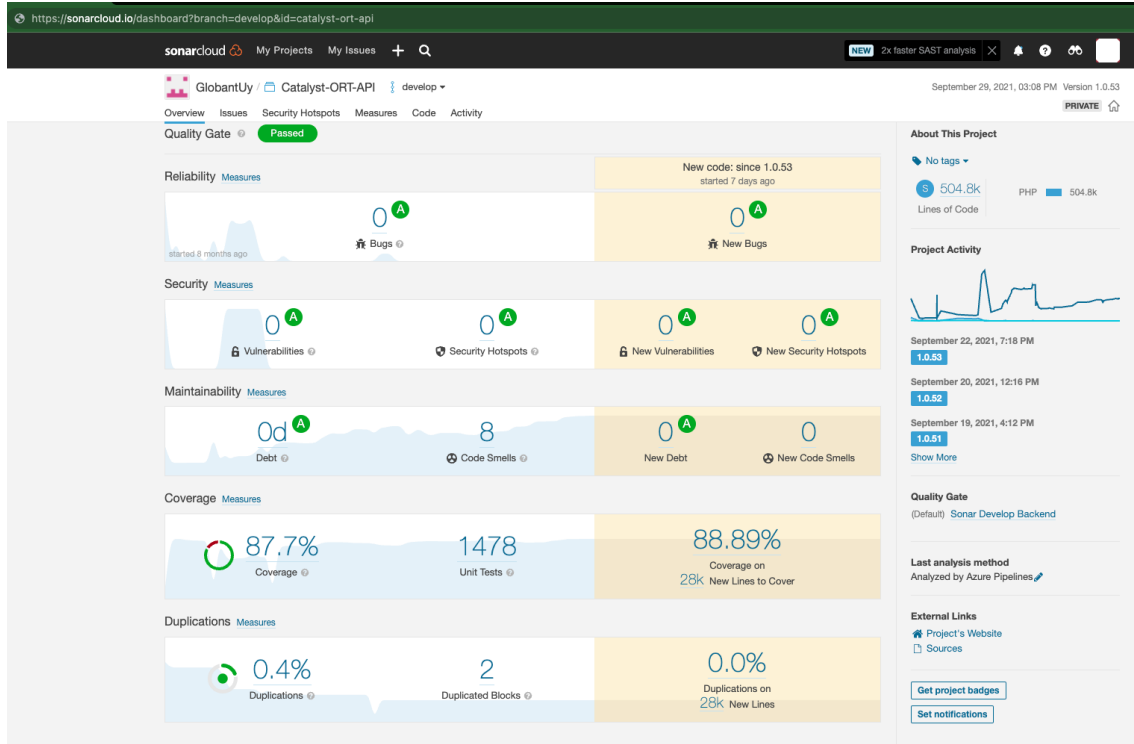


Figura 10.119: Dashboard de SonarQube - Backend.

10.8.4. Reglas de calidad definidas para pruebas de SonarQube - *Backend*

The screenshot shows the SonarQube interface for configuring quality gates. The page title is 'Sonar Develop Backend'. On the left, a sidebar lists quality gates: 'Sonar default', 'Sonar Develop Backend' (marked as 'DEFAULT'), 'Sonar Develop Front', and 'Sonar way' (marked as 'BUILT-IN'). The main content area is titled 'Conditions' and includes an 'Add Condition' button. It is divided into two sections: 'Conditions on New Code' and 'Conditions on Overall Code'. Each section contains a table of metrics with their respective operators and values, along with edit and delete icons.

Metric	Operator	Value	Edit	Delete
Coverage	is less than	80.0%		
Duplicated Lines (%)	is greater than	6.0%		
Maintainability Rating	is worse than	A		
Reliability Rating	is worse than	A		
Security Hotspots Reviewed	is less than	100%		
Security Rating	is worse than	A		

Metric	Operator	Value	Edit	Delete
Coverage	is less than	80.0%		
Duplicated Lines (%)	is greater than	6.0%		
Maintainability Rating	is worse than	A		
Reliability Rating	is worse than	A		
Security Hotspots Reviewed	is less than	100%		
Security Rating	is worse than	A		

Figura 10.120: *Gate rules* en SonarQube - *Backend*.

10.9. Anexo 9 - Información Globant

10.9.1. Acerca de Globant

Empresa líder en el sector de desarrollo de *software* e innovación a nivel mundial. Fundada en 2003 por cuatro amigos que querían crear una multinacional en Argentina, inspirados en grandes compañías localizadas en India, decidieron crear una compañía que genera y añade valor en América Latina, construyendo las bases de la innovación y el desarrollo del *software*.

A lo largo de los años Globant ha ido creciendo a paso constante, generando valor para sus clientes, a través de la tecnología y de la diferenciación de sus servicios

Estos son algunos de los clientes que manejan, tienen una cartera muy extensa, a través de la base innovadora y el desarrollo de software logran trabajar sobre cualquier rubro en diferentes locaciones geográficas.

Su misión es convertirse en la empresa líder de desarrollo de *software* en el mundo.



Figura 10.121: Clientes de Globant evidenciando industrias.

10.9.2. Ubicación geográfica

Cuenta con más de 5000 empleados localizados en las oficinas de Argentina, Brasil, Estados Unidos, Colombia, España, México, Uruguay, India, Chile, Reino Unido, Belarús y Rumania.



Figura 10.122: Ubicación de oficinas en el mundo.

10.9.3. Estructura interna

Mediante la separación de unidades operativas llamadas Studios que fomentan la creatividad y innovación al mismo tiempo, le permite desarrollar, mejorar y consolidar la experiencia en torno a una variedad de tecnologías emergentes. Cada uno de los Studios tiene conocimientos de dominio específicos y ofrece soluciones personalizadas en desafíos tecnológicos específicos.

Estos Studios se separan dentro de estas 5 categorías:

- Sustainable Business Strategy:
 - Business hacking.
 - Digital sales.
- Culture & Agile:
 - Cultural hacking.
 - Agile delivery.
- Technology And Data:
 - CloudOps.
 - IOT.
- Adaptive Organizations:
 - Salesforce.
 - Process optimization.



Figura 10.123: Estructura interna de Studios 1.

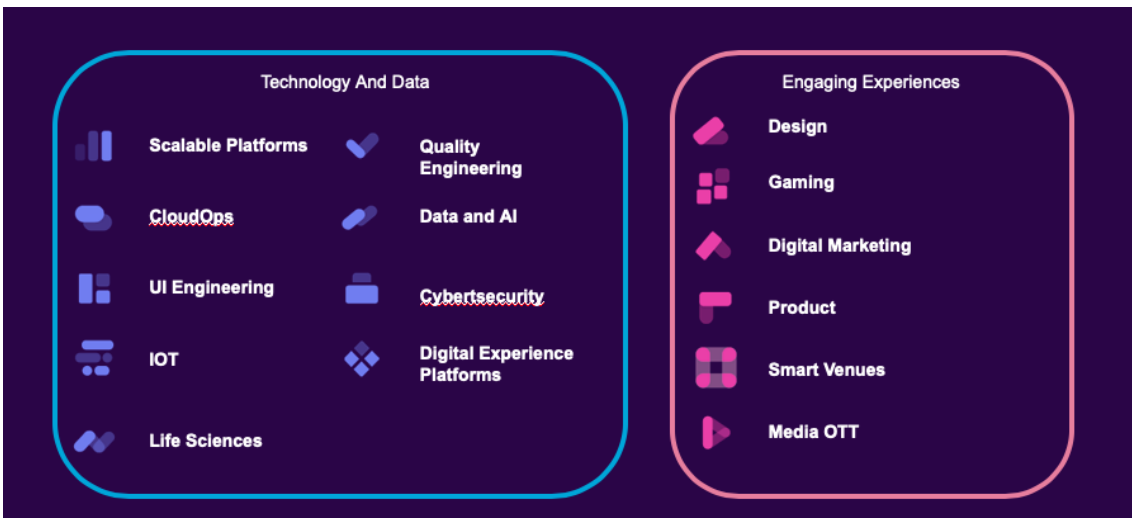


Figura 10.124: Estructura interna de Studios 2.

10.9.4. Labs y Catalyst

Labs es un estudio transversal que consume la innovación de todos los Studios de forma conjunta, o sea toma de cada uno (tiene a disposición) lo mejor que tiene cada Studio (expertos y recursos) y unifica en un proceso de *POC* para salida a un *MVP* para poder presentar la idea al top management y bussiness.

Dentro de Labs se encuentra alojado el programa Catalyst, encargo de llevar a cabo todos los desarrollos internos de la empresa, ya sean innovadores para ser presentados en un futuro a los clientes, de uso dentro de las oficinas o hasta tutorías tecnológicas para personas que quieran incursionar sobre la tecnología.

En otras palabras vendría a ser como una incubadora de tecnología dentro de la estructura interna del cliente.

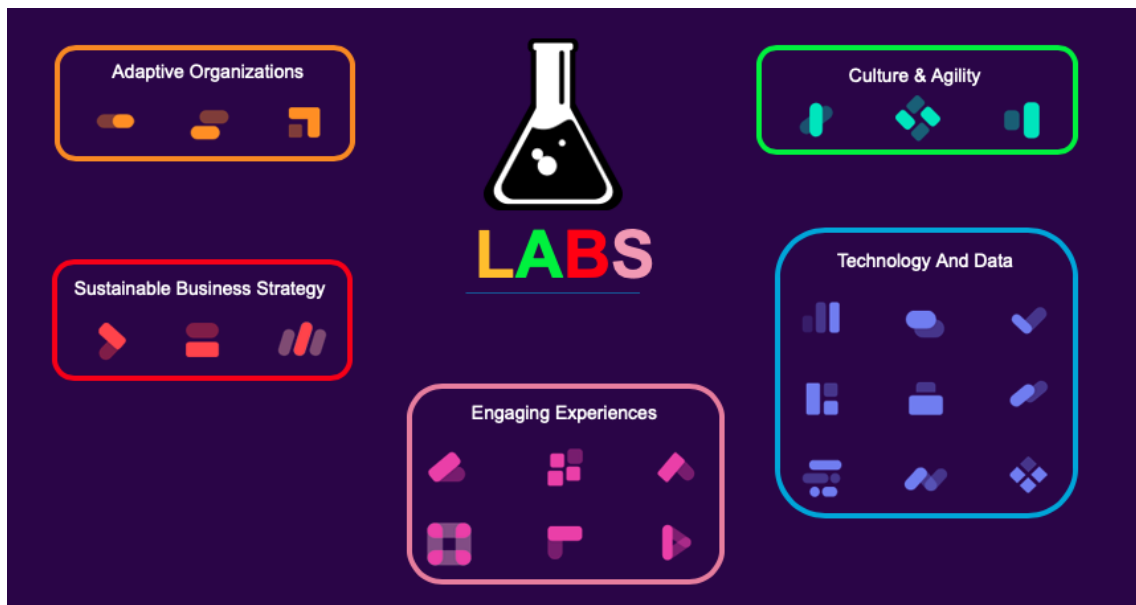


Figura 10.125: Labs dentro de la estructura interna de Globant.

10.10. Anexo 10 - Gestión de Incidentes

ID	Descripción	Reportado por	Tipo	Gravedad	Estado
I000	Aplicación no se ejecuta en emulador iOS	FB	Defecto	Alta	Resuelto
I001	Aplicación se cierra inesperadamente	FB	Defecto	Alta	Resuelto
I002	Descuentos de producto no muestran imagenes	PP	Defecto	Media	Resuelto
I003	Botones de agregar amigos desaparecen luego de navegar hacia drawer y volver	MS	Defecto	Media	Resuelto
I004	Endpoint de relaciones no devuelve todos los usuarios	MS	Defecto	Alta	Resuelto
I005	Cierre inesperado al hacer click en un descuento durante una búsqueda	FB	Defecto	Alta	Resuelto
I006	Usuario no desaparece de lista de usuarios agregables luego de ser agregado	MS	Defecto	Media	Resuelto
I007	Código QR generado no se almacena en backend	FB	Defecto	Alta	Resuelto
I008	Al redireccionar luego de compartir un descuento se abre community en lugar del inicio de stack de discounts	FB	Defecto	Media	Resuelto
I009	Store no se muestra correctamente sobre la tienda, se está mostrando la store contigua	PP	Defecto	Alta	Resuelto
I010	Al compartir ubicación la lista de usuarios queda seleccionada si se vuelve a compartir ubicación	FB	Defecto	Media	Resuelto
I011	Loading screen no muestra timeout si se corta la conexión al backend	PP	Defecto	Media	Resuelto
I012	Botón de cerrar búsqueda de usuarios no responde	FB	Defecto	Media	Resuelto
I013	Íconos no se muestran en Android luego de reordenación de autolink	PP	Defecto	Media	Resuelto
I014	Mapa no muestra ubicación del usuario - localización interna	MS	Defecto	Alta	Resuelto
I015	Redirección interrumpida después de login en iOS	MS	Defecto	Alta	Resuelto
I016	Store no es agregada a favoritos	FB	Defecto	Media	Resuelto
I017	Debido a actualización a iOS 15 la aplicación no se ejecuta	MS	Defecto	Alta	Resuelto

Figura 10.126: Planilla de gestión de incidentes.

10.11. Anexo 11 - Módulos HM-10

El HM-10 es un módulo *bluetooth* 4.0 que se utiliza para establecer comunicaciones de datos inalámbricas.

El mismo fue diseñado utilizando *chipset* Texas Instruments CC2540 o CC2541 *Bluetooth low energy* (BLE) system on chip (SOC) pero el diseño y *firmware* originario de Jinan Huamao Technology Company

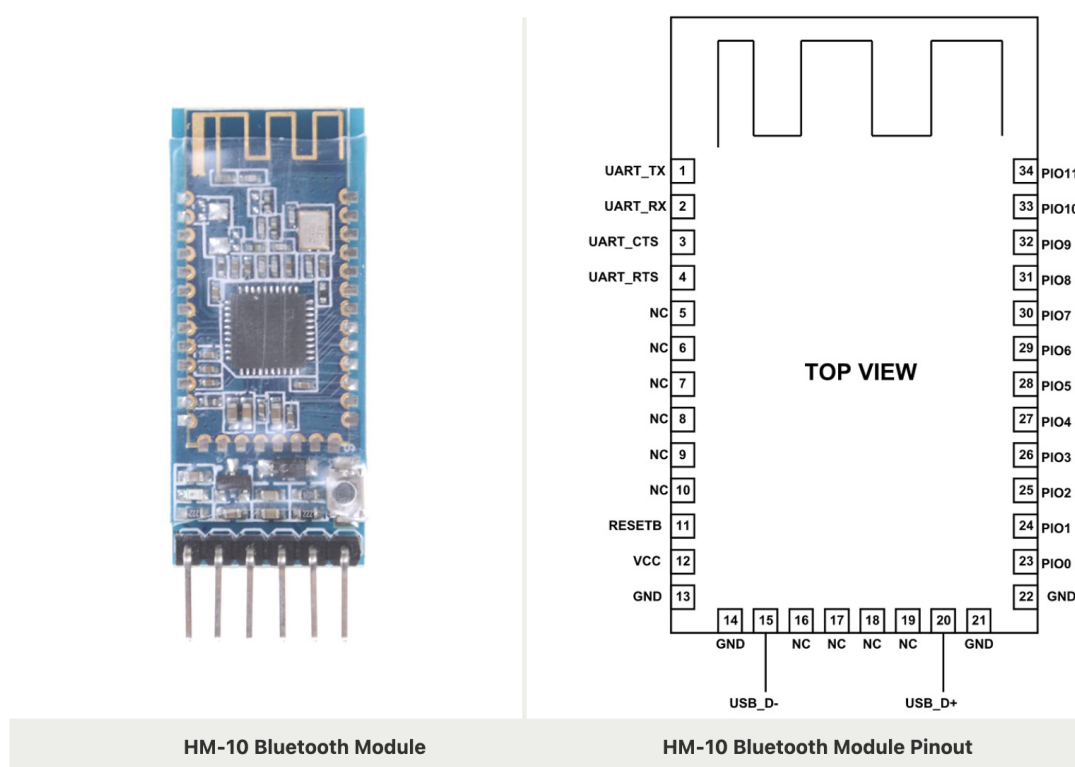


Figura 10.127: HM-10 modulo *pinout*.

El HM-10 es un módulo de 34 pines, en donde la gran mayoría no son de uso obligatorio, solamente necesitamos utilizar 4 de estos pines para poder establecer comunicación, a continuación se describe la función de cada uno de los pin:

Número	Nombre	Descripción
1	UART_TX	<i>UART interface-Transmit</i>
2	UART_RX	<i>UART interface-Receive</i>
3	UART_CTS	<i>UART interface</i>
4	UART_RTS	<i>UART interface</i>
5	NC	<i>No connection</i>
6	NC	<i>No connection</i>
7	NC	<i>No connection</i>
8	NC	<i>No connection</i>
9	NC	<i>No connection</i>
10	NC	<i>No connection</i>
11	RESETB	<i>Reset if low for 100ms</i>
12	VCC	3.3V
13	GND	<i>Ground</i>
14	GND	<i>Ground</i>
15	USB_D-	USB interface
16	NC	<i>No connection</i>
17	NC	<i>No connection</i>
18	NC	<i>No connection</i>
19	NC	<i>No connection</i>
20	USB_D+	USB interface
21	GND	<i>Ground</i>
22	GND	<i>Ground</i>
23	PIO0	<i>System key</i>
24	PIO1	<i>System LED</i>
25	PIO2	<i>Programmable input/output line</i>
26	PIO3	<i>Programmable input/output line</i>
27	PIO4	<i>Programmable input/output line</i>
28	PIO5	<i>Programmable input/output line</i>
29	PIO6	<i>Programmable input/output line</i>
30	PIO7	<i>Programmable input/output line</i>
31	PIO8	<i>Programmable input/output line</i>
32	PIO9	<i>Programmable input/output line</i>
33	PIO10	<i>Programmable input/output line</i>
34	PIO11	<i>Programmable input/output line</i>

Un resumen de sus características son las siguientes:

- *Bluetooth version: V4.0.*
- *BLE chip: bluetooth low energy consumption.*
- Envío y recibo sin límites de bytes.
- *UART serial interface.*
- *Full-speed USB interface.*
- *Long range: 100 metros en ambientes abiertos.*

A continuación se muestra como se realizó la conexión desde el módulo HM-10 hacia el controlador (Node MSU - CP2012 - con puente USB TTL CP2021), para poder conectarnos por USB y utilizar el Arduino IDE.

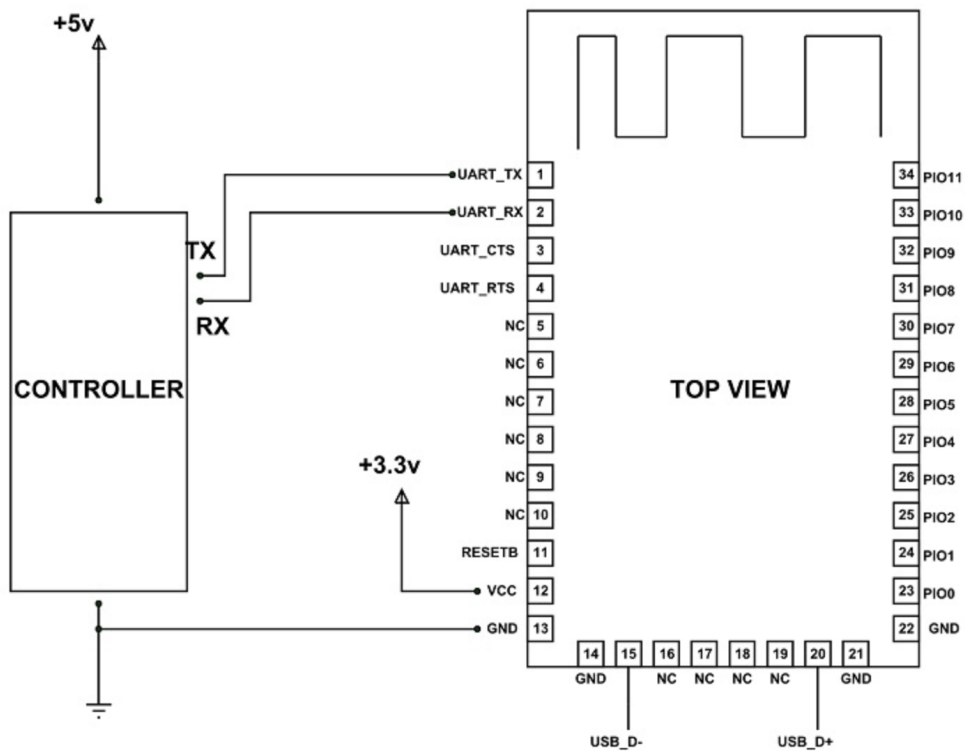


Figura 10.128: HM-10 conexión con controlador.

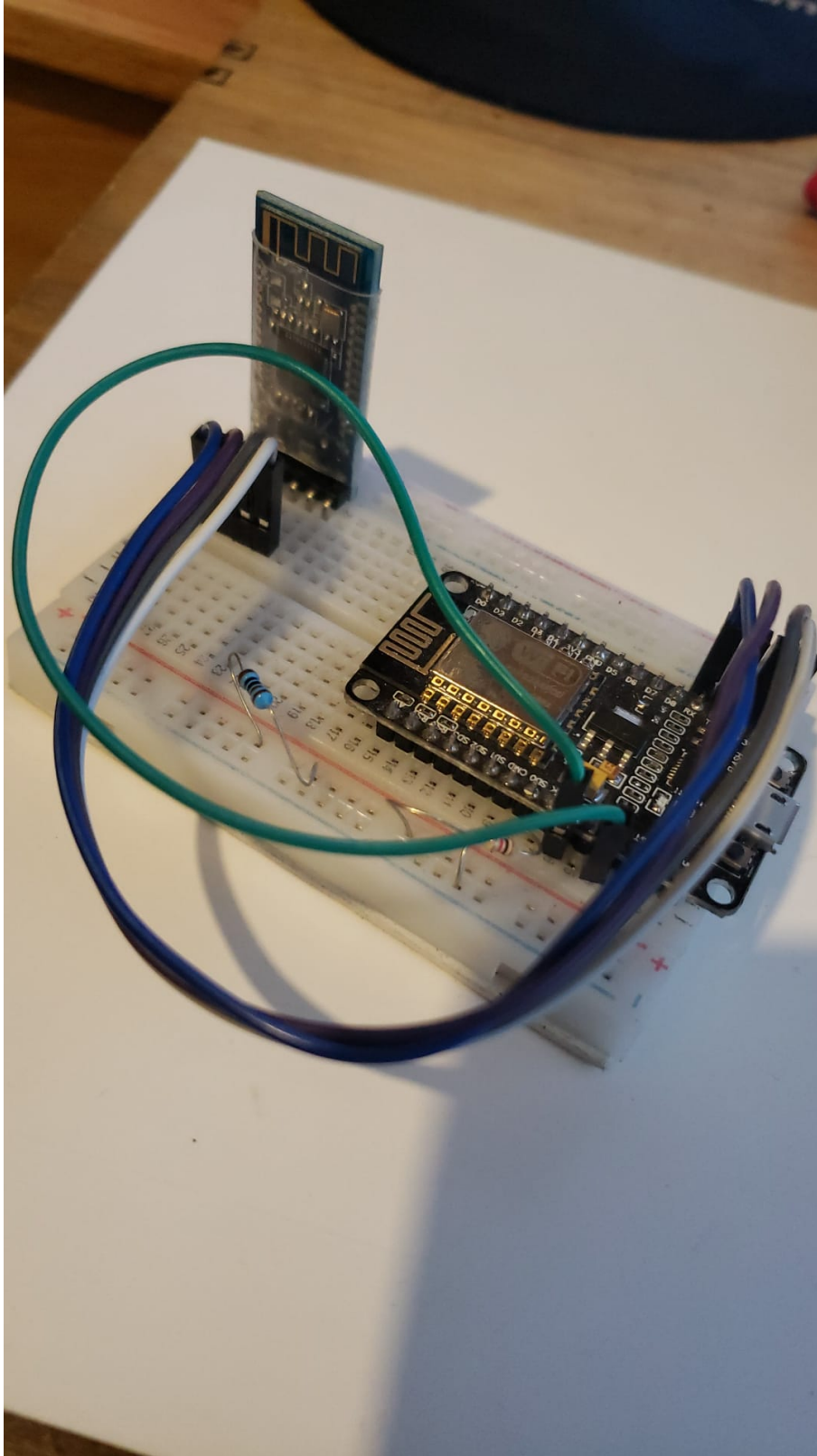


Figura 10.129: HM-10 conectado con nuestro controlador para poder configurarlo.

