

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACIÓN MÓVIL PROTOTIPO PARA RECOMENDAR
ESTABLECIMIENTOS COMERCIALES DE QUITO

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
EN SISTEMAS INFORMÁTICOS Y CIENCIAS DE LA COMPUTACIÓN

DANIEL ENRIQUE FREIRE GONZÁLEZ

dgz1606@gmail.com

WILMER VINICIO GUEVARA FARINANGO

wilmer.quevara@outlook.es

DIRECTOR: MSC. MARITZOL TENEMAZA

maritzol.tenemaza@epn.edu.ec

CO-DIRECTOR: MSC. SEBASTIAN GUERRERO

sebastian.guerrero@epn.edu.ec

Quito, noviembre 2019

DECLARACIÓN

Nosotros, Daniel Enrique Freire González y Wilmer Vinicio Guevara Farinango declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Daniel Enrique Freire González

**Wilmer Vinicio Guevara
Farinango**

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Daniel Enrique Freire González y Wilmer Vinicio Guevara Farinango, bajo nuestra supervisión.

MSc. Maritzol Tenemaza

DIRECTOR DE PROYECTO

MSc. Sebastian Guerrero

**CO-DIRECTOR DE
PROYECTO**

DEDICATORIA

Dedico este trabajo principalmente a Dios, por haberme dado a mis padres que con su amor, paciencia y esfuerzo me han permitido llegar a este momento tan importante de mi formación profesional.

A la memoria de Antonio Farinango mi abuelito un hombre dedicado y luchador, una admirable persona quien durante el desarrollo de esta tesis fue fuente de inspiración para cada día despertarme con ganas de alcanzar el éxito.

Finalmente, a mis amigos y amigas por extender su mano en momentos difíciles.

Wilmer Guevara

AGRADECIMIENTO

A Dios por ser la luz incondicional que ha guiado mi camino.

A mis padres por su apoyo incondicional, a mis hermanas por estar siempre pendiente de mí.

A mis profesores de la prestigiosa Escuela Politécnica Nacional.

A nuestra directora de tesis MSc. Maritzol Tenemaza y nuestro codirector Msc. Sebastian Guerrero por su guía durante el desarrollo del proyecto.

Finalmente, a Daniel Freire con quien realizamos este proyecto de manera exitosa.

Wilmer Guevara

ÍNDICE DE CONTENIDO

1	<u>INTRODUCCIÓN</u>	1
1.1	PLANTEAMIENTO DEL PROBLEMA.....	1
1.2	OBJETIVOS.....	2
1.2.1	OBJETIVO GENERAL	2
1.2.2	OBJETIVOS ESPECÍFICOS.....	2
1.3	MARCO TEÓRICO	2
1.3.1	SERVICIOS REST	2
1.3.2	PLATAFORMA COMO SERVICIO (PAAS).....	3
1.3.3	PATRÓN MVC	4
1.3.4	PRINCIPALES ENTIDADES DEL SISTEMA Y SU RELACIÓN	5
1.4	HERRAMIENTAS, FRAMEWORKS Y BIBLIOTECAS DE DESARROLLO	7
1.5	SCRUM.....	9
1.5.1	EQUIPO SCRUM (<i>SCRUM TEAM</i>).....	9
1.5.2	ARTEFACTOS DE SCRUM.....	10
1.5.3	EVENTOS DE SCRUM	10
1.5.4	SPRINT 0	11
2	<u>METODOLOGÍA</u>	11
2.1	SPRINT 0	11
2.1.1	ARQUITECTURA	12

2.1.2	DEFINICIÓN DE ROLES.....	12
2.1.3	HISTORIAS ÉPICAS	12
2.1.4	PRODUCT BACKLOG	13
2.1.5	PLANIFICACIÓN DEL RELEASE	13
2.2	SPRINT 1	14
2.2.1	SPRINT PLANNING	14
2.2.2	EJECUCIÓN DEL SPRINT.....	15
2.2.3	SPRINT REVIEW.....	20
2.2.4	SPRINT RETROSPECTIVE	22
2.3	SPRINT 2	22
2.3.1	SPRINT PLANNING	22
2.3.2	EJECUCIÓN DEL SPRINT.....	24
2.3.3	SPRINT REVIEW.....	25
2.3.4	SPRINT RETROSPECTIVE	27
2.4	SPRINT 3	27
2.4.1	SPRINT PLANNING	27
2.4.2	EJECUCIÓN DEL SPRINT.....	29
2.4.3	SPRINT REVIEW.....	34
2.4.4	SPRINT RETROSPECTIVE	35
2.5	SPRINT 4	36

2.5.1	SPRINT PLANNING	36
2.5.2	EJECUCIÓN DEL SPRINT.....	38
2.5.3	SPRINT REVIEW.....	45
2.5.4	SPRINT RETROSPECTIVE	46
2.6	SPRINT 5.....	47
2.6.1	SPRINT PLANNING	47
2.6.2	EJECUCIÓN DEL SPRINT.....	48
2.6.3	SPRINT REVIEW.....	51
2.6.4	SPRINT RETROSPECTIVE	53
3	<u>RESULTADOS Y DISCUSIÓN</u>	53
3.1	PRODUCTO FINAL	53
3.1.1	RECOMENDAR UN ESTABLECIMIENTO.....	53
3.1.2	CANJEAR LOS RECOPUNTOS POR UNA PROMOCIÓN	57
3.2	PRUEBAS CON USUARIOS FINALES	62
3.2.1	INTERÉS GENERADO	63
3.2.2	UTILIDAD PERCIBIDA	64
3.2.3	FACILIDAD DE USO PERCIBIDA	65
4	<u>CONCLUSIONES Y RECOMENDACIONES</u>	66
4.1	CONCLUSIONES	66
4.2	RECOMENDACIONES.....	67

5	REFERENCIAS BIBLIOGRÁFICAS	68
6	ANEXOS	70
6.1	ANEXO I: DIAGRAMA DE CLASES	70
6.2	ANEXO II: HISTORIAS ÉPICAS	71
6.3	ANEXO III: HISTORIAS DE USUARIO DEL SPRINT 1	71
6.4	ANEXO IV: HISTORIAS DE USUARIO DEL SPRINT 2	72
6.5	ANEXO V: HISTORIAS DE USUARIO DEL SPRINT 3	74
6.6	ANEXO VI: HISTORIAS DE USUARIO DEL SPRINT 4	75
6.7	ANEXO VII: HISTORIAS DE USUARIO DEL SPRINT 5	76
6.8	ANEXO VIII: CREACIÓN DE LOS SERVICIOS AWS	77
6.9	ANEXO IX: GLOSARIO	83
6.10	ANEXO X: GUÍA DE USO	86
6.11	ANEXO XI: ENCUESTAS REALIZADAS	91

ÍNDICE DE FIGURAS

Figura 1. Representación de PaaS en la nube [7].	3
Figura 2. Arquitectura del proyecto.	12
Figura 3: Diagrama de clases con la relación entre las preferencias del usuario y los establecimientos.	16
Figura 4. Comandos para crear las variables de entorno en Linux.	17
Figura 5. Adaptador por defecto para la conexión a la base de datos.	17
Figura 6. Configuración personalizada para tener acceso a las credenciales de AWS. ..	18
Figura 7: Línea de código para sobrescribir la contraseña del usuario con el hash de esta.	18
Figura 8. Arquitectura de la aplicación.	19
Figura 9. Estructura del proyecto.	20
Figura 10. Registro de intereses del usuario.	24
Figura 11. Validación para que el usuario escoja mínimo un interés y una colección.	25
Figura 12. Validaciones de la petición HTTP para obtener los establecimientos cercanos.	30
Figura 13. Pantalla principal.	31
Figura 14. Ventana emergente con todas las colecciones.	32
Figura 15. Ventana emergente con establecimientos <i>premium</i> de una colección seleccionada.	33
Figura 16. Algoritmo para calificar a un establecimiento.	38
Figura 17. Algoritmo para recomendar a un establecimiento.	39

Figura 18. Perfil del establecimiento.	40
Figura 19. Menú del establecimiento.	41
Figura 20. Comentarios del establecimiento (sección vacía).	41
Figura 21. Comentarios del establecimiento.	42
Figura 22. Galería del establecimiento.	42
Figura 23. Ubicación del establecimiento.	43
Figura 24. Ruta dibujada hacia el establecimiento.	43
Figura 25. Algoritmo para obtener la ruta hacia un establecimiento específico.	44
Figura 26. Algoritmo para canjear recopuntos.	49
Figura 27. Algoritmo para generar una imagen que contiene un código QR.	50
Figura 28. Algoritmo para cerrar la pantalla con el código QR generado.	51
Figura 29. Perfil del establecimiento con cero recomendaciones.	54
Figura 30. Opciones para recomendar del establecimiento.	55
Figura 31. Perfil del establecimiento con una recomendación.	56
Figura 32. Cantidad de recopuntos en el perfil de usuario.	57
Figura 33. Pantalla de promociones con los establecimientos cercanos.	58
Figura 34. Pantalla de promociones con promociones de un establecimiento seleccionado.	59
Figura 35. Ventana emergente de confirmación para canjear los recopuntos.	60
Figura 36. Pantalla con el código QR de la promoción y su información.	61
Figura 37. Pantalla de promociones luego de haber canjeado recopuntos.	62

Figura 38. Resultados de las preguntas sobre el interés generado.....	63
Figura 39. Diagrama de clases.	70
Figura 40. Pantalla para crear una cuenta en AWS.	77
Figura 41. Pantalla para crear la llave privada.	78
Figura 42. Descarga de la llave privada.....	79
Figura 43. Pantalla para seleccionar el motor de base de datos.	79
Figura 44. Datos de conexión a la base de datos.	80
Figura 45. Consola de administración de Amazon S3.....	81
Figura 46. Pantalla para buscar servicios de AWS.	81
Figura 47. Consola de administración de Amazon EC2.....	82
Figura 48. Formulario para el interés generado.....	91
Figura 49. Formulario para la utilidad percibida.....	92
Figura 50. Formulario para la facilidad de uso percibida.....	93
Figura 51. Respuestas de la pregunta 1 del interés generado.	94
Figura 52. Respuestas de la pregunta 2 del interés generado.	94
Figura 53. Respuestas de la pregunta 3 del interés generado.	95
Figura 54. Respuestas de la pregunta 1 para la utilidad percibida.	95
Figura 55. Respuestas de la pregunta 2 para la utilidad percibida.	96
Figura 56. Respuestas de la pregunta 3 para la utilidad percibida.	96
Figura 57. Respuestas de la pregunta 1 para la facilidad de uso percibida.	97
Figura 58. Respuestas de la pregunta 2 para facilidad de uso percibida.	97

Figura 59. Respuestas de la pregunta 3 para facilidad de uso percibida. 98

ÍNDICE DE TABLAS

Tabla 1. Roles del equipo Scrum en el Proyecto.	12
Tabla 2. Historias Épicas.	12
Tabla 3. <i>Product Backlog</i> inicial.....	13
Tabla 4. Planificación del release.....	14
Tabla 5. Historias de usuario escogidas para el sprint 1 (anexo III, sección 6.3).	14
Tabla 6. Sprint backlog del sprint 1.	14
Tabla 7: Pruebas de aceptación del sprint 1.....	20
Tabla 8. Product Backlog adaptado tras finalizar el sprint 1.....	21
Tabla 9. Historias de usuario escogidas para el sprint 2 (anexo IV, sección 6.4).....	22
Tabla 10. Sprint backlog del sprint 2.	23
Tabla 11. Pruebas de aceptación del sprint 2.	25
Tabla 12. Product Backlog adaptado tras finalizar el sprint 2.....	26
Tabla 13. Historias de usuario escogidas para el sprint 3 (anexo V, sección 6.5).....	28
Tabla 14. Sprint backlog del sprint 3.	28
Tabla 15. Pruebas de aceptación del sprint 3.	34
Tabla 16. Product backlog adaptado tras finalizar el sprint 3.	35
Tabla 17. Historias de usuario escogidas para el sprint 4 (anexo VI, sección 6.6).....	36
Tabla 18. Sprint backlog del sprint 4.	37
Tabla 19. Pruebas de aceptación del sprint 4.	45
Tabla 20. Product backlog adaptado tras finalizar el sprint 4.	46

Tabla 21. Historias de usuario escogidas para el sprint 3 (anexo VII, sección 6.7).....	47
Tabla 22. Sprint backlog del sprint 5.	48
Tabla 23. Pruebas de aceptación del sprint 5.	51
Tabla 24. Product backlog adaptado tras finalizar el sprint 5.	52
Tabla 25. Historia Épica HUE1.	71
Tabla 26. Historia Épica HUE2.	71
Tabla 27. Historia Épica HUE3.	71
Tabla 28. Historia de Usuario HU1-1.....	71
Tabla 29. Historia de Usuario HU2-1.....	72
Tabla 30. Historia de Usuario HU3-1.....	72
Tabla 31. Historia de Usuario HU4-1.....	73
Tabla 32. Historia de Usuario HU5-1.....	74
Tabla 33. Historia de Usuario HU1-2.....	75
Tabla 34. Historia de usuario HU2-2.	76
Tabla 35. Historia de Usuario HU1-3.....	76
Tabla 36. Historia de Usuario HU2-3.....	77

ÍNDICE DE GRÁFICOS

Gráfico 1. Modelo vista controlador.	5
Gráfico 2. Burndown chart del sprint 1.....	22
Gráfico 3. Burndown chart del sprint 2.....	27
Gráfico 4. Burndown chart del sprint 3.....	36
Gráfico 5. Burndown chart del sprint 4.....	47
Gráfico 6. Burndown chart del sprint 5.....	53
Gráfico 7. Promedio por pregunta de la utilidad percibida en el producto final.	64
Gráfico 8. Promedio por pregunta de la facilidad de uso percibida en el producto final. .	65

RESUMEN

El presente trabajo ofrece una solución a establecimientos que quieren llegar a más personas a través de aplicaciones móviles; y a personas que quieren recomendar lugares a sus amigos y así mismo conocer los lugares que ellos frecuentan.

La solución es una aplicación móvil para plataformas iOS que permitirá a los establecimientos registrados promocionar sus productos; y a las personas aprovechar las soluciones.

La realización de esta solución inicia con la introducción en donde se detalla cuál es la problemática, los objetivos, las tecnologías y las herramientas que se usarán.

En el segundo capítulo se mencionan los aspectos más importantes de cada etapa en el proceso de desarrollo de la aplicación.

En el tercer capítulo se presentan los resultados obtenidos una vez desarrollada la aplicación, así como los resultados de las pruebas realizadas.

Finalmente, en el cuarto capítulo se presentan las conclusiones de este trabajo, así como las recomendaciones y sugerencias para trabajos futuros.

Palabras clave: Recomendar, algoritmos, API, recopuntos, intereses, colecciones, iOS.

ABSTRACT

The present work offers a solution to establishments that want to reach out to more people through mobile applications and to people who want to recommend places to their friends and also know the places they frequent.

The solution is a mobile application for iOS platforms that will allow registered establishments to promote their products; and for people to take advantage of the solutions.

The realization of this solution begins with the introduction where the problem, objectives, technologies and tools to be used are detailed.

In the second chapter, the most important aspects of each stage in the application development process are mentioned.

In the third chapter, the results obtained once the application has been developed are presented, as well as the results of the tests performed.

Finally, the conclusions of this work are presented in the fourth chapter, as well as the recommendations and suggestions for future work.

Key words: Recommend, algorithms, API, recopuntos, interests, collections, iOS.

1 INTRODUCCIÓN

1.1 Planteamiento del problema

El incremento constante de turistas nacionales y extranjeros, ha traído como consecuencia que los establecimientos comerciales en Quito busquen nuevas medidas o herramientas para incrementar su competitividad y darse a conocer. En este sentido, los establecimientos pueden optar por utilizar aplicaciones en las que se registran y esperar a que usuarios registrados en dicha aplicación los busquen. Posteriormente, el usuario podrá comentar o calificar al establecimiento. Sin embargo, llegar a las personas de forma directa y posicionarse en el top de los resultados de búsqueda en la aplicación puede llegar a ser difícil y muy costoso ya que además de que la persona lo vea debe ingresar a dicha aplicación web.

Por otra parte, cabe destacar que el 33,7% de la población ecuatoriana tiene un teléfono inteligente (smartphone) [1], en la provincia de Pichincha el 69.5% cuenta con uno [2] y en la ciudad de Quito, el 66.9% [3]. Actividades cotidianas como pedir comida a domicilio o reservar un hotel se pueden realizar usando un smartphone y esta tendencia crece de la misma manera que el desarrollo de aplicaciones móviles.

Así mismo, hoy en día existen aplicaciones que muestran reseñas de diferentes destinos como Tripadvisor [4] o reseñas de locales comerciales en general como Yelp [5], estas reseñas son hechas por los usuarios de dichas aplicaciones. Sin embargo, no existe una aplicación que permita recomendar establecimientos comerciales puntuales a un grupo de amigos, familia, etc. O una aplicación que nos brinde algún tipo de recompensa por realizar las recomendaciones.

Este proyecto plantea el desarrollo de una aplicación móvil prototipo que permita al usuario recomendar a sus contactos los establecimientos comerciales. Además, se implementará un mecanismo de recompensas por realizar las recomendaciones. Estas recompensas serán los “recopuntos” que podrán ser acumulados y posteriormente gastados en todos los locales comerciales afiliados a la aplicación. Además, los usuarios podrán recibir recomendaciones de establecimientos sin tener la aplicación instalada ya que el sistema de recomendaciones se realizará utilizando las redes Facebook y WhatsApp.

El no contar con esta aplicación provocará que la situación actual se mantenga. De llegar a usarse este sistema los establecimientos obtendrán un medio diferente para publicitarse y los usuarios podrán recomendarlos a sus amigos y recibir recomendaciones de ellos.

Recomiendo será una aplicación que permitirá conectar oportunidades entre establecimientos comerciales y usuarios, con perfiles relacionados a variedad de industrias de alimentos, bebidas, hospedaje y diversión.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar una aplicación móvil prototipo para la plataforma iOS que permita realizar recomendaciones de establecimientos comerciales específicos a usuarios.

1.2.2 Objetivos Específicos

- Establecer los requerimientos funcionales y no funcionales, el negocio en general, mediante reuniones con el *product owner*, para la correcta implementación del prototipo.
- Configurar los equipos necesarios con el software adecuado para el desarrollo de una aplicación iOS.
- Definir la metodología a utilizar con el análisis de los requerimientos para la respectiva implementación del prototipo.
- Implementar un algoritmo que muestre los establecimientos comerciales que un usuario visualiza en la pantalla de inicio con base en: sus calificaciones; sus recomendaciones; y su ubicación para promover las promociones de los establecimientos afiliados.

1.3 Marco Teórico

1.3.1 Servicios REST

REST, o *Representational State Transfer*, es un estilo arquitectónico para proporcionar estándares entre sistemas informáticos en la web, lo que facilita la comunicación entre los sistemas. En la arquitectura REST, los clientes envían solicitudes para recuperar o modificar recursos, y los servidores envían respuestas a estas solicitudes [6].

1.3.2 Plataforma como servicio (PaaS)

Las plataformas como servicio eliminan la necesidad de las compañías de administrar la infraestructura subyacente (normalmente hardware y sistemas operativos) y le permiten centrarse en la implementación y la administración de sus aplicaciones. Esto contribuye a mejorar su eficacia, pues no tiene que preocuparse del aprovisionamiento de recursos, la planificación de la capacidad, el mantenimiento de software, los parches, ni de las demás arduas tareas que conlleva la ejecución de su aplicación [7].

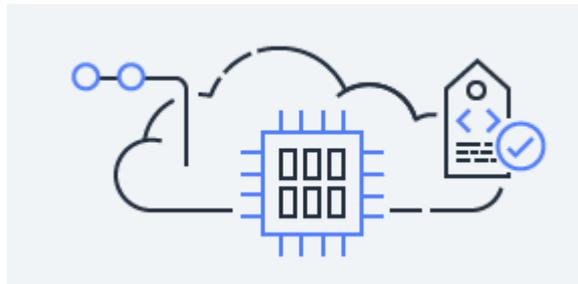


Figura 1. Representación de PaaS en la nube [7].

1.3.2.1 RDS

Con Amazon Relational Database Service (Amazon RDS), es sencillo configurar, utilizar y escalar una base de datos relacional en la nube. El servicio suministra capacidad rentable y escalable al mismo tiempo que automatiza las arduas tareas administrativas, como el aprovisionamiento de hardware, la configuración de bases de datos, la implementación de parches y la creación de copias de seguridad. RDS libera de estas tareas para que pueda concentrarse en sus aplicaciones y darles el rendimiento rápido, la alta disponibilidad, la seguridad y la compatibilidad que necesitan [8].

1.3.2.2 EC2

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable. Está diseñado para simplificar el uso de la informática en la nube a escala web para los desarrolladores.

La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar capacidad con una fricción mínima. Proporciona un control completo sobre los recursos informáticos y puede ejecutarse en el entorno informático acreditado de Amazon. Amazon EC2 reduce el tiempo necesario para obtener e iniciar nuevas instancias de servidor en

cuestión de minutos, lo que permite escalar rápidamente la capacidad, ya sea aumentándola o reduciéndola, en función de sus necesidades [9].

1.3.2.3 S3

Amazon *Simple Storage Service* (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector. Esto significa que clientes de todos los tamaños y sectores pueden utilizarlo para almacenar y proteger cualquier cantidad de datos para diversos casos de uso, como sitios web, aplicaciones móviles, procesos de copia de seguridad y restauración, operaciones de archivado, aplicaciones empresariales, dispositivos IoT y análisis de *big data*. Amazon S3 proporciona características de administración fáciles de utilizar que le permiten organizar los datos y configurar sofisticados controles de acceso con objeto de satisfacer sus requisitos empresariales, organizativos y de conformidad. Amazon S3 está diseñado para ofrecer una durabilidad del 99,999999999 % y almacena datos de millones de aplicaciones para empresas de todo el mundo [10].

1.3.3 Patrón MVC

MVC (Modelo Vista Controlador) es un patrón de diseño de aplicación que consta de tres partes interconectadas. Incluyen el modelo (datos), la vista (GUI, CLI o API) y el controlador (procesos que manejan la entrada).

El patrón MVC se usa comúnmente para desarrollar interfaces de usuario modernas. Proporciona las piezas fundamentales para diseñar programas para computadoras de escritorio o dispositivos móviles, así como aplicaciones web. Funciona bien con la programación orientada a objetos, ya que los diferentes modelos, vistas y controladores pueden tratarse como objetos y reutilizarse dentro de una aplicación [11].

Modelo

Un modelo son datos utilizados por un programa. Puede ser una base de datos, un archivo o un objeto simple, como un ícono o un personaje en un videojuego.

Vista

Una vista es el medio de mostrar objetos dentro de una aplicación.

Controlador

Un controlador actualiza ambos, los modelos y las vistas. Acepta entradas y realiza la actualización correspondiente.

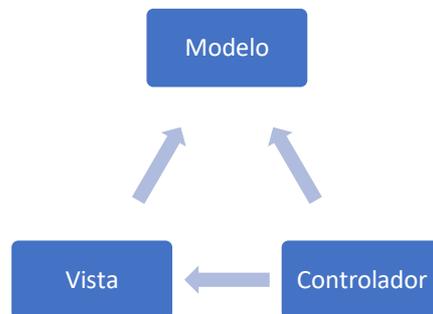


Gráfico 1. Modelo vista controlador.

Las tres partes de MVC están interconectadas. La vista muestra el modelo para el usuario. El controlador acepta la entrada del usuario y actualiza el modelo y la vista en consecuencia. Si bien MVC no es necesario en el diseño de aplicaciones, muchos lenguajes de programación e IDEs son compatibles con la arquitectura MVC, por lo que es una opción común para los desarrolladores [11].

1.3.4 Principales entidades del sistema y su relación

El sistema cuenta con 4 entidades principales que son: Establecimientos, Usuarios, Intereses y Colecciones.

Establecimientos

Son los locales comerciales que a través de una aplicación web se registran en el sistema. Estos establecimientos, para objeto de la estrategia del negocio, se diferencian con 3 estados que son: regular, *sponsor* y *premium*. Todo establecimiento tiene el estado de regular cuando es creado y dependiendo del plan o membresía que contrate puede adquirir uno o los dos estados restantes. El estado *sponsor* permite al establecimiento mostrar sus promociones en la pantalla de *home* de la aplicación móvil. El estado *premium* permite al establecimiento aparecer en la segunda ventana emergente que se muestra en la pantalla de *home* de la aplicación móvil.

Al crear un establecimiento se le debe asignar obligatoriamente por lo menos un interés y una colección. Esto es indispensable ya que es la base de la relación que existirá con los usuarios.

Para efecto de la culminación del prototipo y de este proyecto toda la información referente a los establecimientos fue ingresada al sistema directamente por el equipo de desarrollo, dado que el desarrollo de la aplicación web no es parte del proyecto.

Usuarios

Son las personas que tengan acceso a la aplicación móvil y se registren en el sistema. Los usuarios una vez que hayan registrado sus datos personales deberán escoger obligatoriamente por lo menos un interés y una colección. Esto es indispensable ya que de esto depende lo que la aplicación móvil muestre.

Intereses

De acuerdo a lo expresado por el *product owner* los intereses, en el sistema, representan las cosas o eventos que les pueden atraer a los usuarios y se puedan ofrecer en un establecimiento como, por ejemplo, la música en vivo, la cocina gourmet, la comida italiana, los cocteles, entre otros.

Dado que la lista completa de intereses no fue proporcionada por el *product owner*, para efecto de desarrollo y pruebas, el equipo de desarrollo definió la siguiente lista de intereses:

- comida italiana
- comida india
- cocteles
- comida china
- comida mexicana
- música en vivo
- comida rápida
- cocina gourmet

Colecciones

De acuerdo a lo expresado por el *product owner* las colecciones, en el sistema, representan los planes que pueden tener los usuarios para realizar con otras personas en un establecimiento. Cuando un usuario selecciona una o más colecciones es para informar a la aplicación acerca de cuál es su objetivo al usar la aplicación. Algunas colecciones pueden ser: salida en pareja, salida de familia, entre otras.

La lista completa de colecciones proporcionada por el *product owner* es la siguiente:

- salida con amigos
- salida en pareja
- salida de chicas
- salida de chicos
- salida de familia
- salida de niños

Los intereses y las colecciones conforman las preferencias del usuario.

La relación que existe entre estas cuatro entidades y parte de la lógica del sistema viene dada de la siguiente manera:

1. Todo establecimiento registrado en el sistema tiene por lo menos un interés y una colección asociadas.
2. Todo usuario registrado en el sistema igualmente tiene por lo menos un interés y una colección asociadas.
3. Los establecimientos que se muestren en la aplicación móvil dependerán de que sus intereses y colecciones empaten con los del usuario que está usando la aplicación.

Finalmente, hay que mencionar que un establecimiento **no** tiene interacción alguna con la aplicación móvil, los únicos que interactúan con ella son los usuarios.

1.4 Herramientas, frameworks y bibliotecas de desarrollo

Sails.js

Sails.js o Sails es un *framework* ligero que se encuentra sobre Express, otro *framework*. Su conjunto de pequeños módulos trabaja para proporcionar simplicidad, facilidad de mantenimiento y convenciones estructurales a las aplicaciones de Node.js [12]. Este *framework* fue utilizado para construir el *backend* de la aplicación.

Directions API de Google Maps

Es un servicio que calcula direcciones entre dos ubicaciones. Se puede realizar la búsqueda para varios modos de transporte, incluidos el transporte público, la conducción,

el caminar o el andar en bicicleta [13]. Mediante este servicio se obtienen las rutas a los diferentes establecimientos.

AWS SDK para JavaScript en Node.js

Es un conjunto de desarrollo de software que ayuda a eliminar la complejidad de la codificación al proporcionar objetos JavaScript para los servicios AWS (*Amazon Web Services*), incluidos Amazon S3, Amazon EC2, entre otros [14]. Este SDK (*Software Development Kit*) fue usado principalmente para la comunicación con el servicio Amazon S3 donde se almacenan las imágenes utilizadas en la aplicación.

Socket.IO

Es una biblioteca que permite la comunicación en tiempo real, bidireccional y basada en eventos entre el cliente, que puede ser un navegador o un dispositivo móvil, y el servidor [15]. Esta biblioteca fue usada para el manejo las características de la aplicación que son en tiempo real como la creación de nuevos establecimientos.

WebStorm

Es un IDE (*Integrated Development Environment*) de desarrollo que provee soporte completo para JavaScript, TypeScript, HTML, CSS, así como para *frameworks* como React, Angular y Vue.js. Ayuda al desarrollo del lado del servidor con Node.js [16]. Este IDE fue una herramienta bastante útil por su fácil manejo sin la necesidad de complementos.

Xcode

Es un IDE (*Integrated Development Environment*) de Apple. Se utiliza para crear aplicaciones para productos Apple, incluidos iPad, iPhone, Apple Watch, Apple TV y Mac. Proporciona herramientas para administrar todo el flujo de trabajo de desarrollo, desde crear la aplicación hasta probarla, optimizarla y enviarla a la App Store [17]. Este IDE fue utilizado para la realización de la aplicación móvil.

Bitbucket

Es un sistema para alojar repositorios de control de versiones propiedad de Atlassian, utiliza el VCS (*Version Control System*) *Git* o *Mercurial*. Un VCS, también conocido como control de revisión o sistema de control de fuente, es una utilidad de software que rastrea

y administra los cambios en un sistema de archivos [18]. Durante el progreso del proyecto, se utilizó Bitbucket para el almacenamiento y versionado del código.

Postman

Es una plataforma de colaboración para el desarrollo de API. Sus características simplifican cada paso de la creación de una API y agilizan la colaboración para que pueda crear mejores API, más rápido [19]. Se utilizó esta herramienta para la prueba del servicio web y de los servicios de Google.

1.5 Scrum

Scrum es un marco de proceso que se ha utilizado para gestionar el trabajo en productos complejos desde principios de los años noventa. Scrum no es un proceso, técnica o método definitivo. Más bien, es un marco dentro del cual puede emplear diversos procesos y técnicas. Scrum deja en claro la eficacia relativa de la gestión de su producto y las técnicas de trabajo para que pueda mejorar continuamente el producto, el equipo y el entorno de trabajo [20].

1.5.1 Equipo Scrum (*Scrum Team*)

El equipo Scrum consta de un *Product Owner*, *The Development Team* y un *Scrum Master*. Los equipos Scrum son autoorganizados y multifuncionales. Los equipos autoorganizados eligen la mejor manera de realizar su trabajo, en lugar de ser dirigidos por otros fuera del equipo. Los equipos interfuncionales tienen todas las competencias necesarias para realizar el trabajo sin depender de otros que no forman parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad [20].

- El *Product Owner*. Es responsable de maximizar el valor del producto resultante del trabajo del equipo de desarrollo [20].
- El *Development Team* está formado por profesionales que realizan el trabajo de entregar un Incremento potencialmente liberable del producto "Listo" al final de cada *Sprint*. Se requiere un incremento "Listo" en el Sprint review [20].
- El *Scrum Master* es responsable de promover y apoyar Scrum como se define en la Guía Scrum. Los Scrum Masters hacen esto al ayudar a todos a comprender la teoría, las prácticas, las reglas y los valores de Scrum [20].

1.5.2 Artefactos de Scrum

- *Product Backlog* es una lista ordenada de todo lo que se sabe que se necesita en el producto. Es la única fuente de requisitos para cualquier cambio que se realice en el producto. El propietario del producto es responsable de la acumulación de productos, incluido su contenido, disponibilidad y pedidos [20].
- *Sprint backlog* es el conjunto de elementos del *product backlog* seleccionados para el *Sprint*, más un plan para entregar el Incremento del producto y alcanzar el Objetivo Sprint. El *sprint backlog* es un pronóstico del *Development Team* sobre qué funcionalidad estará en el próximo Incremento y el trabajo necesario para entregar esa funcionalidad en un Incremento "Listo" [20].
- El Incremento es la suma de todos los elementos del *product backlog* completados durante un *Sprint* y el valor de los incrementos de todos los *Sprints* anteriores. Al final de un *Sprint*, el nuevo Incremento debe estar "Listo", lo que significa que debe estar en condiciones de uso y cumplir con la definición del *Scrum Team* de "Listo" [20].

1.5.3 Eventos de Scrum

Los eventos prescritos se usan en Scrum para crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son eventos de tiempo, de modo que cada evento tiene una duración máxima.

- (Sprint) El corazón de Scrum es un Sprint, un lapso de tiempo de un mes o menos durante el cual se crea un Incremento de producto "Listo", utilizable y potencialmente liberable. Los sprints tienen duraciones constantes a lo largo de un esfuerzo de desarrollo. Un nuevo Sprint comienza inmediatamente después de la conclusión del Sprint anterior [20].
- Cancelación de un Sprint (Cancelling a Sprint) Un Sprint puede cancelarse antes de que termine el tiempo de Sprint. Solo el propietario del producto tiene la autoridad para cancelar el Sprint, aunque puede hacerlo bajo la influencia de las partes interesadas, el equipo de desarrollo o el Scrum Master [20].
- La planificación de Sprint (Sprint planning) tiene un límite de tiempo de hasta ocho horas por un mes de Sprint. Para Sprints más cortos, el evento suele ser más corto. El Scrum Master asegura que el evento se lleve a cabo y que los asistentes comprendan su propósito [20].

- Daily Scrum es un evento de 15 minutos para el equipo de desarrollo. El Daily Scrum se lleva a cabo todos los días del Sprint. En él, el Equipo de Desarrollo planea trabajar para las próximas 24 horas. Esto optimiza la colaboración y el rendimiento del equipo al inspeccionar el trabajo desde el último Daily Scrum y pronosticar el próximo trabajo de Sprint [20].
- Se realiza una Revisión de Sprint (Sprint Review) al final del Sprint para inspeccionar el Incremento y adaptar la Cartera de Producto si es necesario. Durante la Revisión de Sprint, el Equipo Scrum y las partes interesadas colaboran sobre lo que se hizo en Sprint. En base a eso y a cualquier cambio en el Backlog del Producto durante el Sprint, los asistentes colaboran en las siguientes cosas que podrían hacerse para optimizar el valor. Esta es una reunión informal, no una reunión de estado, y la presentación del Incremento tiene como objetivo obtener comentarios y fomentar la colaboración [20].
- La Retrospectiva de Sprint (The sprint Retrospective) ocurre después de la Revisión de Sprint y antes de la próxima Planificación de Sprint. Esto es como máximo una reunión de tres horas para Sprints de un mes. Para Sprints más cortos, el evento suele ser más corto. El Scrum Master asegura que el evento se lleve a cabo y que los asistentes comprendan su propósito [20].

1.5.4 Sprint 0

Un Sprint 0 es el nombre que a menudo se le da a un breve esfuerzo para crear una visión y una acumulación de productos en bruto que permite crear una estimación de la liberación de un producto. No hay nada malo en hacerlo, siempre y cuando todos los interesados tengan claro que el plan está sujeto a cambios, ya que se sabe más como resultado de la inspección del resultado de los *sprints*. En resumen, esa actividad no cumple con la definición de Sprint en Scrum, por lo que es mejor no llamarlo así [21].

2 METODOLOGÍA

2.1 Sprint 0

Este “*sprint 0*” tuvo una duración aproximada de 1 mes. Aquí se recopiló toda la información necesaria antes de empezar el proyecto, se diseñó su arquitectura global y un *mockup* de la aplicación móvil, se definieron los roles del *Scrum Team*, las historias épicas, el *Product Backlog* y finalmente el número de *sprints* con su duración.

2.1.1 Arquitectura

La **Figura 2** define la infraestructura global del proyecto mediante una arquitectura Modelo-vista-controlador (MVC). El modelo corresponde a los datos, de donde se obtiene toda la información de los establecimientos registrados; el controlador es un servidor que procesa las solicitudes HTTP; y la vista es la aplicación móvil.

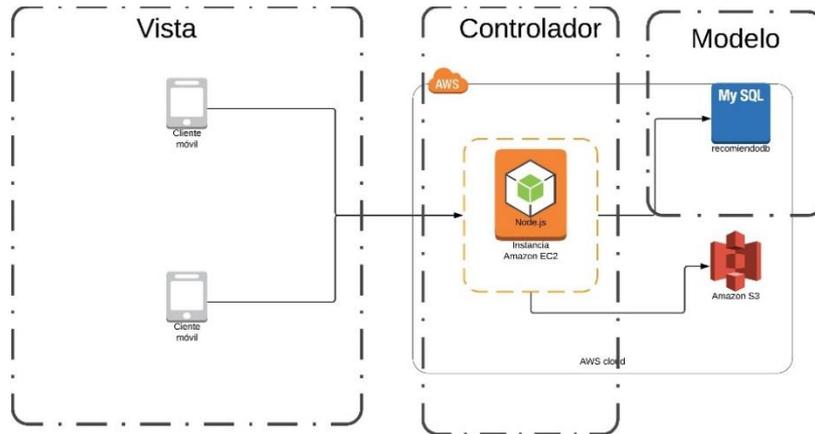


Figura 2. Arquitectura del proyecto.

2.1.2 Definición de roles

Tabla 1. Roles del equipo Scrum en el Proyecto.

Equipo Scrum	
Rol	Responsable
<i>Product Owner</i>	Fundadores de la empresa Optimus Social
<i>Scrum Master</i>	Maritzol Tenemaza, Sebastian Guerrero
<i>Development Team</i>	Daniel Freire, Wilmer Guevara

2.1.3 Historias Épicas

En la **Tabla 2** se pueden apreciar las historias épicas identificadas para el desarrollo de todo el proyecto. El detalle de cada una se puede visualizar en el anexo II (sección **6.2**).

Tabla 2. Historias Épicas.

CÓDIGO	TÍTULO	PRIORIDAD
HUE1	Gestión de usuarios.	Alta

HUE2	Gestión de establecimientos.	Alta
HUE3	Gestión de promociones y recopuntos.	Alta

2.1.4 Product Backlog

Las historias de usuario épicas se dividen en historias de usuario, la dimensión o esfuerzo de cada una se definió de acuerdo con la técnica de *Planning Poker* donde la medida de estimación está basada en puntos. Se ha tomado como pivote la historia de usuario HU2-2 asignándole 4 puntos y en base a la misma se ha estimado el resto.

En la **Tabla 3** se puede apreciar el primer *product backlog* creado en el sprint 0.

Tabla 3. *Product Backlog* inicial.

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Prioridad
HUE1	HU1-1	Registrar usuarios.	20	Alta
	HU2-1	Registrar preferencias de usuario.	7	Alta
	HU3-1	Inicio de sesión de usuarios.	7	Alta
	HU4-1	Navegación de usuario en la aplicación.	6	Alta
	HU5-1	Visualizar promociones, establecimientos y colecciones.	20	Alta
HUE2	HU1-2	Perfilar el establecimiento.	13	Alta
	HU2-2	Visualizar la ubicación de un establecimiento.	7	Alta
HUE3	HU1-3	Definición de promociones por establecimiento.	13	Alta
	HU2-3	Asignación de promociones.	7	Alta

2.1.5 Planificación del Release

Tras establecer un total de 9 historias de usuario con sus respectivas estimaciones de esfuerzo las cuales sumadas tienen un total de 100 puntos se decidió dividir el esfuerzo en 5 sprints, así cada uno debería tener un promedio de 20 puntos de esfuerzo, con esto

garantizamos un equilibrio de carga de esfuerzo en cada Sprint. Se estima que cada sprint durará entre 1 y 2 semanas obteniendo en el peor de los casos 10 semanas que son dos meses y medio para el desarrollo de la aplicación.

Tabla 4. Planificación del release.

Planificación del release									
Sprint 1		Sprint 2		Sprint 3		Sprint 4		Sprint 5	
ID	Esfuerzo	ID	Esfuerzo	ID	Esfuerzo	ID	Esfuerzo	ID	Esfuerzo
HU1-1	20	HU2-1	7	HU5-1	20	HU1-2	13	HU1-3	13
		HU3-1	7			HU2-2	7	HU2-3	7
		HU4-1	6						
TOTAL	20		20		20		20		20

2.2 Sprint 1

2.2.1 Sprint Planning

Objetivo del Sprint

Construir un servicio RESTful para el registro de usuarios con el fin de almacenar sus datos.

Historias de Usuario

En la **Tabla 5** se encuentran las historias de usuario que fueron seleccionadas para el primer sprint. En el anexo III se encuentran desglosadas con más detalle.

Tabla 5. Historias de usuario escogidas para el sprint 1 (anexo III, sección 6.3).

ID	Título	Esfuerzo	Estado
HU1-1	Registrar usuarios.	20	Por implementar

Sprint Backlog

Tabla 6. Sprint backlog del sprint 1.

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
HU1-1	Determinar las entidades y sus relaciones.

Crear el modelo de base datos.
Crear el repositorio en Bitbucket.
Instalar el <i>framework</i> Sails de manera global.
Generar un proyecto vacío de Sails.
Codificar los modelos representantes de las entidades del modelo de base de datos.
Levantar el servidor y verificar que escuche solicitudes http.
Creación de una cuenta en AWS.
Crear la llave de acceso en AWS.
Creación del RDS.
Configuración del RDS
Creación del S3
Configuración del S3
Creación de la Máquina virtual en EC2 en AWS.
Configuración de la instancia.
Verificar que los ambientes en AWS funcionen correctamente.
Crear el <i>endpoint</i> para registrar usuarios.
Validar que en la base se almacene el <i>hash</i> de la contraseña y no la contraseña en sí.
Validar que, en caso de error en el registro, si el usuario subió una imagen esta se borre.
Crear un proyecto <i>single view app</i> en Xcode.
Añadir el proyecto al repositorio.
Crear los modelos necesarios.
Implementar funciones genéricas para consumir los <i>endpoints</i> disponibles en el <i>backend</i> .
Implementar funcionalidad de carga de imagen del perfil de usuario.
Implementar el diseño de la pantalla de registro de usuarios.
Implementar la funcionalidad de la pantalla de registro de usuario.

2.2.2 Ejecución del Sprint

Durante la ejecución del *sprint* se pueden mencionar los siguientes detalles importantes.

Diagrama de clases para el registro de usuarios y sus preferencias

La **Figura 3** muestra el diagrama de clases correspondiente a las tablas involucradas en el registro de usuarios y sus preferencias, así como la relación que existe entre estas preferencias y los establecimientos. Las preferencias del usuario vienen dadas por los interés y colecciones que escojan, estas preferencias son fundamentales para que la aplicación muestre al usuario establecimientos que sean de su interés ya que estos establecimientos también estarán ligados a intereses y colecciones específicos.

En el anexo I, sección 6.1 se encuentra el diagrama completo.

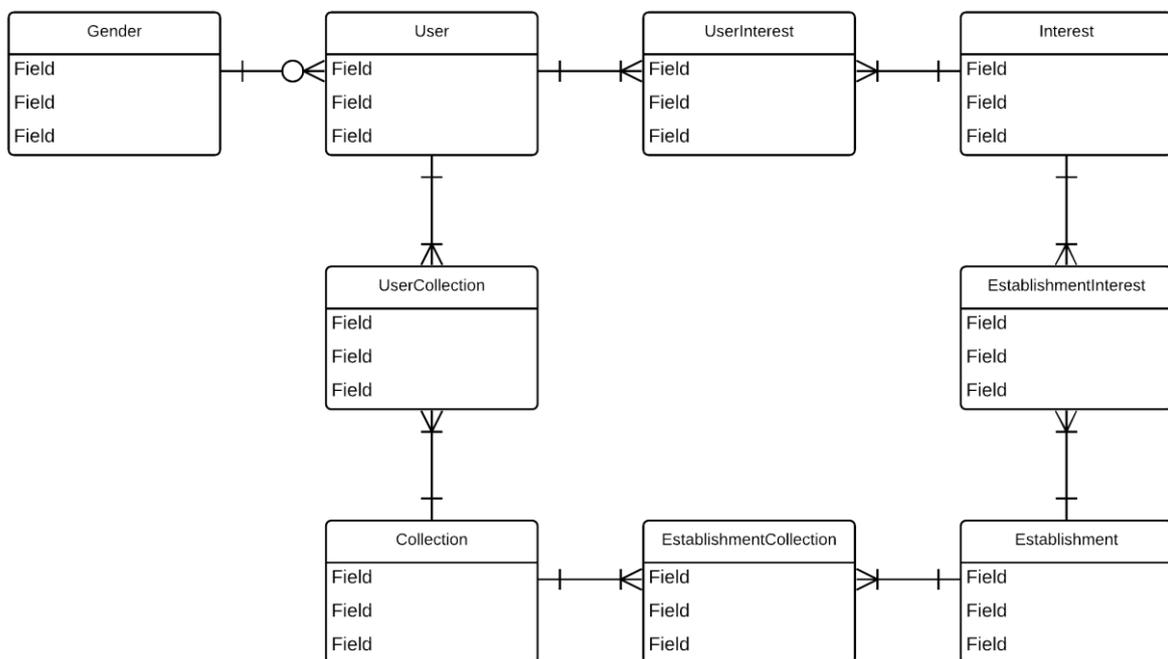


Figura 3: Diagrama de clases con la relación entre las preferencias del usuario y los establecimientos.

Los siguientes puntos se encuentran en el anexo VIII, sección 6.8.

Creación de una cuenta en AWS.

Creación y configuración del servicio RDS.

Creación del servicio S3

Creación del EC2

Integración de los servicios AWS con el backend

Lo primero que se realizó fue crear las variables de entorno en el servidor EC2 como se ve en la **Figura 4**.

```
export RECOMIENDO_DB_URL=xxxxxxx
export ASW_S3_ACCESS_KEY_ID=xxxxxxx
export AWS_S3_SECRET_ACCESS_KEY=xxxxxxx
export AWS_S3_REGION=xxxxxxx
export AWS_S3_BUCKET=xxxxxxx
```

Figura 4. Comandos para crear las variables de entorno en Linux.

- RECOMIENDO_DB_URL: es la URL de conexión a la base de datos alojada en el RDS.

Las siguientes variables son específicamente para la conexión con el contenedor S3.

- AWS_S3_ACCESS_KEY_ID: es el ID de clave de acceso.
- AWS_S3_SECRET_ACCESS_KEY: es la clave de acceso secreta.
- AWS_S3_REGION: es la región en la que fue creada el contenedor.
- AWS_S3_BUCKET: es el nombre del contenedor.

Luego, en el archivo de configuración *datastores.js* se cambió la URL por defecto para que sea obtenida de las variables de entorno como se ve en la **Figura 5**; **Error! No se encuentra el origen de la referencia.**

```
module.exports.datastores = {
  ...
  default: {
    adapter: 'sails-mysql',
    url: process.env.RECOMIENDO_DB_URL,
  },
};
```

Figura 5. Adaptador por defecto para la conexión a la base de datos.

Finalmente, en el archivo de configuración *custom.js* se agrega un nuevo valor con las credenciales de AWS para que igualmente sean obtenidas de las variables de entorno como se observa en la **Figura 6**.

```

module.exports.custom = {
  aws: {
    credentials: {
      accessKeyId: process.env.ASW_S3_ACCESS_KEY_ID,
      secretAccessKey: process.env.AWS_S3_SECRET_ACCESS_KEY,
      region: process.env.AWS_S3_REGION,
    },
    domain: 's3.amazonaws.com',
    bucket: process.env.AWS_S3_BUCKET,
  },
};

```

Figura 6. Configuración personalizada para tener acceso a las credenciales de AWS.

Endpoint para la creación o registro de usuarios

En este punto hay que destacar que, antes de crear al usuario con los datos solicitados, primero se sobrescribe su contraseña como se muestra en la **Figura 7**, esto para que en la base se almacene el *hash* de la contraseña y no la contraseña con el fin de incrementar la seguridad de los datos del usuario.

```

user.password = await sails.helpers.passwords.hashPassword(user.password);

```

Figura 7: Línea de código para sobrescribir la contraseña del usuario con el hash de esta.

También se toma en cuenta de que en caso de que haya ocurrido un error durante la creación del usuario se proceda a eliminar la imagen subida al S3 en caso de ser necesario.

Comunicación de la aplicación móvil con el backend

La aplicación se comunica con los servidores para intercambiar datos y lo hace consumiendo un *API RESTful*. La aplicación envía solicitudes HTTP a los servidores y estos responden a la aplicación en formato JSON o archivos de imagen. Una representación se muestra en la **Figura 8**.

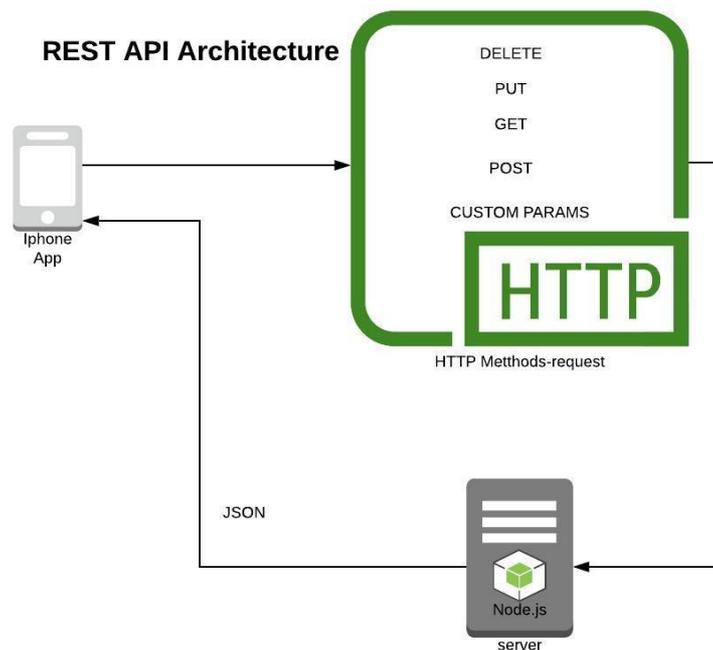


Figura 8. Arquitectura de la aplicación.

Uso del patrón de diseño modelo vista controlador (MVC)

El desarrollo de la aplicación se guía en el patrón de diseño MVC. Esto se logra creando: los modelos necesarios como son el modelo Usuario, el modelo Establecimiento, etcétera. Las vistas son las pantallas que visualiza el usuario y finalmente los controladores donde se realiza toda la lógica que implica a la vista y al modelo. En la estructura del proyecto de la aplicación móvil, como se muestra en la **Figura 9** se puede observar e identificar claramente cada uno de los conceptos de MVC aplicados.

Modelo -> Carpeta Models

Vista -> Carpeta Views

Controlador -> Carpeta ViewControllers

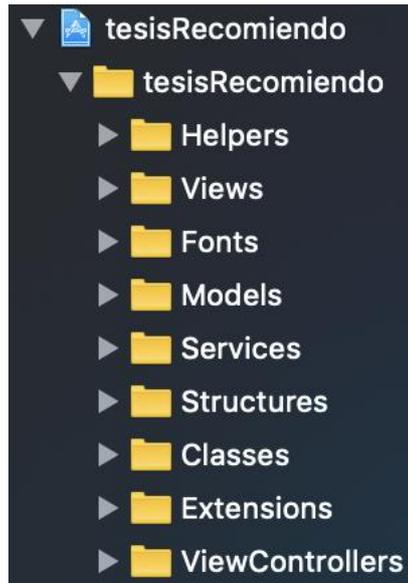


Figura 9. Estructura del proyecto.

2.2.3 Sprint Review

El objetivo del *Sprint* se alcanzó con facilidad gracias al fácil uso de las herramientas utilizadas.

Pruebas de aceptación

Tras revisar el trabajo realizado en el primer *Sprint*, en la **Tabla 7** se puede verificar que se cumplieron todos los criterios de aceptación de las historias de usuario.

Tabla 7: Pruebas de aceptación del sprint 1.

Historia de Usuario	Criterio de aceptación	Cumplido
HU1-1	La primera vez que el usuario ingrese al sistema deberá registrarse. En la pantalla de inicio de sesión tendrá un acceso Registrarse.	Sí
	Una vez que el usuario escoja Registrarse se abrirá una nueva pantalla con campos disponibles para registrar su nombre, apellido, género, fecha de nacimiento, dirección de correo electrónico y contraseña.	Sí
	En la misma pantalla de registro, el usuario tendrá la opción de escoger una imagen de su galería o tomar una nueva foto para usarla como imagen de perfil.	Sí

	La contraseña deberá ser mínimo de 6 caracteres y deberá contener por lo menos una letra y un número. En caso de que no se cumpla esto, la aplicación informará al usuario con una alerta.	Sí
--	--	----

Adaptación del Product Backlog

Tabla 8. Product Backlog adaptado tras finalizar el sprint 1.

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Estado
HUE1	HU1-1	Registrar usuarios.	20	Terminado
	HU2-1	Registrar preferencias de usuario.	7	Por implementar
	HU3-1	Inicio de sesión de usuarios.	7	Por implementar
	HU4-1	Navegación de usuario en la aplicación.	6	Por implementar
	HU5-1	Visualizar promociones, establecimientos y colecciones.	20	Por implementar
HUE2	HU1-2	Perfilar el establecimiento.	13	Por implementar
	HU2-2	Visualizar la ubicación de un establecimiento.	7	Por implementar
HUE3	HU1-3	Definición de promociones por establecimiento.	13	Por implementar
	HU1-3	Asignación de promociones.	7	Por implementar

2.2.4 Sprint Retrospective

En el **Gráfico 2** se observa el *Burndown chart* donde se presenta el progreso de trabajo realizado durante todo el sprint. En donde podemos visualizar que se inician con 27 tareas y se finaliza en el décimo día el cual está dentro del tiempo establecido.

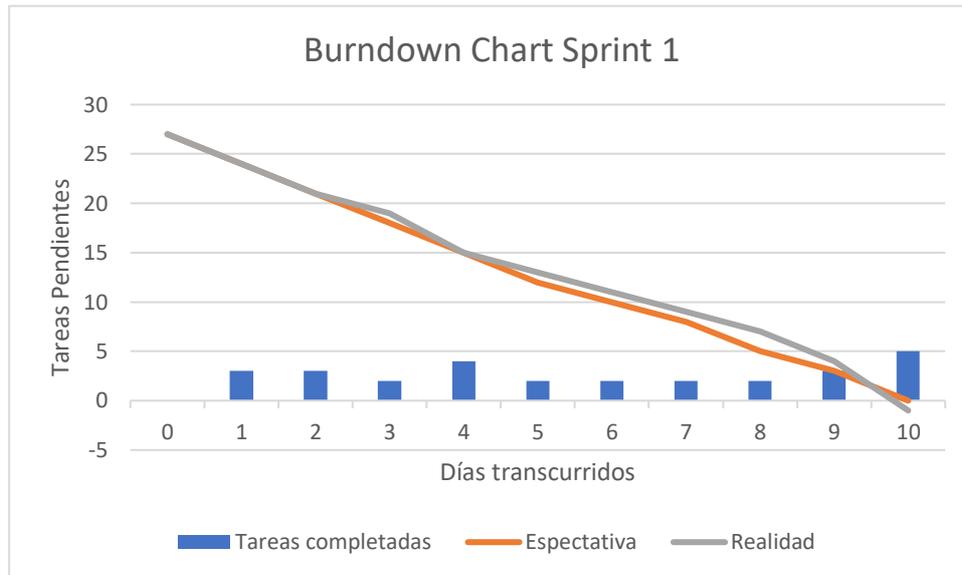


Gráfico 2. Burndown chart del sprint 1.

2.3 Sprint 2

2.3.1 Sprint Planning

Objetivo del Sprint

Crear pantallas de registro de preferencias del usuario, inicio de sesión y del menú de la aplicación para permitir al usuario registrar sus preferencias, iniciar sesión y visualizar las pantallas internas de la aplicación.

Historias de Usuario

Tabla 9. Historias de usuario escogidas para el sprint 2 (anexo IV, sección 6.4).

ID	Título	Esfuerzo	Estado
HU2-1	Registrar preferencias de usuario.	7	Por implementar

HU3-1	Inicio de sesión de usuarios.	7	Por implementar
HU4-1	Navegación de usuario en la aplicación.	6	Por implementar

Sprint Backlog

Tabla 10. Sprint backlog del sprint 2.

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
HU2-1	Crear el <i>endpoint</i> para registrar las preferencias del usuario.
	Verificar que las preferencias del usuario hayan sido creadas correctamente.
	Implementar el diseño de la pantalla para registro de preferencias del usuario.
	Implementar la funcionalidad de la pantalla de registro de preferencias del usuario.
HU3-1	Crear el <i>endpoint</i> para autenticar usuarios e iniciar sesión.
	Instalar el paquete <i>jsonwebtoken</i> para la autenticación de los usuarios.
	Verificar que la respuesta del servidor incluya los datos del usuario y un <i>token</i> válido.
	Implementar el diseño de la pantalla de inicio de sesión.
	Implementar la funcionalidad de la pantalla de inicio de sesión.
HU4-1	Implementar el diseño de la pantalla del menú de la aplicación.
	Implementar la funcionalidad de la pantalla del menú de la aplicación.
	Implementar el diseño de la pantalla con información sobre el sistema de recopuntos.
	Implementar la funcionalidad de la pantalla con información sobre el sistema de recopuntos.
	Implementar el diseño de la pantalla con los comentarios del usuario.
	Consultar todos los comentarios del usuario.
	Consultar las promociones de los establecimientos que son <i>sponsor</i> .
	Implementar la funcionalidad para editar los datos personales del

usuario.

2.3.2 Ejecución del Sprint

Durante la ejecución del segundo *sprint* se pueden rescatar los siguientes puntos importantes:

Endpoint para el registro de las preferencias del usuario

Las preferencias del usuario constan de intereses y colecciones. Las dos se registran de la misma forma. Primero, se eliminan todos los registros anteriores para después proceder con el registro de las nuevas preferencias. En la **Figura 10** se observa el registro de los intereses.

```
await UserInterest.destroy({ userId: inputs.userId });
const userInterests = inputs.interestIds.map(id => {
  | return { userId: inputs.userId, interestId: id };
});
await UserInterest.createEach(userInterests);
```

Figura 10. Registro de intereses del usuario.

Una vez completado el registro de las preferencias del usuario el servidor devuelve un mensaje de éxito en formato JSON siempre y cuando no haya habido errores. Caso contrario, devuelve el error correspondiente.

Validar que un usuario escoja mínimo un interés y una colección en la pantalla de registro de preferencias.

Para que el usuario pueda avanzar de la pantalla donde registra sus preferencias es imperativo que escoja por lo menos un interés y una colección para que de esta manera la aplicación pueda mostrarle contenido en la pantalla principal o *home*. En caso de no hacerlo, la aplicación le mostrará el mensaje correspondiente. En la **Figura 11** se observa esta validación.

```

fileprivate func registerInterestsAndCollections() {
    if selectedInterestIds.count == 0 && selectedCollectionIds.count == 0 {
        showError(errorMessage: "\ (String.customError.selectedInterests)
        | \n\ (String.customError.selectedCollections)")
        return
    }
    if selectedInterestIds.count == 0 {
        showError(errorMessage: String.customError.selectedInterests)
        return
    }
    if selectedCollectionIds.count == 0 {
        showError(errorMessage: String.customError.selectedCollections)
        return
    }
}

```

Figura 11. Validación para que el usuario escoja mínimo un interés y una colección.

2.3.3 Sprint Review

Pruebas de aceptación

Tabla 11. Pruebas de aceptación del sprint 2.

Historia de Usuario	Criterio de aceptación	Cumplido
HU2-1	Una vez que el usuario presiona Registrarse en la pantalla de registro se abrirá una nueva pantalla con la lista completa de intereses y colecciones disponibles.	Sí
	El usuario debe seleccionar por lo menos un interés y una colección. En caso de que esto no se cumpla, la aplicación informará al usuario con una alerta.	Sí
HU3-1	En la pantalla de inicio de sesión se mostrarán dos campos disponibles para que el usuario ingrese una dirección de correo electrónico y una contraseña.	Sí
	La dirección de correo electrónico ingresada debe ser una dirección válida, es decir, que se asemeje en estructura a una dirección de correo electrónico. Si se ingresa una dirección incorrecta, la aplicación informará al usuario con una alerta.	Sí
	Si la dirección de correo electrónico no existe o la contraseña es incorrecta, la aplicación informará al usuario con una alerta.	sí

HU4-1	En la pantalla principal el usuario tendrá disponible una barra de pestañas para navegar entre las pantallas: principal, buscar, perfil, promociones y últimos.	Sí
	En la pantalla de perfil, el usuario tendrá acceso a sus comentarios, a información sobre cómo ganar recopuntos y a configuraciones de su cuenta.	Sí
	Cuando el usuario acceda a la pantalla de sus comentarios la aplicación mostrará todos los comentarios del usuario y al final las promociones de los establecimientos que son <i>sponsor</i> .	Sí
	En la pantalla de información sobre recopuntos, cuando el usuario seleccione alguna de las opciones de información la aplicación deberá redirigirlo al perfil del último establecimiento que visitó.	Sí

Adaptación del Product Backlog

Tabla 12. Product Backlog adaptado tras finalizar el sprint 2.

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Estado
HUE1	HU1-1	Registrar usuarios.	20	Terminado
	HU2-1	Registrar preferencias de usuario.	7	Terminado
	HU3-1	Inicio de sesión de usuarios.	7	Terminado
	HU4-1	Navegación de usuario en la aplicación.	6	Terminado
	HU5-1	Visualizar promociones, establecimientos y colecciones.	20	Por implementar
HUE2	HU1-2	Perfilar el establecimiento.	13	Por implementar
	HU2-2	Visualizar la ubicación de un establecimiento.	7	Por implementar

HUE3	HU1-3	Definición de promociones por establecimiento.	13	Por implementar
	HU2-3	Asignación de promociones.	7	Por implementar

2.3.4 Sprint Retrospective

En el **Gráfico 3** se observa el *Burndown chart* donde se presenta el progreso de trabajo realizado durante todo el sprint. Se puede observar que las tareas propuestas se finalizaron sin ninguna novedad.

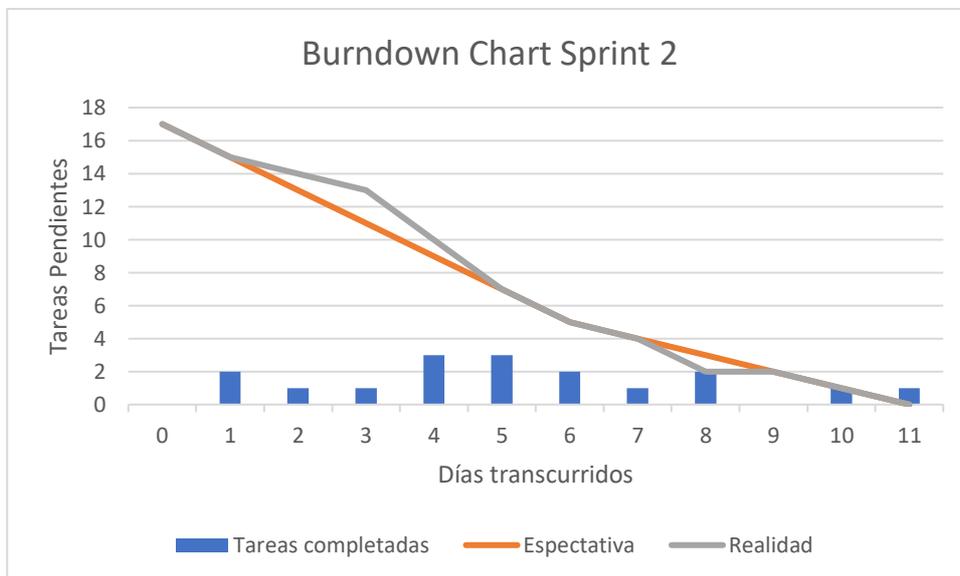


Gráfico 3. Burndown chart del sprint 2.

2.4 Sprint 3

2.4.1 Sprint Planning

Objetivo del Sprint

Desarrollar la pantalla principal con información de los establecimientos registrados, número de *recopuntos* y colecciones seleccionadas para mostrar al usuario información de acuerdo a sus preferencias.

Historias de Usuario

Tabla 13. Historias de usuario escogidas para el sprint 3 (anexo V, sección 0).

ID	Título	Esfuerzo	Estado
HU5-1	Visualizar promociones, establecimientos y colecciones.	7	Por implementar

Sprint Backlog

Tabla 14. Sprint backlog del sprint 3.

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
HU5-1	Crear el <i>endpoint</i> para obtener los establecimientos cercanos.
	Desarrollar el algoritmo para el cálculo de la distancia entre la ubicación del usuario y el establecimiento.
	Desarrollar el algoritmo para determinar qué establecimientos son los que devuelve el sistema.
	Validar que el cuerpo de la petición HTTP contenga los parámetros mínimos requeridos para la obtención de los establecimientos cercanos.
	Implementar el diseño de la pantalla principal para mostrar número de recopuntos del usuario.
	Implementar funciones para consultar el número de recopuntos con los que cuenta el usuario.
	Implementar el diseño para mostrar promociones de los establecimientos que son <i>sponsor</i> .
	Implementar funcionalidad para mostrar promociones de los establecimientos que son <i>sponsor</i> .
	Implementar diseño y funcionalidad para mostrar los establecimientos cercanos según las preferencias del usuario.
	Implementar diseño para mostrar las colecciones seleccionadas por el usuario.
	Implementar funcionalidad para mostrar colecciones seleccionadas por el usuario.

	Implementar diseño y funcionalidad del menú de la pantalla principal.
	Implementar ventana emergente para mostrar todas las colecciones disponibles.
	implementar ventana emergente para los establecimientos <i>premium</i> pertenecientes a cada colección disponible.

2.4.2 Ejecución del Sprint

Durante el tercer *sprint* se pueden rescatar los siguientes puntos:

Criterio para obtener los establecimientos cercanos

Con el fin de que el usuario de la aplicación móvil visualice los establecimientos que se encuentran cerca y que sean de su interés se tomaron en cuenta varias cosas. Primero, la solicitud HTTP realizada por la aplicación móvil contiene tres datos indispensables que son: el ID del usuario y su ubicación determinada por coordenadas geográficas. Luego, se estableció la distancia mínima en 5 Km.

Para determinar qué establecimientos visualizará el usuario se tomaron en cuenta sus preferencias que vienen dadas por los intereses y colecciones que haya seleccionado en el proceso de registro. De la misma manera, se toma en cuenta el perfil de cada establecimiento para saber a qué intereses y colecciones está asociado.

Validación de la petición HTTP

La realización de la validación fue relativamente sencilla gracias al *framework* Sails que nos permite indicar qué validaciones necesita cada parámetro de la petición como se observa en la **Figura 12**.

```

inputs: {
  userId: {
    type: 'number',
    required: true
  },
  latitude: {
    type: 'number',
    required: true,
    min: -(Math.atan(Math.sinh(Math.PI)) * 180 / Math.PI),
    max: Math.atan(Math.sinh(Math.PI)) * 180 / Math.PI
  },
  longitude: {
    type: 'number',
    required: true,
    min: -180,
    max: 180
  },
},
},

```

Figura 12. Validaciones de la petición HTTP para obtener los establecimientos cercanos.

De esta manera se verifica que:

- *userId*: es un número y es obligatorio.
- *latitude*: es un número, es requerido y tiene como valor mínimo y máximo 85.0511 aproximadamente con su signo respectivo.
- *longitude*: es un número, es obligatorio y tiene como valor mínimo y máximo 180 con su signo respectivo.

Criterio de selección de promociones de establecimientos *sponsor*

Como se mencionó en la sección **1.3.4** los establecimientos pueden adquirir el estado de *sponsor* lo que les permitirá mostrar sus promociones en la parte inicial de la pantalla principal como se muestra en la **Figura 13**.

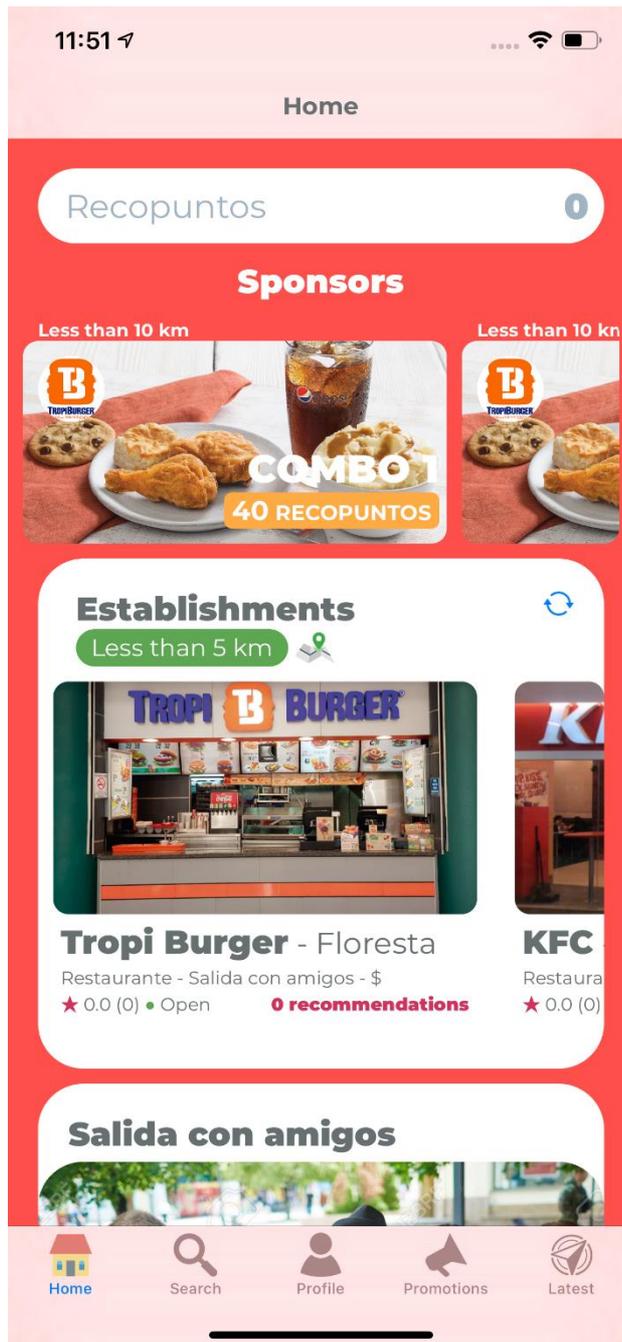


Figura 13. Pantalla principal.

A continuación, se tiene la sección de los establecimientos, estos están ubicados a una distancia de 5 km de la ubicación del usuario y consideran sus preferencias. En esta sección se muestra información general de los establecimientos.

Finalmente, se tiene la sección de las colecciones seleccionadas por el usuario la cual muestra el nombre de la colección con su respectiva imagen.

Justificación de la creación de las ventanas emergentes:

Las ventanas emergentes son para visualizar todas las colecciones existentes con el objetivo de mostrar al usuario más establecimientos, esta ventana emergente solo será presentada cuando la aplicación inicie.

La primera ventana emergente, como se ve en la **Figura 14**, es para presentar todas las colecciones que tiene la aplicación.

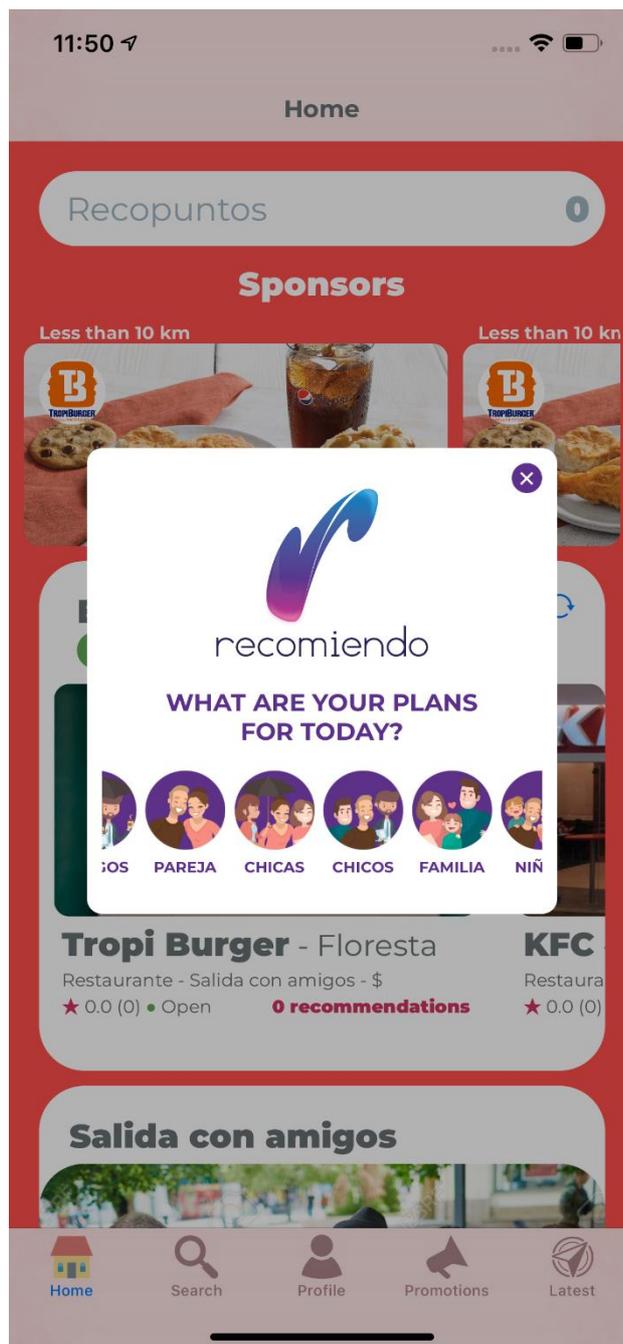


Figura 14. Ventana emergente con todas las colecciones.

En la **Figura 15** se muestra la segunda ventana emergente que aparece cuando el usuario selecciona una de las colecciones en la primera ventana emergente, aquí se visualizarán hasta un máximo de tres establecimientos que pertenezcan a dicha colección y tengan el estado *premium*.

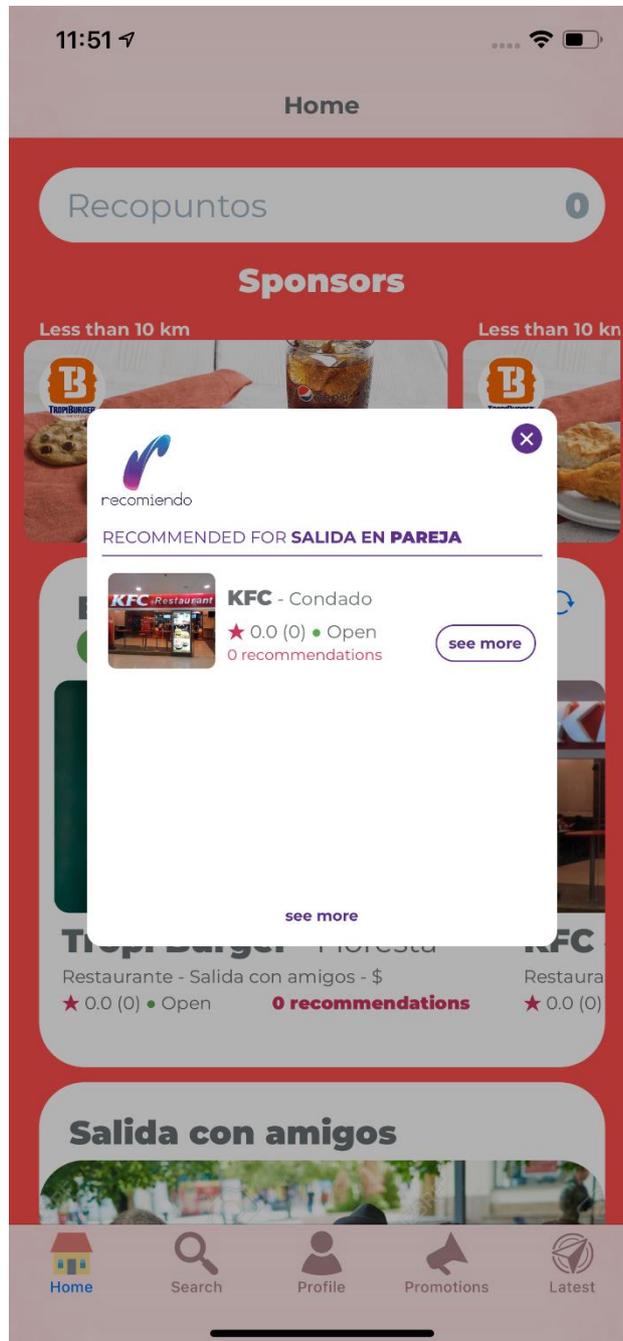


Figura 15. Ventana emergente con establecimientos *premium* de una colección seleccionada.

2.4.3 Sprint Review

Pruebas de aceptación

Tabla 15. Pruebas de aceptación del sprint 3.

Historia de Usuario	Criterio de aceptación	Cumplido
HU5-1	Después del registro, una vez que el usuario guarde sus intereses y colecciones se mostrará la pantalla principal. En futuras ocasiones la aplicación siempre se abrirá en esta pantalla.	Sí
	En la pantalla principal, en la parte superior se mostrará el número de recopuntos con los que cuenta el usuario.	Sí
	En la pantalla principal, debajo del número de recopuntos del usuario se mostrarán las promociones de los establecimientos que son <i>sponsor</i> .	Sí
	En la pantalla principal, debajo de las promociones se mostrarán los establecimientos cercanos según las preferencias del usuario.	Sí
	Los establecimientos que se mostrarán serán los que se encuentren en un radio de 5 km a la redonda respecto a la ubicación del usuario.	Sí
	Los establecimientos se mostrarán en orden descendente tomando en cuenta primero el número de recomendaciones que posee y luego su calificación.	Sí
	En la pantalla principal, debajo de los establecimientos se mostrarán las colecciones seleccionadas por el usuario.	Sí
	Cuando la aplicación se abra, en la pantalla principal se mostrará una ventana emergente con todas las colecciones disponibles.	Sí
	Si el usuario selecciona alguna de las colecciones mostradas en la ventana emergente, se abrirá una nueva ventana emergente y se mostrarán todos los establecimientos <i>premium</i> pertenecientes a la colección seleccionada.	sí

Adaptación del Product Backlog

Tabla 16. Product backlog adaptado tras finalizar el sprint 3.

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Estado
HUE1	HU1-1	Registrar usuarios.	<u>20</u>	Terminado
	HU2-1	Registrar preferencias de usuario.	<u>7</u>	Terminado
	HU3-1	Inicio de sesión de usuarios.	<u>7</u>	Terminado
	HU4-1	Navegación de usuario en la aplicación.	<u>6</u>	Terminado
	HU5-1	Visualizar promociones, establecimientos y colecciones.	<u>20</u>	Terminado
HUE2	HU1-2	Perfilar el establecimiento.	13	Por implementar
	HU2-2	Visualizar la ubicación de un establecimiento.	7	Por implementar
HUE3	HU1-3	Definición de promociones por establecimiento.	13	Por implementar
	HU2-3	Asignación de promociones.	7	Por implementar

2.4.4 Sprint Retrospective

En el **Gráfico 4** se observa el *Burndown chart* donde se presenta el progreso de trabajo realizado durante todo el sprint, en donde podemos visualizar que se inicia con un progreso mejor a la expectativa, pero a partir del tercer día se encuentran inconvenientes, pero al ser tan poco significantes se logra completar todas las tareas dentro del tiempo esperado.

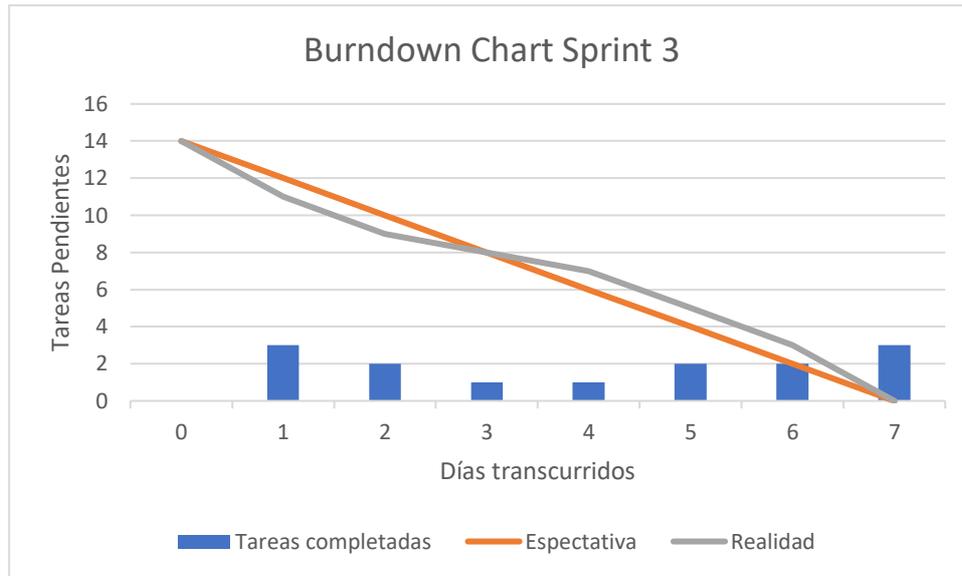


Gráfico 4. Burndown chart del sprint 3.

2.5 Sprint 4

2.5.1 Sprint Planning

Objetivo del Sprint

Desarrollar las pantallas del perfil del establecimiento para mostrar su información y su ubicación en un mapa para que el usuario pueda llegar al establecimiento de manera fácil.

Historias de Usuario

Tabla 17. Historias de usuario escogidas para el sprint 4 (anexo VI, sección 0).

ID	Título	Esfuerzo	Estado
HU1-2	Perfilar el establecimiento.	13	Por implementar
HU2-2	Visualizar la ubicación de un establecimiento.	7	Por implementar

Sprint Backlog

Tabla 18. Sprint backlog del sprint 4.

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
HU1-2	Crear el <i>endpoint</i> para calificar un establecimiento.
	Desarrollar el algoritmo para calificar un establecimiento.
	Crear el <i>endpoint</i> para recomendar un establecimiento.
	Desarrollar el algoritmo para recomendar un establecimiento.
	Implementar diseño para listar productos y/o servicios que ofrece el establecimiento.
	Implementar funcionalidad para listar productos y/o servicios que ofrece el establecimiento.
	Implementar diseño para ver, crear o eliminar comentarios en el perfil del establecimiento.
	Implementar funcionalidad para ver, crear o eliminar comentarios en el perfil del establecimiento.
	Implementar diseño para mostrar la galería del establecimiento.
	Implementar funcionalidad para mostrar la galería del establecimiento.
	Implementar diseño para mostrar la información del establecimiento como: nombre, calificación, si se encuentra abierto, a qué tipo pertenece, su colección principal, estimación de costo y dirección.
	Implementar diseño para recomendar el establecimiento mediante WhatsApp y Facebook.
	Implementar diseño para calificar el establecimiento.
	Implementar funcionalidad para calificar el establecimiento.
HU2-2	Configurar el API Directions de Google Maps.
	Mostrar ubicación del establecimiento con un ícono personalizado.
	Generar la ruta desde la ubicación del usuario hasta la del establecimiento.

2.5.2 Ejecución del Sprint

Durante el cuarto *sprint* se pueden rescatar los siguientes puntos:

Algoritmo para calificar a un establecimiento

En el algoritmo de la **Figura 16** primero se consulta cuántos recopuntos ganaría el usuario al calificar un establecimiento. El usuario gana recopuntos únicamente la primera vez que califica, en caso de que ya haya calificado al establecimiento que desea, solo se actualizará su calificación. En caso de que sea la primera vez que califica a un establecimiento se genera un registro y se aumentan los respectivos recopuntos al usuario.

```
fn: async inputs => {
  const settings = await Settings.find();
  const setting = settings[0];
  const earnedRecopuntos = setting.ratingRecopuntos;
  const customerRating = await CustomerRating.findOne({
    userId: inputs.userId,
    establishmentId: inputs.establishmentId,
  });
  if (customerRating) {
    let updatedCustomerRating = await CustomerRating.updateOne({ id: customerRating.id })
      .set({ rating: inputs.rating });
    updatedCustomerRating.earnedRecopuntos = 0;
    return updatedCustomerRating;
  }
  const user = await User.findOne({ id: inputs.userId });
  const newCustomerRating = await CustomerRating.create({
    rating: inputs.rating,
    earnedRecopuntos,
    userId: inputs.userId,
    establishmentId: inputs.establishmentId,
  }).fetch();
  await User.update({ id: inputs.userId })
    .set({ recopuntos: user.recopuntos + earnedRecopuntos });
  return newCustomerRating;
}
```

Figura 16. Algoritmo para calificar a un establecimiento.

Algoritmo para recomendar un establecimiento

Un usuario puede recomendar un establecimiento mediante dos medios que serán Facebook y WhatsApp. La recompensa de recopuntos dependerá del medio utilizado. Existe un tiempo de espera después del cual el usuario podrá ganar nuevamente recopuntos por recomendar el mismo establecimiento mediante el mismo medio.

En el algoritmo de la **Figura 17** primero se procede a consultar la última recomendación del usuario al establecimiento por el medio especificado por el mismo, esto para verificar el tiempo mínimo de espera y proceder o no con la asignación de recopuntos al usuario. Luego, se genera un registro de la recomendación y se actualizan los recopuntos del usuario.

```
fn: async inputs => {
  const myLastRecommendation = await Recommendation.find({
    where: { userId: inputs.userId, establishmentId: inputs.establishmentId, through: inputs.through },
    limit: 1,
    sort: 'createdAt DESC'
  });
  const settings = await Settings.find();
  let recopuntos = settings[0][inputs.through + 'Recopuntos'];
  if (myLastRecommendation.length > 0) {
    const waitingTime = settings[0][inputs.through + 'WaitingTime'];
    const lastRecommendationTime = new Date().getTime() - myLastRecommendation[0].createdAt;
    if (lastRecommendationTime < waitingTime) recopuntos = 0;
  }
  await Recommendation.create({
    through: inputs.through,
    options: inputs.options.join(', '),
    earnedRecopuntos: recopuntos,
    userId: inputs.userId,
    establishmentId: inputs.establishmentId,
  });
  const user = await User.findOne({ id: inputs.userId });
  const updatedUser = await User.updateOne({ id: inputs.userId })
    .set({ recopuntos: user.recopuntos + recopuntos });
  let recommendationsAmount = await Recommendation.count({ establishmentId: inputs.establishmentId });
  return { userRecopuntos: updatedUser.recopuntos, recommendationsAmount };
}
```

Figura 17. Algoritmo para recomendar a un establecimiento.

Descripción de la implementación de la pantalla de perfil del establecimiento:

La **Figura 18** muestra la pantalla de perfil del establecimiento donde el usuario observará toda la información disponible de ese establecimiento como puede ser: el nombre, la dirección, el tipo de establecimiento, por ejemplo si es un restaurante o una discoteca, la colección con la que se identifica, la referencia de precio, la cantidad de veces que los usuarios han visitado ese perfil, la cantidad de recomendaciones que tiene, la calificación y más. De la misma manera podrá realizar diferentes acciones como calificar, recomendar o dejar un comentario.

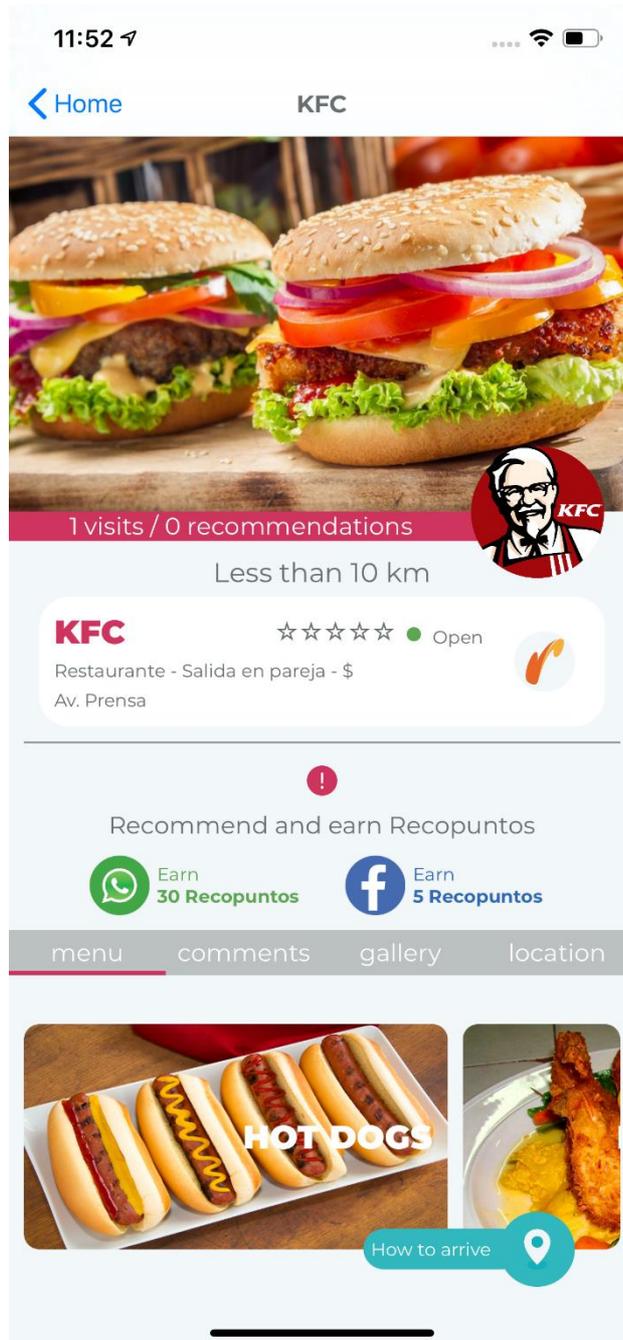


Figura 18. Perfil del establecimiento.

En la implementación de la pantalla del perfil del establecimiento se tuvieron los siguientes puntos relevantes:

- Menú: en esta sección se visualizan los servicios que brinda el establecimiento mediante una imagen representativa como se ve en la **Figura 19**. Además, cuenta con un botón que le indicará al usuario la ubicación del establecimiento en una nueva pantalla.

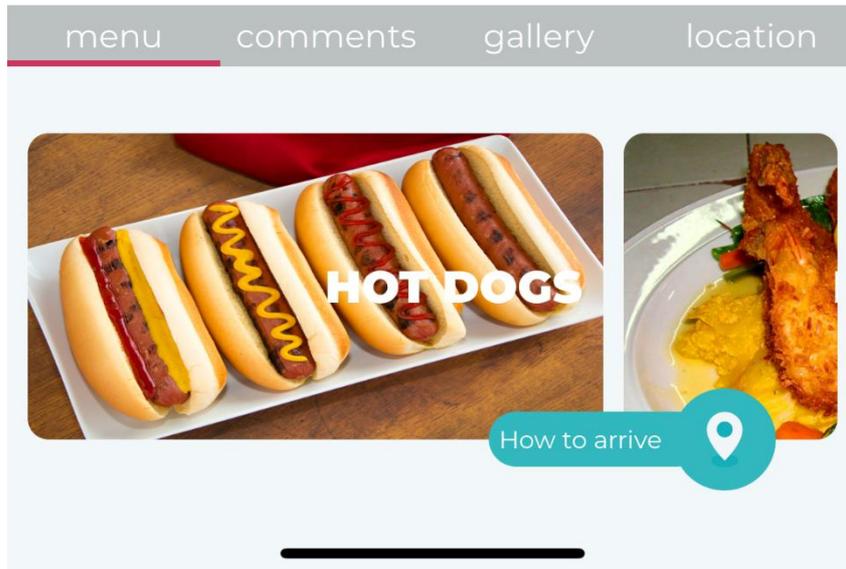


Figura 19. Menú del establecimiento.

- Comentarios: en la sección de comentarios el usuario podrá ver comentarios del establecimiento y realizar o borrar sus comentarios.

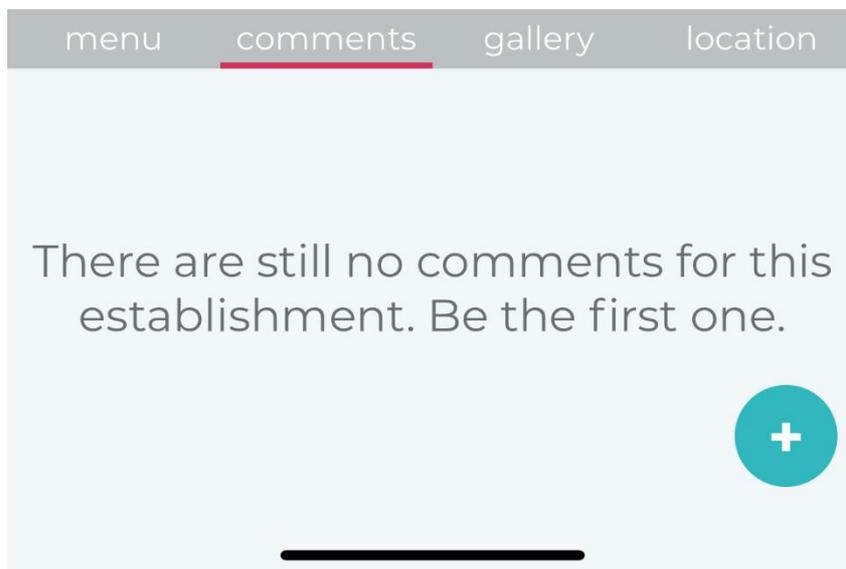


Figura 20. Comentarios del establecimiento (sección vacía).

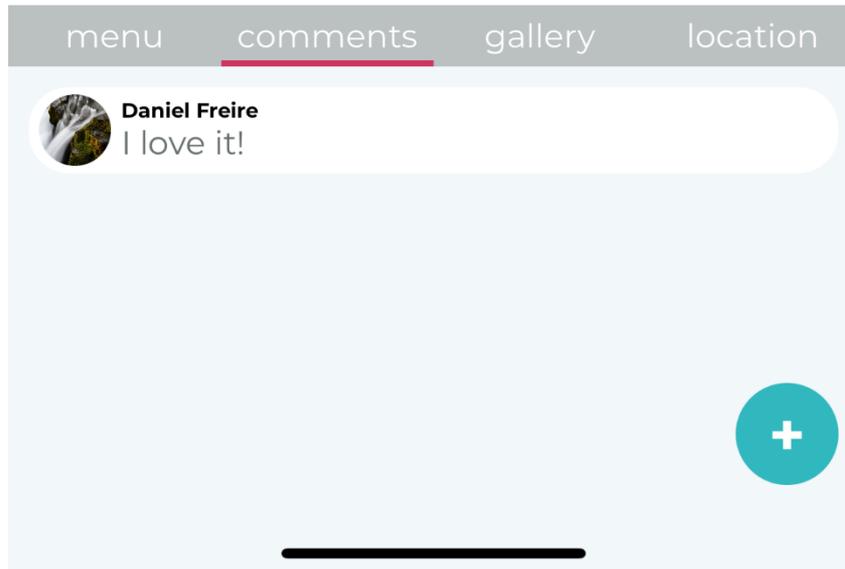


Figura 21. Comentarios del establecimiento.

- Galería: en esta sección puede visualizar fotos publicadas por el establecimiento como se ve en la **Figura 22**.

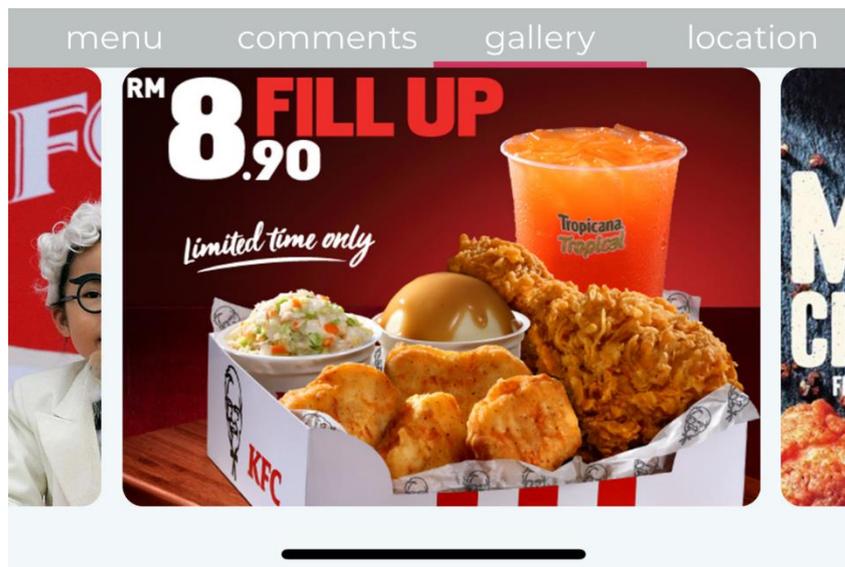


Figura 22. Galería del establecimiento.

- Ubicación: en esta sección podrá ver la ubicación del establecimiento en un mapa y, al presionar el botón *show*, generar la ruta desde la ubicación del usuario hasta la del establecimiento. Adicionalmente, si el usuario desea una ruta guiada puede presionar el botón *GO* y se abrirá la aplicación de Google Maps y en caso de que no la tenga instalada se abrirá un mapa en el navegador.

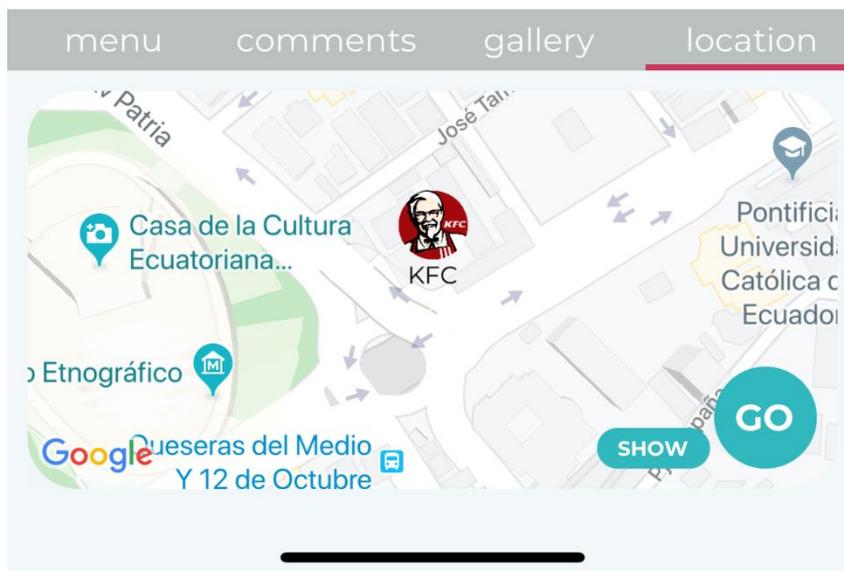


Figura 23. Ubicación del establecimiento.

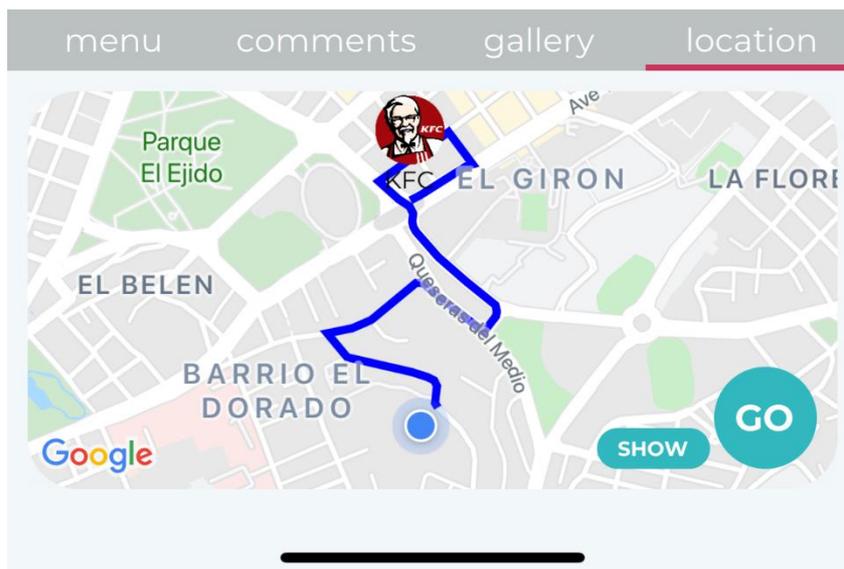


Figura 24. Ruta dibujada hacia el establecimiento.

- Ganar recopuntos: desde el perfil del establecimiento el usuario podrá recomendarlo a través de Facebook y WhatsApp para ganar recopuntos.
- Signo de dólar "\$": el signo de dólar ubicado en la sección de datos del establecimiento es para representar los precios de sus productos o servicios de la siguiente manera:
 - \$ = bajos.
 - \$\$ = promedio.
 - \$\$\$ = altos.

Configuración del API Directions de Google Maps.

Para implementar el mapa provisto por Google en la aplicación hay que instalar la dependencia GoogleMaps lo que nos permite usar el SDK de Google Maps para iOS. Esto se realizó utilizando el administrador de dependencia CocoaPods. Luego se generó un API key que es necesaria para el uso de los mapas. Posteriormente, lo que se requirió fue mostrar la ruta desde la ubicación del usuario hasta la del establecimiento y para lograr esto fue necesario utilizar el API Directions de Google Maps.

```
@objc func getDirections(sender: UIButton) {
    guard
        let location = locationManager.location?.coordinate,
        let latitude = establishment?.latitude,
        let longitude = establishment?.longitude
    else { return }

    let origin = "\(location.latitude), \(location.longitude)"
    let destination = "\(latitude), \(longitude)"
    let key = "API_KEY"
    let start = "https://maps.googleapis.com/maps/api/directions/json?"
    let urlString = "\(start)origin=\(origin)&destination=\(destination)&key=\(key)"

    let polyline = GMSPolyline()

    mapView.setDirections(from: urlString, for: polyline)
}
```

Figura 25. Algoritmo para obtener la ruta hacia un establecimiento específico.

En la **Figura 25** se muestra el procedimiento realizado para obtener la ruta hacia el establecimiento. Primero se obtiene la ubicación del usuario dada por la latitud y longitud y se almacena en la variable llamada *origin*. Luego se obtiene la ubicación del establecimiento, igualmente dada por la latitud y longitud, se almacena en la variable llamada *destination*. Conociendo la ubicación de los dos puntos y con el API key ya se puede generar la URL que se usará para consumir el API Directions.

2.5.3 Sprint Review

Pruebas de aceptación

Tabla 19. Pruebas de aceptación del sprint 4.

Historia de Usuario	Criterio de aceptación	Cumplido
HU1-2	Después del registro, una vez que el usuario guarde sus intereses y colecciones se mostrará la pantalla principal. En futuras ocasiones la aplicación siempre se abrirá en esta pantalla.	Sí
	En la pantalla principal, en la parte superior se mostrará el número de recopuntos con los que cuenta el usuario.	Sí
	En la pantalla principal, debajo del número de recopuntos del usuario se mostrarán las promociones de los establecimientos que son <i>sponsor</i> .	Sí
	En la pantalla principal, debajo de las promociones se mostrarán los establecimientos cercanos según las preferencias del usuario.	Sí
	Los establecimientos que se mostrarán serán los que se encuentren en un radio de 5 km a la redonda respecto a la ubicación del usuario.	Sí
	Los establecimientos se mostrarán en orden descendente tomando en cuenta primero el número de recomendaciones que posee y luego su calificación.	Sí
	En la pantalla principal, debajo de los establecimientos se mostrarán las colecciones seleccionadas por el usuario.	Sí
	Cuando la aplicación se abra, en la pantalla principal se mostrará una ventana emergente con todas las colecciones disponibles.	Sí
	Si el usuario selecciona alguna de las colecciones mostradas en la ventana emergente, se abrirá una nueva ventana emergente y se mostrarán todos los establecimientos <i>premium</i> pertenecientes a la colección seleccionada.	Sí
HU2-2	El usuario podrá ver la ubicación del establecimiento en	Sí

	un mapa. En la pantalla del perfil del establecimiento habrá un acceso ¿Cómo llegar?	
	El usuario podrá ver la ruta marcada en el mapa desde su ubicación hasta la del establecimiento al seleccionar el botón Mostrar.	Sí

Adaptación del Product Backlog

Tabla 20. Product backlog adaptado tras finalizar el sprint 4.

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Estado
HUE1	HU1-1	Registrar usuarios.	20	Terminado
	HU2-1	Registrar preferencias de usuario.	7	Terminado
	HU3-1	Inicio de sesión de usuarios.	7	Terminado
	HU4-1	Navegación de usuario en la aplicación.	6	Terminado
	HU5-1	Visualizar promociones, establecimientos y colecciones.	20	Terminado
HUE2	HU1-2	Perfilar el establecimiento.	13	Terminado
	HU2-2	Visualizar la ubicación de un establecimiento.	7	Terminado
HUE3	HU1-3	Definición de promociones por establecimiento.	13	Por implementar
	HU2-3	Asignación de promociones.	7	Por implementar

2.5.4 Sprint Retrospective

En el **Gráfico 5** se observa el *Burndown chart* donde se presenta el progreso de trabajo realizado durante todo el sprint, en donde podemos visualizar que entre el día 2 y 12 la curva se aleja de lo esperado, pero en los días 13 y 14 al realizar más de una tarea al día se logra el objetivo dentro del tiempo esperado.

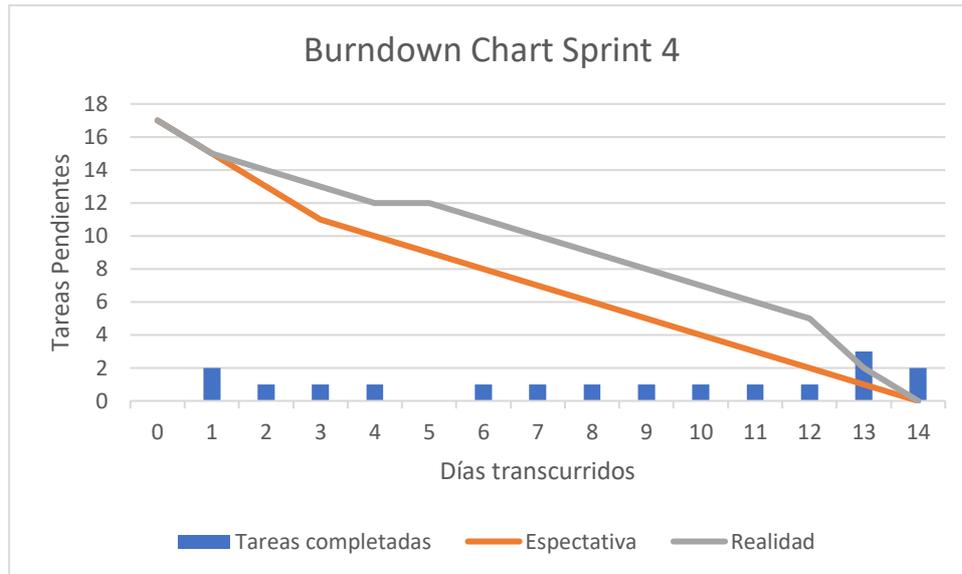


Gráfico 5. Burndown chart del sprint 4.

2.6 Sprint 5

2.6.1 Sprint Planning

Objetivo del Sprint

Desarrollar las pantallas de visualización y canjeo de promociones para que el usuario pueda ver y adquirir productos a menor precio.

Historias de Usuario

Tabla 21. Historias de usuario escogidas para el sprint 3 (anexo VII, sección 6.7).

ID	Título	Esfuerzo	Estado
HU1-3	Creación de la pantalla con la lista de promociones disponibles.	13	Por implementar
HU2-3	Asignación de promociones	7	Por implementar

Sprint Backlog

Tabla 22. Sprint backlog del sprint 5.

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
HU1-3	Crear el <i>endpoint</i> para canjear recopuntos.
	Desarrollar el algoritmo para canjear recopuntos.
	Consultar los establecimientos cercanos que empaten con las preferencias del usuario preferencias.
	Consultar las promociones de cada uno de los establecimientos cercanos.
	Implementar la funcionalidad que permita al usuario navegar por las promociones de los diferentes establecimientos.
	Implementar la funcionalidad que permita al usuario ver un mapa con la ubicación del establecimiento sin tener que salir de la pantalla de promociones.
HU2-3	Implementar la funcionalidad para generar un código QR a partir del código único de la promoción.
	Implementar el diseño de la pantalla que contiene toda la información de una promoción.
	Implementar la funcionalidad que haga que, después de cierto tiempo, se cierre la pantalla con el código QR generado.

2.6.2 Ejecución del Sprint

Durante el quinto *sprint* se pueden rescatar los siguientes puntos:

Algoritmo para canjear recopuntos

Todos los establecimientos pueden crear promociones que los usuarios podrán visualizar en la aplicación. Estas promociones podrán ser reclamadas directamente en los diferentes establecimientos una vez generado el cupón para el usuario y que este haya

intercambiado sus recopuntos por la promoción. Las promociones serán limitadas por un número máximo de cupones que se pueden generar y por una fecha de caducidad. Si el número máximo de cupones es alcanzado, la promoción pasará a un estado de inactiva e igualmente si se alcanza la fecha de caducidad de esta.

```
fn: async inputs => {
  const user = await User.findOne({ id: inputs.userId });
  const promotion = await Promotion.findOne({
    id: inputs.promotionId,
    isActive: true,
  });
  if (!promotion)
    throw { unavailablePromotion: { message: 'This promotion is no longer available.' } };
  if (user.recopuntos < promotion.recopuntos) {
    throw {
      notEnoughRecopuntos: {
        message: 'You don\'t have enough recopuntos.',
        data: { recopuntos: user.recopuntos },
      }
    };
  }
  const recopuntos = user.recopuntos - promotion.recopuntos;
  await User.update({ id: inputs.userId }).set({ recopuntos });
  const numberOfCoupons = promotion.numberOfCoupons - 1;
  let promotionData = { numberOfCoupons };
  if (numberOfCoupons === 0) promotionData.isActive = false;
  await Promotion.update({ id: inputs.promotionId }).set(promotionData);
  await Coupon.create({ userId: inputs.userId, promotionId: inputs.promotionId });
  return {
    message: 'Recopuntos exchanged.',
    data: { recopuntos, numberOfCoupons },
  };
}
```

Figura 26. Algoritmo para canjear recopuntos.

En el algoritmo de la **Figura 26** primero se procede a consultar si la promoción en la que el usuario quiere gastar sus recopuntos se encuentra activa, caso contrario se arroja una excepción indicándole que la promoción ya no se encuentra disponible. Posteriormente, se verifica que el usuario tenga suficientes recopuntos para canjear la promoción, caso contrario se arroja una excepción indicándole que no tiene suficientes recopuntos. Si todo procede correctamente se sustraen los recopuntos del usuario, se resta el número de cupones disponibles de la promoción y en caso de que el número de cupones sea 0 se procede a inactivar la promoción. Finalmente, se genera el registro del cupón y se retorna un mensaje y datos relevantes para la aplicación móvil.

Criterio para obtener las promociones

Las promociones dependen directamente de los establecimientos, entonces lo primero que se hace es consultar los establecimientos cercanos que empatan con las preferencias del usuario, por lo que este visualizará una lista de establecimientos para posteriormente obtener las promociones del establecimiento que seleccione.

Generación del código QR

El SDK de iOS facilitó enormemente la generación del código QR ya que cuenta con un generador de código QR incorporado que a partir de un texto nos genera una imagen. Su implementación se puede observar en la **Figura 27**.

```
func setImageFromStringCode(stringCode: String) {
    let data = stringCode.data(using: String.Encoding.ascii)
    guard let filter = CIFilter(name: "CIQRCodeGenerator") else { return }
    filter.setValue(data, forKey: "inputMessage")
    let transform = CGAffineTransform(scaleX: 5, y: 5)
    guard let output = filter.outputImage?.transformed(by: transform) else { return }
    image = UIImage(ciImage: output)
    stopLoading(in: loadingView)
}
```

Figura 27. Algoritmo para generar una imagen que contiene un código QR.

Algoritmo para cerrar la pantalla con el código QR generado

Una vez que el usuario acepte y esté seguro de gastar sus recopuntos en la promoción se le presentará una pantalla con un código QR generado a partir del código único de la promoción el cual deberá presentar en el local correspondiente en el tiempo que le indique la pantalla. Esta pantalla es únicamente informativa.

El tiempo en el que se cerrará la pantalla puede ser configurado por el administrador, al momento se encuentra establecido en 5 minutos y viene dado en milisegundos. En el algoritmo de la **Figura 28** se puede observar cómo obtenemos este tiempo y lo convertimos a minutos para posteriormente con la ayuda de contadores simular la cuenta regresiva. Cuando la cuenta llegue a 0 la pantalla se cerrará.

```

fileprivate func setupRemainingTimeValue() {
    var time = Helpers.getMinutesFromMilliseconds(value: remainingTime)
    let _ = Timer.scheduledTimer(withTimeInterval: 1, repeats: true) { [weak self] timer in
        self?.remainingTimeValueLabel.text =
            "\(time.minutes > 0 ?
            "\(time.minutes):\(time.seconds < 10 ? "0\(time.seconds)" : "\(time.seconds)" :
            "\(time.seconds)"")"
        if time.minutes == 0 && time.seconds == 0 {
            timer.invalidate()
            self?.navigationController?.popViewController(animated: true)
        }
        if time.seconds == 0 {
            time.minutes -= 1
            time.seconds = 60
        }
        time.seconds -= 1
    }
}
}

```

Figura 28. Algoritmo para cerrar la pantalla con el código QR generado.

2.6.3 Sprint Review

Pruebas de aceptación

Tabla 23. Pruebas de aceptación del sprint 5.

Historia de Usuario	Criterio de aceptación	Cumplido
HU1-3	El usuario podrá ver la ubicación del establecimiento en un mapa sin tener que ingresar al perfil de este. Cada promoción contará con un acceso ¿Cómo llegar?	Sí
	Una vez que el usuario escoja ¿Cómo llegar? se abrirá una pantalla que muestre el número de recopuntos del usuario y un mapa con la ubicación del establecimiento.	Sí
	En la pantalla de promociones el usuario visualizará una lista de establecimientos. Cuando seleccione uno se desplegará una sublista con las promociones de este y podrá ver el precio en dinero y recopuntos de la promoción.	Sí
	Si el usuario al intentar reclamar una promoción no tiene suficientes recopuntos la aplicación mostrará una alerta.	Sí
	Si el número de cupones disponibles en la promoción se agotan luego de la última transacción, la promoción se deshabilita y si un usuario quiere canjearla la aplicación	Sí

	muestra una alerta.	
HU2-3	La aplicación generará un código QR una vez que el usuario reclame una promoción en la pantalla de promociones.	Sí
	El usuario al ingresar en la pantalla de la promoción podrá ver toda la información referente a esta.	Sí
	La pantalla con el código QR e información de la promoción mostrará al usuario un temporizador de 5 minutos que cuando llegue a cero cerrará la pantalla.	Sí

Adaptación del Product Backlog

Tabla 24. Product backlog adaptado tras finalizar el sprint 5.

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Estado
HUE1	HU1-1	Registrar usuarios.	20	Terminado
	HU2-1	Registrar preferencias de usuario.	7	Terminado
	HU3-1	Inicio de sesión de usuarios.	7	Terminado
	HU4-1	Navegación de usuario en la aplicación.	6	Terminado
	HU5-1	Visualizar promociones, establecimientos y colecciones.	20	Terminado
HUE2	HU1-2	Perfilar el establecimiento.	13	Terminado
	HU2-2	Visualizar la ubicación de un establecimiento.	7	Terminado
HUE3	HU1-3	Definición de promociones por establecimiento.	13	Terminado
	HU2-3	Asignación de promociones.	7	Terminado

2.6.4 Sprint Retrospective

En el **Gráfico 6** se observa el *Burndown chart* donde se presenta el progreso de trabajo realizado durante todo el *sprint*. Se puede observar que el avance es lo esperado.

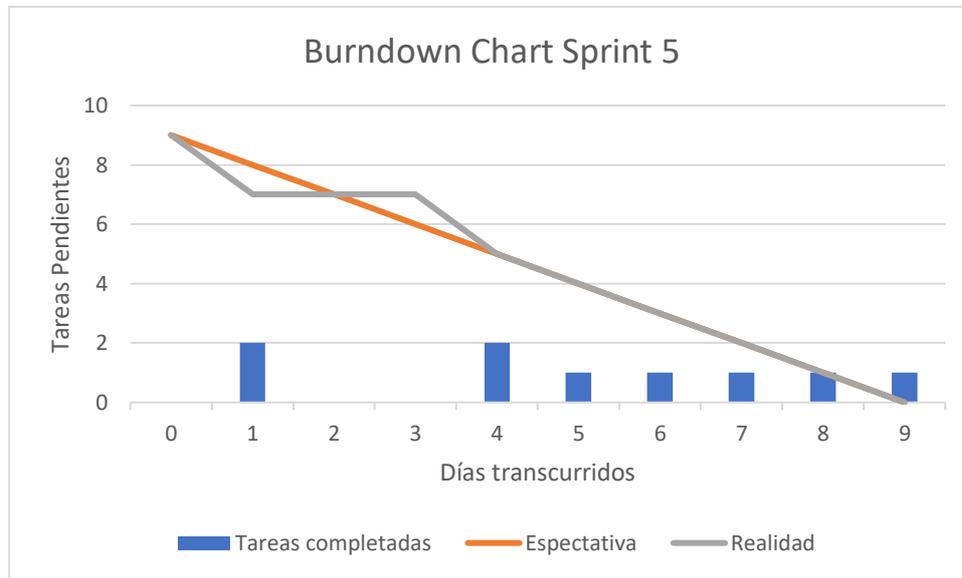


Gráfico 6. Burndown chart del sprint 5.

3 RESULTADOS Y DISCUSIÓN

3.1 Producto Final

El producto final del actual proyecto es la aplicación prototipo desarrollada para la plataforma iOS. La aplicación funciona como se esperaba, existen establecimientos que el usuario puede recomendar, el usuario gana recopuntos cuando recomienda un establecimiento y finalmente el usuario puede gastar estos recopuntos en las promociones de los diferentes establecimientos.

3.1.1 Recomendar un Establecimiento

En el perfil del establecimiento se puede observar las visitas y las recomendaciones del establecimiento, como se puede ver en la **Figura 29** en primera instancia el establecimiento tiene 0 recomendaciones.

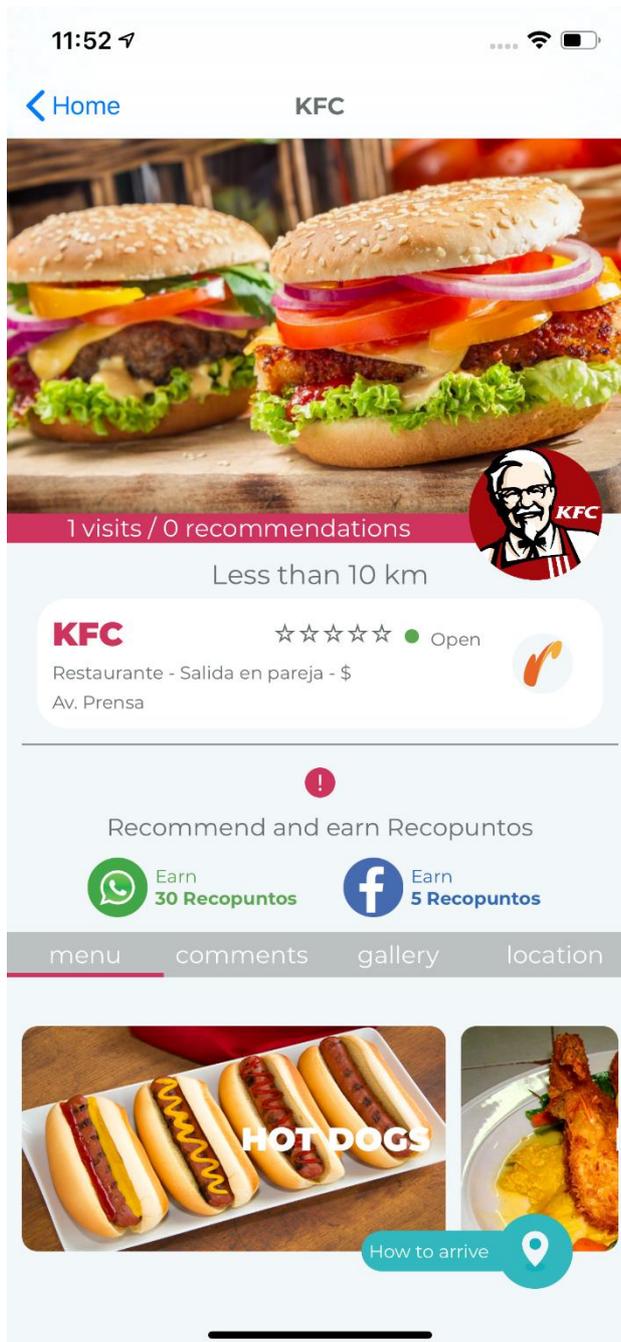


Figura 29. Perfil del establecimiento con cero recomendaciones.

Al seleccionar un medio para recomendar, se muestran las opciones que se pueden recomendar del establecimiento como se puede observar en la **Figura 30**.

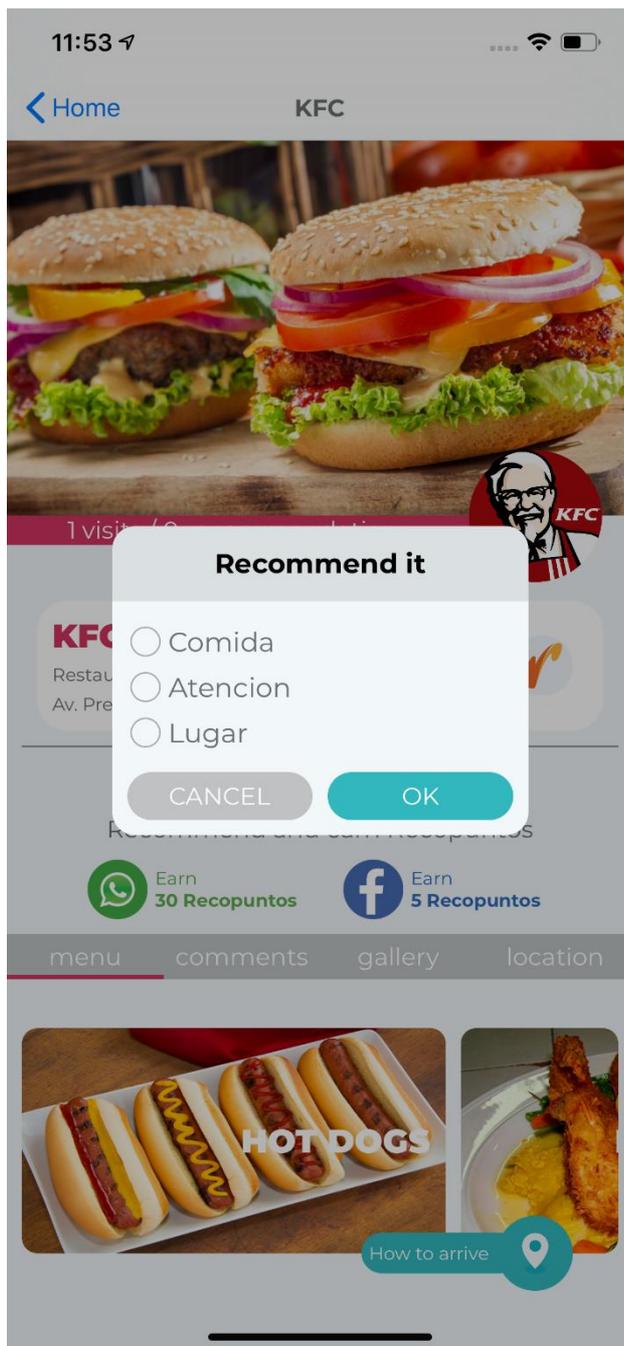


Figura 30. Opciones para recomendar del establecimiento.

Luego de realizar la recomendación se puede ver en la **Figura 31** como el establecimiento ya cuenta con una recomendación.

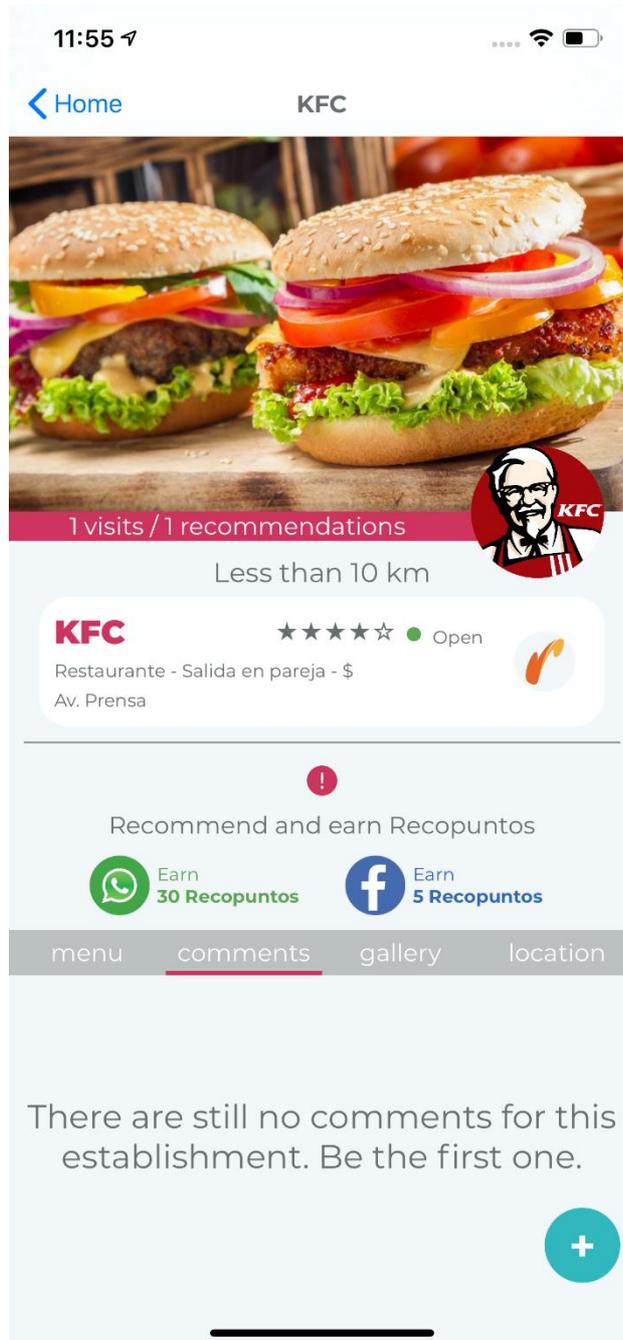


Figura 31. Perfil del establecimiento con una recomendación.

Finalmente, uno de los lugares en los que se puede ver cuántos recopuntos se tiene es en la pantalla del perfil de usuario. En la **Figura 32** se puede observar que el usuario ya cuenta con 50 recopuntos.



Figura 32. Cantidad de recopuntos en el perfil de usuario.

3.1.2 Canjear los recopuntos por una promoción

Al ingresar a la pantalla de las promociones como se ve en la **Figura 33** el usuario visualiza con cuántos recopuntos cuenta y una lista de los establecimientos cercanos, cada uno indicando la cantidad de cupones de las promociones con los que cuenta.

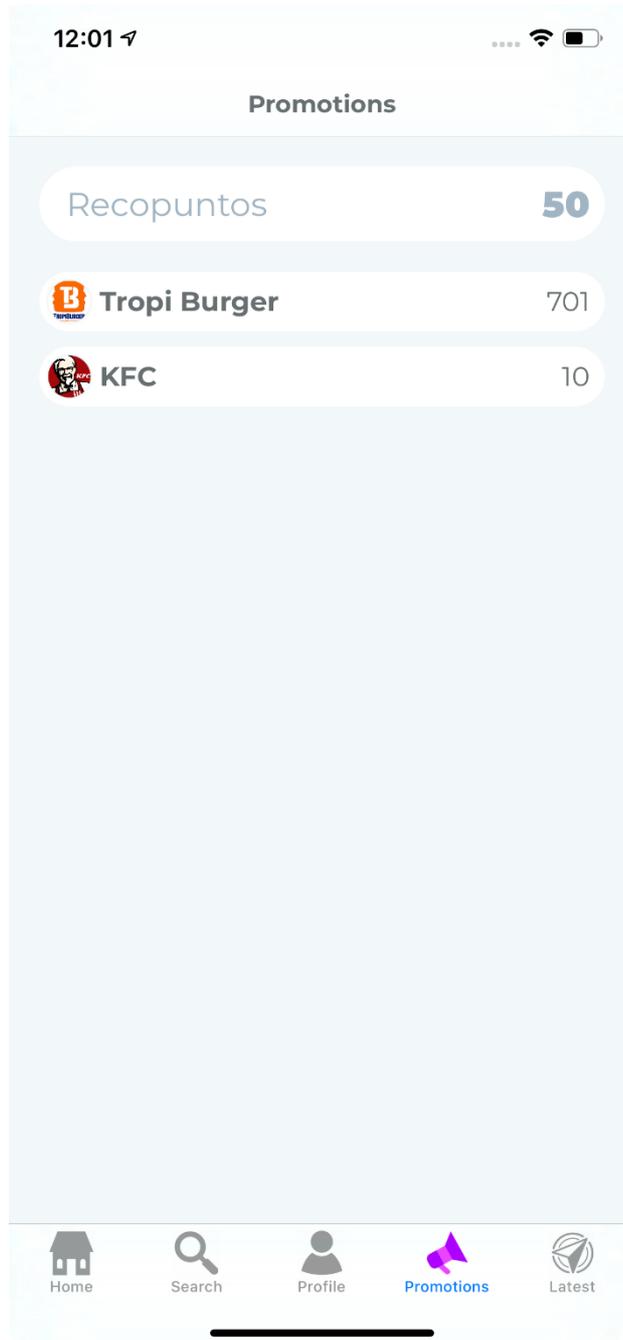


Figura 33. Pantalla de promociones con los establecimientos cercanos.

Al seleccionar cualquiera de los establecimientos se muestran las promociones con las que cuenta el establecimiento seleccionado, cada una indicando su valor en recopuntos aparte de otra información como se ve en la **Figura 34**.

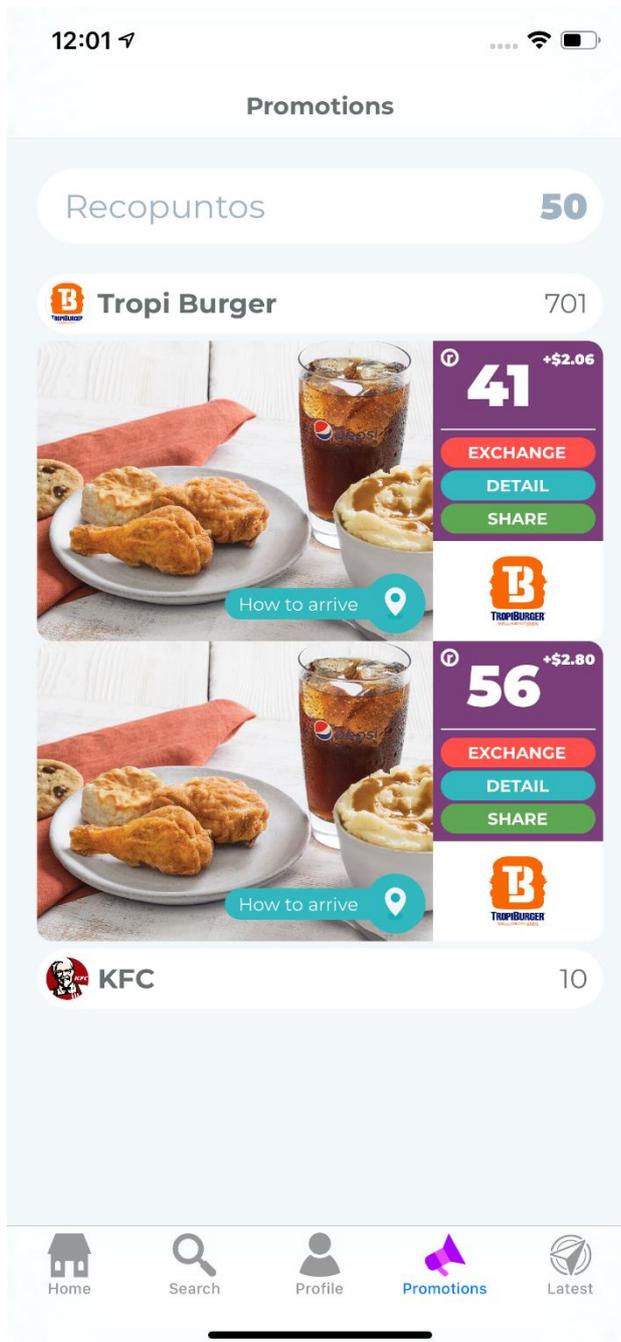


Figura 34. Pantalla de promociones con promociones de un establecimiento seleccionado.

Luego, al presionar el botón *EXCHANGE* se muestra una ventana emergente de confirmación con información extra como se ve en la **Figura 35**.

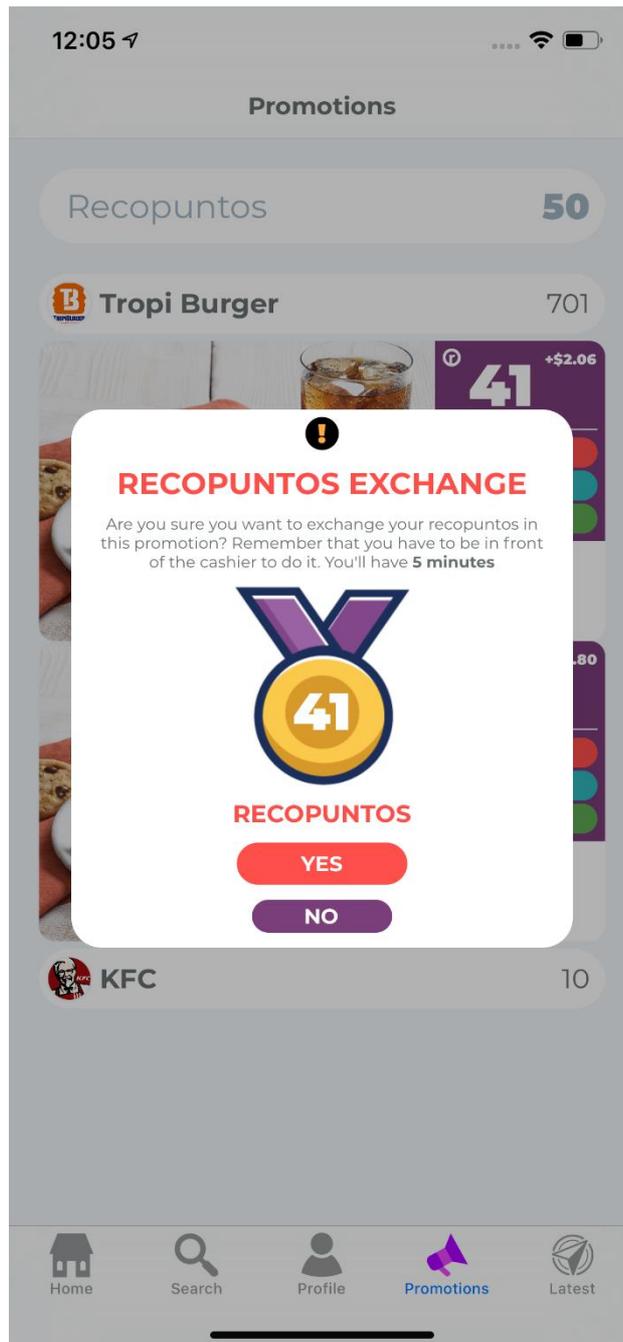


Figura 35. Ventana emergente de confirmación para canjear los recopuntos.

Posteriormente, si el usuario acepta se muestra la pantalla con el código QR generado con toda la información de la promoción, además de un contador en la parte superior que es el tiempo que le queda para reclamar la promoción como se ve en la **Figura 36**.



Figura 36. Pantalla con el código QR de la promoción y su información.

Finalmente, al regresar a la pantalla de promociones, se observa como la cantidad de cupones del establecimiento en el que el usuario canjeó los recopuntos ha disminuido, así como la cantidad de recopuntos del usuario como se ve en la **Figura 37**.

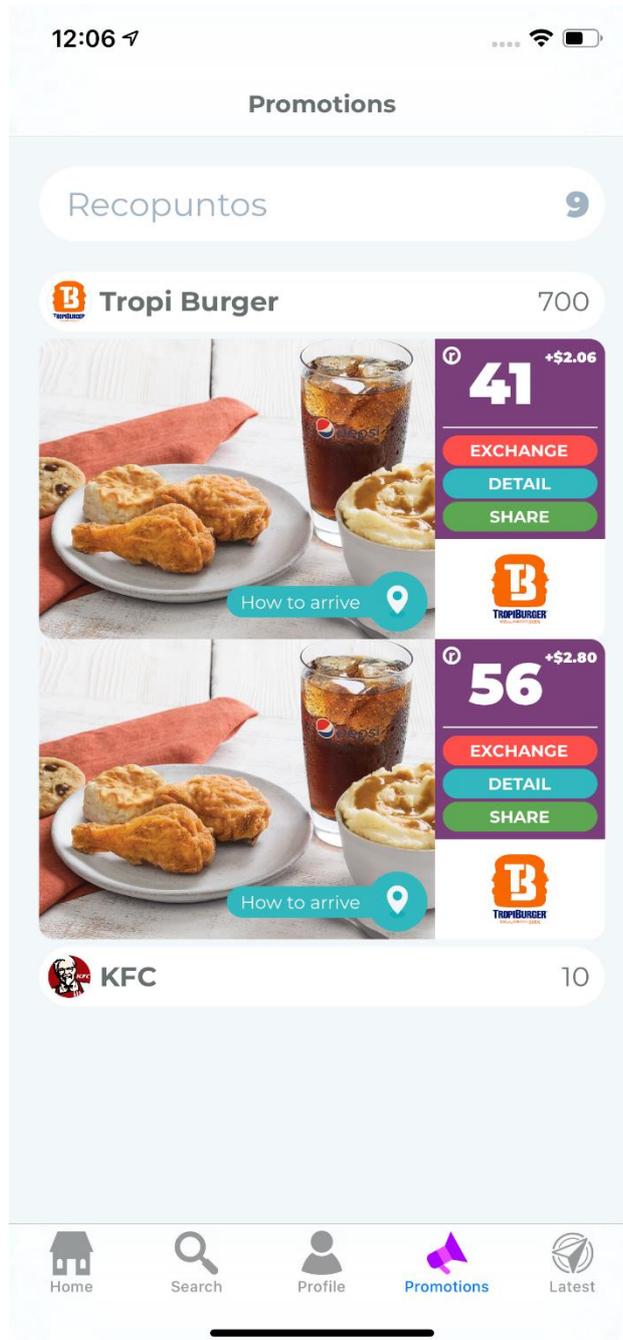


Figura 37. Pantalla de promociones luego de haber canjeado recopuntos.

3.2 Pruebas con usuarios finales

Se realizaron pruebas del producto final a un total de 20 personas, estudiantes que cursan la materia de iOS. Se evaluó el interés, la utilidad y la facilidad de uso percibidas para realizar recomendaciones y canjear los recopuntos. Se efectuaron un total de 9 preguntas por cada usuario, 3 para el interés que genera la aplicación, 3 para la utilidad percibida y 3 para la facilidad de uso. Las 2 primeras preguntas sobre el interés generado fueron de selección múltiple y la última de selección única, las respuestas de utilidad y

facilidad de uso se ajustaron a una escala del 1 al 7 donde 1 es la peor calificación y 7 es la mejor. Las preguntas realizadas se encuentran en el anexo VIII, sección 6.8.

3.2.1 Interés generado

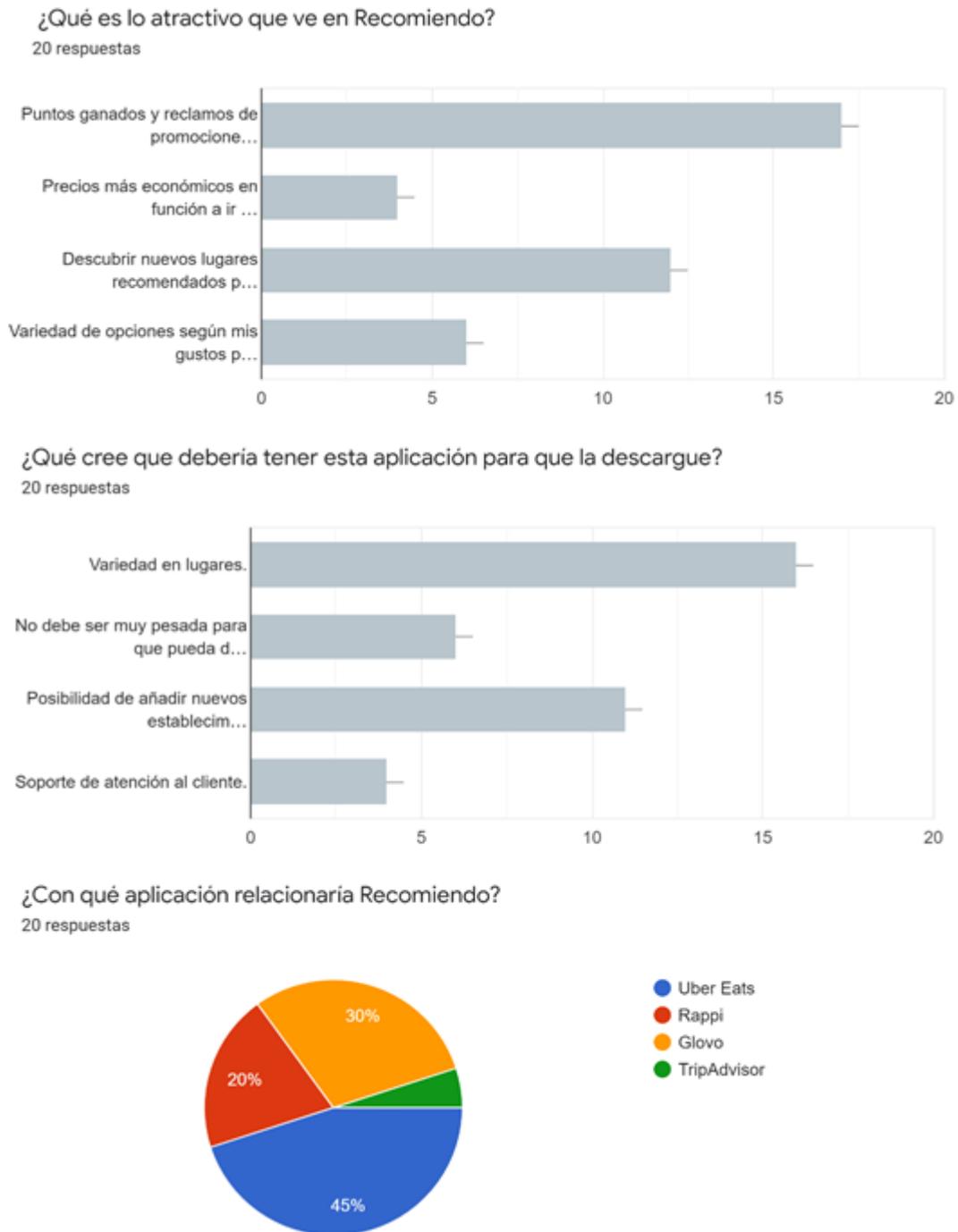


Figura 38. Resultados de las preguntas sobre el interés generado.

En la **Figura 38** se puede visualizar que: en la pregunta 1 el interés que mayor genera al usuario son los puntos generados y reclamo de promociones con un porcentaje del 85%; en la pregunta 2 en el que se puede deducir que el interés generado es conocer variedad de lugares con una calificación del 80%; y en la pregunta 3 en el que se preguntó con qué aplicación relacionaría recomendando, el 45% de usuarios respondió con Uber Eats.

3.2.2 Utilidad percibida

El **Gráfico 7** muestra el promedio de respuesta por cada pregunta con respecto a la utilidad percibida. La pregunta 1 se refiere a si el usuario visitaría un establecimiento que un amigo le recomiende, y obtuvo un promedio de 5.65/7 por lo que se entiende que sí existe un interés en conocer los lugares que sus amigos frecuentan y consideran buenos.

En la pregunta 2 relacionada con el tiempo en que la pantalla con el código de la promoción permanece visible, se observa que obtuvo el menor puntaje promedio de las tres (4.75/7) lo que hace pensar que el tiempo actual de 5 minutos no es suficiente. Sin embargo, hay que tomar en cuenta que más del 65% estuvo de acuerdo con el tiempo.

La mayor calificación es 6.3/7 que corresponde a la pregunta 3 que trata de averiguar si la idea de ganar recopuntos incentiva a las personas a recomendar y posteriormente visitar los establecimientos. Se observa que las personas estuvieron muy interesadas con esta idea y efectivamente recomendarían los establecimientos para recibir recopuntos.

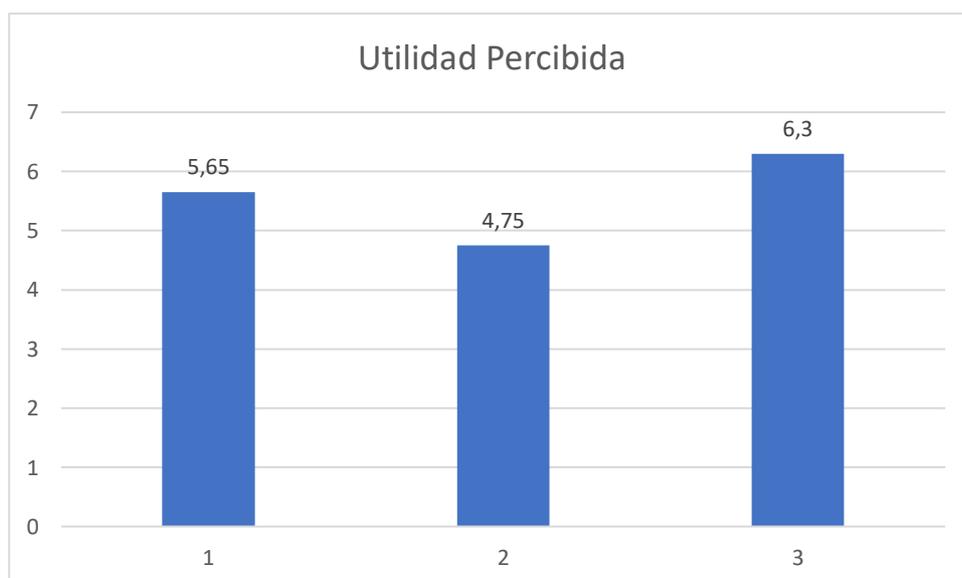


Gráfico 7. Promedio por pregunta de la utilidad percibida en el producto final.

3.2.3 Facilidad de uso percibida

Los promedios de cada pregunta se encuentran en el **Gráfico 8**. En este caso, los resultados son iguales o mayores a 6, lo que quiere decir que el producto es fácil de usar. La pregunta 1 se refiere a si le resultó fácil al usuario realizar una recomendación a sus amigos, muestra una calificación de 6.15/7 debido a que fue bastante intuitivo donde el flujo.

La pregunta 2 se refiere al funcionamiento y uso de recopuntos y tiene una calificación de 6/7, esto indica que los usuarios entendieron rápidamente el concepto y funcionamiento de los recopuntos.

La pregunta 3 se refiere al concepto de intereses y colecciones que usa la aplicación para personalizar las preferencias de los usuarios. esta es la que tiene la mayor calificación con 6.2/7 ya que se entendió rápidamente a qué se referían cada una de las diferentes preferencias mostradas para escoger.

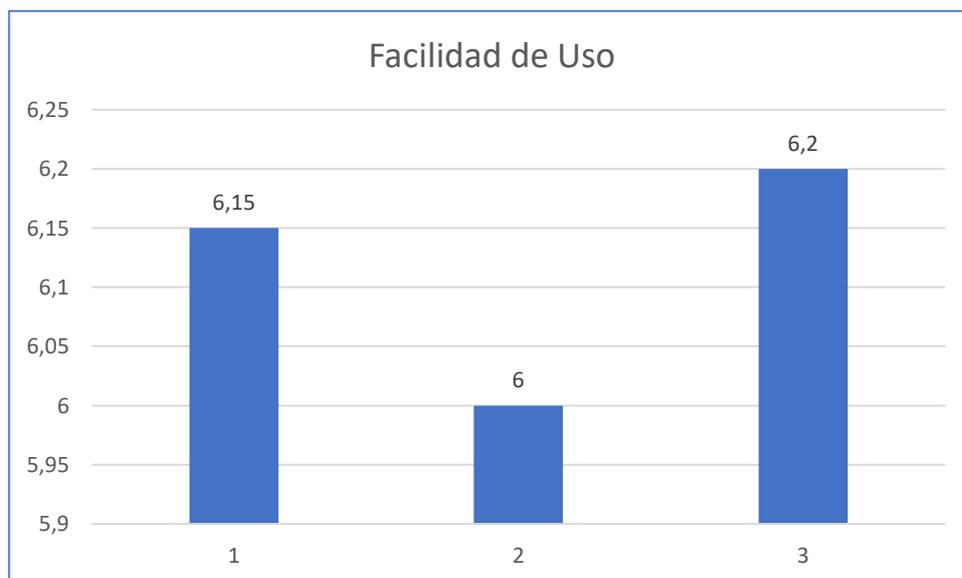


Gráfico 8. Promedio por pregunta de la facilidad de uso percibida en el producto final.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Lo expuesto a lo largo de este trabajo permite arribar las siguientes conclusiones:

- La definición de los requerimientos funcionales y no funcionales con la participación del *product owner* permitió conocer de manera clara las necesidades del cliente.
- La utilización de herramientas compatibles con la plataforma iOS facilitó la configuración de los equipos, instalación y la interoperabilidad entre cada una. De manera que se logró el desarrollo de un prototipo nativo.
- Con base en los requerimientos, se decidió el uso del *framework* SCRUM en el proyecto para desarrollar una aplicación basada en el valor del negocio, flexibilidad y adaptación respecto a las necesidades del cliente.
- El uso de las preferencias del usuario y su ubicación permitió implementar un algoritmo que muestre los establecimientos comerciales en la pantalla principal.
- Los establecimientos que visualiza el usuario varían de acuerdo con su ubicación, sus intereses y la calificación y cantidad de recomendaciones del establecimiento.
- La implementación de recopuntos como recompensa por interactuar con la aplicación llama la atención a más usuarios.
- La aplicación permite a los usuarios recomendar los establecimientos de su interés y ganar recopuntos al hacerlo.
- La aplicación permite a los establecimientos llegar a más personas a través de las diferentes opciones que ofrece la aplicación para ofrecer sus promociones.
- Para evitar el abuso de ganar recopuntos a cada momento, al recomendar un establecimiento se decidió establecer un tiempo de espera entre cada recomendación tomando en cuenta el medio utilizado.
- La parametrización de los tiempos de espera y la cantidad de recopuntos que un usuario puede ganar facilita en gran medida los cambios que se quieran realizar a futuro con respecto al valor de los recopuntos sin necesidad de realizar cambios en la aplicación.
- Con base en la experiencia del *product owner* se determinó que las promociones no pueden ser reclamadas únicamente con recopuntos, sino que también deben incluir un valor monetario.

4.2 Recomendaciones

Para trabajos futuros se recomienda:

- Implementar el pago directo de las promociones desde la aplicación para mejorar el proceso de compra y pago.
- Implementar el registro de un usuario mediante a través de Facebook y Google para mejorar el tiempo de registro.
- Desarrollar una aplicación web para gestionar y proveer los datos necesarios para el funcionamiento del sistema.
- Transformar los recopuntos a monedas virtuales para facilitar la transacción de pago en los establecimientos.
- Implementar la aplicación para la plataforma Android, para que de esta manera llegar a más personas y establecimientos.
- Encriptar los datos personales del usuario mediante un algoritmo de encriptación, en caso de que la información sea hurtada.
- Agregar las entidades necesarias al modelo para implementar un método de pago a través de membresías.
- Implementar inteligencia artificial en el algoritmo para obtener la lista de establecimientos que muestra la aplicación.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] M. d. T. y. S. d. I. Informacion, Libro blanco de la Sociedad de Informacion del conocimiento, 2018.
- [2] INEC, «Tecnologías de la Información y Comunicación,» 2017.
- [3] I. d. E. y. Censos, «Tras las cifras de quito,» Instituto Nacional de Estadisticas y cesos, [En línea]. Available: <http://www.ecuadorencifras.gob.ec/tras-las-cifras-de-quito>. [Último acceso: 26 02 26].
- [4] Tripadvisor, «MediaRoom,» [En línea]. Available: <https://tripadvisor.mediaroom.com/US-about-us>. [Último acceso: 2018 10 07].
- [5] Yelp, «Yelp,» [En línea]. Available: <https://www.yelp.com/about>. [Último acceso: 07 11 2018].
- [6] c. investors, «codecademy,» [En línea]. Available: <https://www.codecademy.com/articles/what-is-rest>. [Último acceso: 21 9 2019].
- [7] A. W. services, «Tipos de informática en la nube,» [En línea]. Available: <https://aws.amazon.com/es/types-of-cloud-computing/>. [Último acceso: 21 09 2019].
- [8] AWS, «Amazon Relational Database Service,» [En línea]. Available: <https://aws.amazon.com/es/rds/?nc=sn&loc=0>. [Último acceso: 21 09 2019].
- [9] A. W. Service, «Amazon Web Service,» [En línea]. Available: <https://aws.amazon.com/es/ec2/>. [Último acceso: 21 09 2019].
- [10] Amazon Web Services, «Amazon S3,» 2019. [En línea]. Available: <https://aws.amazon.com/s3/>. [Último acceso: 18 September 2019].
- [11] TechTerms, «MVC,» 7 March 2018. [En línea]. Available: <https://techterms.com/definition/mvc>. [Último acceso: 21 September 2019].
- [12] The Sails Company, «Sails features,» 18 June 2019. [En línea]. Available: <https://sailsjs.com/features>. [Último acceso: 12 September 2019].
- [13] Google Developers, «Directions API,» 5 August 2019. [En línea]. Available: <https://developers.google.com/maps/documentation/directions/start>. [Último acceso: 18 September 2019].
- [14] Amazon Web Services, «AWS SDK for JavaScript in Node.js,» 2019. [En línea]. Available: <https://aws.amazon.com/sdk-for-node-js/>. [Último acceso: 18 September 2019].

- [15] Socket.IO, «What Socket.IO is,» 2019. [En línea]. Available: <https://socket.io/docs>. [Último acceso: 18 September 2019].
- [16] JetBrains, «Getting Started with WebStorm,» 2019. [En línea]. Available: <https://www.jetbrains.com/help/webstorm/getting-started-with-webstorm.html>. [Último acceso: 18 September 2019].
- [17] Apple, «Welcome to Xcode,» 2019. [En línea]. Available: <https://help.apple.com/xcode/mac/current/#/devc8c2a6be1>. [Último acceso: 18 September 2019].
- [18] Atlassian, «Version control software for professional teams,» 2019. [En línea]. Available: <https://bitbucket.org/product/version-control-software>. [Último acceso: 18 September 2019].
- [19] Postman, Inc., «The Collaboration Platform for API Development,» 2019. [En línea]. Available: <https://www.getpostman.com>. [Último acceso: 18 September 2019].
- [20] K. S. a. J. Sutherland, «Scrumguides.org,» [En línea]. Available: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>. [Último acceso: 21 09 2019].
- [21] A. B. Latre, «Scrum.org,» [En línea]. Available: <https://www.scrum.org/resources/blog/scrum-myths-it-ok-have-sprint-0-design-sprint-hardening-sprint>. [Último acceso: 21 09 2019].
- [22] M. Martinez, «Plan de marketing digital para pyme,» 2014.
- [23] F. D. Davis, «Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology,» *MIS Quarterly*, vol. 13, pp. 319-340, 1989.

6 ANEXOS

6.1 Anexo I: Diagrama de Clases

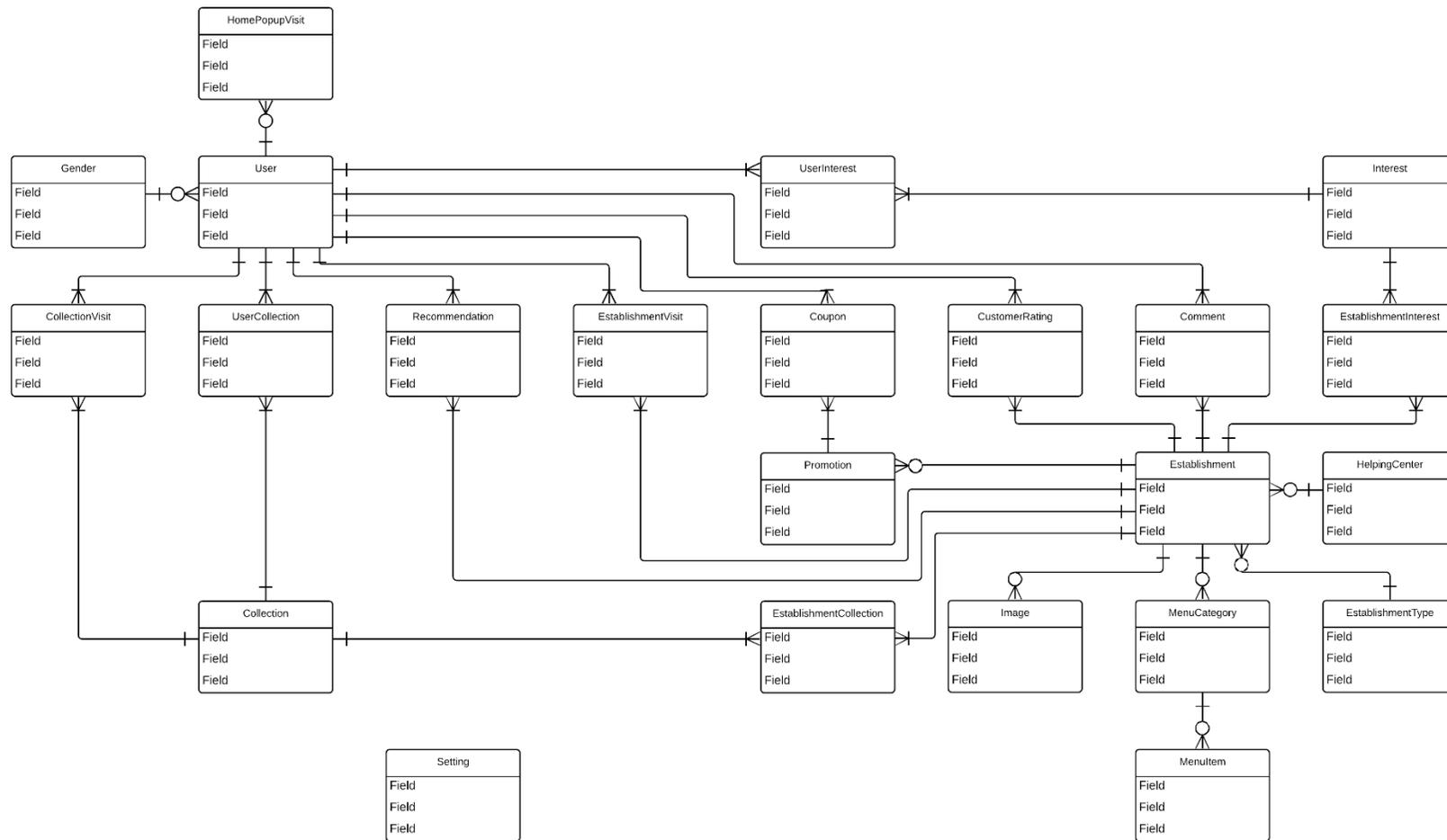


Figura 39. Diagrama de clases.

6.2 Anexo II: Historias Épicas

Tabla 25. Historia Épica HUE1.

HISTORIA DE USUARIO ÉPICA	
ID: HUE1	Prioridad: Alta
Título: Gestión de usuarios.	
Descripción: Como usuario, quiero conocer establecimientos en un radio no tan amplio y que se ajusten a mis preferencias para no perder tiempo en la búsqueda de estos.	

Tabla 26. Historia Épica HUE2.

HISTORIA DE USUARIO ÉPICA	
ID: HUE2	Prioridad: Alta
Título: Gestión de establecimientos.	
Descripción: Como usuario, necesito visualizar la información de un establecimiento para ver lo que ofrece y, si me agrada, visitarlo y recomendarlo y de esta manera obtener un beneficio.	

Tabla 27. Historia Épica HUE3.

HISTORIA DE USUARIO ÉPICA	
ID: HUE3	Prioridad: Alta
Título: Gestión de promociones y recopuntos.	
Descripción: Como usuario, necesito visualizar las promociones de los diferentes establecimientos que me muestra la aplicación para canjear mis recopuntos en alguna de estas.	

6.3 Anexo III: Historias de Usuario del Sprint 1

Tabla 28. Historia de Usuario HU1-1.

HISTORIA DE USUARIO		
ID: HU1-1	Esfuerzo: 20	
Título: Registrar usuarios.		
Sprint N°: 1	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario necesito registrarme en la aplicación para tener acceso a todas sus funcionalidades.		
Criterios de	La primera vez que el usuario ingrese al sistema deberá registrarse. En	

aceptación	la pantalla de inicio de sesión tendrá un acceso Registrarse.
	Una vez que el usuario escoja Registrarse se abrirá una nueva pantalla con campos disponibles para registrar su nombre, apellido, género, fecha de nacimiento, dirección de correo electrónico y contraseña.
	En la misma pantalla de registro, el usuario tendrá la opción de escoger una imagen de su galería o tomar una nueva foto para usarla como imagen de perfil.
	La contraseña deberá ser mínimo de 6 caracteres y deberá contener por lo menos una letra y un número. En caso de que no se cumpla esto, la aplicación informará al usuario con una alerta.

6.4 Anexo IV: Historias de Usuario del Sprint 2

Tabla 29. Historia de Usuario HU2-1.

HISTORIA DE USUARIO		
ID: HU2-1		Esfuerzo: 7
Título: Registrar preferencias de usuario.		
Sprint N°: 2	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario necesito ingresar mis preferencias para personalizar mi cuenta.		
Criterios de aceptación	Una vez que el usuario presiona Registrarse en la pantalla de registro de datos personales se abrirá una nueva pantalla con la lista completa de intereses y colecciones disponibles.	
	El usuario debe seleccionar por lo menos un interés y una colección. En caso de que esto no se cumpla, la aplicación informará al usuario con una alerta.	

Tabla 30. Historia de Usuario HU3-1.

HISTORIA DE USUARIO		
ID: HU3-1		Esfuerzo: 7
Título: Inicio de sesión de usuarios.		
Sprint N°: 2	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario puedo iniciar sesión para poder acceder a mi información.		
Criterios de aceptación	En la pantalla de inicio de sesión se mostrarán dos campos disponibles para que el usuario ingrese una dirección de correo electrónico y una	

	contraseña.
	La dirección de correo electrónico ingresada debe ser una dirección válida. Si se ingresa una dirección incorrecta, la aplicación informará al usuario con una alerta.
	Si la dirección de correo electrónico no existe o la contraseña es incorrecta, la aplicación informará al usuario con una alerta.

Tabla 31. Historia de Usuario HU4-1.

HISTORIA DE USUARIO		
ID: HU4-1		Esfuerzo: 6
Título: Navegación de usuario en la aplicación.		
Sprint N°: 2	Prioridad: Media	Estado: Por implementar
Descripción: Como usuario puedo editar mis datos personales, obtener información sobre el funcionamiento del sistema de recopuntos de la aplicación, revisar mis comentarios y buscar colecciones.		
Criterios de aceptación	En la pantalla principal el usuario tendrá disponible una barra de pestañas para navegar entre las pantallas: principal, buscar, perfil, promociones y últimos.	
	En la pantalla de perfil, el usuario tendrá acceso a sus comentarios, a información sobre cómo ganar recopuntos y a configuraciones de su cuenta.	
	Cuando el usuario acceda a la pantalla de sus comentarios la aplicación mostrará todos los comentarios del usuario y al final las promociones de los establecimientos que son <i>sponsor</i> .	
	En la pantalla de información sobre recopuntos, cuando el usuario seleccione alguna de las opciones de información la aplicación deberá redirigirlo al perfil del último establecimiento que visitó.	

6.5 Anexo V: Historias de Usuario del Sprint 3

Tabla 32. Historia de Usuario HU5-1.

HISTORIA DE USUARIO		
ID: HU5-1		Esfuerzo: 20
Título: Visualizar promociones, establecimientos y colecciones.		
Sprint N°: 3	Prioridad: Alta	Estado: Por implementar
<p>Descripción: Como usuario puedo ver cuantos recopuntos tengo, colecciones seleccionadas y todos los establecimientos que se encuentren en mi cercanía de acuerdo con mis preferencias para visitarlos.</p>		
Criterios de aceptación	Después del registro, una vez que el usuario guarde sus intereses y colecciones se mostrará la pantalla principal. En futuras ocasiones la aplicación siempre se abrirá en esta pantalla.	
	En la pantalla principal, en la parte superior se mostrará el número de recopuntos con los que cuenta el usuario.	
	En la pantalla principal, debajo del número de recopuntos del usuario se mostrarán las promociones de los establecimientos que son <i>sponsor</i> .	
	En la pantalla principal, debajo de las promociones se mostrarán los establecimientos cercanos según las preferencias del usuario.	
	Los establecimientos que se mostrarán serán los que se encuentren en un radio de 5 km a la redonda respecto a la ubicación del usuario.	
	Los establecimientos se mostrarán en orden descendente tomando en cuenta primero el número de recomendaciones que posee y luego su calificación.	
	En la pantalla principal, debajo de los establecimientos se mostrarán las colecciones seleccionadas por el usuario.	
	Cuando la aplicación se abra, en la pantalla principal se mostrará una ventana emergente con todas las colecciones disponibles.	
	Si el usuario selecciona alguna de las colecciones mostradas en la ventana emergente, se abrirá una nueva ventana emergente y se mostrarán todos los establecimientos <i>premium</i> pertenecientes a la colección seleccionada.	

6.6 Anexo VI: Historias de Usuario del Sprint 4

Tabla 33. Historia de Usuario HU1-2.

HISTORIA DE USUARIO		
ID: HU1-2		Esfuerzo: 13
Título: Perfilar el establecimiento.		
Sprint N°: 4	Prioridad: Alta	Estado: Por implementar
<p>Descripción: Como usuario necesito visualizar la información de un establecimiento para ver lo que ofrece y, si me agrada, visitarlo y recomendarlo y de esta manera obtener un beneficio.</p>		
Criterios de aceptación	En la pantalla principal, cuando un usuario seleccione algún establecimiento se abrirá la pantalla con el perfil del establecimiento seleccionado.	
	En la pantalla del perfil del establecimiento, se mostrará la información del establecimiento como: nombre, calificación, si se encuentra abierto, a qué tipo pertenece, su colección principal, estimación de costo y dirección.	
	En la pantalla del perfil del establecimiento, el usuario podrá ver lista de productos y/o servicios que ofrece el establecimiento.	
	En la pantalla del perfil del establecimiento, el usuario podrá ver, crear o eliminar comentarios.	
	En la pantalla del perfil del establecimiento, el usuario podrá ver la galería con imágenes proporcionadas por el establecimiento.	
	En la pantalla del perfil del establecimiento, el usuario podrá calificar al establecimiento con una valoración entre 1 y 5, siendo 1 la más baja y 5 la más alta.	
	En la pantalla del perfil del establecimiento, el usuario podrá recomendarlo a través de WhatsApp o Facebook, dependiendo de la opción que elija.	
	Luego de realizar una recomendación, la aplicación mostrará una alerta informando al usuario cuántos recopuntos ha ganado.	

Tabla 34. Historia de usuario HU2-2.

HISTORIA DE USUARIO		
ID: HU2-2		Esfuerzo: 7
Título: Visualizar la ubicación de un establecimiento.		
Sprint N°: 4	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario necesito ver la ubicación para llegar a los diferentes establecimientos.		
Criterios de aceptación	El usuario podrá ver la ubicación del establecimiento en un mapa. En la pantalla del perfil del establecimiento habrá un acceso ¿Cómo llegar?	
	El usuario podrá ver la ruta marcada en el mapa desde su ubicación hasta la del establecimiento al seleccionar el botón Mostrar.	

6.7 Anexo VII: Historias de Usuario del Sprint 5

Tabla 35. Historia de Usuario HU1-3.

HISTORIA DE USUARIO		
ID: HU1-3		Esfuerzo: 13
Título: Definición de promociones por establecimiento.		
Sprint N°: 5	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario puedo ver las promociones de los diferentes establecimientos de acuerdo con mis preferencias y canjear mis recopuntos.		
Criterios de aceptación	El usuario podrá ver la ubicación del establecimiento en un mapa sin tener que ingresar al perfil de este. Cada promoción contará con un acceso ¿Cómo llegar?	
	Una vez que el usuario escoja ¿Cómo llegar? se abrirá una pantalla que muestre el número de recopuntos del usuario y un mapa con la ubicación del establecimiento.	
	En la pantalla de promociones el usuario visualizará una lista de establecimientos. Cuando seleccione uno se desplegará una sublista con las promociones de este y podrá ver el precio en dinero y recopuntos de la promoción.	
	Si el usuario al intentar reclamar una promoción no tiene suficientes recopuntos la aplicación mostrará una alerta.	
	Si el número de cupones disponibles en la promoción se agotan luego de la última transacción, la promoción se deshabilita y si un usuario quiere canjearla la aplicación muestra una alerta.	

Tabla 36. Historia de Usuario HU2-3.

HISTORIA DE USUARIO		
ID: HU2-3	Esfuerzo: 7	
Título: Asignación de promociones.		
Sprint N°: 5	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario puedo canjear mis recopuntos por las diferentes promociones ofertadas.		
Criterios de aceptación	La aplicación generará un código QR una vez que el usuario reclame una promoción en la pantalla de promociones.	
	El usuario al ingresar en la pantalla de la promoción podrá ver toda la información referente a esta.	
	La pantalla con el código QR e información de la promoción mostrará al usuario un temporizador de 5 minutos que cuando llegue a cero cerrará la pantalla.	

6.8 Anexo VIII: Creación de los servicios AWS

Creación de una cuenta en AWS.

El primer paso y requisito indispensable para utilizar los servicios de AWS es crear una cuenta como se muestra en la **Figura 40**.

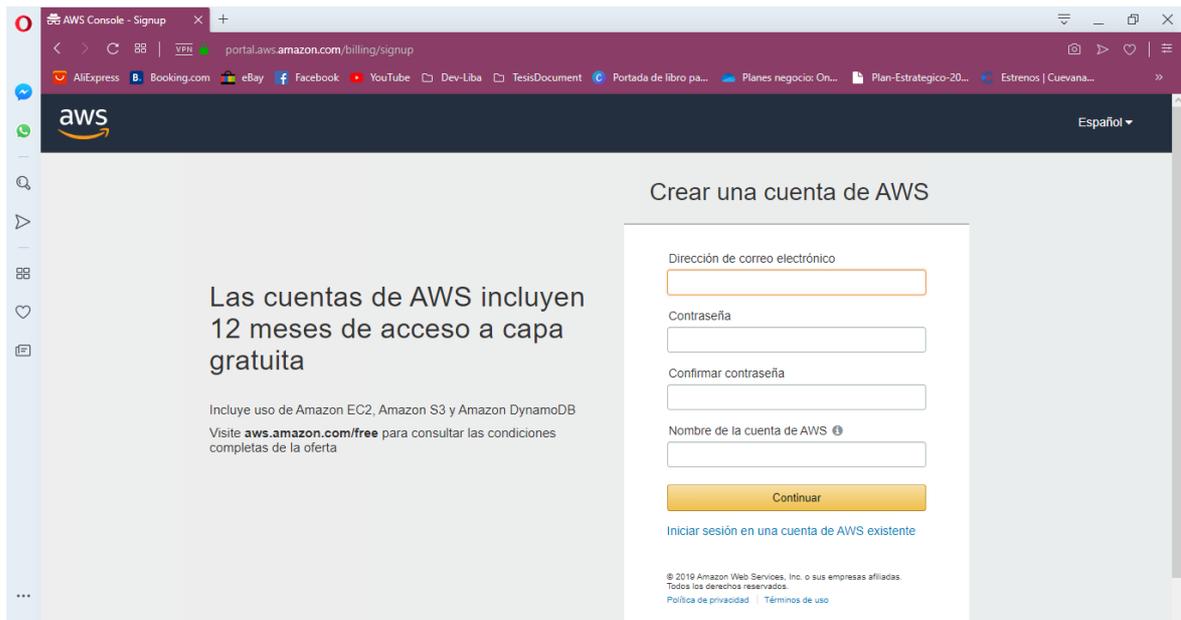


Figura 40. Pantalla para crear una cuenta en AWS.

Creación y configuración del servicio RDS.

Ya que se va a usar AWS se escogió la opción RDS que ofrece la base de datos MySQL que se usó en el proyecto. Para la creación del servicio de base de datos RDS se debe configurar una llave privada.

La **Figura 41** muestra el menú del *dashboard* de la consola en el que se despliega el menú seguridad y red *Network and Security* y al seleccionar *Key Pairs* aparece una pantalla.

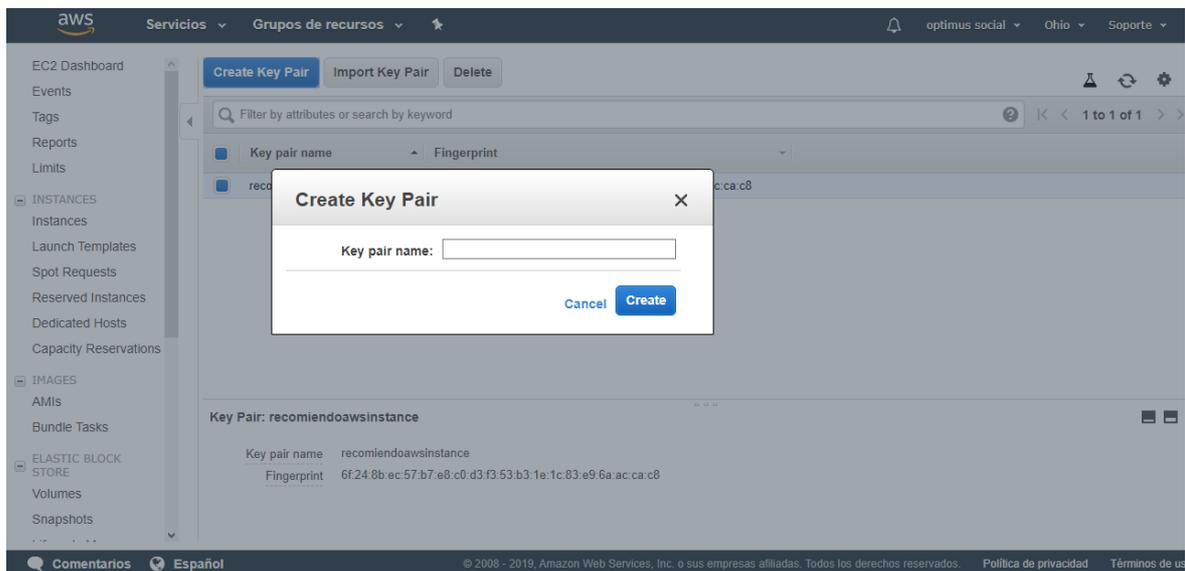


Figura 41. Pantalla para crear la llave privada.

Después de ingresar el nombre a la llave secreta se presiona el botón *create*; enseguida se descargará un archivo como se muestra en la **Figura 42** nombreLlave.pem el cual nos permitirá conectar a nuestros servicios de Amazon.

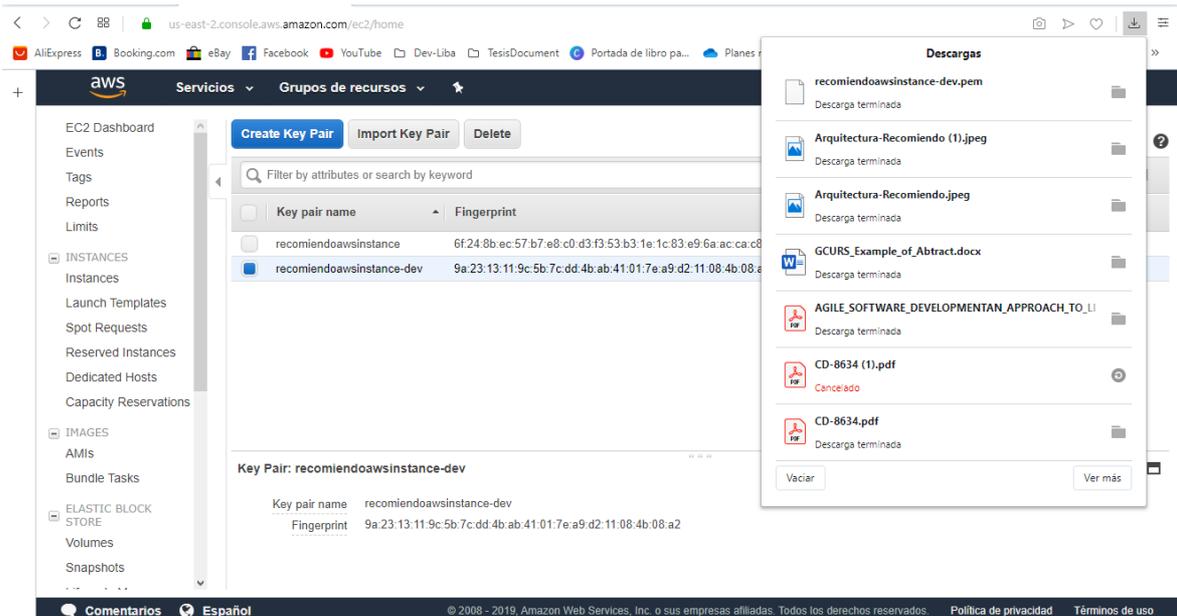


Figura 42. Descarga de la llave privada.

Una vez creada la llave de acceso se procede a crear el servicio RDS para lo cual se seleccionó una base de datos MySQL como se muestra en la **Figura 43**.

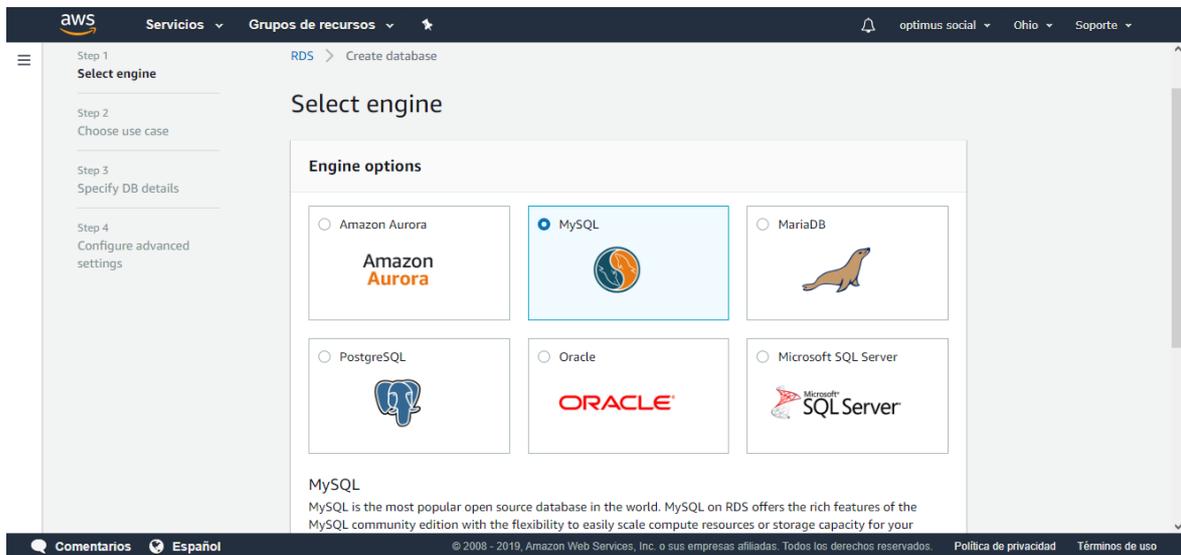


Figura 43. Pantalla para seleccionar el motor de base de datos.

Para conectarse desde un cliente de base de datos o una aplicación, es necesario utilizar los datos de conexión que se encuentran en la sección *Connectivity & Security* como se muestra en la **Figura 44**.

tutorial-db-instance

Summary

DB identifier tutorial-db-instance	CPU
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port

Endpoint
tutorial-db-instance. [redacted].us-west-2.rds.amazonaws.com

Port
3306

Network

Availability
us-west-

VPC
tutorial-

Subnet c

Figura 44. Datos de conexión a la base de datos.

Creación del servicio S3

Se creó un contenedor de AWS S3 debido a que la aplicación debe almacenar imágenes del establecimiento y del cliente. Mediante este servicio se logra rendimiento, escalabilidad y disponibilidad.

Para crear un contenedor de S3 se debe iniciar sesión en la consola de administración de AWS y abrir la consola de Amazon S3 como se muestra en la **Figura 45**.

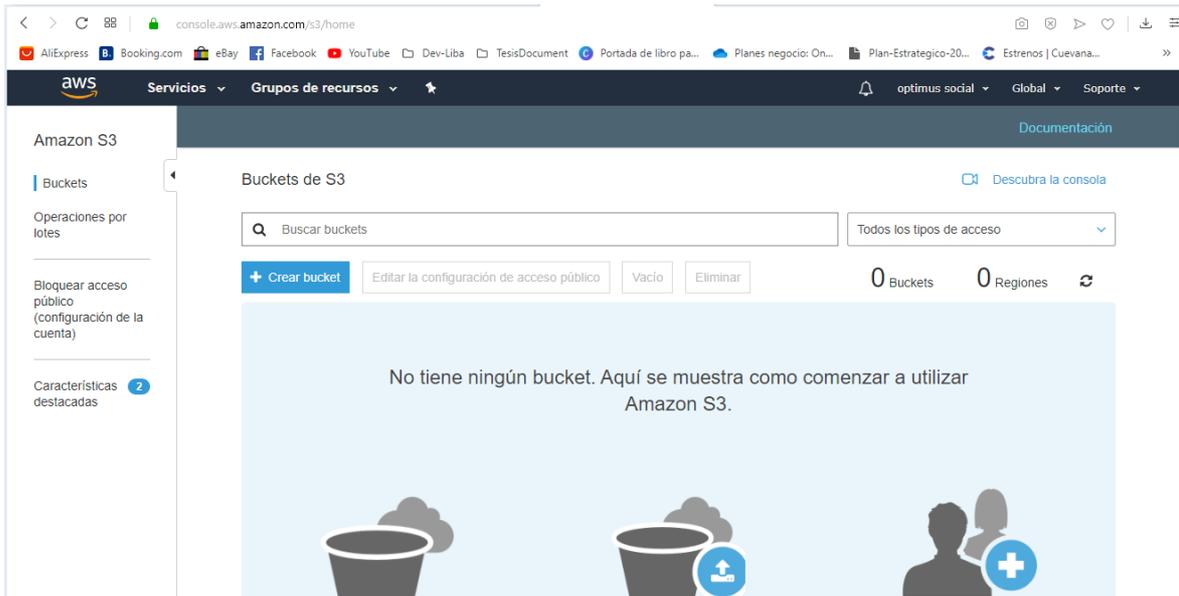


Figura 45. Consola de administración de Amazon S3.

Creación del EC2

La solución que AWS ofrece para capacidad computacional es Amazon *Elastic Compute Cloud* (EC2).

Para crear el EC2 se debe ingresar a la consola de administración del AWS e ingresar en el buscador la palabra clave EC2 como se muestra en la **Figura 46**.

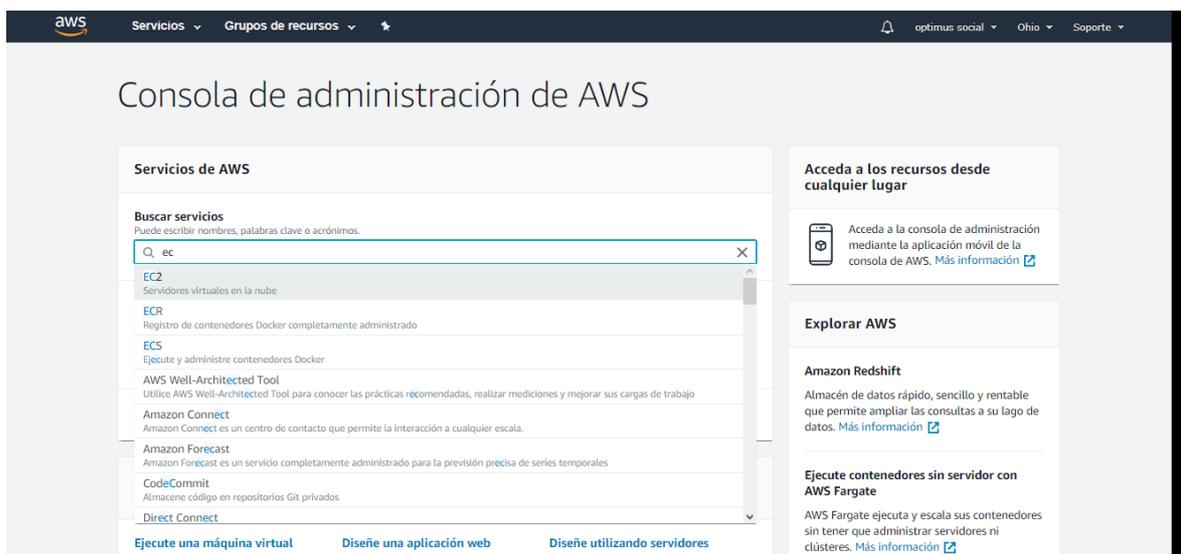


Figura 46. Pantalla para buscar servicios de AWS.

Una vez creada la instancia se tendrá algo como se muestra en la **Figura 47** en la cual podrá visualizar la IP pública, memoria y entre otras especificaciones de la instancia.

The screenshot shows the AWS Management Console interface for EC2 instances. The top navigation bar includes the AWS logo, 'Servicios', 'Grupos de recursos', and user information 'optimus social'. The left sidebar contains navigation options like 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES', 'Launch Templates', 'Spot Requests', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'IMAGES', 'AMIs', 'Bundle Tasks', 'ELASTIC BLOCK STORE', 'Volumes', and 'Snapshots'. The main content area shows a table of instances with columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, and Status Checks. Two instances are listed: 'recomiendoweb' and 'recomiendows'. The 'recomiendows' instance is selected, and its details are shown below, including Instance ID, Instance state (stopped), Instance type (t2.micro), Elastic IPs (3.15.253.151), Public DNS (IPv4), IPv4 Public IP (3.15.253.151), IPv6 IPs (-), and Private DNS (ip-172-31-28-24.us-east-2.compute.internal).

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
recomiendoweb	i-02594ef331175d742	t2.micro	us-east-2b	stopped	N...
recomiendows	i-0adbd74148ac2dec0	t2.micro	us-east-2b	stopped	N...

Instance: i-0adbd74148ac2dec0 (recomiendows)		Elastic IP: 3.15.253.151	
Instance ID	i-0adbd74148ac2dec0	Public DNS (IPv4)	ec2-3-15-253-151.us-east-2.compute.amazonaws.com
Instance state	stopped	IPv4 Public IP	3.15.253.151
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs	3.15.253.151*	Private DNS	ip-172-31-28-24.us-east-2.compute.internal

Figura 47. Consola de administración de Amazon EC2.

6.9 Anexo IX: Glosario

API (Application Programming Interface): Es una interfaz o protocolo de comunicación entre partes diferentes de un programa de computación con la finalidad de simplificar la implementación y el mantenimiento del software.

Recomiendo: Aplicación móvil prototipo descrita en el documento.

Interés: Representan las cosas o eventos que les pueden atraer a los usuarios y se puedan ofrecer en un establecimiento como, por ejemplo, la música en vivo, la cocina gourmet, la comida italiana, los cocteles, entre otros.

Colección: Representan los planes que pueden tener los usuarios para realizar con otras personas en un establecimiento.

Recopuntos: Representan las recompensas que el usuario de la aplicación obtiene al recomendar o calificar a un establecimiento.

Sponsors: Establecimientos que han realizado un pago para poder mostrar sus promociones en la pantalla principal de la aplicación.

App: término utilizado para referirse a aplicaciones móviles.

GPS: es un sistema que permite determinar en toda la tierra la posición de cualquier objeto (una persona, un vehículo)

Git: software de control de versiones, se define control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración.

AWS: (Amazon Web Services) es una colección de servicios de computación en la nube.

iOS: Sistema operativo móvil de la multinacional Apple Inc.

REST: es una interfaz para conectar varios sistemas basados en el protocolo HTTP.

JSON: (JavaScript Object Notation) es un formato de texto sencillo para el intercambio de datos.

Apple store: es una cadena de tiendas de Apple que venden ordenadores y otros productos de la marca.

Swift: es un lenguaje de programación multiparadigma creado por Apple enfocado en el desarrollo de aplicaciones para iOS.

Backend: es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos.

Frontend: son todas las tecnologías de diseño y desarrollo que se ejecutan en el navegador o dispositivo que se encargan de la interactividad con los usuarios.

Multiparadigma de programación: soporta más de un paradigma de programación, es decir permite crear programas usando mas de un estilo de programación.

Paradigma de programación: representa un enfoque particular o filosofía para la construcción del software.

Aplicación: es un programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de tareas.

Base de datos: conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización.

FTP: es un protocolo de transferencia de archivos.

SSH: es un protocolo de administración remota que le permite a los usuarios controlar y modificar sus servidores remotos a través de internet a través de un mecanismo de autenticación.

Puerto: punto de conexión entre redes.

MVC: es un estilo de arquitectura de software que separa los datos de una aplicación.

Servidor (Hardware): un servidor basado en hardware es una máquina física integrada en una red informática.

Servidor (software): un servidor basado en software es un programa que ofrece un servicio especial que otros programas denominados clientes (*clients*) pueden usar a nivel local o a través de una red

HTTP: protocolo de transferencia de hipertexto.

HTTPS: protocolo seguro de transferencia de hipertexto.

Hipertexto: es una estructura no secuencial que permite crear, agregar, enlazar y compartir información de diversas fuentes por medio de enlaces asociativos y redes sociales.

SCRUM: Scrum es un marco de trabajo de procesos (*framework*) que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90.

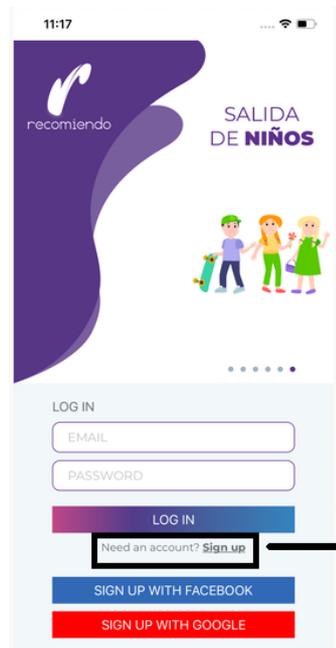
Sprint: El corazón de Scrum es el Sprint, es un bloque de tiempo (*time-box*) de un mes o menos durante el cual se crea un incremento de producto "Terminado" utilizable y potencialmente desplegable.

CLOUD: son tecnologías que habilita al usuario el consumo de servicios pagando solo por lo que se consume, de forma escalable, elástica, flexible y por medio de un portal de auto aprovisionamiento.

6.10 Anexo X: Guía de uso

- **Registrar usuario y configurar el perfil.**

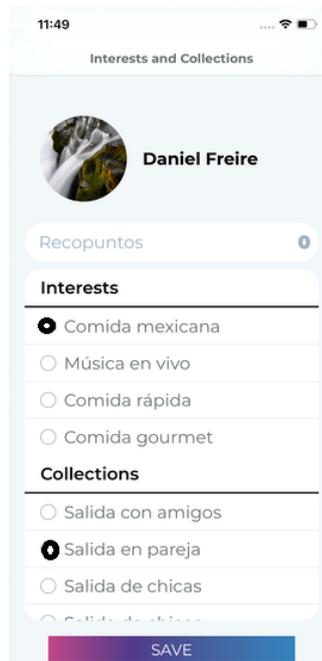
En la pantalla de inicio, presionar en Sign up.



Ingresar los datos personales y presionar en SIGN IN.



Para el correcto funcionamiento seleccionar por lo menos un interés y una colección.



- **Recomendar y ganar recopuntos.**

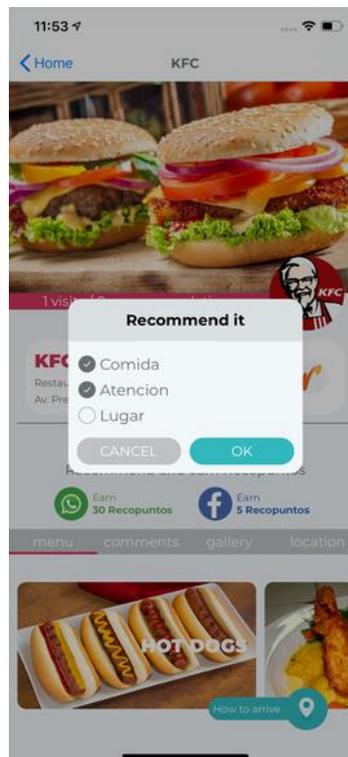
En la pantalla principal seleccionar un establecimiento.



Recomendar y ganar recopuntos compartiendo en WhatsApp o Facebook.



Presionar el medio en el que desea compartir; recomendar y presionar en ok tal como se muestra a continuación.

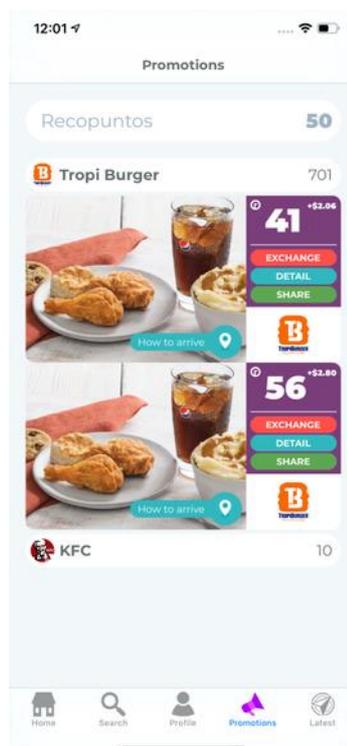


- **Canjear recopuntos.**

Ingresa a la pantalla de promociones y seleccionar el establecimiento en el que desea canjear los recopuntos.



Seleccionar la promoción en la que desea canjear los recopuntos.



Para canjear los recopuntos asegurarse de estar en el establecimiento.



Mostrar el código QR generado.



6.11 Anexo XI: Encuestas Realizadas

Cuestionario del interés generado.

General

Encuesta para conocer la opinión de recomiendo después de usarla

***Obligatorio**

¿Qué es lo atractivo que ve en Recomendando? *

- Puntos ganados y reclamos de promociones.
- Precios más económicos en función a ir al lugar sin recomendando.
- Descubrir nuevos lugares recomendados por mis amigos y la gente.
- Variedad de opciones según mis gustos previamente ingresados.

¿Qué cree que debería tener esta aplicación para que la descargue? *

- Variedad en lugares.
- No debe ser muy pesada para que pueda descargarme.
- Posibilidad de añadir nuevos establecimientos.
- Soporte de atención al cliente.

¿Con qué aplicación relacionaría Recomendando? *

- Uber Eats
- Rappi
- Glovo
- Otro: _____

Siguiente

Figura 48. Formulario para el interés generado.

Cuestionario de utilidad percibida

General

*Obligatorio

Utilidad percibida

¿Visitaría los establecimientos que sus amigos le recomienden mediante la aplicación Recomendando? *

1 2 3 4 5 6 7

De ninguna manera Absolutamente

¿Le parece suficiente el tiempo de espera de 5 minutos antes de que se cierre la pantalla con el código de la promoción? *

1 2 3 4 5 6 7

De ninguna manera Absolutamente

¿La idea de ganar recopuntos le incentivaron a visitar y recomendar establecimientos? *

1 2 3 4 5 6 7

De ninguna manera Absolutamente

Atrás

Siguiente

Figura 49. Formulario para la utilidad percibida.

Cuestionario de facilidad de uso percibida

General

*Obligatorio

Facilidad de uso percibida

¿Le resulta fácil recomendar un establecimiento a sus amigos? *

1 2 3 4 5 6 7

De ninguna manera Absolutamente

¿El funcionamiento y uso de recopuntos le parecieron claros y entendibles? *

1 2 3 4 5 6 7

De ninguna manera Absolutamente

¿Se entendió lo que son las colecciones e intereses? *

1 2 3 4 5 6 7

De ninguna manera Absolutamente

[Atrás](#) [Enviar](#)

Figura 50. Formulario para la facilidad de uso percibida.

Resultados de las encuestas

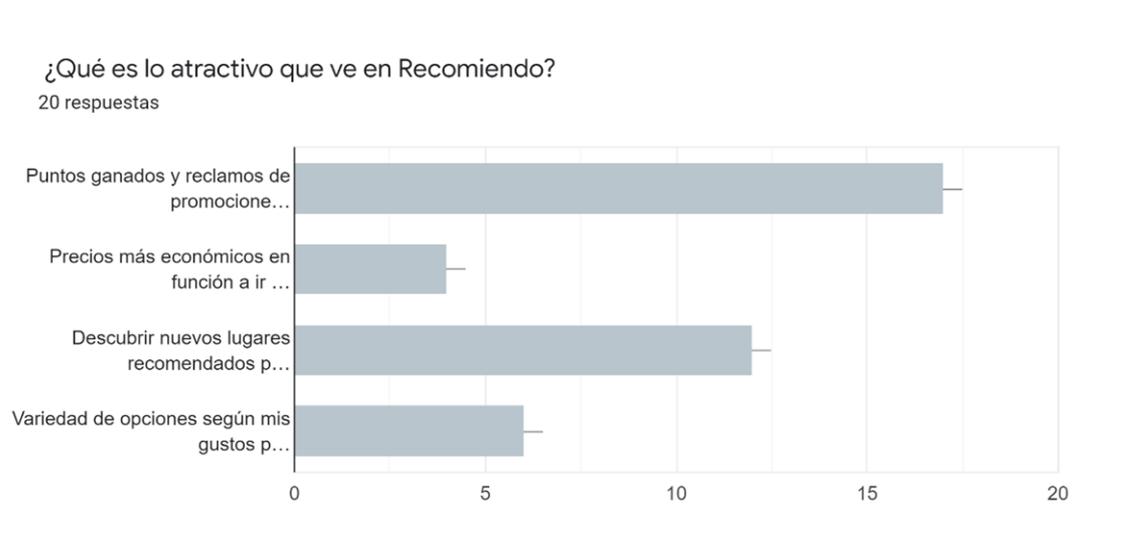


Figura 51. Respuestas de la pregunta 1 del interés generado.

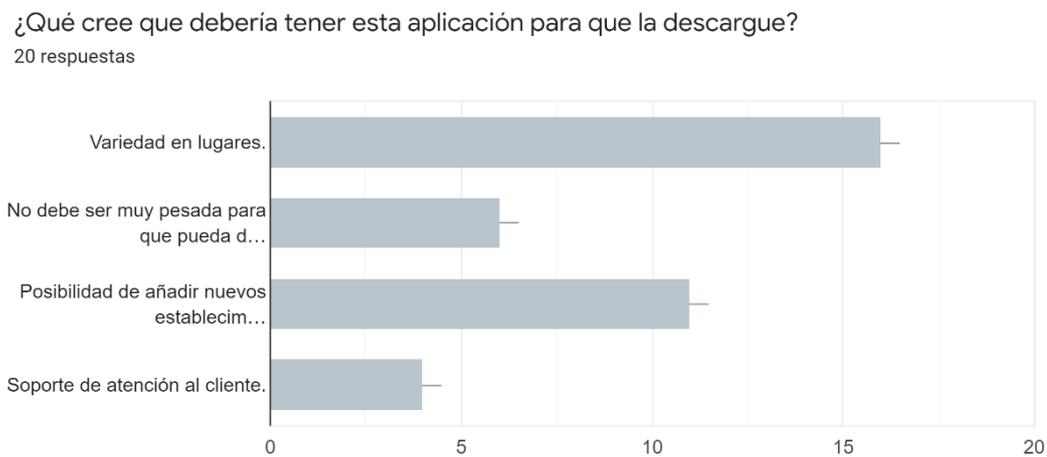


Figura 52. Respuestas de la pregunta 2 del interés generado.

¿Con qué aplicación relacionaría Recomendando?

20 respuestas

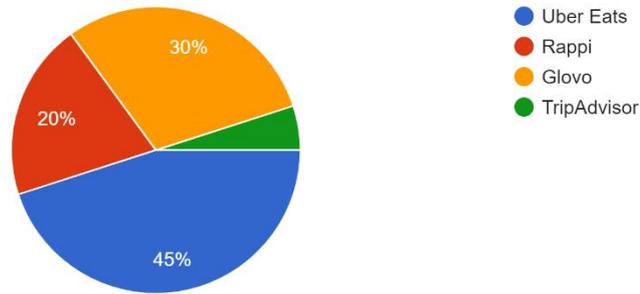


Figura 53. Respuestas de la pregunta 3 del interés generado.

¿Visitaría los establecimientos que sus amigos le recomiendan mediante la aplicación Recomendando?

20 respuestas

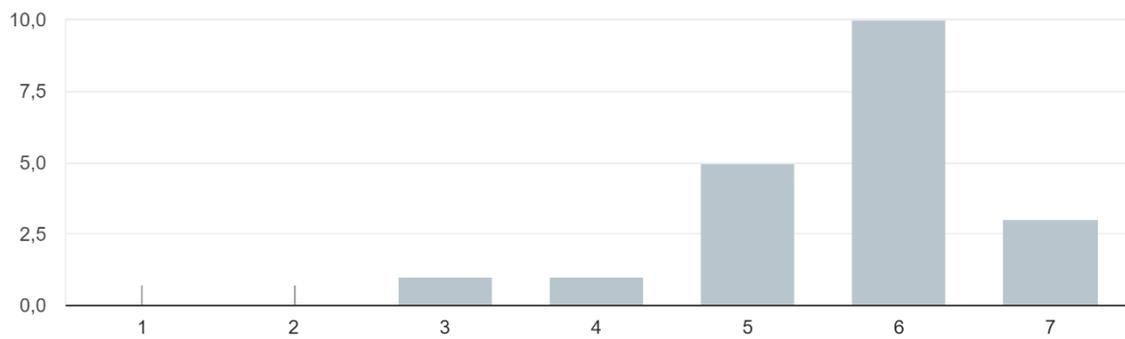


Figura 54. Respuestas de la pregunta 1 para la utilidad percibida.

¿Le parece suficiente el tiempo de espera de 5 minutos antes de que se cierre la pantalla con el código de la promoción?

20 respuestas

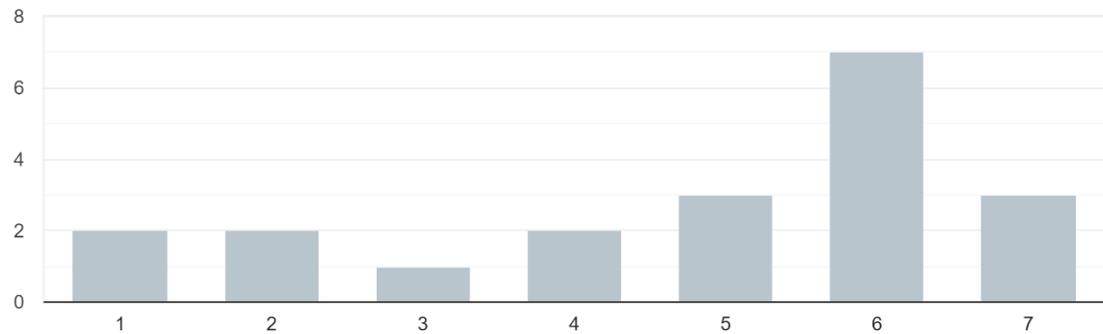


Figura 55. Respuestas de la pregunta 2 para la utilidad percibida.

¿La idea de ganar recopuntos le incentivaron a visitar y recomendar establecimientos?

20 respuestas

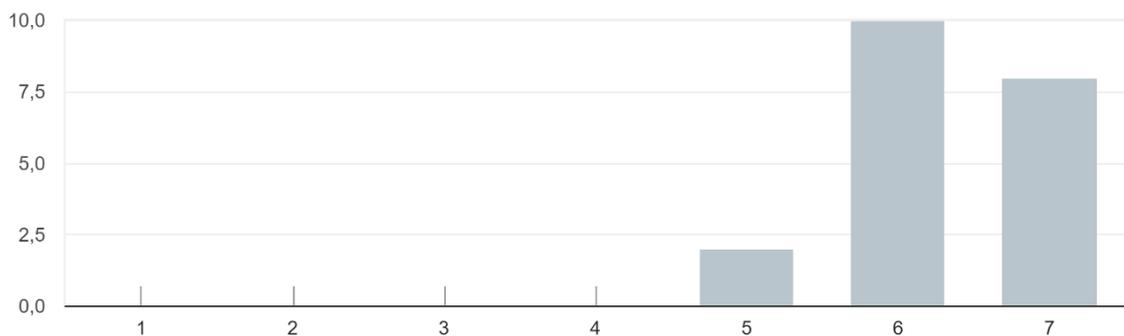


Figura 56. Respuestas de la pregunta 3 para la utilidad percibida.

¿Le resulta fácil recomendar un establecimiento a sus amigos?

20 respuestas

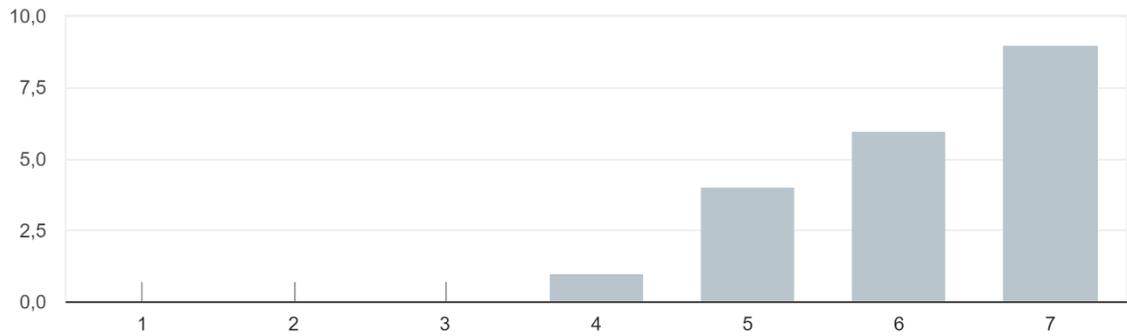


Figura 57. Respuestas de la pregunta 1 para la facilidad de uso percibida.

¿La idea de ganar recopuntos le incentivaron a visitar y recomendar establecimientos?

20 respuestas

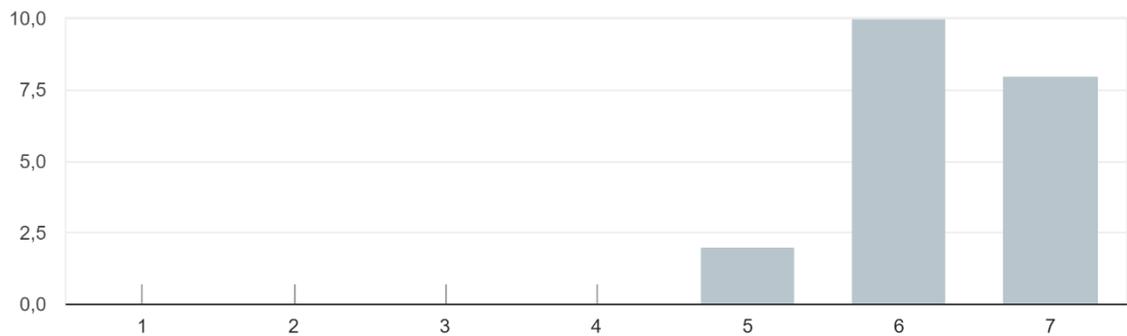


Figura 58. Respuestas de la pregunta 2 para facilidad de uso percibida.

¿Se entendió lo que son las colecciones e intereses?

20 respuestas

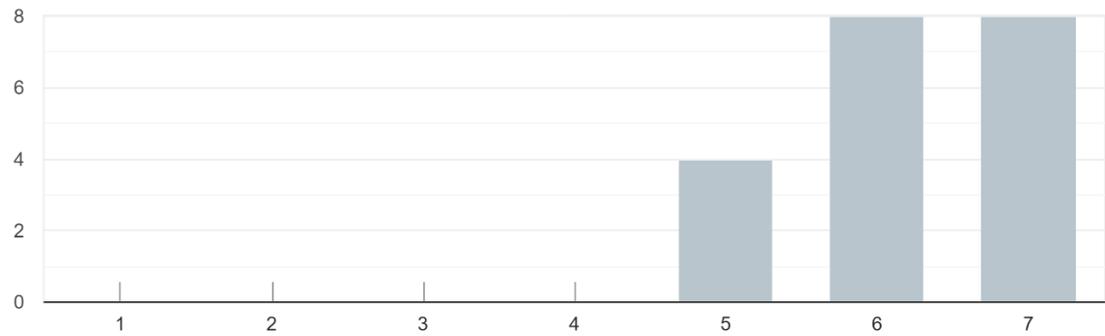


Figura 59. Respuestas de la pregunta 3 para facilidad de uso percibida.