



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESARROLLO DE UNA APLICACIÓN MÓVIL EN ANDROID PARA PERSONAS CON POCOS RECURSOS

Autor: Jose Rafael Martín Torre

Director: Benjamin C.Lok

Co-Director: Jacob Stuart

Julio 2022, Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

**DESARROLLO DE UNA APLICACIÓN MÓVIL EN ANDROID PARA
PERSONAS CON POCOS RECURSOS**

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2021/22 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.

Fdo.: Jose Rafael Martín Torre

Fecha: 06 / 06 / 2022



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Benjamin C.Lok

Fecha: 06 / 06 / 2022



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESARROLLO DE UNA APLICACIÓN MÓVIL EN ANDROID PARA PERSONAS CON POCOS RECURSOS

Autor: Jose Rafael Martín Torre

Director: Benjamin C.Lok

Co-Director: Jacob Stuart

Julio 2022, Madrid

Agradecimientos

En primer lugar, quería agradecer a Benjamin C.Lok, director del proyecto y a Jacob L.Stuart, codirector del proyecto, por toda la ayuda brindada durante estos nueve meses en los que hemos trabajado juntos para desarrollar el proyecto.

También agradecer a mi familia todo el esfuerzo que han hecho durante estos años para ofrecerme la mejor educación posible y brindarme la oportunidad de haber podido estudiar Ingeniería de Telecomunicaciones en ICAI y este cuarto curso en la Universidad de Florida.

Gracias.

DESARROLLO DE UNA APLICACIÓN MÓVIL EN ANDROID PARA PERSONAS CON POCOS RECURSOS

Autor: Jose Rafael Martín Torre

Director: Benjamin C.Lok

Entidad Colaboradora: Universidad de Florida – UF (Gainesville, Florida - USA)

RESUMEN DEL PROYECTO

Este proyecto consiste en el desarrollo de una aplicación para móvil (F4U - Food For You) destinada a personas con dificultades económicas para ayudarlas a poder adquirir alimentos y artículos de primera necesidad en establecimientos adheridos a un programa de ayuda para personas que sufren de inseguridad alimentaria.

El objetivo principal ha sido ayudar a estas personas a encontrar este tipo de establecimientos de una manera rápida y sencilla a través de un mapa en el que puedan localizar todos los establecimientos para poder elegir entre ellos en función de su localización.

Palabras clave: FIREBASE, Front-End, Back-End, ANDROID STUDIO, SDK Maps, API

1. Introducción

Con el paso de los años, la pobreza en los Estados Unidos ha ido aumentando, dejando a familias sin la posibilidad de conseguir todos los alimentos necesarios para una dieta equilibrada y balanceada.

Según cifras de la USDA (US Department of Agriculture) en el año 2020, 12,7 millones de los hogares estadounidenses (10,5% de todos los hogares estadounidenses) se encontraban por debajo del umbral de seguridad alimentaria baja o muy baja. [1]

La Doctorada en Ciencias de la Alimentación y Nutrición Humana de la Universidad de Florida, Karla Shelnut, quiso buscar una solución a este problema, y junto a un grupo de personas del programa ampliado de educación alimentaria y nutricional de la Universidad de Florida, presentó su idea al Dr. Benjamin C.Lok. La idea era desarrollar una aplicación para móvil (Android) que permitiera a las personas que están pasando por una situación de inseguridad alimentaria, encontrar, tanto Bancos de Alimentos, como Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria (SNAP), cercanos a su ubicación, para poder visitarlos y conseguir comida gratuita en esos Bancos de Alimentos, o a un menor

precio debido a los descuentos aplicados que pueden utilizar aquellas personas que se encuentran dentro del programa SNAP (Programa que permite a familias de al menos 3 personas, que tienen unos ingresos mensuales no superiores a 1.800\$, disfrutar de esos descuentos anteriormente mencionados).

Por los motivos de sencillez, rapidez y comodidad, una aplicación para móvil era la mejor solución a la necesidad debido a su portabilidad y uso en cualquier instancia y lugar. La mayor parte de las personas en la actualidad poseen un dispositivo móvil, y con tan solo el acceso a su ubicación, la aplicación será capaz de dirigirles al Banco de Alimentos o al Comercio Minorista del Programa de Asistencia Nutricional Suplementaria más cercano. Se les facilitará información acerca de los horarios de los establecimientos, podrán obtener las direcciones exactas para llegar a ellos, podrán visitar sus páginas Web y llamar por teléfono... todo esto desde un dispositivo móvil.

Para poder asegurar ese tiempo de respuesta del que se ha hablado antes, se ha hecho uso de la plataforma, FIREBASE. Esta plataforma permite manejar toda la información mediante el almacenamiento en la nube. Se ha hecho uso del Cloud Computing, otorgando a los usuarios de capacidades informáticas que no requieren de interacción humana con el proveedor del servicio, que es capaz de servir a múltiples clientes según la demanda sin que estos tengan noción alguna de la ubicación de estos recursos. Mediante el uso de esta tecnología, se consigue un mayor control y optimización de los procesos. [2]

2. Definición del proyecto

Inicialmente se llevó a cabo un proceso de análisis de las plataformas tecnológicas que se ofrecían en el mercado para poder desarrollar el proyecto. Tras ese análisis se escogieron las que se consideraron más adecuadas para alcanzar los requisitos establecidos.

- La primera plataforma que se escogió fue GOOGLE MAPS PLATFORM. Esta plataforma se usa para situar mapas en aplicaciones de móvil. Se escogió ya que Google es el principal proveedor de servicios de mapas.
- La segunda plataforma que se escogió fue ANDROID STUDIO. Existen dos plataformas principales para el desarrollo de aplicaciones de móvil, ANDROID STUDIO y REACT. Se escogió ANDROID STUDIO sobre REACT debido a que

ANDROID STUDIO utiliza como lenguaje de desarrollo un lenguaje de programación con el que ya se ha trabajado anteriormente, Java.

- La tercera plataforma que se escogió fue FIREBASE. Existen dos plataformas principales para el uso de la nube y el almacenamiento de datos, AMAZON WEB SERVICES (AWS) y FIREBASE. Se escogió FIREBASE sobre AMAZON WEB SERVICES (AWS) debido a que FIREBASE es propiedad de Google, y al utilizar la plataforma de GOOGLE MAPS PLATFORM la integración de ambas plataformas era más sencilla.

Para poder hacer uso de estas tres plataformas, ha sido necesario llevar a cabo un proceso previo de autoformación basado en estudio de documentación extraída de las propias plataformas tecnológicas y de la visualización de tutoriales en formato de vídeo.

Las características principales de las tres plataformas seleccionadas para el desarrollo de la aplicación para móvil son las siguientes:

- ANDROID STUDIO. Es el Framework de desarrollo con el que se ha llevado a cabo tanto el diseño del interfaz de la aplicación con la que el usuario interactúa, como el desarrollo de la parte del servidor para poder manejar las peticiones y acciones de los usuarios cuando interactúan con la aplicación. Se ha hecho uso tanto del lenguaje XML para la parte del diseño de la interfaz a través de las “activities” que son las distintas pantallas que tiene la aplicación, como del lenguaje JAVA para la parte de desarrollo del código fuente de esas “activities”, es decir, para darles funcionalidades a los elementos de las pantallas de la aplicación.
- GOOGLE MAPS PLATFORM. Mediante esta plataforma se ha podido usar la API de Maps SDK para Android, con la que se pudo añadir un mapa en la aplicación. En el mapa se sitúan tanto la localización del usuario, como la de los marcadores de los establecimientos que se ofrecen a los usuarios (Bancos de Alimentos y Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria). A través de este mapa los usuarios son capaces de ver qué establecimientos tienen cerca de su ubicación y obtener información sobre ellos.

- FIREBASE. Plataforma de Google que permite el manejo y el almacenamiento de los usuarios mediante una base de datos en tiempo real del tipo NoSQL (base de datos que almacena los elementos con el formato clave-valor, en lugar de almacenar los elementos en tablas con el formato entidad-relación como se hace con SQL), utilizando la plataforma Cloud de Google. Todo esto a través de distintos SDK que permiten la interacción del cliente con estos servicios. A través de FIREBASE se lleva a cabo también la autenticación de los usuarios de la aplicación, para verificar que los mismos utilizan correos electrónicos reales.

El desarrollo de la aplicación se dividió principalmente en dos fases, y mediante la metodología Agile, en la que cada semana se establecían las tareas a realizar, se ha podido desarrollar el proyecto de manera satisfactoria y en los plazos esperados.

- La primera fase consistió en el diseño de la aplicación. A partir de los requisitos funcionales definidos, se llevaron a cabo, mediante el uso de la plataforma Visual Paradigm Online, los diagramas de casos de uso y de navegación para poder tener una idea clara del funcionamiento de la aplicación y las acciones que puede llevar a cabo el usuario con ella. Durante esta primera fase también se llevaron a cabo, mediante el uso de la plataforma Figma, distintos diseños de la interfaz gráfica de la aplicación, de manera que, tras una evaluación de los mismos, se pudo elegir el diseño más adecuado, sencillo y atractivo visualmente.
- En la segunda fase se realizaron dos tipos de tareas. Por un lado, se llevó a cabo el desarrollo puro de la aplicación mediante el uso de ANDROID STUDIO, GOOGLE MAPS PLATFORM y FIREBASE. Esto permitió conseguir una aplicación completamente funcional, sencilla y fácil de usar.
Por otra parte, se llevaron a cabo las tareas de testing. Inicialmente se probó la aplicación con un grupo de personas del programa ampliado de educación alimentaria y nutricional de la Universidad de Florida. Durante esta sesión de testing, mediante el uso de la aplicación se pudieron identificar fallos, procesos poco lógicos y aspectos a mejorar de la misma. Después de esta fase de testing, se desarrollaron otras dos fases conocidas, la fase de testing de Friends & Family, en la que un ámbito reducido de personas (usuarios de prueba) se descargó un archivo .APK de la aplicación de móvil para comprobar su funcionamiento e identificar errores, y la fase

de “User Testing”, en la que se subió la aplicación al GOOGLE PLAY CONSOLE para poder llevar a cabo la distribución de la aplicación, de tal manera que solo algunas personas (usuarios reales) mediante una autorización previa pudieron descargarse la aplicación desde el Google Play Store.

El usuario cuando se encuentra dentro de la aplicación tiene la posibilidad de navegar por tres pantallas.

- En una pantalla es capaz de ver en un mapa su ubicación y la ubicación de distintos establecimientos (Bancos de Alimentos y Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria) mediante el uso de marcadores. Pinchando en los marcadores pueden obtener información acerca de esos establecimientos, como información acerca de los horarios, direcciones exactas para llegar a ellos, página Web, número de teléfono... Además, el usuario también es capaz de filtrar por el tipo de establecimiento que desea ver, o Bancos de Alimentos, o Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria, o ambos, y, además, según lo que introduzca el usuario en la barra de búsqueda puede ver aquellos establecimientos que contengan los caracteres introducidos.
- En otra segunda pantalla es capaz de obtener, en forma de lista, la información de los horarios, direcciones exactas para llegar, página Web, número de teléfono... de los establecimientos (Bancos de Alimentos y Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria), con la posibilidad de ordenar por nombre o distancia hasta su ubicación esos establecimientos. Además, el usuario también es capaz de filtrar por el tipo de establecimiento que desea ver, o Bancos de Alimentos, o Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria, o ambos, y además, según lo que introduzca el usuario en la barra de búsqueda, puede ver aquellos establecimientos que contengan los caracteres introducidos.
- Y en otra tercera pantalla es capaz de configurar la aplicación. Puede cambiar el idioma entre inglés o español, puede seleccionar el tipo de notificaciones que desea recibir y puede cambiar el tiempo que tiene que transcurrir entre ellas. En esta pantalla también es capaz de eliminar su usuario de la plataforma.

Además de lo anterior, existen otras tres pantallas por las que pueden navegar los usuarios, asociadas con el mantenimiento de los mismos en la aplicación.

- En una pantalla puede iniciar sesión y entrar en la aplicación. Para ello el usuario debe estar registrado anteriormente y debe haber autenticado su cuenta. Para esto, se hace uso de FIREBASE, con su real-time database, para almacenar esos usuarios registrados y verificar que existen cuando intenten iniciar sesión, y con su plataforma de autenticación, para comprobar que los usuarios están autenticados cuando intenten iniciar sesión. Además, se permite a los usuarios iniciar sesión con su cuenta de Google, de manera que toda la parte de la autenticación la lleva Google, pero la información del usuario sigue añadiéndose a la real-time database.
- En otra segunda pantalla puede registrarse en la aplicación rellenando unos campos obligatorios que son: correo electrónico, nombre de usuario y contraseña. Como se ha dicho anteriormente, ese nuevo usuario registrado se añade a la base de datos, pero hasta que no autentica su usuario, no se le permite iniciar sesión y entrar a la aplicación. Además, se les permite a los usuarios registrarse con su cuenta de Google, de manera que toda la parte de la autenticación la lleva Google, pero la información del usuario sigue añadiéndose a la real-time database.
- En otra tercera pantalla puede solicitar un cambio de contraseña introduciendo su correo electrónico. A través de FIREBASE, se le enviará un correo con un enlace en el que podrá cambiar su contraseña. En ese instante, se guarda de nuevo el usuario en la real-time database con su nueva contraseña.

3. Descripción de las herramientas y del sistema

- FIREBASE. Esta herramienta permite usar elementos de análisis, autenticación, almacenamiento en bases de datos en tiempo real, mensajería en tiempo real... Todo esto mediante el uso de los SDK que se proporcionan y a través de los cuales se lleva a cabo la interacción entre la parte del cliente (Aplicación) y la parte del Back-End, sin necesidad de establecer un intermediario para el uso de estos servicios, es decir, el formato request-response del modelo HTTP ya no se utiliza. Esto permite a los desarrolladores simplificar su trabajo, ya que no se necesita gestionar los servidores ni escribir APIs, FIREBASE actúa como servidor y como API. [3]

De todas las plataformas que ofrece FIREBASE, se ha hecho uso de dos de ellas.

- La base de datos en tiempo real, con la que se lleva a cabo el almacenamiento de los usuarios que acceden a la aplicación usando un formato NoSQL.
- La plataforma de autenticación de usuarios, que sirve para corroborar que los usuarios que se dan de alta en la aplicación son reales y tienen un correo electrónico real.
- **GOOGLE MAPS PLATFORM** se obtiene una API con la que se puede integrar en la aplicación un mapa mediante el uso de la SDK de Google Maps para Android. Con este SDK, en el mapa se puede observar tanto la localización del usuario, como la de los distintos establecimientos que se ofrecen a los usuarios (Bancos de Alimentos y Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria). Además, ofrece un análisis de métricas acerca del número de solicitudes que recibe la API para poder ver cuántas fueron satisfactorias y cuántas erróneas.
El funcionamiento de esta plataforma es muy simple. Lo que hay que hacer es obtener una credencial de clave API, que servirá para poder integrar el SDK de Maps en la aplicación, y añadir la clave API a nuestro proyecto de ANDROID STUDIO en el manifest. A partir de ese momento ya se puede utilizar y visualizar el mapa en la aplicación Android.
- **SISTEMA RESULTANTE.** Como resultado, se obtiene el sistema final, que es la aplicación, mediante el uso de ANDROID STUDIO, FIREBASE y GOOGLE MAPS PLATFORM. El usuario a través de la aplicación se comunica por separado con FIREBASE y GOOGLE MAPS PLATFORM. Todo esto a través de todas las funcionalidades desarrolladas con ANDROID STUDIO. Por un lado, con el uso de las SDK de FIREBASE se pueden manejar los usuarios que usan la aplicación, de manera que se habilita la capacidad de autenticar y guardar la información del usuario que está utilizando la aplicación y, por otro lado, con el uso de la SDK de Google Maps, se puede mostrar el mapa al usuario que está usando la aplicación con toda la información.

4. Resultados

El resultado del proyecto ha sido una aplicación para móvil (ANDROID) totalmente funcional y sencilla de utilizar que cumple con todos los requisitos definidos por los promotores de la idea.

Esta aplicación para móvil se encuentra subida a la plataforma GOOGLE PLAY CONSOLE. Por medio de esta plataforma se puede llevar a cabo la distribución de la aplicación de tal manera que las personas la pueden descargar desde Google Play Store. Actualmente, solo determinadas personas previamente autorizadas, pueden descargarse la aplicación ya que se encuentra en proceso final de testing.

A continuación, se van a exponer algunos de los aspectos más destacados de la aplicación desarrollada.

- **Vista de la Aplicación**

La validación de la interfaz gráfica de la aplicación de móvil y el correcto funcionamiento de la misma, se llevó a cabo a través de la plataforma de ANDROID STUDIO. Esta plataforma permitió comprobar el correcto funcionamiento de la aplicación de móvil gracias a su herramienta de “run/debug” y mediante el uso de un dispositivo móvil Android, que se modificó para que tuviera opciones de desarrollador, fue posible correr la aplicación en el mismo.

A continuación se presentan las pantallas de la aplicación.

- Ilustración 1 pantalla del mapa

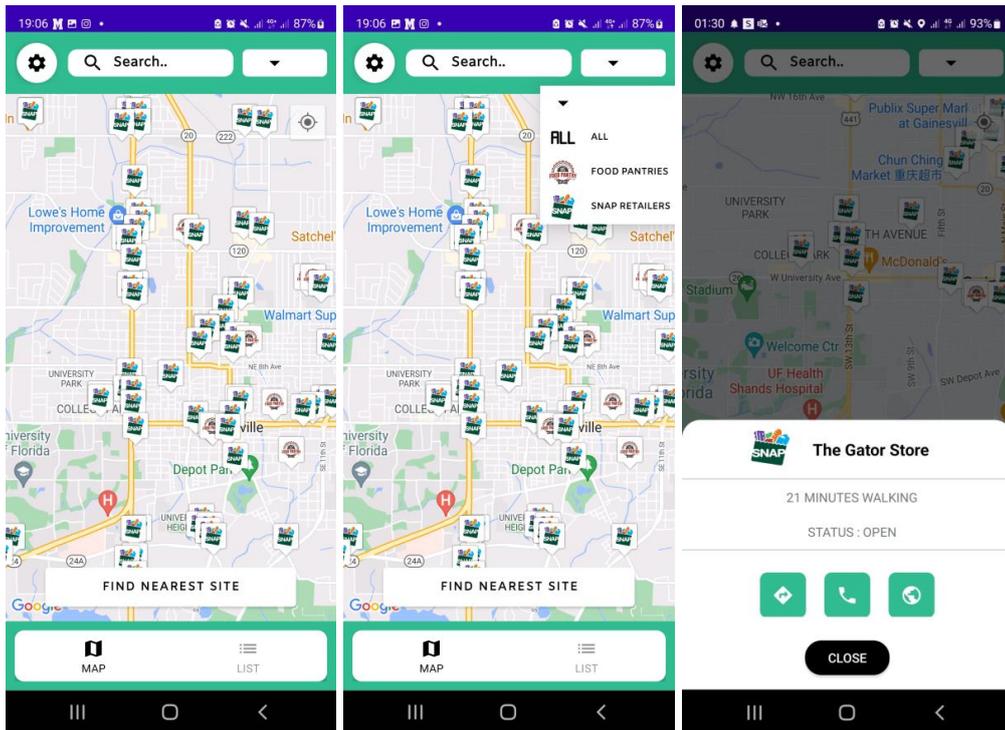


Ilustración 1: Pantalla del Mapa

- Ilustración 2 pantalla de la lista

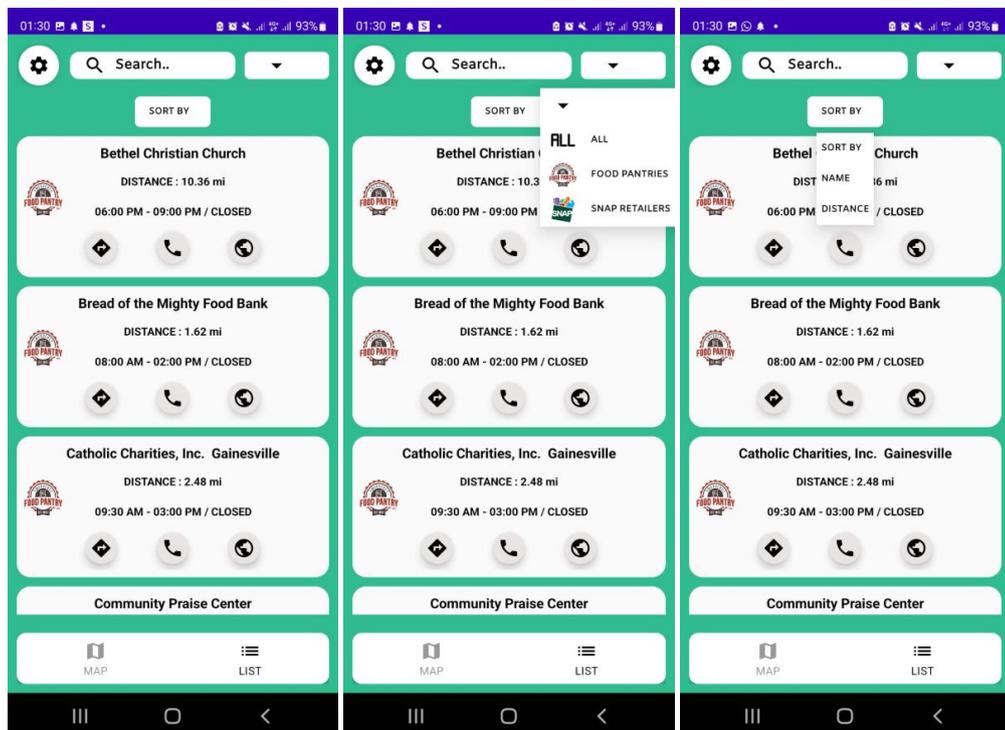


Ilustración 2: Pantalla de la Lista

- Ilustración 3 pantalla de configuración

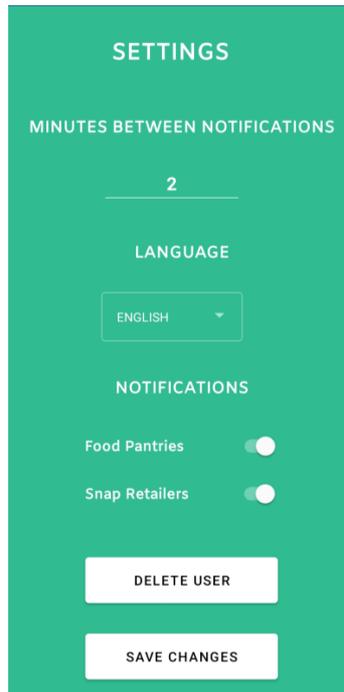


Ilustración 3: Pantalla de Configuración

- Ilustración 4 pantallas de inicio de sesión, registro de usuarios y recuperación de contraseña

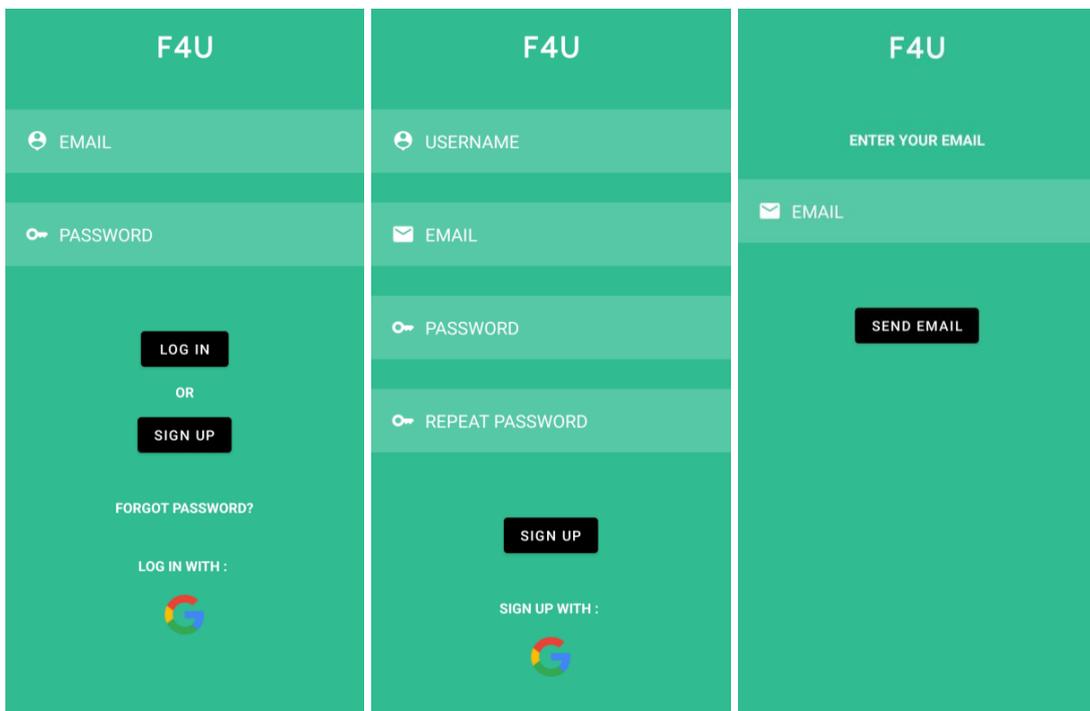


Ilustración 4: Pantallas de Inicio de Sesión, Registro y Recuperación de Contraseña

- **Persistencia**

Las comprobación de las funcionalidades relacionadas con el almacenamiento y la autenticación de los usuarios que usan la aplicación móvil se han verificado mediante el uso de herramientas de análisis disponibles en la plataforma FIREBASE. En la Ilustración 5 se muestran los resultados gráficos obtenidos en el proceso de verificación.

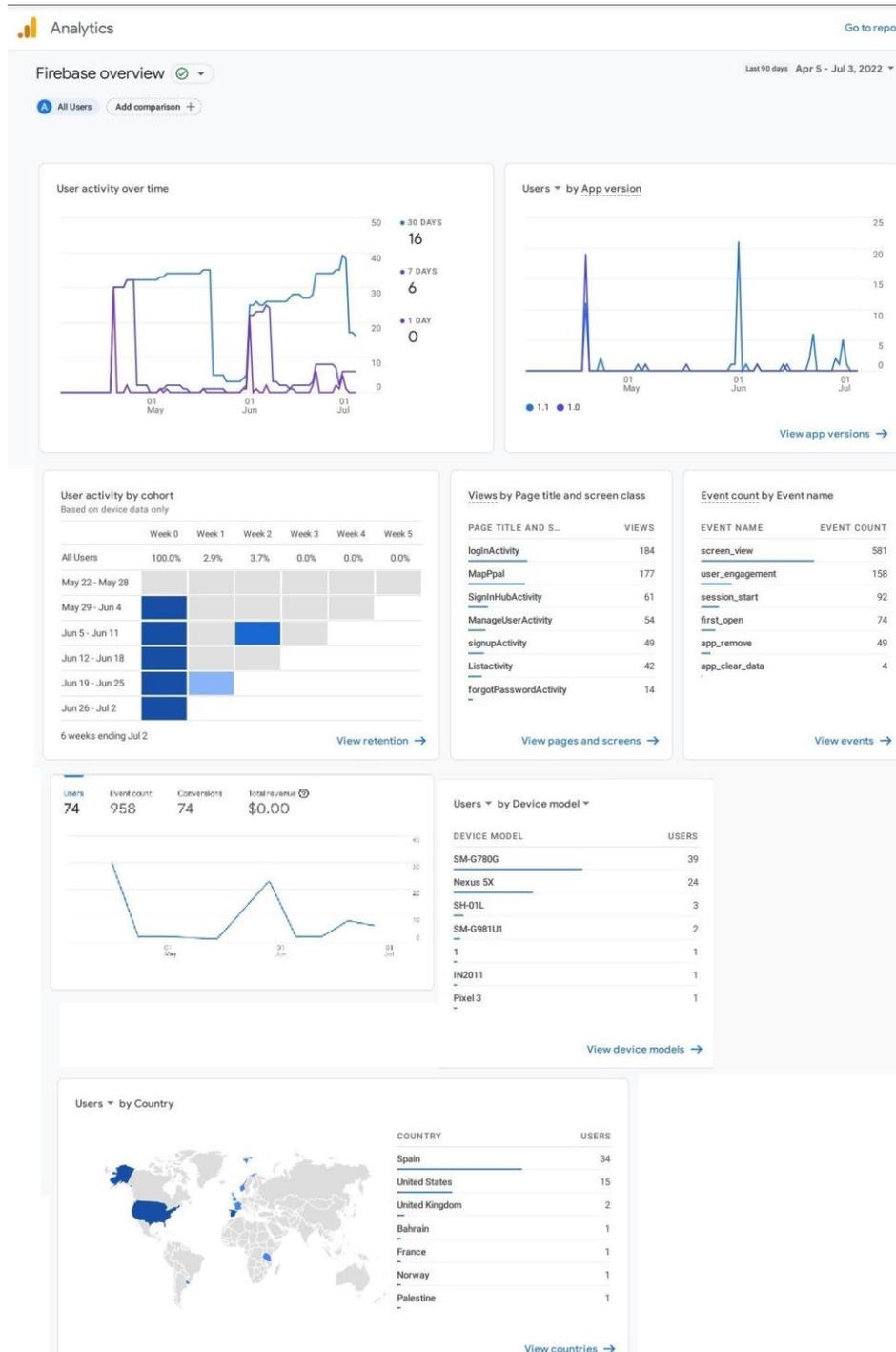


Ilustración 5. Plataforma de Análisis de FIREBASE

- **Solicitudes de Carga de Mapas**

La comprobación de las solicitudes de carga de mapas se llevó a cabo a través de la herramienta de visualización del tráfico por código de respuesta que ofrece GOOGLE MAPS PLATFORM. En la Ilustración 6 se pueden observar las solicitudes de carga de mapas de los últimos 30 días.

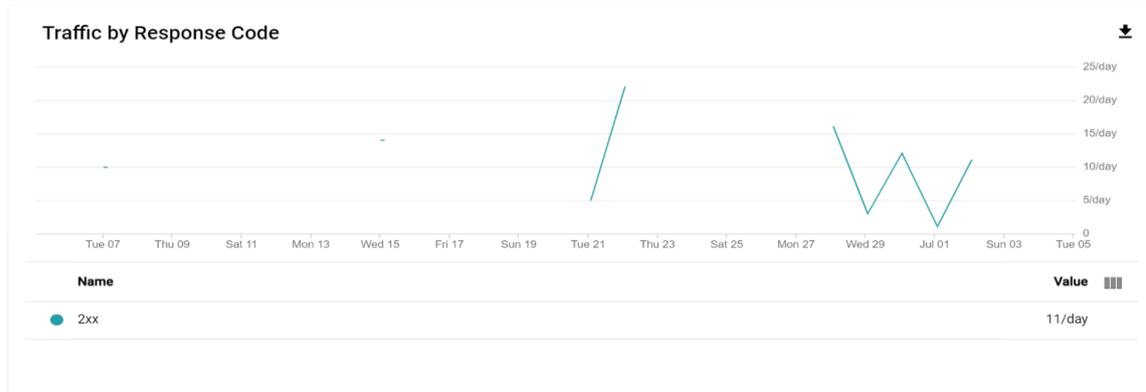


Ilustración 6. Tráfico por código de Respuesta

- **Usuarios de Prueba y Usuarios Reales**

Gracias a los diferentes procesos de testing que se han realizado durante la fase de construcción de la aplicación de móvil se han podido identificar aspectos necesarios a corregir de la misma. En estos procesos han participado usuarios de prueba y usuarios reales que han podido realizar sugerencias a cerca de la usabilidad de la aplicación.

5. Conclusiones

Para poder sacar conclusiones, se deben comparar los objetivos principales definidos al principio del proyecto y su cumplimiento tras el desarrollo del mismo.

- Desarrollar una aplicación para móvil sencilla y “friendly” que los usuarios puedan usar. Este objetivo se ha cumplido.
- Implementar la SDK de Google Maps para poder situar un mapa en la aplicación. Este objetivo se ha cumplido.
- Permitir que la aplicación envíe notificaciones a los usuarios. Este objetivo se ha cumplido.
- **Desarrollar una aplicación en los idiomas español e inglés. Este objetivo se ha cumplido.**

- Diseñar la base de datos para el almacenamiento de la información de los usuarios que accedan a la aplicación. Este objetivo se ha cumplido.
- Diseñar el sistema de autenticación y autorización de los usuarios que accedan a la aplicación. Este objetivo se ha cumplido.
- Llevar a cabo una fase final de testing de la aplicación con usuarios de prueba y usuarios reales de la aplicación. Este objetivo falta por finalizarse.

Tras todo el trabajado realizado, se puede afirmar que se han completado la gran mayoría de los objetivos principales, ya que se ha conseguido desarrollar una herramienta de ayuda para las personas con inseguridad alimentaria y problemas económicos, que les permite localizar y obtener información acerca de los establecimientos dispuestos en la aplicación para conseguir alimentos y otros productos de primera necesidad de una manera rápida y sencilla.

Como trabajos futuros se pueden destacar fundamentalmente tres:

- Finalizar la fase de testing de la aplicación con usuarios reales de la aplicación censados por la Universidad de Florida.
- Creación de una base de datos en FIREBASE para el almacenamiento de la información de los establecimientos.
- Desarrollo de una plataforma de mantenimiento de los datos de los establecimientos (Alta, baja, modificación) para facilitar la actualización de los datos sin necesidad de intervención de personal técnico.

Estos tres grupos de tareas se van a realizar a partir del mes de septiembre conjuntamente con la Universidad de Florida.

6. Referencias

[1] Alisha Coleman-Jensen y Laura Hales, U.S Department of Agriculture.

<https://www.ers.usda.gov/topics/food-nutrition-assistance/food-security-in-the-u-s/interactive-charts-and-highlights/>

[2] Bill Loeffler, “Cloud Computing: What is Infrastructure as a Service”.

[https://docs.microsoft.com/en-us/previous-versions/technet-magazine/hh509051\(v=msdn.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/technet-magazine/hh509051(v=msdn.10)?redirectedfrom=MSDN)

[3] Golap Gunjan Barman, “What is Firebase? All you need to know about Firebase”.

<https://androidapps-development-blogs.medium.com/what-is-firebase-all-you-need-to-know-about-firebase-68f8a8a363d0>

[4] Nathan Thomas, “How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website”.

<https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>

DEVELOPMENT OF AN ANDROID MOBILE APPLICATION FOR PEOPLE WITH LIMITED RESOURCES

Author: Jose Rafael Martín Torre

Supervisor: Benjamin C.Lok

Collaborating Entity: University of Florida – UF (Gainesville, Florida)

ABSTRACT

This project consists of the development of a mobile application (F4U - Food For You) aimed at people with economic difficulties to help them acquire food and basic necessities in establishments that are part of a program to help people suffering from food insecurity.

The main objective has been to help these people to find such establishments quickly and easily through a map where they can locate all the establishments to choose from them according to their location.

Keywords: FIREBASE, Front-End, Back-End, ANDROID STUDIO, SDK Maps, API

1. Introduction

Over the years, poverty in the United States has been increasing, leaving families unable to get all the food they need for a balanced diet.

According to USDA (US Department of Agriculture) figures in 2020, 12,7 million US households (10,5% of all US households) were below the low or very low food security threshold. [1]

University of Florida Food Science and Human Nutrition Ph.D. graduate Karla Shelnett wanted to find a solution to this problem, and along with a group of people from the University of Florida's expanded food and nutrition education program, presented her idea to Dr. Benjamin C.Lok. The idea was to develop a mobile (Android) application that would allow people who are experiencing food insecurity to find both Food Banks and Supplemental Nutrition Assistance Program (SNAP) Retail Stores near their location to visit and get free food at those Food Banks or at a lower cost price due to the discounts applied that can be used by those people who are within the SNAP program (Program that allows

families of at least 3 people, who have a monthly income of no more than \$1.800, to enjoy those aforementioned discounts).

For the reasons of simplicity, speed, and convenience, a mobile application was the best solution to the need due to its portability and use in any instance and place. Most people today own a mobile device, and by simply accessing their location, the application will be able to direct them to the nearest Food Bank or Supplemental Nutrition Assistance Program Retailer. They will be provided with information about the opening hours of the establishments, they will be able to get exact directions to them, they will be able to visit their websites and call them by phone... all this from a mobile device.

To ensure the response time mentioned above, the FIREBASE platform has been used. This platform makes it possible to manage all the information through cloud storage. Use has been made of Cloud Computing, providing users with computing capabilities that do not require human interaction with the service provider, who can serve multiple customers on-demand without them having any notion of the location of these resources. With this technology, greater control and optimization of processes are achieved. [2]

2. Project Definition

Initially, a process of analysis of the technological platforms offered in the market was carried out to develop the project. After this analysis, the most suitable platforms were chosen to meet the established requirements.

- The first platform chosen was GOOGLE MAPS PLATFORM. This platform is used to place maps in mobile applications. It was chosen since Google is the main provider of map services.
- The second platform chosen was ANDROID STUDIO. There are two main platforms for mobile application development, ANDROID STUDIO and REACT. ANDROID STUDIO was chosen over REACT since ANDROID STUDIO uses as its development language a programming language that has been worked with before, Java.
- The third platform chosen was FIREBASE. There are two main platforms for cloud usage and data storage, AMAZON WEB SERVICES (AWS) and FIREBASE.

FIREBASE was chosen over AMAZON WEB SERVICES (AWS) because FIREBASE is owned by Google, and by using the GOOGLE MAPS PLATFORM the integration of both platforms was easier.

To make use of these three platforms, it has been necessary to carry out a previous process of self-training based on the study of documentation extracted from the technology platforms themselves and the viewing of tutorials in video format.

The main characteristics of the three platforms selected for the development of the mobile application are the following:

- **ANDROID STUDIO.** This is the development framework with which both the design of the application interface with which the user interacts and the development of the server-side to handle the requests and actions of users when interacting with the application have been carried out. The XML language has been used for the interface design through the "activities" that are the different screens of the application, as well as the JAVA language for the development of the source code of these "activities" to give functionalities to the elements of the application screens.
- **GOOGLE MAPS PLATFORM.** Through this platform, it has been possible to use the Maps SDK API for Android, with which it was possible to add a map to the application. On the map are placed both the location of the user, as well as the location of the markers of the establishments that are offered to users (Food Banks and Retailers of the Supplemental Nutrition Assistance Program). Through this map, users can see what establishments are near their location and obtain information about them.
- **FIREBASE.** Google platform that allows the management and storage of users through a real-time database of the NoSQL type (database that stores the elements with the key-value format, instead of storing the elements in tables with the entity-relationship format as it is done with SQL), using the Google Cloud platform. All this through different SDKs that allow client interaction with these services. Through FIREBASE, the authentication of the application users is also carried out, to verify that they are using real e-mails.

The development of the application was mainly divided into two phases, and through the Agile methodology, in which each week the tasks to be performed were established, it has been possible to develop the project satisfactorily and within the expected deadlines.

- The first phase consisted of the design of the application. Based on the defined functional requirements, using the Visual Paradigm Online platform, the use case and navigation diagrams were created to have a clear idea of how the application works and the actions that the user can carry out with it. During this first phase, different designs of the graphical interface of the application were also carried out using the Figma platform, so that, after an evaluation of them, the most appropriate, simple, and visually attractive design could be chosen.
- In the second phase, two types of tasks were carried out. On the one hand, the pure development of the application was carried out using ANDROID STUDIO, GOOGLE MAPS PLATFORM, and FIREBASE. This allowed us to achieve a fully functional, simple, and easy-to-use application.

On the other hand, testing tasks were carried out. Initially, the application was tested with a group of people from the extended food and nutrition education program of the University of Florida. During this testing session, by using the application, bugs, illogical processes, and aspects to be improved could be identified. After this testing phase, two other known phases were developed, the Friends & Family testing phase, in which a reduced scope of people (test users) downloaded an .APK file of the mobile application to test its operation and identify bugs, and the of "User Testing", in which the application was uploaded to the GOOGLE PLAY CONSOLE to carry out the distribution of the application so that only some people (real users) through a previous authorization could download the application from the Google Play Store.

The user when inside the application can navigate through three screens.

- On one screen they can see on a map their location and the location of different establishments (Food Banks and Supplemental Nutrition Assistance Program Retailers) through the use of markers. By clicking on the markers they can get information about those establishments, such as information about opening hours, exact directions to get to them, website, phone number... In addition, the user is also able to filter by the type of establishment they want to see, either Food Banks, or

Supplemental Nutrition Assistance Program Retailers, or both, and, in addition, depending on what the user enters in the search bar they can see those establishments that contain the characters entered.

- On a second screen, the user can obtain, in the form of a list, information on the opening hours, exact directions, website, telephone number... of the establishments (Food Banks and Retailers of the Supplemental Nutrition Assistance Program), with the possibility of sorting by name or distance to the location of these establishments. In addition, the user is also able to filter by the type of establishment they wish to view, either Food Banks, or Supplemental Nutrition Assistance Program Retailers, or both, and in addition, depending on what the user enters in the search bar, they can view those establishments that contain the characters entered.
- And on another third screen, you can configure the application. You can change the language between English and Spanish, you can select the type of notifications you want to receive, and you can change the time that has to elapse between them. On this screen, you are also able to remove your user from the platform.

In addition to the above, there are three other screens that users can navigate through, associated with maintaining them in the application.

- On one screen you can log in and enter the application. To do this, the user must be previously registered and must have authenticated their account. For this, use is made of FIREBASE, with its real-time database, to store those registered users and verify that they exist when they try to log in, and with its authentication platform, to verify that users are authenticated when they try to log in. In addition, users are allowed to log in with their Google account, so all the authentication part is handled by Google, but the user information is still added to the real-time database.
- In another second screen you can register in the application by filling in some mandatory fields which are: e-mail, user name and password. As mentioned above, that new registered user is added to the database, but until he/she authenticates his/her user, he/she is not allowed to log in and enter the application. In addition, users are

allowed to register with their Google account, so all the authentication part is handled by Google, but the user information is still added to the real-time database.

- On a third screen, they can request a password change by entering their email address. Through FIREBASE, you will be sent an email with a link where you can change your password. The user is then saved again in the real-time database with the new password.

3. Description of the tools and the system

- FIREBASE. This tool allows the use of analysis elements, authentication, real-time database storage, real-time messaging... All this by using the SDKs provided and through which the interaction between the client-side (Application) and the Back-End side is carried out, without the need to establish an intermediary for the use of these services, i.e. the request-response format of the HTTP model is no longer used. This allows developers to simplify their work, as there is no need to manage servers or write APIs, FIREBASE acts as both server and API. [3]

Of all the platforms offered by FIREBASE, use has been made of two of them.

- The real-time database, with which the storage of users accessing the application is carried out using a NoSQL format.
- The user authentication platform, which is used to corroborate that the users who register in the application are real and have a real e-mail address.
- GOOGLE MAPS PLATFORM an API is obtained with which a map can be integrated into the application by using the Google Maps SDK for Android. With this SDK, the map can show both the user's location and the location of the different establishments that are offered to users (Food Banks and Retailers of the Supplemental Nutrition Assistance Program). In addition, it offers an analysis of metrics about the number of requests received by the API to see how many were successful and how many were unsuccessful.
The operation of this platform is very simple. What you have to do is to obtain an API key credential, which will be used to integrate the Maps SDK in the application

and add the API key to your ANDROID STUDIO project in the manifest. From that moment on, the map can be used and visualized in the Android application.

- **RESULTING SYSTEM.** As a result, we obtain the final system, which is the application, using ANDROID STUDIO, FIREBASE, and GOOGLE MAPS PLATFORM. The user through the application communicates separately with FIREBASE and GOOGLE MAPS PLATFORM. All this through all the functionalities developed with ANDROID STUDIO. On the one hand, with the use of the FIREBASE SDKs it is possible to manage the users that use the application, so that the ability to authenticate and save the information of the user that is using the application is enabled and, on the other hand, with the use of the Google Maps SDK, the map can be shown to the user that is using the application with all the information.

4. Results

The result of the project has been a fully functional and easy-to-use mobile application (ANDROID) that meets all the requirements defined by the promoters of the idea.

This mobile application is uploaded to the GOOGLE PLAY CONSOLE platform. Through this platform, the distribution of the application can be carried out in such a way that people can download it from the Google Play Store. Currently, only certain previously authorized persons can download the application as it is in the final testing process.

The following are some of the highlights of the developed application.

- **Application View**

The validation of the graphical interface of the mobile application and its correct functioning was carried out through the ANDROID STUDIO platform. This platform allowed checking the correct operation of the mobile application thanks to its "run/debug" tool and through the use of an Android mobile device, which was modified to have developer options, it was possible to run the application on it.

The screenshots of the application are presented below.

- Illustration 1 Map screen

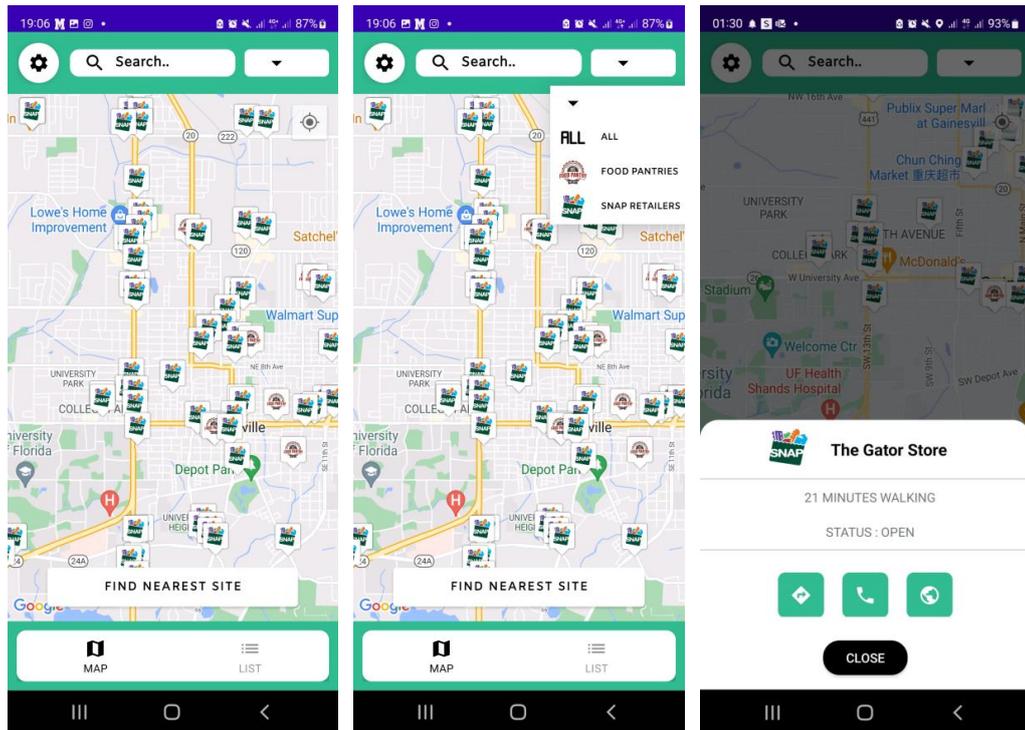


Illustration 1: Map screen

- Illustration 2 List screen

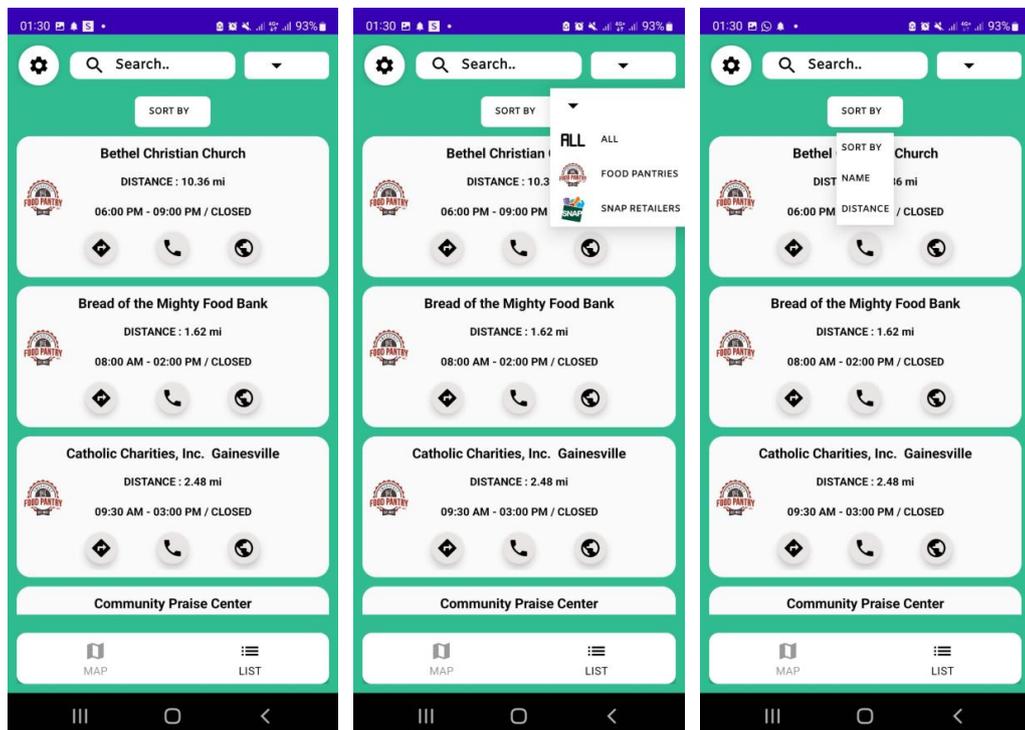


Illustration 2: List screen

- Illustration 3 Configuration screen

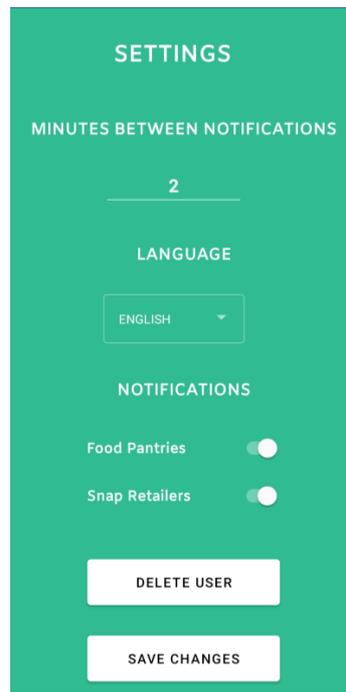


Illustration 3: Configuration screen

- Illustration 4 login, user registration, and password recovery screens

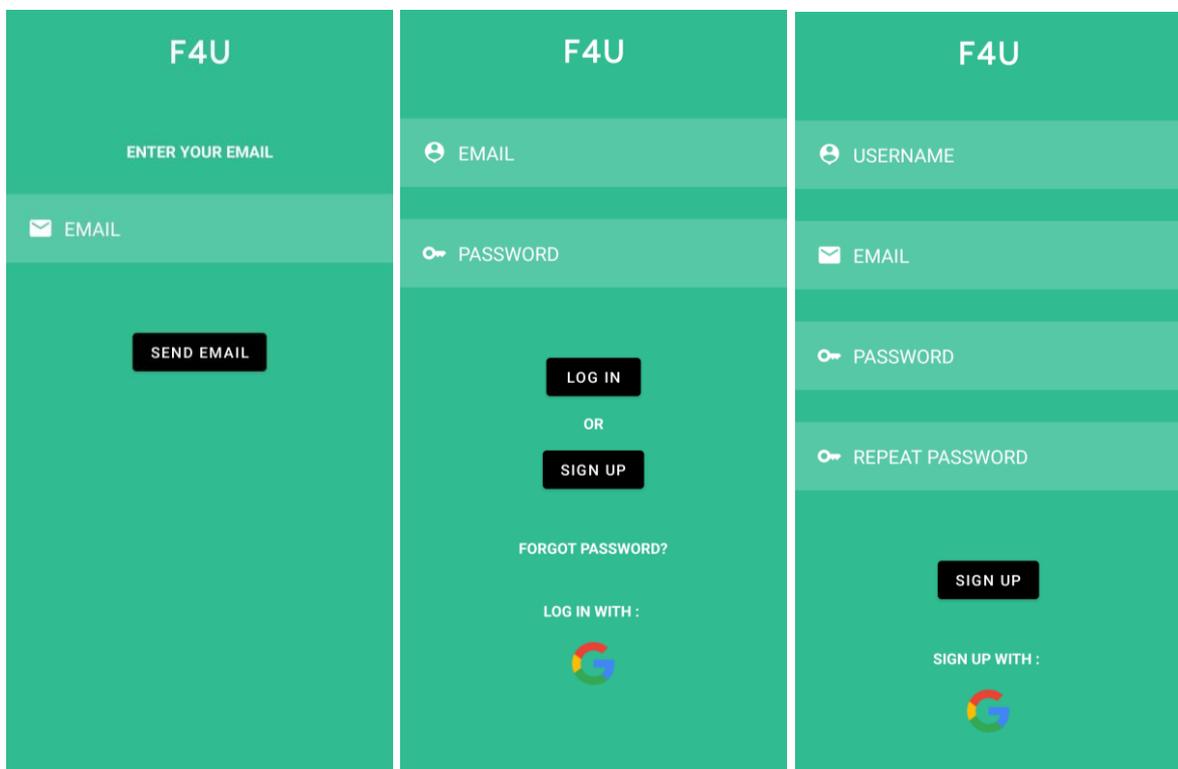


Illustration 4: Login, registration, and password recovery screens.

- Persistence

The verification of the functionalities related to the storage and authentication of users using the mobile application have been verified by using analysis tools available on the FIREBASE platform. Illustration 5 shows the graphical results obtained in the verification process.

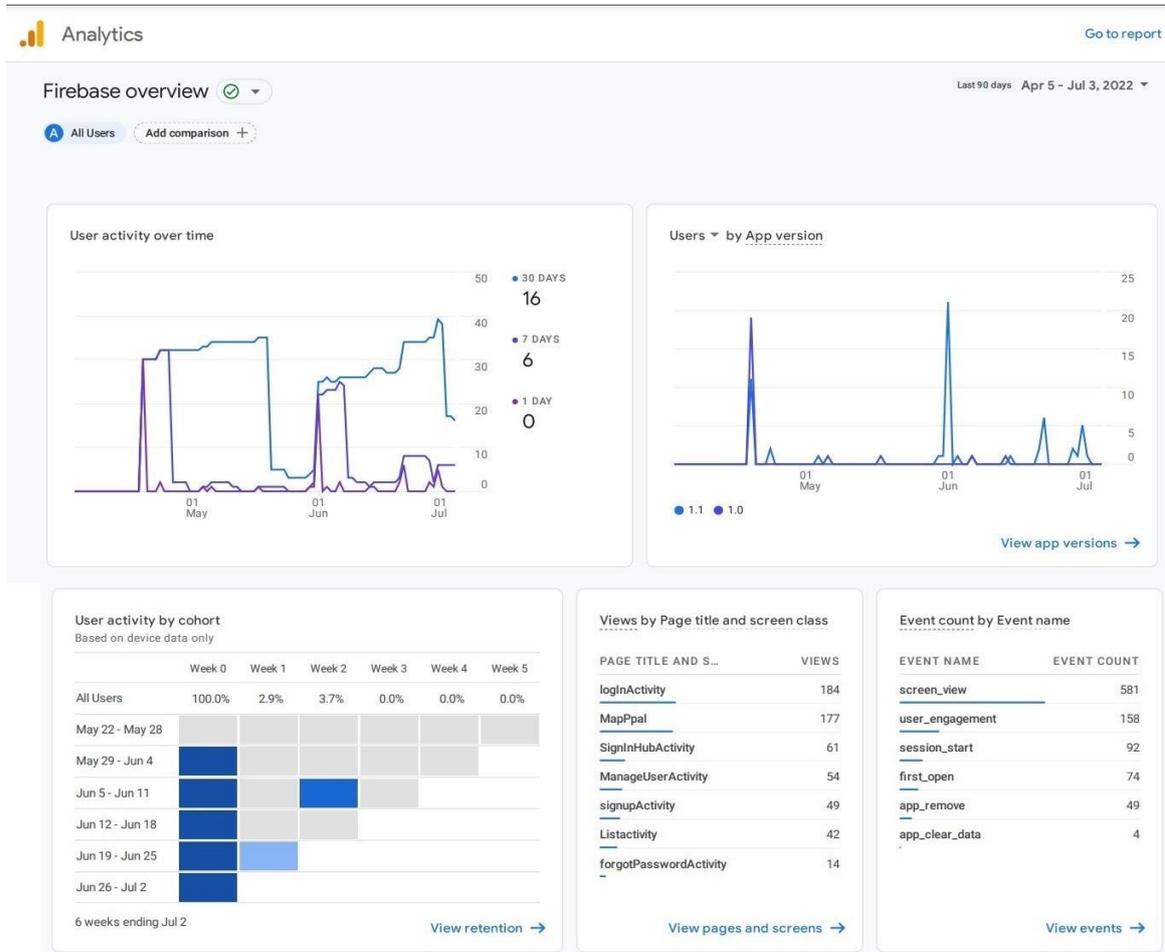


Illustration 5. FIREBASE Analysis Platform

- Map Upload Requests

The verification of the map upload requests was carried out through the traffic visualization tool by response code offered by GOOGLE MAPS PLATFORM. Illustration 6 shows the map loading requests for the last 30 days.



Illustration 6. Traffic by Response Code

- Test Users and Real Users

Thanks to the different testing processes that have been carried out during the construction phase of the mobile application, it has been possible to identify aspects of the application that need to be corrected. These processes have involved test users and real users who have been able to make suggestions about the usability of the application.

5. Conclusions

To conclude, the main objectives defined at the beginning of the project should be compared with their fulfillment after the development of the project.

- To develop a simple and "friendly" mobile application that users can use. This objective has been met.
- Implement the Google Maps SDK to be able to place a map in the application. This objective has been met.
- Allow the application to send notifications to users. This objective has been met.
- Develop an application in Spanish and English. This objective has been met.
- Design the database for storing information on users accessing the application. This objective has been met.
- Design the authentication and authorization system for users accessing the application. This objective has been met.
- To carry out a final testing phase of the application with test users and real users of the application. This objective is yet to be completed.

After all the work done, it can be said that most of the main objectives have been completed, since it has been possible to develop a tool to help people with food and economic insecurity, which allows them to locate and obtain information about the establishments available in the application to obtain food and other basic necessities quickly and easily.

For future work we can highlight three main areas of work:

- Completion of the testing phase of the application with real users of the application surveyed by the University of Florida.
- Creation of a database in FIREBASE for the storage of the information of the establishments.
- Development of an establishment data maintenance platform (registration, cancellation, modification) to facilitate the updating of data without the need for intervention of technical personnel.

These three groups of tasks will be carried out starting in September in conjunction with the University of Florida.

6. References

[1] Alisha Coleman-Jensen and Laura Hales, U.S Department of Agriculture.

<https://www.ers.usda.gov/topics/food-nutrition-assistance/food-security-in-the-u-s/interactive-charts-and-highlights/>

[2] Bill Loeffler, “Cloud Computing: What is Infrastructure as a Service”.

[https://docs.microsoft.com/en-us/previous-versions/technet-magazine/hh509051\(v=msdn.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/technet-magazine/hh509051(v=msdn.10)?redirectedfrom=MSDN)

[3] Golap Gunjan Barman, “What is Firebase? All you need to know about Firebase”.

<https://androidapps-development-blogs.medium.com/what-is-firebase-all-you-need-to-know-about-firebase-68f8a8a363d0>

[4] Nathan Thomas, “How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website”.

<https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>

Índice de la Memoria

<i>Índice de Figuras</i>	3
<i>Notación</i>	5
Capítulo 1. Introducción	6
1.1 Introducción.....	6
1.2 Motivación del Proyecto	7
Capítulo 2. Descripción de las Tecnologías	9
2.1 Fase de Diseño.....	9
2.1.1 Visual Paradigma Online.....	9
2.1.2 Figma.....	11
2.2 Fase de Desarrollo	13
2.2.1 ANDROID STUDIO	13
2.2.2 GOOGLE MAPS SDK.....	16
2.2.3 FIREBASE	17
Capítulo 3. Estado de la Cuestión	19
3.1 Food For All.....	19
3.2 Fresh Food Connect.....	20
3.3 Food For You – F4U	20
Capítulo 4. Definición del Trabajo	22
4.1 Justificación.....	22
4.2 Objetivos	24
4.3 Metodología.....	25
4.3.1 Fall Semester	26
4.3.1.1 Fase de Diseño.....	26
4.3.1.2 Fase de Desarrollo	27
4.3.1.3 Fase de Evaluación	29
4.3.2 Spring Semester.....	29

4.3.2.1 Fase de Desarrollo	29
4.3.2.2 Fase de Testing	31
4.4 Planificación y Estimación Económica	32
4.4.1 Planificación	32
4.4.2 Estimación Económica	32
Capítulo 5. Sistema/Modelo Desarrollado	36
5.1 Análisis y Diseño del Sistema	36
5.1.1 Requisitos Principales	37
5.1.2 Requisitos Adicionales	37
5.2 Desarrollo e Implementación del Sistema	42
5.2.1 Log In	42
5.2.2 Sign Up	46
5.2.3 Reset Password	49
5.2.4 Mapa	50
5.2.5 Lista	56
5.2.6 Configuración	63
Capítulo 6. Análisis de Resultados	71
6.1 Vista de la Aplicación	71
6.2 Persistencia	75
6.3 Solicitudes de Carga de Mapas	78
6.4 Usuarios de Prueba y Usuarios Reales	79
Capítulo 7. Conclusiones y Trabajos Futuros	80
6.1 Conclusiones	80
6.1 Trabajos Futuros	81
Capítulo 8. Bibliografía	83
ANEXO I: Alineación del Proyecto con los ODS	85
ANEXO II: Código de Búsqueda del Establecimiento más Cercano	87
ANEXO III: Código de Ordenación por Distancia y Orden Alfabético	94

Índice de figuras

<i>Figura 1. Diagrama de Casos de Usos</i>	<i>10</i>
<i>Figura 2. Diagrama de Navegación.....</i>	<i>11</i>
<i>Figura 3. Diseño 1 de la Interfaz de la Aplicación</i>	<i>12</i>
<i>Figura 4. Diseño 2 de la Interfaz de la Aplicación</i>	<i>12</i>
<i>Figura 5. Diseño 3 de la Interfaz de la Aplicación</i>	<i>13</i>
<i>Figura 6. Activity de ANDROID STUDIO</i>	<i>14</i>
<i>Figura 7. Entorno de Desarrollo de ANDROID STUDIO</i>	<i>15</i>
<i>Figura 8. Interfaz de GOOGLE MAPS PLATFORM.....</i>	<i>16</i>
<i>Figura 9. Interfaz de FIREBASE.....</i>	<i>18</i>
<i>Figura 10. Real-time Database de FIREBASE</i>	<i>18</i>
<i>Figura 11. Diagrama de Gantt del Proyecto</i>	<i>32</i>
<i>Figura 12. Diagrama de Casos de Uso.....</i>	<i>39</i>
<i>Figura 13. Diagrama de Navegación.....</i>	<i>40</i>
<i>Figura 14. Diseño 1 de la Interfaz de la Aplicación</i>	<i>41</i>
<i>Figura 15. Diseño 2 de la Interfaz de la Aplicación</i>	<i>41</i>
<i>Figura 16. Diseño 3 de la Interfaz de la Aplicación</i>	<i>42</i>
<i>Figura 17. Pantalla de Inicio de Sesión.....</i>	<i>43</i>
<i>Figura 18. Pantalla de Inicio de Sesión con Google</i>	<i>44</i>
<i>Figura 19. Plataforma FIREBASE Log In</i>	<i>44</i>
<i>Figura 20. Pantalla de Registro.....</i>	<i>47</i>
<i>Figura 21. Plataforma FIREBASE Sign Up.....</i>	<i>47</i>
<i>Figura 22. Pantallas de Cambio de Contraseña.....</i>	<i>49</i>
<i>Figura 23. Pantalla del Mapa</i>	<i>51</i>
<i>Figura 24. Marcador del Mapa Seleccionado</i>	<i>52</i>
<i>Figura 25. Establecimiento más Cercano</i>	<i>53</i>
<i>Figura 26. Búsqueda de Establecimientos en el Mapa</i>	<i>54</i>
<i>Figura 27. Filtro de Establecimientos en el Mapa.....</i>	<i>55</i>
<i>Figura 28. Pantalla de la Lista</i>	<i>57</i>

<i>Figura 29. Búsqueda de Establecimientos en la Lista 1</i>	58
<i>Figura 30. Búsqueda de Establecimientos en la Lista 2</i>	59
<i>Figura 31. Ordenación de Establecimientos en la Lista</i>	60
<i>Figura 32. Pantalla de Configuración</i>	63
<i>Figura 33. Ejemplo de Notificación</i>	64
<i>Figura 34. Aplicación en Español</i>	65
<i>Figura 35. Plataforma FIREBASE Delete User</i>	66
<i>Figura 36. Pantalla del Mapa</i>	72
<i>Figura 37. Pantalla de la Lista</i>	73
<i>Figura 38. Pantalla de Configuración</i>	74
<i>Figura 39. Pantallas de Inicio de Sesión, Registro y Recuperación de Contraseña</i>	75
<i>Figura 40. Plataforma de Análisis de FIREBASE 1</i>	77
<i>Figura 41. Plataforma de Análisis de FIREBASE 2</i>	78
<i>Figura 42. Tráfico por Código de Respuesta</i>	78
<i>Figura 43. 17 Objetivos de Desarrollo Sostenible</i>	85

Notación

SDK = “Software Developer Kit” (grupo de herramientas que permiten la programación de aplicaciones para móviles).

API = “Application Programming Interfaces” (normas y protocolos utilizados para crear e integrar software de aplicación, que permiten la comunicación basada en reglas entre dos programas informáticos).

HTTP = “Hypertext Transfer Protocol” (establece los criterios de sintaxis y semántica informática para el establecimiento de la comunicación entre los diferentes elementos que constituyen la arquitectura Web: servidores y clientes).

Framework = Herramienta que provee de una estructura base que sirve de punto de partida, para que los desarrolladores de software puedan elaborar y desarrollar sus proyectos.

XML = “Extensible Markup Language” (lenguaje de marcado que define un conjunto de reglas para la codificación de documentos).

SHA-1 = Función hash criptográfica que genera un 160 bit (20 bytes) hash a partir de cualquier valor de entrada.

Manifest = Archivo de manifiesto que describe información esencial de las aplicaciones de Android.

Agile = La metodología Agile planea el trabajo desde el principio dividiéndolo en “sprints”.

APK = Paquete de Aplicación Android, forma en la que está empaquetada una aplicación de Android con todos los archivos necesarios para poder instalarse.

Bundle = Formato de publicación de una aplicación Android que incluye todos los recursos y el código fuente compilado de la misma.

Testing = Acto de examinar los artefactos y el comportamiento del software mediante la validación y la verificación.

Pop-Up = Pequeños cuadros de diálogo que aparecen en la pantalla del móvil que bloquean temporalmente las interacciones con la vista principal de la aplicación.

Run/Debug = Run es iniciar la aplicación de manera normal. Debug es iniciar la aplicación en modo debug para poder encontrar bugs en la aplicación gracias a unas herramientas.

Capítulo 1. INTRODUCCIÓN

1.1 INTRODUCCIÓN

Millones de personas en todo el mundo sufren de inseguridad alimentaria y viven cerca de la línea de la pobreza. Tras la devastadora epidemia del Covid-19, ha surgido una crisis económica que está agravando el problema de la inseguridad alimentaria, dejando al descubierto la falta de acceso que tienen algunas personas a los bienes de primera necesidad. [1]

Particularmente, en los Estados Unidos la pobreza ha ido aumentando con el paso de los años, dejando a familias sin la posibilidad de conseguir todos los alimentos necesarios para una dieta equilibrada y balanceada.

Los Bancos de Alimentos durante estos años de Covid-19 en los Estados Unidos, han tenido que llevar a cabo mayores esfuerzos para poder llegar al mayor número de personas posible. La inflación, los problemas de suministros de alimentos, la finalización de cheques de ayuda... son los principales culpables de esta situación, que ha obligado a 60 millones de estadounidenses a buscar ayuda. [2]

Según cifras de la USDA (US Department of Agriculture) en el año 2020 la situación de seguridad e inseguridad alimentaria en los hogares estadounidenses era la siguiente:

- Hogares con seguridad alimentaria, que tuvieron acceso en todo momento a suficientes alimentos para proporcionar una vida activa y saludable a todos los miembros del hogar, fue del 89,5 por ciento (116,7 millones).
- Hogares con inseguridad alimentaria que no pudieron adquirir en algún momento del año suficientes alimentos para satisfacer las necesidades de todos los miembros de su familia, porque no tenían suficiente dinero u otros recursos para la alimentación, fue del 10,5 por ciento (13,8 millones). De este 10,5 por ciento,

el 6,6 por ciento (8,6 millones) de los hogares estadounidenses tenían una baja seguridad alimentaria, y el 3,9 por ciento (5,1 millones) de los hogares estadounidenses tuvieron una seguridad alimentaria muy baja. [3]

La Doctorada en Ciencias de la Alimentación y Nutrición Humana, Karla Shelnut, quiso buscar una solución a este problema, y junto a un grupo de personas del programa ampliado de educación alimentaria y nutricional de la Universidad de Florida, presento su idea al Dr. Benjamin C.Lok. La idea era desarrollar una aplicación para móvil (Android) que permitiera a las personas que están pasando por una situación de inseguridad alimentaria encontrar, tanto Bancos de Alimentos (organizaciones que se encargan de recuperar excedentes de alimentos y repartirlos entre las personas que más lo necesitan), como Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria (SNAP) (establecimientos donde aquellas personas inscritas en el programa SNAP pueden comprar bienes de primera necesidad con descuentos) cercanos a su ubicación, para poder visitarlos y conseguir comida gratuita en esos Bancos de Alimentos, o a un menor precio debido a los descuentos aplicados que pueden utilizar aquellas personas que se encuentran dentro del programa SNAP (Programa que permite a familias de al menos 3 personas, que tienen unos ingresos mensuales no superiores a 1.800\$, disfrutar de esos descuentos anteriormente mencionados).

De esta manera, con el uso de las nuevas tecnologías, a las cuales la mayor parte de las personas actualmente tiene acceso, se les podrá facilitar información acerca de los horarios de los establecimientos, podrán obtener las direcciones exactas para llegar a ellos, podrán visitar sus páginas Web y llamar por teléfono... Todo esto desde un dispositivo móvil.

1.2 MOTIVACIÓN DEL PROYECTO

El objetivo del proyecto es el desarrollo de una aplicación para móvil debido a su portabilidad y uso en cualquier instancia y lugar. El desarrollo de esta aplicación para móvil permitirá a personas con problemas de inseguridad alimentaria acceder a información acerca de Bancos de Alimentos y de Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria de una manera rápida y sencilla. Todo esto a través de un dispositivo móvil.

La mayor parte de las personas en la actualidad poseen un dispositivo móvil, y con tan solo el acceso a su ubicación, la aplicación será capaz de dirigirles al Banco de Alimentos o Comercio Minorista del Programa de Asistencia Nutricional Suplementaria más cercano. Además, se les ofrecerá a los usuarios información acerca de los establecimientos que deseen visitar, como los horarios, número de teléfono, página Web, distancia a la que se encuentran... De esta forma les será más fácil acceder a estos lugares ya que pasaran a tener toda la información disponible en su dispositivo móvil, ahorrando su tiempo y dinero, para ayudarles a tener un día a día mejor, para ayudarles a tener una vida mejor.

Mediante el Cloud Computing, que va creciendo cada vez más a lo largo de los años, se otorga a los usuarios capacidades informáticas que no requieren de interacción humana con el proveedor del servicio, que es capaz de servir a múltiples clientes según la demanda sin que estos tengan noción alguna de la ubicación de estos recursos. Mediante el uso de esta tecnología, se consigue un mayor control y optimización de los procesos. Es por esto por lo que el acceso a la información necesaria acerca de Bancos de Alimentos y de Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria será instantáneo. Los usuarios tendrán toda la información al instante para que puedan sacar el máximo beneficio de la aplicación y puedan obtener esa ayuda instantánea. [4]

Por estos motivos, sencillez, rapidez y comodidad, una aplicación para móvil sería la mejor solución para la necesidad planteada: la inseguridad alimentaria de las familias y como se les puede ayudar a conseguir comida de manera rápida, segura y al menor coste.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

El desarrollo de la aplicación Android se ha dividido en dos fases.

- Una primera de diseño, en la que a partir de una serie de requisitos funcionales se desarrollaron los diagramas de casos de uso y de navegación. También se desarrollaron tres posibles diseños para la interfaz de la aplicación.
- Una segunda de desarrollo, en la que a través de tres plataformas (ANDROID STUDIO, FIREBASE Y GOOGLE MAPS PLATFORM) se llevó a cabo el desarrollo puro de la aplicación, teniendo en cuenta todo lo anteriormente realizado en la fase de diseño.

2.1 FASE DE DISEÑO

2.1.1 VISUAL PARADIGMA ONLINE

Visual Paradigma Online es una plataforma online que permite a los usuarios generar diversos diagramas, infografías, imprimibles, gráficos Web ... y trabajar juntos en tiempo real en proyectos de diseño.

A través de esta plataforma se pudo llevar a cabo de manera colaborativa el desarrollo de los diagramas de casos de uso y de navegación. Estos diagramas están basados en unos requisitos funcionales definidos previamente.

- El diagrama de casos de uso representa todos los procesos que se pueden llevar a cabo con la aplicación por parte del usuario. También representa todos los procesos propios del sistema.
- El diagrama de navegación representa la estructura del sistema y las posibilidades de que tiene el usuario en el momento de uso del sistema.

Estos diagramas se desarrollaron tras una serie de reuniones con distintas personas de la Universidad de Florida, que tuvieron lugar durante los momentos iniciales del proyecto. En

esas reuniones se definieron cuáles serían los requisitos funcionales de la aplicación, es decir, se definieron todas las funcionalidades que la aplicación debía ofrecer a los usuarios para que estos pudieran obtener el máximo beneficio del uso de la aplicación.

En la Figura 1 se puede observar el diagrama final de casos de uso, y en la Figura 2 el diagrama final de navegación. Se habla de diagramas finales debido a que a lo largo del proceso de desarrollo de la aplicación, se han ido añadiendo más requisitos y algunos de los requisitos iniciales han ido cambiando. Es por esto por lo que el funcionamiento de la aplicación, tal y como se tenía pensado desde el principio, ha ido sufriendo cambios. De esta manera, los diagramas han ido cambiando constantemente hasta el último día de desarrollo de la aplicación.

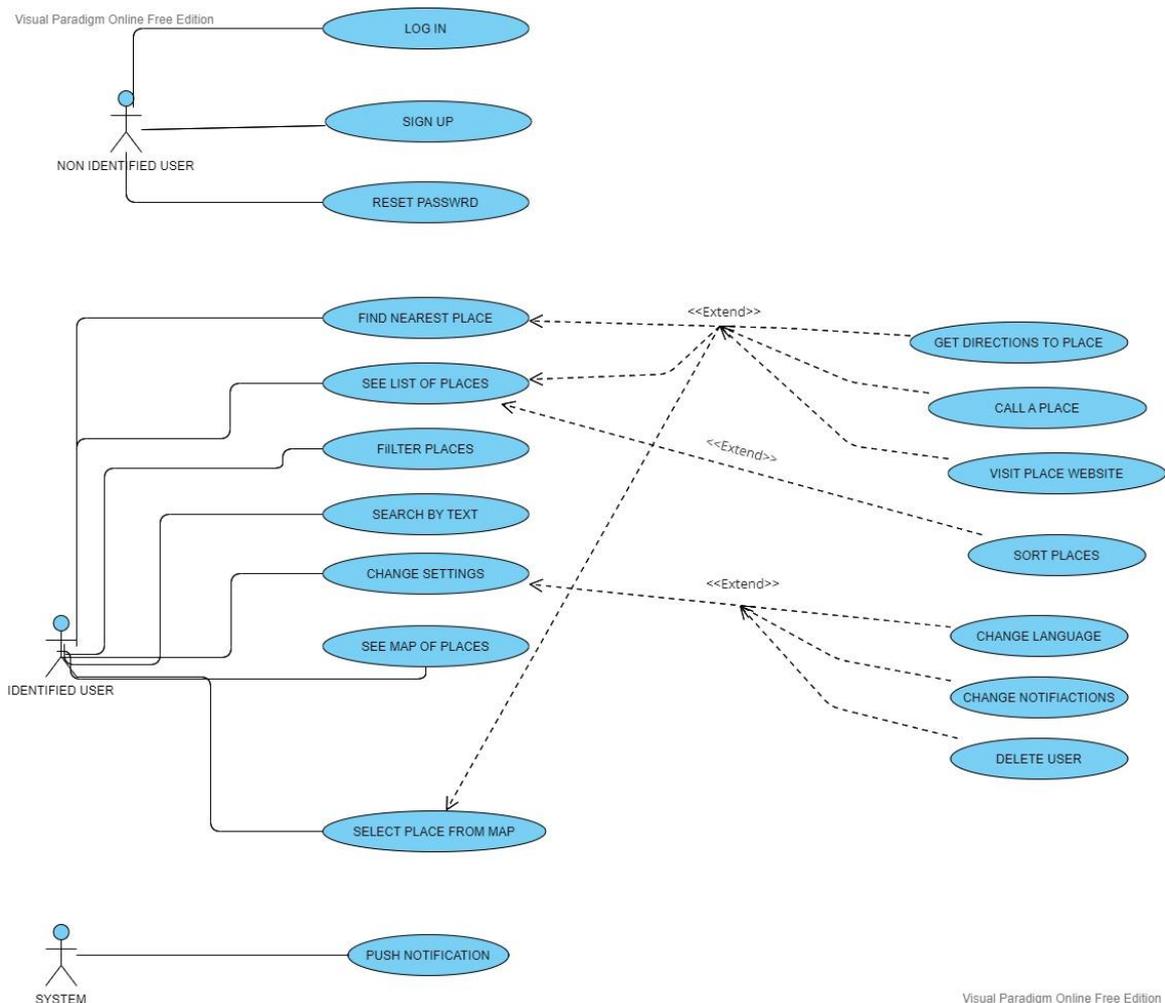


Figura 1. Diagrama de Casos de Usos

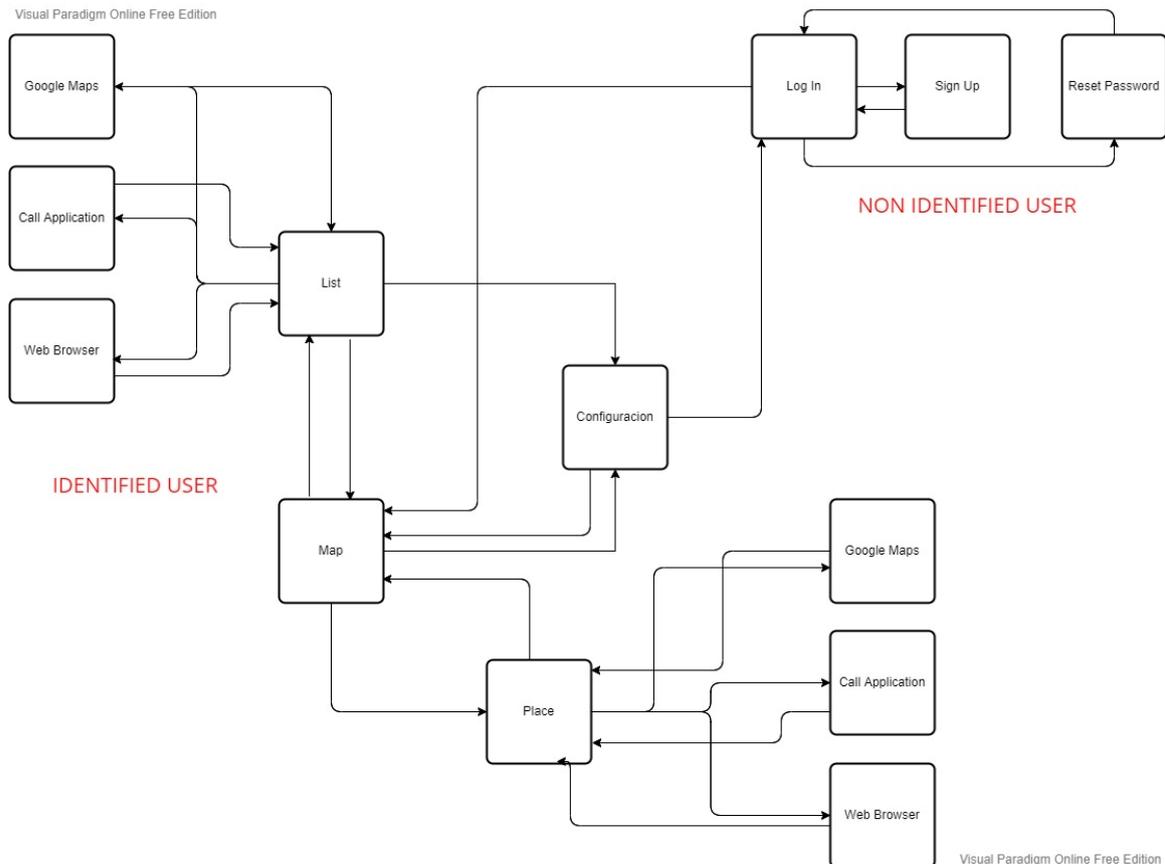


Figura 2. Diagrama de Navegación

2.1.2 FIGMA

Figma es una plataforma tanto Web como de escritorio que sirve para llevar a cabo el desarrollo de prototipos Web de una manera fácil y sencilla. Es una plataforma muy intuitiva que se utiliza principalmente en proyectos colaborativos de diseño.

Mediante el uso de Figma se llevaron a cabo tres posibles diseños de la interfaz de la aplicación. Los diseños eran sencillos y con pocos detalles, ya que eran solo prototipos y se buscaba poder decidir cuál de los tres diseños era el más adecuado, sencillo y atractivo visualmente para posteriormente desarrollar la aplicación basándose en ese diseño. Los tres diseños tienen pantallas similares, que en un principio, eran las que se querían para la aplicación.

En las Figuras 3, 4 y 5 se pueden observar los tres diseños de la interfaz que se realizaron. Las principales funcionalidades que en el momento del diseño de la interfaz se querían cubrir eran:

- Disposición de un mapa en la aplicación.
- Posibilidad de observar todos los lugares en forma de lista.
- Obtener la información a cerca del lugar seleccionado.

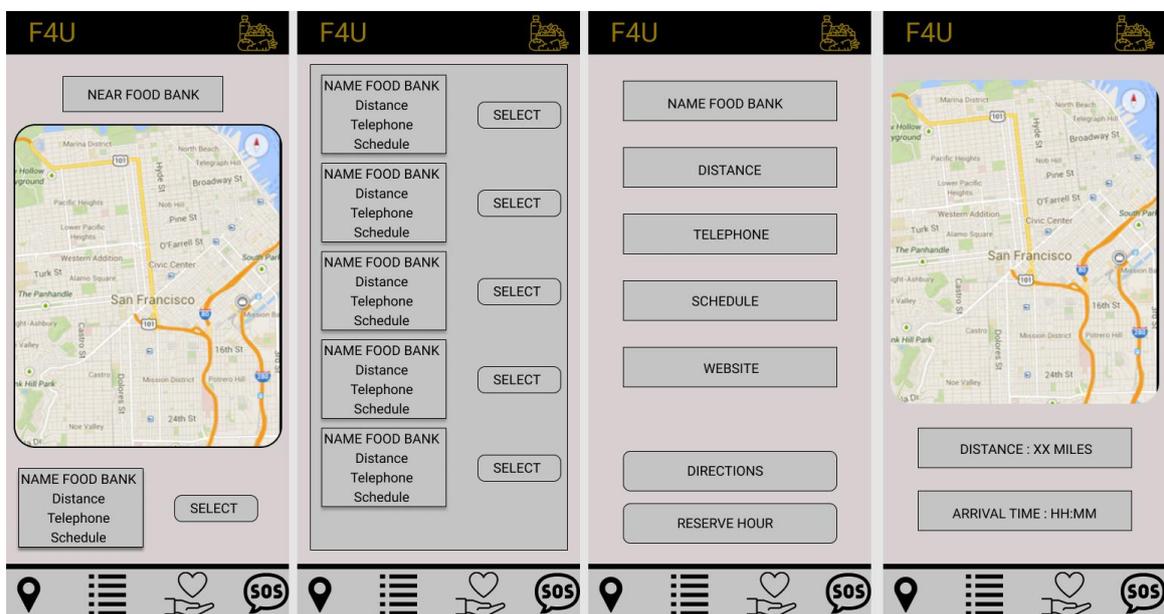


Figura 3. Diseño 1 de la Interfaz de la Aplicación



Figura 4. Diseño 2 de la Interfaz de la Aplicación

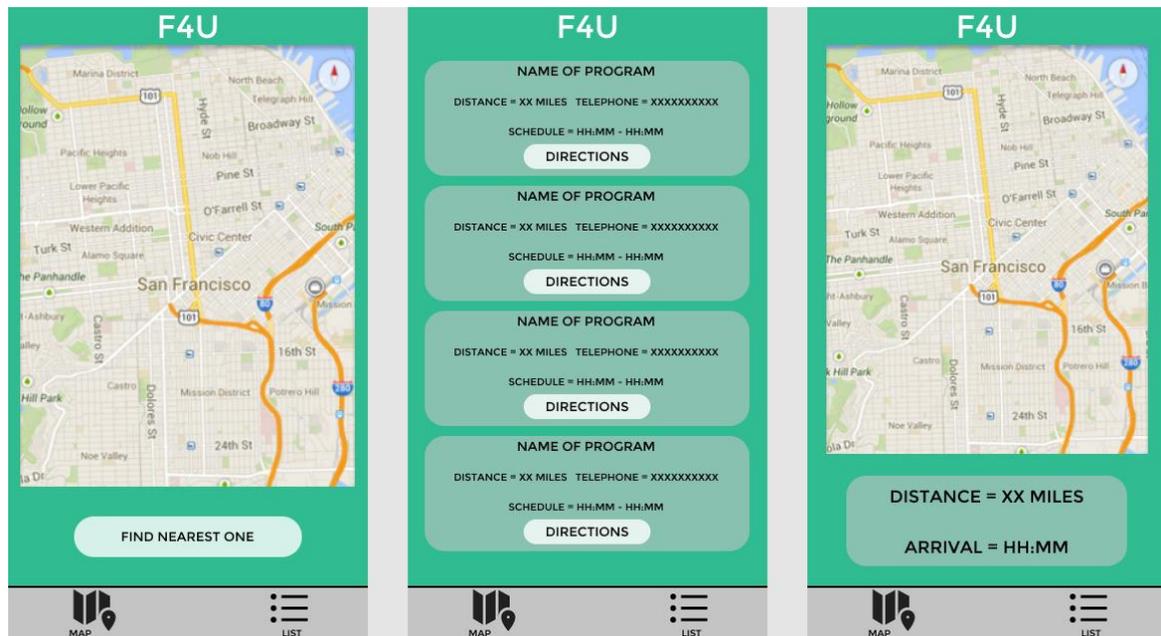


Figura 5. Diseño 3 de la Interfaz de la Aplicación

Tras una evaluación de los tres posibles diseños de la interfaz de la aplicación que se llevó a cabo durante una reunión con distintas personas de la Universidad de Florida, se eligió el tercer diseño de la interfaz como el más adecuado, sencillo y atractivo visualmente. Como se podrá observar posteriormente, se consiguió que el resultado final fuera muy parecido.

2.2 FASE DE DESARROLLO

Inicialmente se llevó a cabo un proceso de análisis de las plataformas tecnológicas que se ofrecían en el mercado para poder desarrollar el proyecto. Tras ese análisis se escogieron las que se consideraron más adecuadas para alcanzar los requisitos establecidos.

Cabe mencionar que para poder hacer uso de las mismas, ha sido necesario llevar a cabo un proceso previo de autoformación basado en estudio de documentación extraída de las propias plataformas tecnológicas y de la visualización de tutoriales en formato de vídeo.

2.2.1 ANDROID STUDIO

ANDROID STUDIO es un Framework de desarrollo con el que se ha llevado a cabo, tanto el desarrollo del interfaz de la aplicación con la que el usuario interactúa, como el desarrollo

de la parte del servidor para poder manejar las peticiones y acciones de los usuarios cuando interactúan con la aplicación.

- Para el desarrollo de la interfaz de la aplicación se han desarrollado múltiples “activities”. Estas “activities” representan cada una de las distintas pantallas de la aplicación para móvil por las que podrá navegar el usuario. Mediante el uso de ANDROID STUDIO y a través de código XML, se han ido desarrollando una por una todas las pantallas. ANDROID STUDIO permite añadir una gran variedad de elementos a estas pantallas como son: Botones, TextViews, EditTexts, Switches, Fragments, Imageviews... y a través de distintos layouts, permite organizar y diseñar las pantallas del modo en que se precise. En la Figura 6 se puede observar el código de una “activity” y el resultado, es decir, como queda la pantalla.

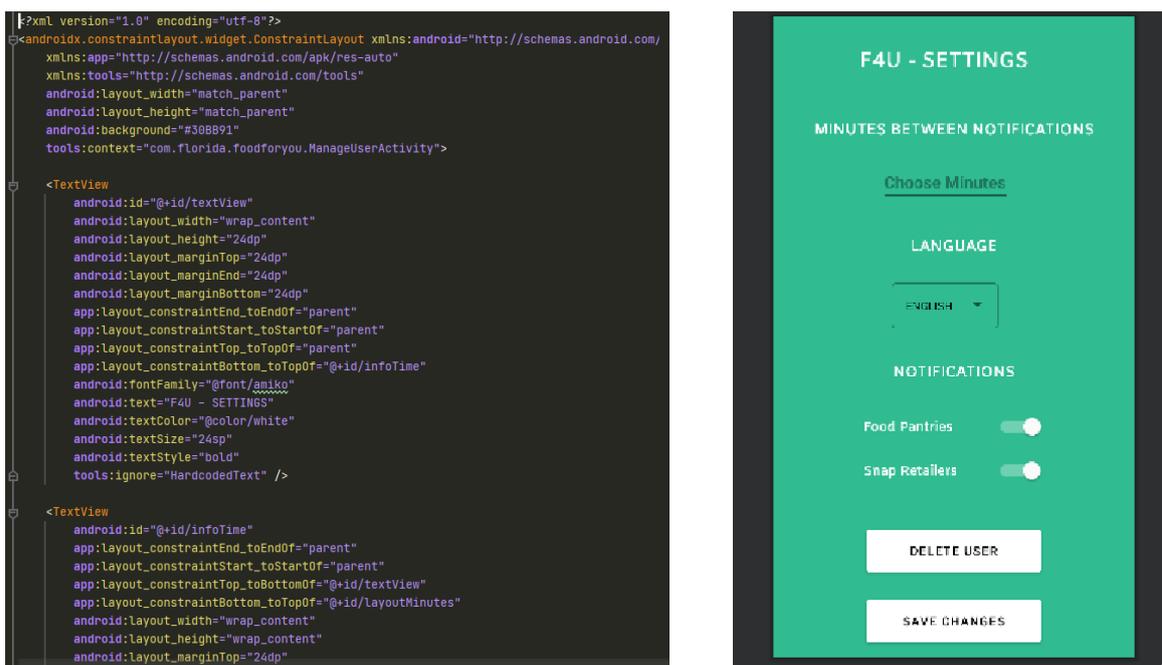


Figura 6. Activity de ANDROID STUDIO

- Para la parte del desarrollo del servidor, es decir del código fuente necesario para poder manejar las peticiones y acciones de los usuarios cuando interactúan con la aplicación, se ha utilizado ANDROID STUDIO, que tiene el lenguaje de programación Java como lenguaje del entorno de desarrollo.

Para llevar a cabo el manejo de esas interacciones del usuario con la aplicación, se desarrolló el código fuente correspondiente de cada una de las “activities” anteriormente mencionadas. El código fuente de cada “activity” se basa en las posibles acciones que pueden llevarse a cabo por parte del usuario sobre los elementos dispuestos en la pantalla. Para que la funcionalidad sea completa se deben contemplar todas las posibles acciones. Por ejemplo: pulsar un botón, moverse en el mapa, cambiar de pantalla, introducir texto...

En la Figura 7 se puede observar cómo es el entorno de desarrollo de ANDROID STUDIO. A la izquierda se encuentra toda la organización del proyecto en distintas carpetas y paquetes. En el centro un ejemplo de código fuente de la clase MapPpal.java. En la parte superior la herramienta que proporciona ANDROID STUDIO para correr la aplicación en un emulador o un dispositivo Android.

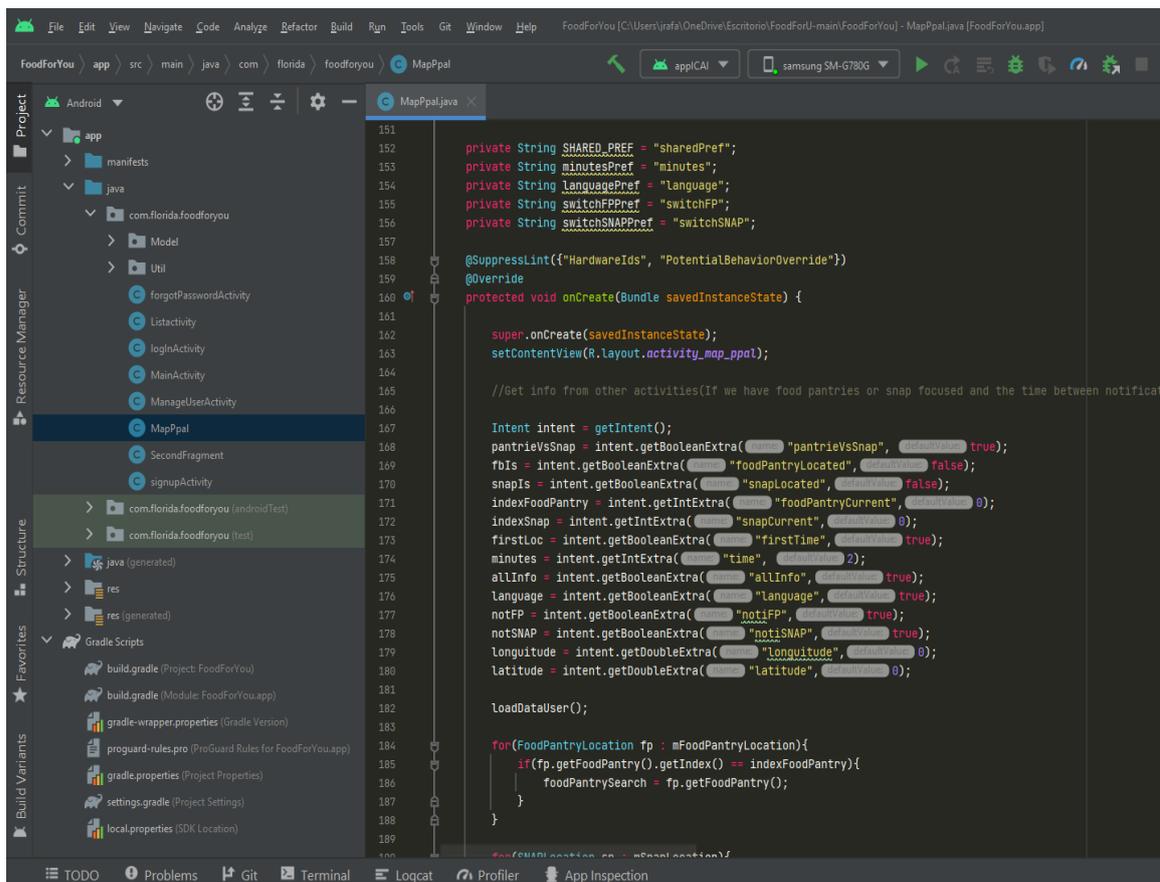


Figura 7. Entorno de Desarrollo de ANDROID STUDIO

2.2.2 GOOGLE MAPS SDK

GOOGLE MAPS PLATFORM es una plataforma que permite obtener una API de Maps SDK para Android que se puede integrar en aplicaciones de móvil. Con este SDK se puede integrar en la aplicación un mapa en el que observar, tanto la localización del usuario, como la de los distintos establecimientos a los que pueden acceder (Bancos de Alimentos y Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria). Además, ofrece un análisis de métricas acerca del número de solicitudes que recibe la API para poder ver cuántas fueron satisfactorias y cuántas erróneas.

En la Figura 8 se puede observar cómo es la interfaz de GOOGLE MAPS PLATFORM y los servicios que ofrece. De todos ellos, se ha hecho uso del Maps SDK para Android. Como se comentó anteriormente, ese SDK es una herramienta que permite añadir un mapa en la aplicación para poder visualizar distintos lugares en ella, además de también poder posicionar la ubicación del usuario en el mapa.

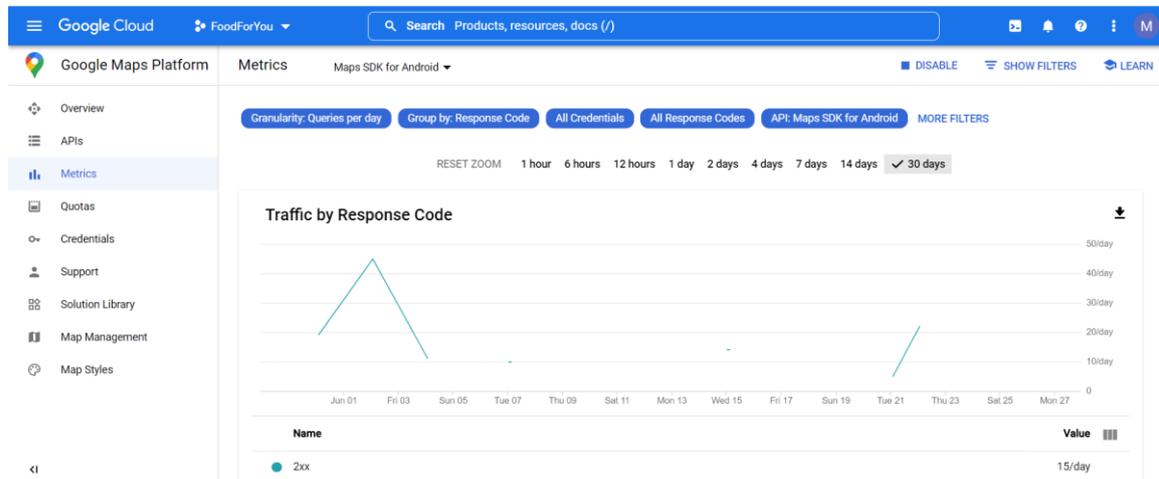


Figura 8. Interfaz de GOOGLE MAPS PLATFORM

El funcionamiento de esta plataforma y su integración en la aplicación es muy simple.

- Lo primero que hay que hacer es obtener una credencial de clave API que servirá para poder integrar el SDK de Maps en la aplicación.
- Tras obtener la API se debe restringir el tipo de uso que se le va a dar. En este caso su finalidad de uso es para una aplicación Android.

- Posteriormente se debe indicar qué aplicación Android específica será la relacionada con la clave API. Para esto se debe obtener la huella digital del certificado de firma SHA-1, y hacerla corresponder con el nombre del paquete del proyecto de la aplicación Android.
- Tras todo esto se añade la clave API a nuestro proyecto de ANDROID STUDIO en el manifest, y a partir de este momento ya se puede utilizar y visualizar el mapa en la aplicación Android.

2.2.3 FIREBASE

FIREBASE permite usar elementos de análisis, autenticación, almacenamiento en bases de datos en tiempo real, mensajería en tiempo real... Todo esto mediante el uso de la plataforma Cloud de Google y los SDK que se proporcionan. A través de esos SDK se lleva a cabo la interacción entre la parte del cliente (Aplicación) y la parte del Back-End sin necesidad de establecer un intermediario para el uso de estos servicios. Es por eso por lo que el formato request-response del modelo HTTP aquí no se utiliza. Esto permite a los desarrolladores simplificar su trabajo, ya que no se necesita gestionar los servidores ni escribir APIs, FIREBASE actúa como servidor y como API. [5]

De todas las plataformas que ofrece FIREBASE, se ha hecho uso de dos de ellas.

- Base de datos en tiempo real. Con la real-time database de FIREBASE se ha podido llevar a cabo el almacenamiento de los usuarios que acceden a la aplicación. Esta real-time database usa un formato NoSQL.
- Plataforma Autenticación de usuarios. Con la plataforma de autenticación de usuarios se ha podido corroborar que los usuarios que se han dado de alta en la aplicación son reales y tienen un correo electrónico real.

En la Figura 9 se puede observar la interfaz de FIREBASE. Se muestran las gráficas de análisis de la aplicación (el número de accesos que ha tenido) y la información sobre la base de datos de tiempo real (el número de descargas de información y de almacenamiento).

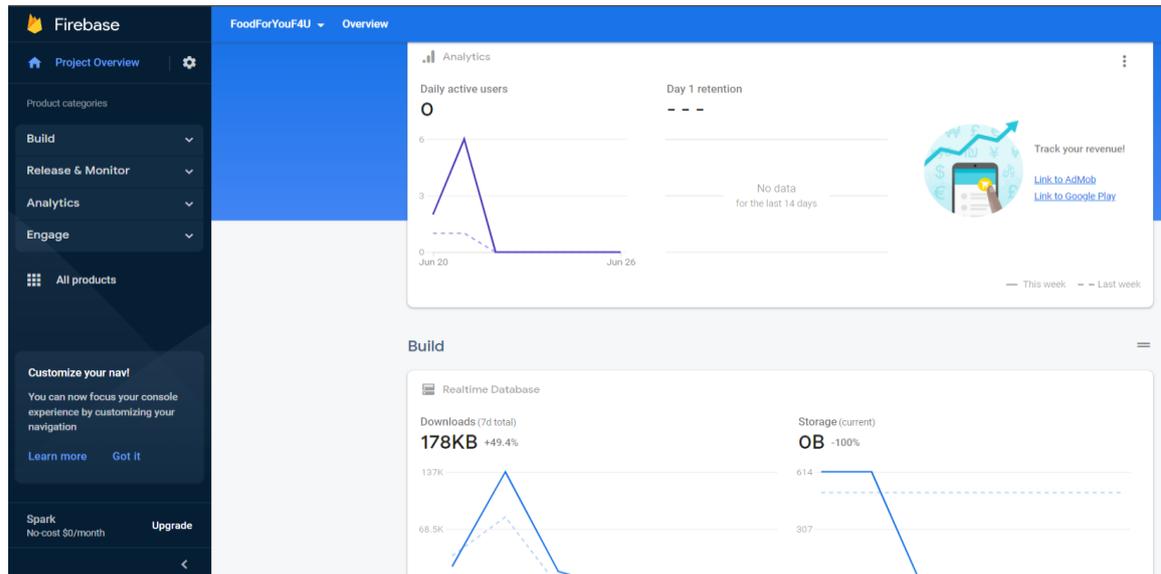


Figura 9. Interfaz de FIREBASE

El formato de almacenamiento NoSQL es muy utilizado actualmente ya que permite guardar información en un formato de clave-valor, dejando a un lado el almacenamiento de información en tablas con el formato entidad-relación. Se ha utilizado este modelo almacenando la información de los usuarios con la relación entre clave, un identificador de cada usuario, y valor, que en este caso son 3 campos: el correo electrónico del usuario, su nombre de usuario y un campo que identifica si el acceso se ha hecho a través del inicio de sesión de Google o a través de la aplicación. Esto se puede observar en la Figura 10. [6]

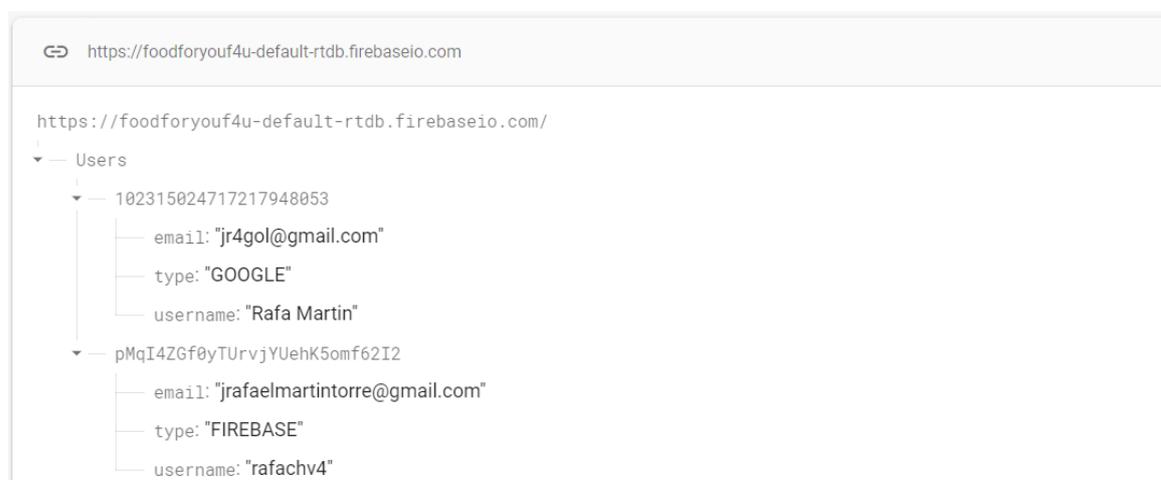


Figura 10. Real-time Database de FIREBASE

Capítulo 3. ESTADO DE LA CUESTIÓN

Este proyecto surge como solución a un problema que siempre ha estado presente en la sociedad, y que no para de incrementarse actualmente tras la época del Covid-19: la inseguridad alimentaria.

La solución tecnológica que se propone al problema, brinda la posibilidad a aquellas familias que sufren de esta inseguridad alimentaria, de ser capaces de a través un dispositivo móvil, encontrar, tanto Bancos de Alimentos, como Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria que estén abiertos y a los que puedan acudir a por comida.

Hay varias aplicaciones de móvil en el mercado que tratan el tema de la inseguridad alimentaria. “Food For All”, que permite a las personas obtener comida al 50% rebajada una hora antes del cierre de algunos restaurantes. Y “Fresh Food Connect”, que permite a las personas obtener alimentos sobrantes de los agricultores y productores de alimentos.

3.1 *FOOD FOR ALL*

Aplicación para móvil desarrollada por Food For All Technologies en el año 2020. La función de la aplicación es la de poner en contacto a los usuarios con restaurantes y cafeterías que ofrecen sus excedentes de comida fresca todos los días con un descuento de al menos el 50%. Los usuarios eligen la comida que quieran, pagan a través de la aplicación, van al restaurante a la hora de la recogida y disfrutan de su comida para llevar.

La aplicación presenta algunos aspectos no suficientemente cubiertos:

- No está teniendo en cuenta a todas las personas que sufren de inseguridad alimentaria. No todas las personas tienen la posibilidad de ir todos los días a un determinado restaurante a por comida para llevarla a casa, ya que desplazarse todos los días también supone un gasto tanto de dinero como de tiempo. Además, no todas

las personas se pueden gastar dinero todos los días en comida, aunque sea al 50% del precio original.

- El tipo de comida de los restaurantes y cafeterías que oferta la aplicación, en algunas ocasiones no es la recomendada para poder llevar a cabo una dieta equilibrada.

3.2 FRESH FOOD CONNECT

Aplicación desarrollada por la organización Fresh Food Connect de Colorado en el año 2019. La función de la aplicación es ayudar a los agricultores y productores de alimentos con sus excesos de producción. El equipo de Fresh Food Connect recogerá esos alimentos y los distribuirá por barrios locales para personas que sufren de esa inseguridad alimentaria.

La aplicación presenta algunos aspectos no suficientemente cubiertos:

- Las personas que sufren de inseguridad alimentaria dependen tanto de los productores de alimentos y sus excedentes de producción, como de las personas de Fresh Food Connect. Se podría dar el caso de que haya días en los que estas personas no dispongan de ningún alimento, ya sea porque no haya alimentos excedentes, o porque la distribución de esos alimentos no se ha podido realizar por parte de Fresh Food Connect.
- Hay personas que aun pasando por dificultades económicas y que se encuentran en una situación de inseguridad alimentaria, les resulta difícil de aceptar que no existe ningún problema en pedir y obtener ayuda, y probablemente no acudirán a estos repartos.

3.3 FOOD FOR YOU – F4U

Este proyecto y la solución que propone incorpora un enfoque algo diferente a lo anteriormente mencionado y sobre todo está orientado específicamente a la actividad

desarrollada por los Bancos de Alimentos y los Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria.

Se diferencia de Food For All y Fresh Food Connect en los siguientes aspectos:

- Se tiene en cuenta a todo el público objetivo con relación a la inseguridad alimentaria, tanto aquellos de nivel de pobreza bajo como de nivel de pobreza alto. Se les brinda la oportunidad de acudir, tanto a Bancos de Alimentos, donde podrán conseguir alimentos de primera necesidad de manera gratuita, como a Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria, donde podrán comprar alimentos básicos de primera necesidad con un descuento sobre el precio original. De esta manera la personas que no quieren acudir a Bancos de Alimentos o repartos de alimentos gratuitos por lo comentado anteriormente, podrán por lo menos acceder a alimentos de primera necesidad pagando un menor precio.
- Al ofertarles estos dos tipos de establecimientos, las comidas las podrán cocinar en sus casas. De esta manera podrán llevar una dieta equilibrada que les aporte todos los alimentos y nutrientes necesarios, dejando a un lado ese tipo de comida rápida y procesada que en algunas ocasiones ofrecen las cadenas de restaurantes y cafeterías.
- Serán los usuarios que utilicen la aplicación los que elijan a qué establecimiento ir en función de la distancia a la que se encuentre el sitio y su horario. No dependen de nadie, ellos eligen, por lo que se les da la posibilidad de conseguir comida todos los días del año gratis o con un descuento aplicado.

Con esta aplicación se va a permitir a todas las personas que pasan dificultades económicas y de inseguridad alimentaria, mantener una dieta balanceada y equilibrada la mayoría de los días del año. Para ello, la aplicación pondrá a disposición de los usuarios distintos establecimientos donde conseguir alimentos y productos de primera necesidad de una manera sencilla y rápida.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

En el apartado anterior se han expuesto dos aplicaciones para móvil que tratan el problema de la inseguridad alimentaria en la sociedad. Estas dos aplicaciones para móvil tienen como objetivo principal ofrecer ayuda a esas personas que sufren de este problema. No obstante, como se mencionaba anteriormente, ninguna de ellas trata ni con Bancos de Alimentos ni con Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria. Además, tienen una serie de funcionalidades que no acaban de solventar del todo el problema de la inseguridad alimentaria en la sociedad.

La aplicación para móvil a desarrollar, trata de ayudar a solventar este problema de una manera distinta, tratando con los Bancos de Alimentos y con los Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria.

- **Bancos de Alimentos.** El uso de Bancos de Alimentos en todo el mundo crece cada año que pasa. En el caso específico de los Estados Unidos, el uso de Bancos de Alimentos como medio para obtener comida ha crecido un 2,2% desde el año 2019, pasando de un 4,4% de familias americanas que usaron Bancos de Alimentos a un 6,6%. [7] Es por esto por lo que se pensó en crear una plataforma que permitiera dar a conocer estos establecimientos, donde están localizados, sus horarios, página Web, número de teléfono... Esta plataforma permitiría el acceso a la información de una manera rápida y sencilla. Además, podría ayudar a que esas cifras relacionadas con el uso de los Bancos de Alimentos siguieran creciendo. Hay que mencionar que debido el Covid-19, se han visto reducidos los voluntariados para el reparto de alimentos y bebidas a personas necesitadas. Es por esto por lo que los Bancos de Alimentos son esenciales hoy en día.

- **Comercios Minoristas del Programa de Asistencia Nutricional.** Los Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria son establecimientos como supermercados, tiendas de alimentos... que están asociados al programa SNAP (programa que permite a familias de al menos 3 personas, que tienen unos ingresos mensuales no superiores a 1.800\$, disfrutar de descuentos aplicados sobre algunos alimentos y productos de primera necesidad). Hay alrededor de 248.000 establecimientos que forman parte de este programa. A través de este proyecto, las personas con inseguridad alimentaria podrán obtener información acerca de qué establecimientos alrededor suyo están dentro de este programa y les ofrecerán descuentos en alimentos y productos de primera necesidad. [8]

Actualmente, la mayor parte de las personas disponen de un dispositivo móvil, de tal manera que una aplicación para móvil es la mejor plataforma para que los usuarios saquen el máximo beneficio de este proyecto. Los usuarios serán capaces de acceder de una manera rápida y sencilla a la información, en cualquier instante y lugar. Serán capaces de observar en un mapa su ubicación y qué establecimientos (Bancos de Alimentos o Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria) se encuentran cercanos a ellos. Se les otorgará la posibilidad de visitar la página Web de los establecimientos, poder llamarlos por teléfono, ver su horario de apertura, buscar por un tipo determinado de establecimiento con una barra de búsqueda... Además, podrán recibir notificaciones automáticas. Estas notificaciones las manda el sistema cuando los usuarios se encuentran cercanos a un establecimiento de los anteriormente mencionados.

Esta aplicación está pensada principalmente para el estado de Florida, pero con el objetivo de expandirla a todos los Estados Unidos. Debido al gran número de personas hispanohablantes que hay en el estado de Florida, la aplicación para móvil ofrecerá la posibilidad de cambiar el idioma de la misma entre español e inglés.

Con todo lo anteriormente mencionado, se puede concluir lo siguiente: esta aplicación para móvil permitirá a los usuarios tener en su mano toda la información de manera instantánea. De esta manera podrán elegir el tipo de establecimiento que desean visitar de una manera

rápida y sencilla. Se les hará más fácil el día a día, ayudándoles a conseguir comida de una manera que nunca se había planteado.

4.2 OBJETIVOS

Los objetivos principales fijados para este proyecto son los siguientes:

- **Desarrollar una aplicación para móvil sencilla y “friendly” que los usuarios puedan usar.** Para ello se utilizará ANDROID STUDIO. Con esta plataforma se llevará a cabo el desarrollo de una aplicación Android. Se desarrollarán las pantallas de acceso a la aplicación y las pantallas de navegación de los usuarios por la misma.
- **Implementar la SDK de Google Maps para poder situar un mapa en la aplicación.** Para ello se utilizará una API. Esta API se consigue mediante la plataforma SDK de Google Maps y permite situar un mapa en la aplicación. En ese mapa se podrá situar tanto al usuario que utiliza la aplicación como aquellos establecimientos que se quieran situar en el mapa.
- **Permitir que la aplicación envíe notificaciones a los usuarios.** Se implementará la funcionalidad que permita a la aplicación enviar notificaciones a los usuarios cuando exista un Banco de Alimentos o un Comercio Minorista del Programa de Asistencia Nutricional cercano a su ubicación. Además, se permitirá que el usuario modifique el tipo de notificación que quiere recibir. Es decir, seleccionar si quiere recibir la notificación por ambos establecimientos, solo por uno de ellos, o no recibir nada.
- **Desarrollar una aplicación en los idiomas español e inglés.** Debido al gran número de hispanohablantes que hay en el estado Florida, se pretende desarrollar una aplicación que permita cambiar el idioma de esta entre español e inglés.

- **Diseñar la base de datos para el almacenamiento de la información de los usuarios que accedan a la aplicación.** Para esto se empleará la plataforma de FIREBASE. Mediante el uso de su real-time database, se almacenará la información en una base de datos NoSQL.
- **Diseñar el sistema de autenticación y autorización de los usuarios que accedan a la aplicación.** Para esto se empleará la plataforma de FIREBASE. Mediante el uso de la plataforma de autenticación de FIREBASE se llevará a cabo la autenticación y verificación de los usuarios. También se llevará a cabo la autenticación y verificación de usuarios a través de Google y su inicio de sesión.
- **Llevar a cabo una fase final de testing de la aplicación con usuarios de prueba y usuarios reales de la aplicación.** Testear el funcionamiento de la aplicación con distintos grupos de personas. Se busca poder sacar conclusiones para poder mejorar tanto el diseño como el funcionamiento de la aplicación.

4.3 *METODOLOGÍA*

El desarrollo temporal del proyecto se divide en dos periodos. El primero transcurrió durante el Fall Semester (semestre de otoño), y el segundo durante el Spring Semester (semestre de primavera). Durante estos dos semestres se trabajó de manera conjunta con la Universidad de Florida, en Estados Unidos

En ambos periodos de trabajo, se ha utilizado la metodología de trabajo “Agile”. Todos los lunes se definían los objetivos de la semana y se exponía y analizaba el trabajo de la semana previa. Esto se llevaba a cabo durante reuniones junto al Dr. Benjamin C.Lok y Jacob L.Stuart.

Todo el trabajo se dividió en cinco fases que estaban comprendidas en los dos periodos temporales anteriormente comentados. La primera fase fue la de diseño, la segunda fue la del primer desarrollo, la tercera fue la de evaluación de la aplicación, la cuarta fue la del segundo de desarrollo, y la quinta fue la de testing.

Para poder llevar a cabo el trabajo de una manera organizada, se creó un documento que se llamó “Project Draft”. En este documento se iban apuntando todas las tareas de las correspondientes fases que se iban completando a lo largo de la semana. También se anotaban todas las tareas que se debían lograr para la siguiente semana. De esta manera se tenía un “record” de todos los “sprints” con las tareas que se debían realizar y las tareas realizadas. Además, en el documento anteriormente mencionado se adjuntan enlaces a videos de YouTube donde se puede apreciar el trabajo realizado cada semana. Esos enlaces se han copiado en este documento.

4.3.1 FALL SEMESTER

4.3.1.1 Fase de Diseño

En esta primera fase se llevó a cabo tanto el diseño de los diagramas de casos de uso y de navegación, como los posibles diseños para la interfaz de la aplicación. Durante la primera semana de trabajo se tuvieron una serie de reuniones con distintas personas de la Universidad de Florida en los que se definieron todos los requisitos funcionales de la aplicación. Con los requisitos definidos y mediante el uso de la plataforma Visual Paradigm Online, se pudo llevar a cabo durante la segunda semana de trabajo el diseño de los diagramas de casos de uso y de navegación. Estos diagramas sirvieron para poder tener una idea clara del funcionamiento de la aplicación y las acciones que puede llevar a cabo el usuario con ella. Después de esto y a través del uso de la plataforma Figma, se llevó a cabo durante la tercera semana de trabajo el diseño de tres posibles interfaces para la aplicación. Tras una evaluación de los mismos, se pudo elegir el diseño más adecuado, sencillo y atractivo visualmente.

Durante esta fase de diseño, la Universidad de Florida facilitó la información de los 230 establecimientos a tener en cuenta (un establecimiento está compuesto por nombre del programa, dirección, horario, número de teléfono y pagina Web). Debido al gran número de establecimientos, se valoró la opción de almacenar la información de estos en una base de datos. La decisión que se tomo fue la de añadir toda la información en el código fuente de la aplicación, dejando la posibilidad de almacenar la información en una base de datos para un futuro, cuando la aplicación estuviera establecida y disponible en el mercado. En ese

momento el mantenimiento de la información acerca de los establecimientos sería crucial, y como se comentó en esta fase de diseño, una base de datos sería la mejor opción.

4.3.1.2 Fase de Desarrollo

En esta segunda fase se llevó a cabo el desarrollo del código fuente de una gran parte de la aplicación. El trabajo se llevó a cabo en 7 semanas en las cuales se implementaron las siguientes funcionalidades:

1ª Semana. Implementación de la API de GOOGLE MAPS SDK y creación de la lista de Bancos de Alimentos y Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria. Se llevó a cabo el desarrollo de dos “activities”. Ese desarrollo implicó tanto el desarrollo del código de las interfaces, como el desarrollo del código fuente necesario para darles las funcionalidades deseadas. Una de ellas es la “activity” en la que se muestra el mapa. En ese mapa es donde se puede ver la localización del usuario y de los distintos establecimientos. La otra “activity” es en la que se pueden ver los establecimientos en forma de lista.
<https://youtu.be/A9RytXob194>

2ª Semana. Implementación de la funcionalidad para obtener direcciones hacia un establecimiento usando Google Maps. Durante esta semana se comenzó a añadir la información sobre los establecimientos. Esa información permite poder situar en el mapa los distintos marcadores y poder visualizar donde están los establecimientos. Además, se depuró un poco más la interfaz de la lista de establecimientos.
https://www.youtube.com/watch?v=Tt_FQ9Z6ZRA

3ª Semana. Implementación de distintas funcionalidades. Ver cuál es el establecimiento más cercano. Mostrar la distancia a la que se encuentra un establecimiento. Filtrar entre los dos tipos de establecimientos. Durante esta semana se desarrolló el código fuente necesario para poder obtener cual era el establecimiento más cercano a la ubicación del usuario. También se implementó la funcionalidad de mostrar un Pop-Up emergente con la información de esa distancia a la que se situaba el establecimiento más cercano. Por último, se añadió la

posibilidad de filtrar entre los dos tipos de establecimientos.
<https://www.youtube.com/watch?v=Hcd0h6IA0-g>

4ª Semana. Desarrollo de la notificación que puede enviar el sistema. Depuración de la aplicación y su interfaz. Durante esta semana se comenzó a implementar la funcionalidad relacionada con la notificación emergente que es capaz de enviar la aplicación al usuario cuando detecta un establecimiento cercano a su ubicación. Además, se llevó a cabo la depuración de la interfaz de la aplicación y el código fuente de esta. Durante esta semana se continuó con la incorporación de la información sobre los establecimientos.

5ª Semana. Finalización de la notificación propia. Implementación de la funcionalidad para configurar el tiempo que tiene que transcurrir entre notificaciones. Durante esta semana se finalizó con la implementación de la funcionalidad de la notificación emergente. La aplicación es capaz de mandar una notificación al usuario cuando detecta que hay un establecimiento cercano a su ubicación. Además, se implementó la funcionalidad de cambiar el tiempo que tiene que transcurrir entre las notificaciones que la aplicación manda al usuario. Durante esta semana se continuó con la incorporación de la información sobre los establecimientos.
<https://www.youtube.com/watch?v=U9JyBsnXc3c>

6ª Semana y 7ª Semana. Detección de bugs en la aplicación. Creación de un archivo .APK. Establecimiento de una fecha para una reunión. Durante estas dos semanas se llevó a cabo una depuración del código fuente. El objetivo de esto era encontrar bugs de la aplicación que hicieran que esta no funcionase correctamente. Tras la depuración del código fuente, se generó el archivo APK para poder probar la aplicación en dispositivos móviles. Además, se acordó una fecha para una reunión con personas del programa ampliado de educación alimentaria y nutricional de la Universidad de Florida. Este meeting tenía como objetivo que esas personas probaran la aplicación a través de la APK. De tal modo que fueran capaces de identificar qué cosas faltaban en la aplicación, cuales estaban bien, que habría que cambiar...

Durante estas dos semanas se continuó con la incorporación de la información sobre los establecimientos.

4.3.1.3 Fase de Evaluación

En esta tercera fase tuvo lugar la reunión de la que se habló anteriormente. Tras esta reunión en el que se probó la aplicación con un grupo de personas del programa ampliado de educación alimentaria y nutricional de la Universidad de Florida, se pudieron sacar muchas ideas en claro sobre que funcionalidades más se tenían que añadir a la aplicación, que partes de la aplicación había que cambiar, y que partes de la aplicación funcionaban y eran correctas.

4.3.2 SPRING SEMESTER

4.3.2.1 Fase de Desarrollo

En esta cuarta fase se llevó a cabo el desarrollo del código fuente de la otra gran parte de la aplicación. El trabajo se llevó a cabo en 10 semanas en las cuales se implementaron las siguientes funcionalidades:

1ª Semana. Posibilidad de cambiar el idioma de la aplicación entre español e inglés. Posibilidad de visitar páginas Web de los establecimientos. Incorporación de la barra de búsqueda a la aplicación. Durante esta semana se implementó la funcionalidad que permite al usuario cambiar el idioma de la aplicación entre inglés y español. También se implementaron, la funcionalidad de poder pinchar en el icono Web de un establecimiento y ser redirigido a su página Web, y la funcionalidad de la barra de búsqueda. Con esta barra de búsqueda se puede filtrar por establecimientos que contengan los caracteres introducidos en ella. Además, se depuro un poco más el diseño de la interfaz cambiando y perfeccionando pequeños detalles.
<https://www.youtube.com/watch?v=ZPIE6-iE8bY>

2ª Semana y 3ª Semana. Implementación de una ventana emergente para los marcadores del mapa. Posibilidad de configurar las notificaciones de los establecimientos. Durante estas dos semanas se implementó la funcionalidad de que

cada vez que un usuario pinchase en un marcador del mapa, saliese una ventana emergente en la parte de abajo de la pantalla. En esa ventana se muestra información sobre el establecimiento seleccionado. También se implementó la funcionalidad de configurar las notificaciones a recibir. El usuario puede elegir entre recibir notificaciones por ambos establecimientos (Bancos de Alimentos o Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria), por solo uno de ellos, o no recibir ninguna notificación. Además, se depuró un poco más el diseño de la interfaz cambiando y perfeccionando pequeños detalles, y se continuó con la incorporación de la información sobre los establecimientos.

<https://www.youtube.com/watch?v=1sbUvc-2OpM>

4ª Semana y 5ª Semana. Cambio de la interfaz principal. Implementación de la posibilidad de guardar las configuraciones en la memoria caché del dispositivo.

Durante la primera semana se tuvo una reunión con distintas personas de la Universidad de Florida. Se decidió hacer un cambio en la interfaz principal de la aplicación. El cambio no fue muy significativo, pero hacía lucir la aplicación más sencilla. Además, se implementó la funcionalidad de guardar toda la información de la configuración personal del usuario en la memoria caché de su dispositivo. Durante estas dos semanas se continuó con la incorporación de la información sobre los establecimientos.

https://www.youtube.com/watch?v=DkQAHzJ_JGs

6ª Semana y 7ª Semana. Ordenación de los establecimientos por nombre o distancia.

Finalización de pequeños detalles de diseño. Durante estas dos semanas se implementó la funcionalidad que permite a los usuarios ordenar en la lista los establecimientos por orden alfabético o por distancia a su ubicación. Además, se finalizó con el perfeccionamiento del diseño de la aplicación y su interfaz, y se terminaron de añadir todos los establecimientos y su información a la aplicación.

<https://www.youtube.com/watch?v=IMpUrR1EKPO>

8ª Semana. Implementación de inicio de sesión, registro y cambio de contraseña. Durante esta semana se llevó a cabo la implementación de las funcionalidades de inicio de sesión, registro y cambio de contraseña. Se desarrollo el código de las interfaces de esas “activities” así como el código fuente para su correcto funcionamiento. Esto supone permitir a los usuarios crear una cuenta, iniciar sesión en la aplicación y poder cambiar su contraseña en caso de olvidarla.

9ª Semana y 10ª Semana. Implementación de FIREBASE en la aplicación para el registro de usuarios y su autenticación. Durante estas dos semanas se llevó a cabo la implementación y la integración de la plataforma FIREBASE en la aplicación. De esta manera, se llevó a cabo el desarrollo del código fuente necesario para poder trabajar con la real-time database y la plataforma de autenticación de FIREBASE. De este modo se consiguieron dos cosas, añadir información de los usuarios a una base de datos para llevar a cabo el manejo de estos, y llevar a cabo la autenticación y la verificación de esos usuarios.

4.3.2.2 Fase de Testing

En esta quinta fase, se han llevado a cabo dos tipos de procesos de testing.

- “Friends & Family”. Proceso de testing en el que un ámbito reducido de personas (usuarios de prueba), entre los que se encontraban tanto compañeros de intercambio, como compañeros de la universidad, como familiares, se descargaron la aplicación en un formato .APK en sus dispositivos móviles para comprobar su funcionamiento e identificar errores. Es una de las fases más importantes ya que en función de los resultados obtenidos se debe seguir con el desarrollo de la aplicación cambiando aquellas funcionalidades que no son correctas, o se puede pasar al siguiente proceso de testing.
- “User Testing”. Tras comprobar el correcto funcionamiento de la aplicación a través del proceso “Friends & Family”, se pasó al proceso “User Testing”. Proceso de testing en el que se generó un archivo bundle de la aplicación y se subió a la plataforma GOOGLE PLAY CONSOLE. Por medio de esta plataforma se puede

llevar a cabo la distribución de la aplicación de tal manera que las personas la puedan descargar desde el Google Play Store. Actualmente, solo determinadas personas (usuarios reales) mediante una autorización previa pueden descargarse la aplicación ya que se encuentra en proceso de testing. En este momento, este es el punto en el que se encuentra el proyecto. Se dejó para más adelante la finalización de este “User Testing” que daría por finalizada la fase de Testing.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

4.4.1 PLANIFICACIÓN

Para poder visualizar de una manera grafica todo el trabajo comentado anteriormente, se ha realizado un diagrama de Gantt. Es un tipo de diagrama de barras que ilustra el cronograma de un proyecto. Se puede observar el diagrama de Gantt con la planificación del proyecto en la Figura 11.

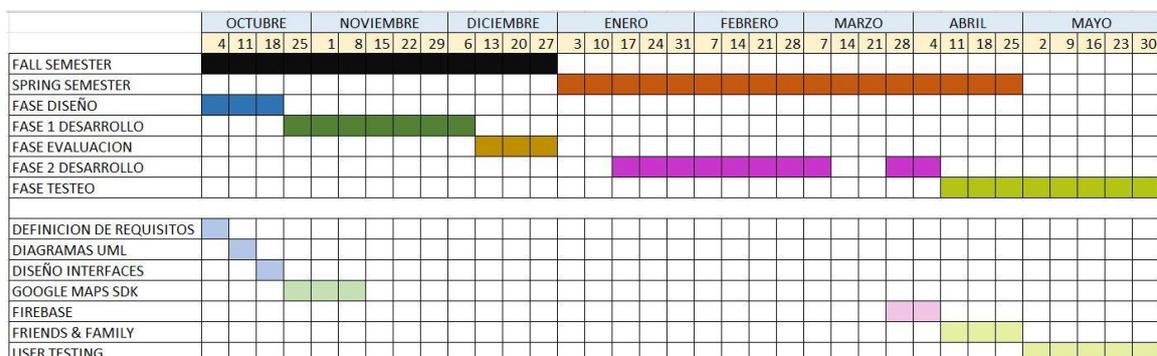


Figura 11. Diagrama de Gantt del Proyecto

4.4.2 ESTIMACIÓN ECONÓMICA

Para la estimación económica del proyecto, habrá que tener en cuenta cuales son los distintos planes de suscripción que tienen las plataformas utilizadas, además de hacer una estimación de la expansión de la aplicación.

- ANDROID STUDIO. Plataforma de uso gratuito.

- **GOOGLE MAPS PLATFORM.** Plataforma que ofrece 28.500 solicitudes de carga de mapas al mes sin coste alguno. A partir de ese número de solicitudes se pagarían 7\$ por cada 1.000 solicitudes. [9]

- **FIREBASE.** Plataforma que dispone de dos planes de uso.
 - **Plan Spark.** Plan de FIREBASE gratuito. Este plan permite llevar a cabo 10.000 autenticaciones de usuarios por mes y 1GB de almacenamiento en la real-time database. Además permite 100 conexiones a la vez con la real-time database.

 - **Plan Blaze.** Plan de FIREBASE de pago. Con este plan se deberán pagar 0,01\$ por cada usuario que se autentica y 5\$ por cada GB de almacenamiento de la real-time database. Este plan permite 200.000 conexiones a la vez con la real-time database. [10]

Actualmente la idea del proyecto es ver como se desarrolla en Gainesville, una ciudad del estado de Florida. Si el proyecto resulta satisfactorio en Gainesville, la idea es ampliarlo a todo el estado de Florida. y si es posible a todos los Estados Unidos.

Se ha llevado a cabo una estimación económica para el caso de Gainesville, ya que es el objetivo principal del proyecto, y para el caso de Florida, que sería el segundo objetivo de este proyecto.

- Gainesville es una ciudad de 133.000 habitantes de los cuales el 29,7% está por debajo del umbral de la pobreza. De esta forma se puede calcular que alrededor de 39.000 personas pueden estar sufriendo de inseguridad alimentaria. Si de esas 39.000 personas se consigue que un 5% se descargue la aplicación en el momento de salida de esta al mercado, supondría que alrededor de 2.000 personas estarían utilizando la aplicación. Si de esas 2.000 personas el uso medio de la aplicación es de 2 días a la semana, supondría un total de 16.000 solicitudes de carga de mapas. [11]

De esta forma el precio a pagar sería de 0 dólares.

GOOGLE MAPS PLATFORM permite 28.500 solicitudes de carga de mapas al mes. Como se ha mencionado anteriormente, se tendrían en torno a 16.000 solicitudes de carga de mapas al mes, por lo que no habría que pagar nada.

Con el plan Spark de FIREBASE se podría autenticar 2.000 usuarios el primer mes sin ningún coste, y posteriormente ir creciendo mes a mes sin problema alguno. Además, de esos 2.000 usuarios es muy poco probable que 100 accedan a la vez a la aplicación para darse de alta, que es el momento en el que se accede a la real-time database. Por último, 2.000 usuarios registrados en la aplicación supondrían un espacio de almacenamiento de 307KB más o menos, hasta llegar a 1GB se tendría espacio de sobra de almacenamiento.

- Florida es un estado de 21,22 millones de habitantes de los cuales el 12,4% está por debajo del umbral de la pobreza. De esta forma se puede calcular que alrededor de 2,6 millones de personas pueden estar sufriendo de inseguridad alimentaria. Si de esos 2,6 millones de personas se consigue que un 2% se descargue la aplicación en el momento de salida de esta al mercado, supondría que alrededor de 52.000 personas estarían utilizando la aplicación. Si de esas 52.000 personas el uso medio de la aplicación es de 2 días a la semana, supondría un total de 416.000 solicitudes de carga de mapas. [11]

De esta forma el precio a pagar sería de alrededor de 3.000\$.

GOOGLE MAPS PLATFORM permite 28.500 solicitudes de carga de mapas al mes. Como se ha mencionado anteriormente, se tendrían en torno a 400.000 solicitudes, por lo que se deben pagar las solicitudes excedentes a 7\$ cada 1.000 solicitudes. Esto hace un total de 2.600\$.

Con el plan Blaze de FIREBASE se tendrían que pagar 520\$ por autenticar 52.000 usuarios, y posteriormente al ir creciendo seguir pagando a 0,01\$ por autenticación de usuario. Este plan ofrece 200.000 conexiones múltiples a la real-time database por

lo que no habría ningún problema en este aspecto. Por último, 52.000 usuarios registrados en la aplicación supondrían un espacio de almacenamiento de 7,9GB más o menos, por lo que habría que pagar 40\$.

Este total de 3.000\$ se podría conseguir buscando financiación pública y privada.

- Pública, como por ejemplo recibir financiación por parte del estado de Florida. Si se les presentase la aplicación como una herramienta que pueden utilizar para aumentar la calidad de vida muchos ciudadanos del estado de Florida que están en riesgo de pobreza y sufriendo de inseguridad alimentaria, podrían ver correcta la inversión de 3.000\$.
- Privada, mediante eventos benéficos como carreras, subastas, cenas... se podría recibir ayuda económica en forma de donación por parte de personas o empresas que quieren ayudar sin ánimo de lucro a que la aplicación se expanda por todo el estado de Florida.

Capítulo 5. SISTEMA/MODELO DESARROLLADO

En este capítulo se va a exponer de manera detallada el sistema resultante del proyecto. Esta explicación se dividirá en dos epígrafes.

- En el primero se describirá como fueron los instantes iniciales del proyecto, en los que se definieron los requisitos y se llevó a cabo todo el diseño del sistema.
- En el segundo se describirá todo el proceso de desarrollo e implementación del sistema.

5.1 ANÁLISIS Y DISEÑO DEL SISTEMA

Durante el periodo inicial del proyecto se llevó a cabo un análisis preliminar de la solución al problema y la herramienta que se quería desarrollar. Este análisis se llevó a cabo junto a la Doctorada en Ciencias de la Alimentación y Nutrición Humana, Karla Shelnutt, y junto a un grupo de personas del programa ampliado de educación alimentaria y nutricional de la Universidad de Florida. Cabe mencionar que Karla Shelnutt fue quien le propuso al Dr. Benjamin C.Lok la idea de llevar a cabo este proyecto.

Tras ese análisis, quedaron definidos los requisitos funcionales de la aplicación, que son aquellos servicios de los que proveerá el sistema a los usuarios. Es decir, la manera en que el sistema reaccionará a entradas particulares.

Como se comentó anteriormente, a lo largo del desarrollo de la aplicación se han ido añadiendo, modificando y eliminando requisitos funcionales. Es por esto por lo que se va a hacer una diferenciación entre los requisitos principales, aquellos que se definieron durante el periodo inicial del proyecto, y los requisitos adicionales, aquellos que se definieron en algún momento posterior durante el desarrollo de la aplicación.

5.1.1 REQUISITOS PRINCIPALES

- La aplicación debe ser capaz de mostrar en un mapa los establecimientos y la ubicación del usuario.
- La aplicación debe ser capaz de enviar una notificación tras detectar un establecimiento cercano a la ubicación del usuario.
- La aplicación debe ser capaz de redirigir al usuario a Google Maps cuando este solicite direcciones hacia un establecimiento.
- La aplicación debe ser capaz de redirigir al usuario a la página Web de un establecimiento cuando este lo solicite.
- La aplicación debe ser capaz de redirigir al usuario a su aplicación de llamadas móviles cuando solicite llamar a un establecimiento.
- El usuario debe ser capaz de visualizar los establecimientos en forma de lista.
- El usuario debe ser capaz de filtrar los establecimientos para visualizar en el mapa o en la lista los que él quiera ver.
- El usuario debe ser capaz de cambiar el idioma de la aplicación entre español e inglés
- El usuario debe ser capaz de buscar un establecimiento determinado.
- El usuario debe ser capaz de ordenar los establecimientos de la lista por nombre o distancia hasta la ubicación del mismo.
- El usuario debe ser capaz de cambiar el tipo de notificación que enviara la aplicación y el tiempo que debe transcurrir entre dos notificaciones.

5.1.2 REQUISITOS ADICIONALES

- La aplicación debe ser capaz de guardar la configuración del usuario en la memoria caché del dispositivo móvil.
- La aplicación debe ser capaz de comunicarse con una base de datos para guardar la información sobre los usuarios y poder autenticarlos.
- El usuario debe ser capaz de llevar a cabo los procesos de Registro, Inicio de Sesión y Cambio de Contraseña.
- El usuario debe ser capaz de eliminar su cuenta.

Tras ese análisis preliminar, y con los requisitos funcionales principales definidos, se llevó a cabo el diseño inicial de la aplicación a través los diagramas de casos de uso y de navegación. Estos diagramas, como se comentó anteriormente, han ido cambiando a lo largo del proceso de desarrollo de la aplicación debido a los cambios que han sufrido los requisitos funcionales.

- El diagrama de casos de uso representa todos los procesos que se pueden llevar a cabo con la aplicación por parte del usuario, así como los procesos propios del sistema. En este caso se diferencian dos tipos usuarios, los usuarios no identificados y los identificados, cada uno con un tipo de funciones diferentes.
- El diagrama de navegación representa la estructura del sistema y las posibilidades que tiene el usuario en el momento de uso del sistema.

Los diagramas finales de casos de uso y navegación se pueden observar en las Figuras 12 y 13.

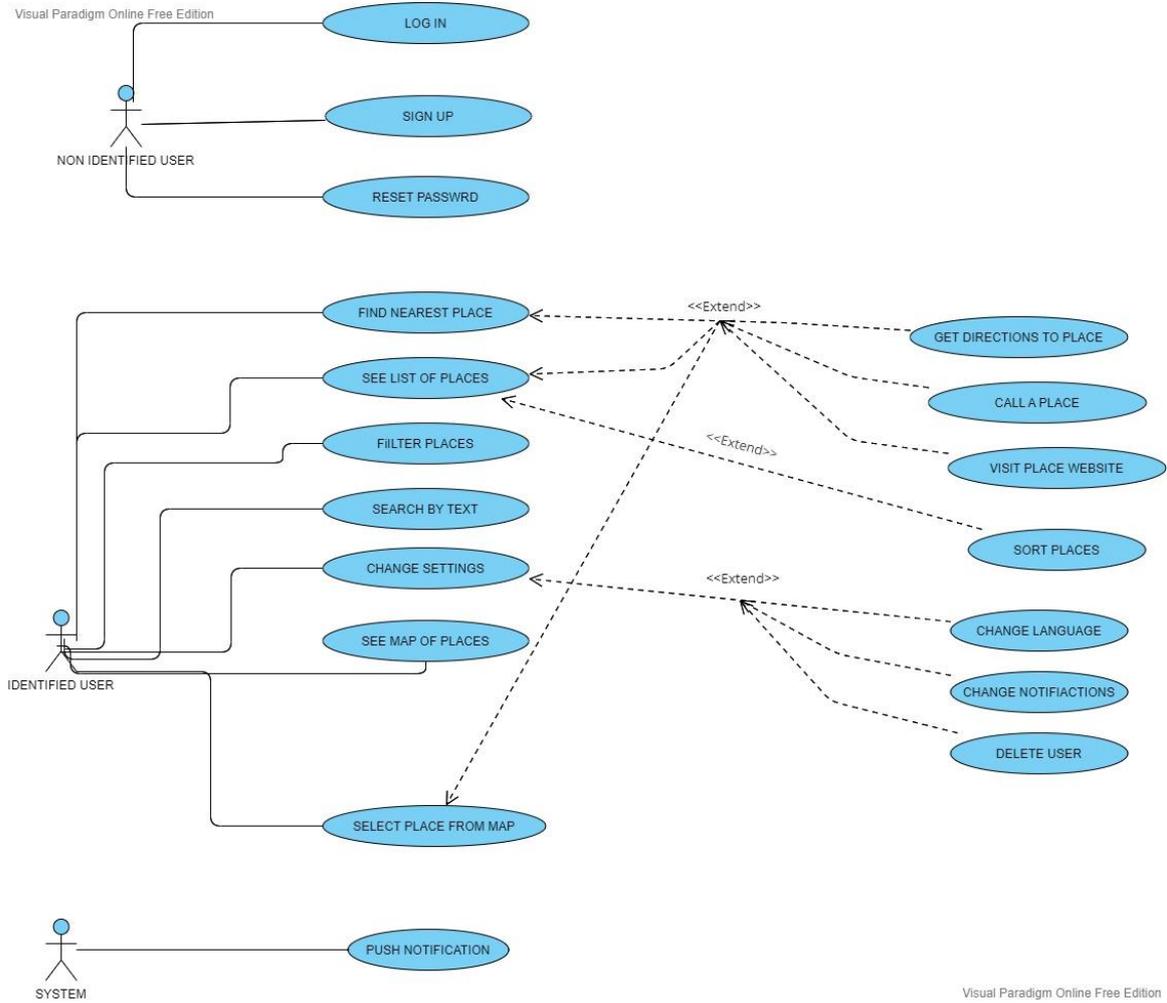


Figura 12. Diagrama de Casos de Uso

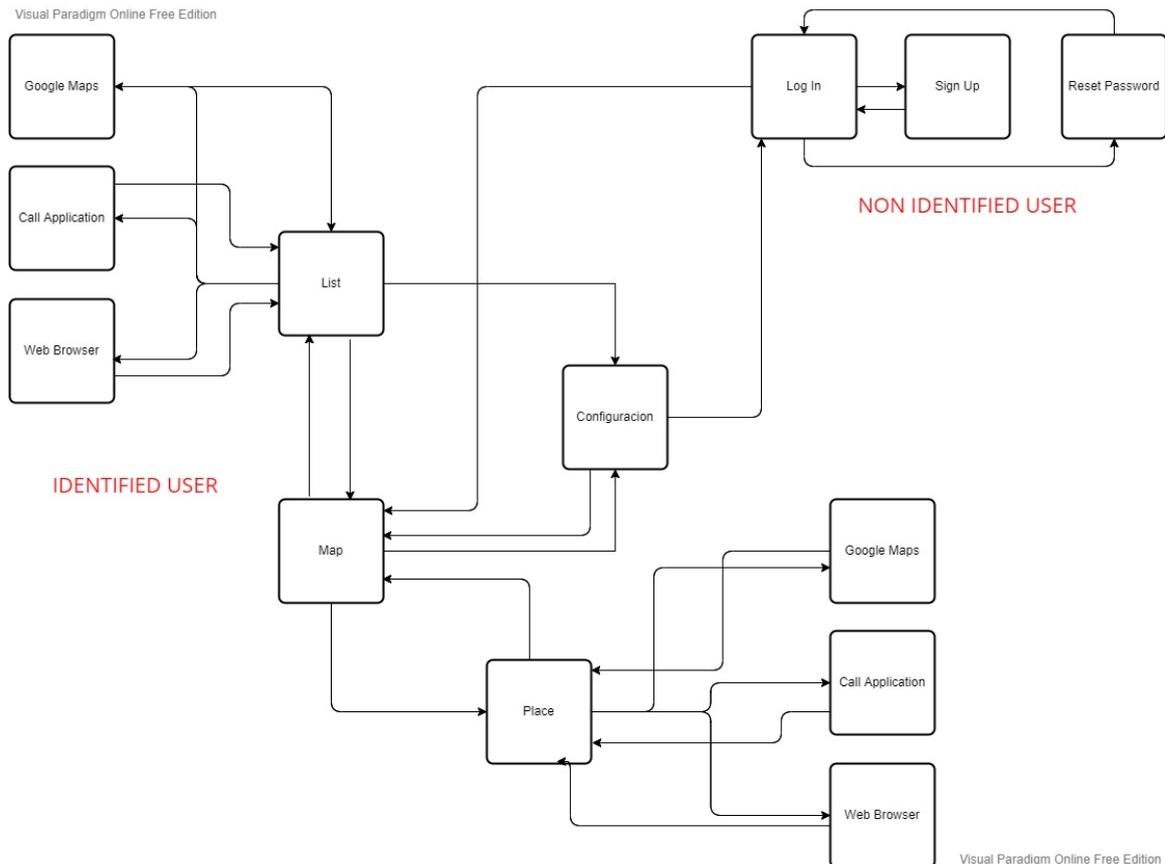


Figura 13 Diagrama de Navegación

Tras el diseño de los diagramas de casos de uso y de navegación, se llevó a cabo el diseño de tres posibles interfaces para la aplicación. Cada una de las interfaces tenía un diseño gráfico distinto, pero todas cumplían con las mismas funcionalidades. Tras una evaluación de los tres posibles diseños de la interfaz de la aplicación que se llevó a cabo durante una reunión con distintas personas de la Universidad de Florida, se eligió el tercer diseño gráfico de la interfaz como el más adecuado, sencillo y atractivo visualmente. Como se podrá observar posteriormente, se consiguió que el resultado final fuera muy parecido al prototipo realizado en esta fase.

En las Figuras 14, 15 y 16 se pueden observar los tres diseños que se realizaron.



Figura 14. Diseño 1 de la Interfaz de la Aplicación



Figura 15. Diseño 2 de la Interfaz de la Aplicación

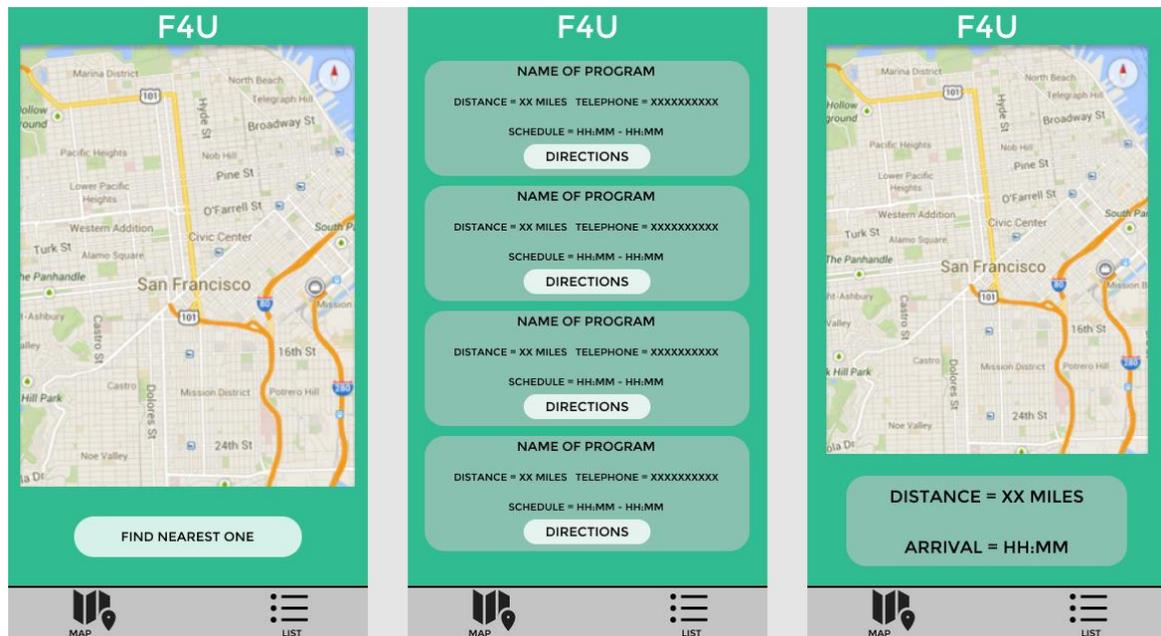


Figura 16. Diseño 3 de la Interfaz de la Aplicación

5.2 DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA

Tras el análisis preliminar de la solución al problema y la herramienta que se quería desarrollar, tras el diseño de los diagramas de casos de uso y de navegación y tras el diseño de la interfaz de la aplicación, comenzó la fase de desarrollo e implementación del sistema.

En este apartado se comentará el resultado final obtenido, por lo que se mostrarán todas las pantallas que componen la aplicación y se describirá su lógica asociada.

5.2.1 LOG IN

La primera pantalla que se le muestra el usuario al acceder a la aplicación es la de inicio de sesión. Se puede observar en la Figura 17.

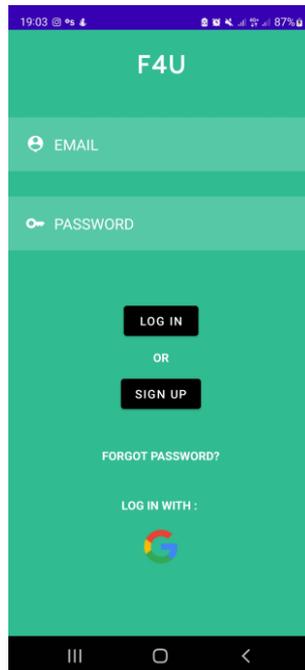


Figura 17. Pantalla de Inicio de Sesión

Aquí el usuario puede realizar diferentes acciones:

- El usuario puede iniciar sesión en la aplicación siempre y cuando tenga una cuenta registrada y autenticada por la plataforma de FIREBASE. En el caso de tener una cuenta registrada pero no autenticada, en el momento de intentar iniciar sesión se le advertirá que primero debe ir a su correo electrónico y autenticar su cuenta. A partir de ese momento, ya podría llevar a cabo el inicio de sesión. También puede iniciar sesión con la plataforma de Google, de tal manera que es Google quien lleva a cabo la autenticación de la cuenta del usuario. Esto se puede observar en la Figura 18. En la Figura 19 se puede observar que en la plataforma de FIREBASE hay un usuario tanto en la base de datos y como en la plataforma de autenticación. Teniendo en cuenta que ese usuario todavía no ha autenticado su cuenta, cuando intente iniciar sesión le aparecerá la advertencia de que primero debe autenticar su cuenta. Tras autenticarla, a través de su correo electrónico, podrá iniciar sesión y entrar a la aplicación.

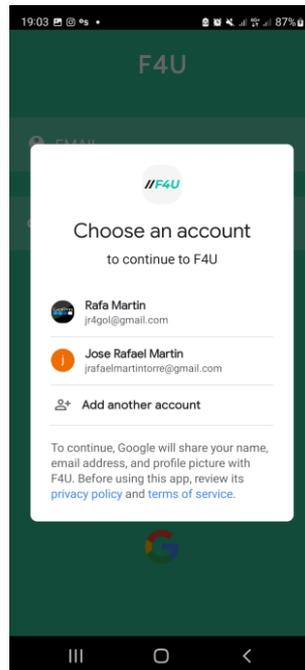


Figura 18. Pantalla de Inicio de Sesión con Google

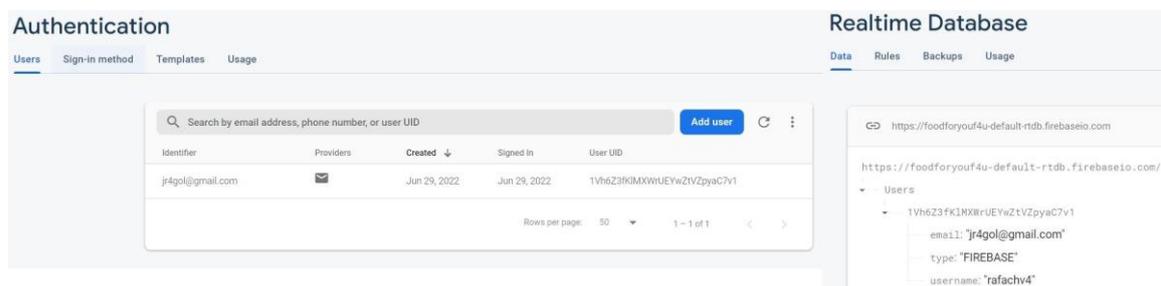


Figura 19. Plataforma FIREBASE Log In

- Si el usuario no tiene una cuenta registrada puede pulsar el botón de Sign Up. En ese momento la aplicación le llevara a la pantalla correspondiente para poder hacer el registro.
- Si el usuario ha olvidado su contraseña y desea cambiarla puede pinchar sobre el texto de “Forgot Password?”. En ese momento la aplicación le llevara a la pantalla correspondiente para poder cambiar la contraseña.

A continuación se muestra la parte de código fuente más importante con respecto a la pantalla de Log In:

COMPROBACIÓN DE FIREBASE PARA EL INICIO DE SESIÓN

Primero se comprueba que las credenciales acreditadas por el usuario (correo electrónico y contraseña) corresponden a las de un usuario de la plataforma de autenticación. Después de esto, se comprueba si el usuario ha autenticado anteriormente su cuenta mediante la verificación de su correo electrónico. Si no está autenticado se le enviará un email de verificación, y si esta autenticado se le dará paso a la siguiente pantalla de la aplicación.

```
mFirebaseAuth.signInWithEmailAndPassword(user,pass)
.addOnCompleteListener(logInActivity.this, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()){
            FirebaseUser user = mFirebaseAuth.getCurrentUser();
            assert user != null;
            if(user.isEmailVerified()){
                Intent intent = new Intent(getApplicationContext(),
MapPpal.class);
                intent.putExtra("language", language);
                startActivity(intent);
            }else{
                user.sendEmailVerification()
                    .addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if (task.isSuccessful()) {
                                Toast.makeText(logInActivity.this, "CHECK YOUR
EMAIL TO VERIFY YOUR ACCOUNT", Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
            }
        }
    }
});

}else{
    Toast.makeText(logInActivity.this, "UNSUCCESSFUL
LOGIN", Toast.LENGTH_SHORT).show();
    username.getText().clear();
    password.getText().clear();
}
});
```

INICIO DE SESIÓN UTILIZANDO LA PLATAFORMA DE GOOGLE

Mediante este mecanismo se coge la información de la cuenta de Google del usuario que ha seleccionado iniciar sesión a través de Google, y se crea un objeto “User” con esa información. Después de esto, el usuario se añade a la base de datos para tener constancia de que un nuevo usuario ha entrado en la aplicación. Por el contrario, esa información del usuario no se añade a la plataforma de autenticación ya que es la plataforma de Google la que realiza la autenticación utilizando sus propios medios.

```
task.getResult(ApiException.class);
GoogleSignInAccount gsa = GoogleSignIn.getLastSignedInAccount(this);

User user = new User(gsa.getEmail(), gsa.getDisplayName(), "GOOGLE");

FirebaseDatabase.getInstance().getReference("Users")
    .child(gsa.getId())
    .setValue(user).addOnCompleteListener(new OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if(task.isSuccessful()){
            Intent intent = new Intent(getApplicationContext(), MapPpal.class);
            intent.putExtra("language", language);
            startActivity(intent);
        }else{
            Toast.makeText(logInActivity.this, "UNSUCCESSFUL LOG
IN", Toast.LENGTH_SHORT).show();
        }
    }
});
```

5.2.2 SIGN UP

Otra pantalla que puede manejar el usuario dentro de la aplicación es la de registro. En esta pantalla el usuario es capaz de crearse una cuenta para poder entrar a la aplicación. Debe rellenar una serie de campos como nombre de usuario, correo electrónico, contraseña... Tras llevar a cabo el registro, el usuario se añade a la base de datos de la plataforma de FIREBASE y a la plataforma de autenticación, donde se almacena su correo electrónico y su contraseña encriptada. Como se comentó anteriormente, no está autenticado hasta que verifica su correo electrónico.

Hay que mencionar, que antes de efectuar el registro, se comprueba que el correo electrónico tenga un dominio correcto, que la contraseña tenga más de 6 caracteres (requerimiento de FIREBASE), y que las dos contraseñas coincidan.

En la Figura 20 se puede apreciar la pantalla de registro, y en la Figura 21 la incorporación del usuario a la base de datos y a la plataforma de autenticación.



Figura 20. Pantalla de Registro

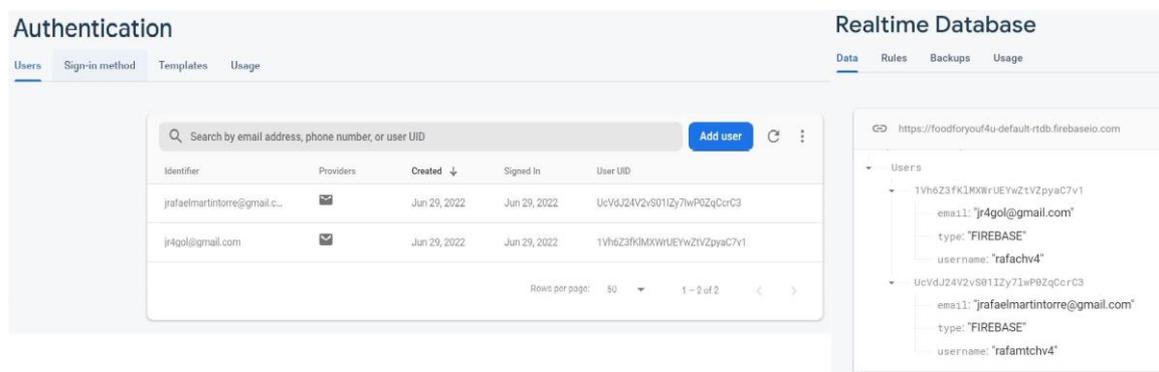


Figura 21. Plataforma FIREBASE Sign Up

Como se puede observar en la anterior Figura, en la base de datos hay un campo que es “type”. Este campo permite identificar a los usuarios en función de si su registro se ha efectuado a través de la aplicación, o a través de la plataforma de Google.

A continuación se muestra la parte de código fuente más importante con respecto a la pantalla de Sign Up:

REGISTRO DE UN NUEVO USUARIO MEDIANTE FIREBASE

Tras realizar las comprobaciones pertinentes, como se comprueba en el código fuente, se añade el usuario con el correo electrónico y la contraseña encriptada a la plataforma de autenticación. Además, se crea un nuevo objeto “User” para añadir a la base de datos ese nuevo usuario que se ha registrado. Como se comentó anteriormente, no está autenticado hasta que verifique su correo electrónico.

```
mFireBaseAuth.createUserWithEmailAndPassword(email, password1)
    .addOnCompleteListener(signupActivity.this, new OnCompleteListener<AuthResult>()
    {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if(task.isSuccessful()){
                User user = new User(email, userName, "FIREBASE");

                FirebaseDatabase.getInstance().getReference("Users")
                    .child(FirebaseAuth.getInstance().getCurrentUser().getUid())
                    .setValue(user).addOnCompleteListener(new
                OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        if(task.isSuccessful()){
                            Intent intent = new Intent(getApplicationContext(),
                loginActivity.class);
                            intent.putExtra("language", language);
                            startActivity(intent);
                        }else{
                            Toast.makeText(signupActivity.this, "UNSUCCESSFUL SIGN
                UP", Toast.LENGTH_SHORT).show();
                        }
                    }
                });
            }else{
                Toast.makeText(signupActivity.this, "UNSUCCESSFUL SIGN
                UP", Toast.LENGTH_SHORT).show();
            }
        }
    });
```

5.2.3 RESET PASSWORD

Otra pantalla que puede manejar el usuario dentro de la aplicación es la del cambio de la contraseña. En esta pantalla, el usuario es capaz de solicitar un cambio de contraseña mediante la introducción de su correo electrónico. La aplicación primero comprueba que el correo electrónico tenga un dominio correcto. Después, a través de la plataforma de FIREBASE comprueba si el correo electrónico se encuentra en la base de datos. Si la comprobación es correcta, el usuario recibirá un email con un enlace para el cambio de contraseña. Tras cambiar la contraseña, FIREBASE de manera automática vuelve a guardar el usuario en la en la plataforma de autenticación con su nueva contraseña.

En la Figura 22 se puede observar la pantalla de cambio de contraseña y como se cambia la contraseña tras pinchar el link que se recibe a través del correo.

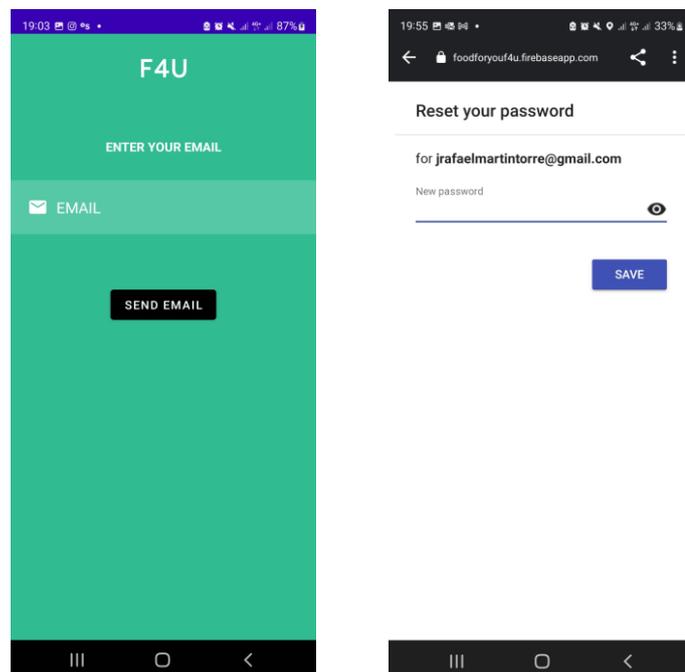


Figura 22. Pantallas de Cambio de Contraseña

A continuación se muestra la parte de código fuente más importante con respecto a la pantalla de Reset Password:

RESETEAR CONTRASEÑA MEDIANTE FIREBASE

Tras comprobar que el usuario existe en la base de datos de FIREBASE, como se comprueba en el código fuente, se le enviará un email al correo electrónico introducido por el usuario. Mediante ese email será capaz de cambiar su contraseña. Cabe mencionar, que la actualización del usuario con su nueva contraseña en la plataforma de autenticación la lleva a cabo FIREBASE de manera automática.

```
mFireBase.sendPasswordResetEmail(email).addOnCompleteListener(new
OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if(task.isSuccessful()){
            Toast.makeText(forgotPasswordActivity.this,"CHECK YOUR EMAIL TO RESET
YOUR PASSWORD",Toast.LENGTH_SHORT).show();

        }else {
            Toast.makeText(forgotPasswordActivity.this,"TRY AGAIN, SOMETHING WENT
WRONG",Toast.LENGTH_SHORT).show();
        }
    }
}
```

5.2.4 MAPA

La primera pantalla a la que accede el usuario tras iniciar sesión en la aplicación es la del mapa. Se puede observar en la Figura 23.

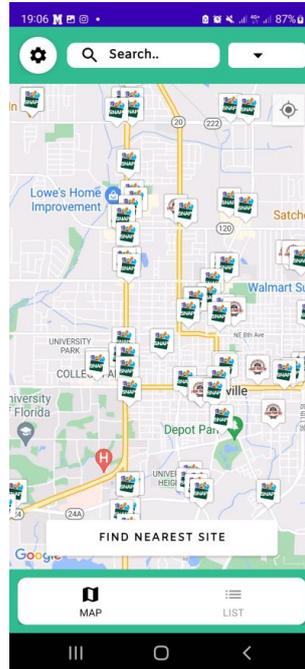


Figura 23. Pantalla del Mapa

Aquí el usuario puede realizar diferentes acciones:

- El usuario puede seleccionar un establecimiento del mapa, de tal modo que podrá observar la aparición de un Pop-Up en la parte inferior de la aplicación. Ese Pop-Up dispondrá de información acerca del tiempo andando que tardaría el usuario hasta el establecimiento, y si se encuentra abierto o cerrado. Además, mediante el uso de los tres botones inferiores que tiene el Pop-Up, el usuario podrá, abrir la aplicación de Google Maps para obtener las direcciones hasta el establecimiento, llamar por teléfono al establecimiento, y visitar su página Web. En la Figura 24 se puede observar todo lo anterior mencionado.

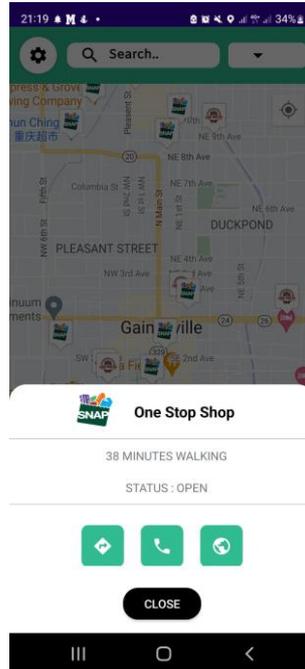


Figura 24. Marcador del Mapa Seleccionado

- El usuario también puede buscar el establecimiento más cercano a su ubicación pulsando el botón de “Find Nearest Site”. En ese momento, aparecerá el mismo Pop-Up anteriormente mencionado pero con la información del establecimiento más cercano a la ubicación del usuario. Esto se puede observar en la Figura 25.

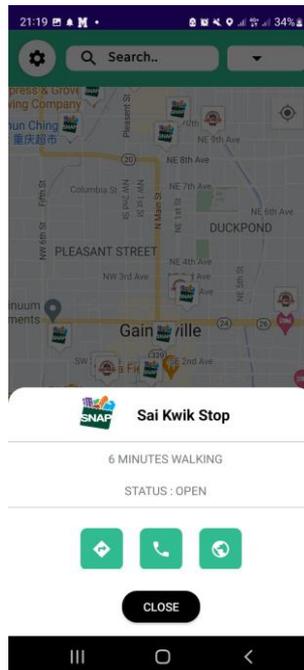


Figura 25. Establecimiento más Cercano

- El usuario también tiene la posibilidad de buscar un establecimiento mediante el uso de la barra de búsqueda. En función de los caracteres introducidos, la aplicación llevará a cabo una búsqueda entre todos los establecimientos que el usuario esté viendo, y mostrará en el mapa solo aquellos que contengan los caracteres introducidos por el usuario. Cabe mencionar, que tras realizar una búsqueda y presionar de nuevo al botón “Find Nearest Site”, se mostrará el establecimiento que, además de contener los caracteres introducidos, sea el más cercano a la ubicación del usuario. Esto se puede observar en la Figura 26, al introducir “Publix”, en el mapa solo se muestran los establecimientos que contienen esos caracteres. Además, al presionar el botón “Find Nearest Site” ocurre lo mencionado anteriormente.

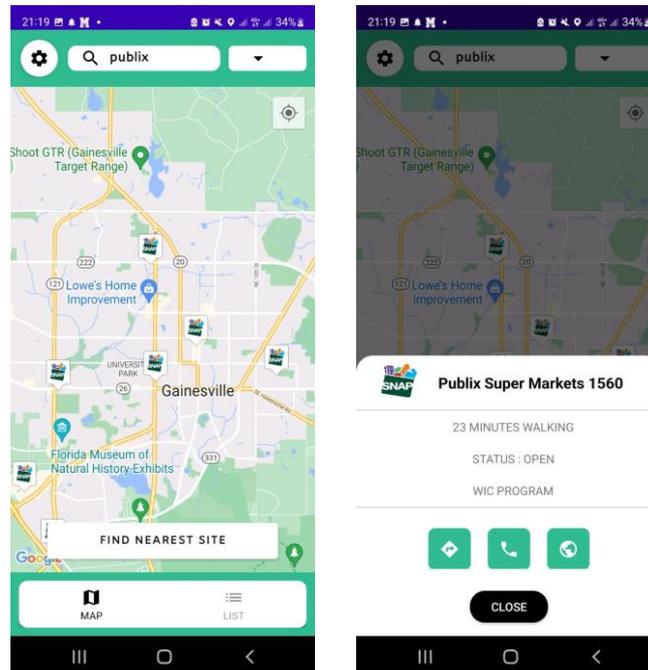


Figura 26. Búsqueda de Establecimientos en el Mapa

- El usuario también tiene la opción de cambiar el tipo de establecimiento que aparece en el mapa en función de lo que él quiera visualizar. Tiene tres posibilidades: ver solo los Bancos de Alimentos, ver solos los Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria, o ver ambos tipos de establecimientos. Como consecuencia de esto, el botón de “Find Nearest Site” que aparecía cuando se visualizaban todos los establecimientos, puede cambiar a “Find Nearest Food Pantry” o “Find Nearest Snap Retailer” en función de lo que se elija. Cabe mencionar, que cuando el usuario realice una búsqueda, esta se efectuará solo sobre el tipo de establecimiento que ha seleccionado ver el usuario. Todo esto se puede observar en la Figura 27.

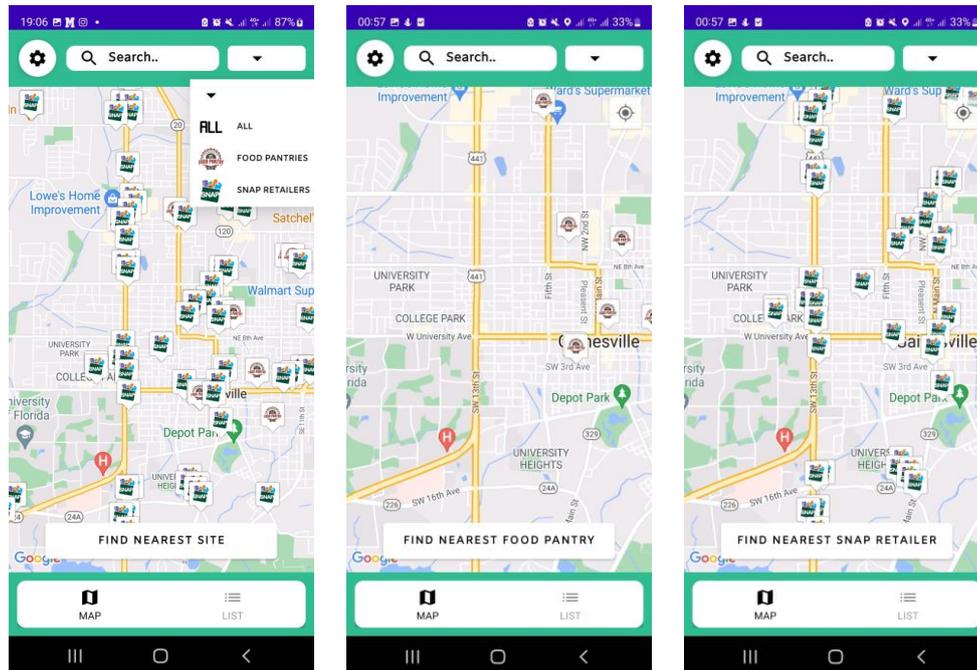


Figura 27. Filtro de Establecimientos en el Mapa

A continuación se muestra la parte de código fuente más importante con respecto a la pantalla del Mapa:

OBTENER LA UBICACIÓN DEL DISPOSITIVO MÓVIL

Para obtener la ubicación del dispositivo móvil se debe llevar a cabo de manera satisfactoria una “task” (tarea). Esa “task” consiste en obtener la última localización del dispositivo móvil. Si se consigue obtener esa última localización, es decir, la tarea se resuelve correctamente, se obtiene la ubicación del dispositivo móvil.

```
private void getDeviceLoctaion(){
    Log.d(TAG, "getDeviceLoctaion: getting current location");
    mFusedProviderClient = LocationServices.getFusedLocationProviderClient(this);
    try{
        Task location = mFusedProviderClient.getLastLocation();
        location.addOnCompleteListener(new OnCompleteListener() {
            @Override
            public void onComplete(@NonNull Task task) {
                if(task.isSuccessful()){

                    Log.d(TAG, "onComplete: found location");
                    Location currentLocation = (Location) task.getResult();

                    current = currentLocation;

                    if(currentLocation != null && firstLoc) {
                        moveCamera(new LatLng(currentLocation.getLatitude(),
currentLocation.getLongitude()), DEFAULT_ZOOM);
                        firstLoc = false;
                    }
                    }else{
                        Toast.makeText(MapPpal.this, "unable to get current
location", Toast.LENGTH_SHORT).show();
                    }
                });
            }catch (SecurityException e){
                Log.e(TAG, "getDeviceLocation : SecurityException : " + e.getMessage());
            }
        }
    }
```

En el Anexo II se puede apreciar una de las funcionalidades principales de la aplicación que se codificó, que es la de buscar el establecimiento más cercano a la ubicación del usuario.

5.2.5 LISTA

El usuario navegando por la aplicación puede llegar a la pantalla de la lista. Se puede observar en la Figura 28.

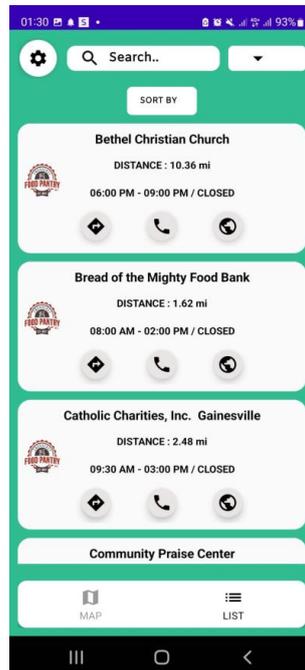


Figura 28. Pantalla de la Lista

Aquí el usuario puede realizar diferentes acciones:

- El usuario puede deslizar la lista para observar los establecimientos que hay en ella y la información que se aporta sobre cada uno de ellos (distancia andando en millas y si se encuentra abierto o cerrado). Además, mediante el uso de los botones de cada uno de los establecimientos, el usuario podrá, abrir la aplicación de Google Maps para obtener las direcciones hasta el establecimiento, llamar por teléfono al establecimiento, y visitar su página Web.
- El usuario también puede buscar un establecimiento mediante el uso de la barra de búsqueda. En función de los caracteres introducidos, la aplicación llevara a cabo una búsqueda entre todos los establecimientos que el usuario este viendo, y mostrará en la lista solo aquellos que contengan los caracteres introducidos por el usuario. Esto se puede observar en la Figura 29. Al introducir “market” en la barra de búsqueda, en la lista solo se muestran los establecimientos que contienen esos caracteres.

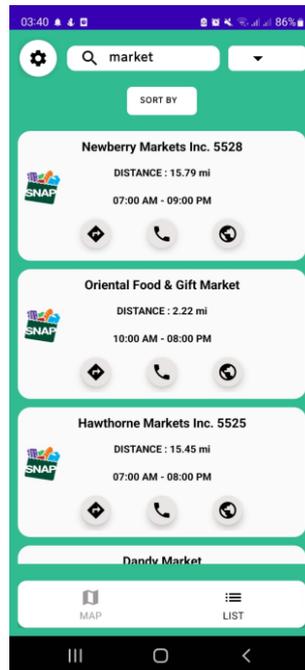


Figura 29. Búsqueda de Establecimientos en la Lista 1

- El usuario también tiene la posibilidad de cambiar el tipo de establecimiento que aparece en la lista en función de lo que él quiera visualizar. Tiene tres posibilidades: ver solo los Bancos de Alimentos, ver solos los Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria, o ver ambos tipos de establecimientos. El funcionamiento es idéntico al de la pantalla del mapa pero trabajando con una lista. Cabe mencionar, que cuando el usuario realice una búsqueda, esta se efectuará solo sobre el tipo de establecimiento que ha seleccionado ver el usuario. Por ejemplo, si el usuario solo está viendo Bancos de Alimentos y busca “Publix”, no aparecerá nada en la lista ya que Publix es un Comercio Minorista del Programa de Asistencia Nutricional Suplementaria. Esto se puede observar en la Figura 30.



Figura 30. Búsqueda de Establecimientos en la Lista 2

- El usuario también tiene la opción de ordenar los establecimientos de la lista por orden alfabético o por distancia hasta su ubicación. Como en todos los casos, se tiene en cuenta el tipo de establecimiento que ha elegido ver el usuario en la lista para llevar a cabo esa ordenación de los mismos. Además, si se realiza una búsqueda, se ordenarán y mostrarán solo los establecimientos que estén dentro de esa búsqueda. Por ejemplo, si se busca “Dollar” y se ordena por distancia aparecerán en orden los establecimientos que contengan “Dollar”. Esto se puede observar en la Figura 31.

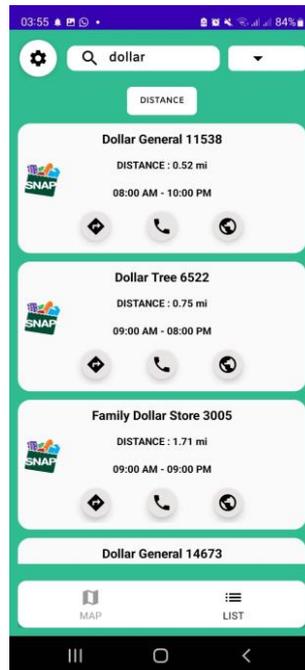


Figura 31. Ordenación de Establecimientos en la Lista

A continuación se muestra la parte de código fuente más importante con respecto a la pantalla de la Lista:

USO DE GOOGLE MAPS, NAVEGADOR WEB Y APP DE LLAMADA

El uso de Google Maps, la aplicación de llamadas y el navegador Web, se hace a través de los “intents”, que son objetos que vinculan dos componentes separados, como son las “activities”, de manera que se puedan pasar información entre ellas. En este caso, a cada “intent” que se crea se le establecen unos datos determinados. Posteriormente, se ejecutan las “activities”, las cuales tendrán una funcionalidad distinta en función del “intent” que se les ha especificado.

```
if(website.contains("https://")){
    Intent intent = new Intent(Intent.ACTION_VIEW).setData(Uri.parse(website));
    startActivity(intent);
}else{
    website = "https://" + website;
    Intent intent = new Intent(Intent.ACTION_VIEW).setData(Uri.parse(website));
    startActivity(intent);
}
overridePendingTransition(10, 10);

Intent intent = new Intent(Intent.ACTION_DIAL);
intent.setData(Uri.parse("tel:"+number));
startActivity(intent);
overridePendingTransition(10, 10);

Uri gmmIntentUri = Uri.parse("geo:0,0?q="+address);
Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
mapIntent.setPackage("com.google.android.apps.maps");
startActivity(mapIntent);
overridePendingTransition(10, 10);
```

Cabe mencionar, que toda la información correspondiente al idioma de la aplicación, tipo de establecimiento que se está visualizando, la localización del usuario... se pasa de una pantalla a otra a través también de los “intents”.

El siguiente código fuente refleja cómo se pasa información de una “activity” a otra:

ENVÍO Y RECEPCIÓN DE INFORMACIÓN

El “intent” sirve para, como se ha mencionado anteriormente, pasar información de una “activity” a otra. Esto se hace mediante el comando `.putExtra`, que permite añadir al “intent” una variable a la que se la asocia una cadena de texto. De esta forma, en el resto de actividades cuando se quiera acceder a esa variable se debe utilizar el comando `.getExtra`, al que se debe indicar la cadena de texto asociada a la variable.

Un ejemplo sería: `minutos = intent.getIntExtra("minutes", 0)`. El segundo valor que se indica es el valor por defecto que se asociara a minutos en el caso de que no haya una variable asociada a la cadena de texto “minute” en el “intent”.

```
Intent intent = new Intent(getApplicationContext(), Listactivity.class);
intent.putExtra("pantrieVsSnap",pantrieVsSnap);
intent.putExtra("foodPantryLocated",fbIs);
intent.putExtra("snapLocated",snapIs);
intent.putExtra("minutes",minutes);
intent.putExtra("allInfo",allInfo);
intent.putExtra("language",language);
intent.putExtra("notiFP",notFP);
intent.putExtra("notiSNAP",notSNAP);
intent.putExtra("firstTime",firstLoc);
intent.putExtra("longitude",longitude);
intent.putExtra("latitude",latitude);
try {
    intent.putExtra("latitude",current.getLatitude());
    intent.putExtra("longitude",current.getLongitude());
}catch(Exception e){
    Log.d(TAG, "onNavigationItemSelected: error " + e.getMessage());
}

if(foodPantrySearch == null){
    intent.putExtra("foodPantryCurrent",0);
}else{
    intent.putExtra("foodPantryCurrent",foodPantrySearch.getIndex());
}

if(snapSearch == null){
    intent.putExtra("snapCurrent",0);
}else{
    intent.putExtra("snapCurrent",snapSearch.getIndex());
}
```

RECEPCIÓN DE INFORMACIÓN

```
Intent intent = getIntent();
pantrieVsSnap = intent.getBooleanExtra("pantrieVsSnap", true);
fbIs = intent.getBooleanExtra("foodPantryLocated", false);
snapIs = intent.getBooleanExtra("snapLocated", false);
indexFoodPantry = intent.getIntExtra("foodPantryCurrent",0);
indexSnap = intent.getIntExtra("snapCurrent",0);
firstLoc = intent.getBooleanExtra("firstTime",true);
minutes = intent.getIntExtra("time", 2);
allInfo = intent.getBooleanExtra("allInfo",true);
language = intent.getBooleanExtra("language",true);
notFP = intent.getBooleanExtra("notiFP",true);
notSNAP = intent.getBooleanExtra("notiSNAP",true);
longitude = intent.getDoubleExtra("longitude",0);
latitude = intent.getDoubleExtra("latitude",0);
```

En el Anexo III se puede apreciar una de las funcionalidades principales de la aplicación que se codificó, que es la de ordenar los establecimientos por distancia.

5.2.6 CONFIGURACIÓN

El usuario navegando por la aplicación puede llegar a la pantalla de configuración. Se puede observar en la Figura 32.

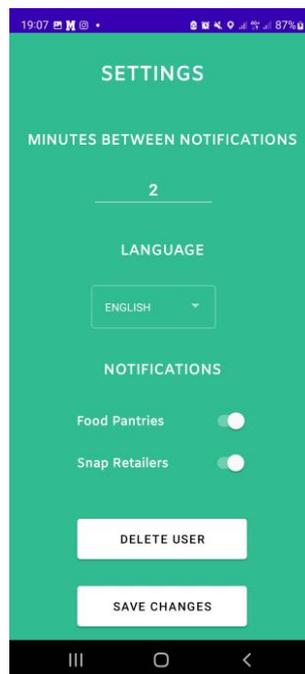


Figura 32. Pantalla de Configuración

Aquí el usuario puede realizar diferentes acciones:

- Con respecto a las notificaciones que manda la aplicación al usuario cuando este se encuentra cerca de Bancos de Alimentos o Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria, el usuario puede configurar dos cosas:
 - El tipo de notificación que quiere recibir. El usuario mediante la interacción con dos switches será capaz de escoger el tipo de notificación a recibir. De Bancos de Alimentos y de Comercios Minoristas del Programa de Asistencia Nutricional Suplementaria, solo de Bancos de Alimentos o Comercios

Minoristas del Programa de Asistencia Nutricional Suplementaria, o ningún tipo de notificación.

- El tiempo que debe transcurrir entre dos notificaciones mandadas por la aplicación. El usuario puede configurar el tiempo que debe transcurrir entre ellas para que la aplicación mande una segunda notificación si sigue detectando algún establecimiento cercano a la ubicación del usuario. En la figura 33 se puede ver el ejemplo de una notificación.

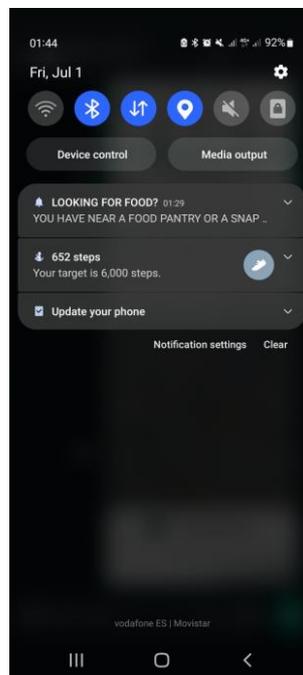


Figura 33. Ejemplo de Notificación

- El usuario también es capaz de cambiar el idioma de la aplicación. Podrá elegir entre español o inglés. A través de los “intents” anteriormente mencionados, esa información del idioma de la aplicación se pasa a las distintas “activities”. Por eso, cuando se inicia una “activity”, lo primero que se hace es una comprobación del ”intent” recibido. De esta manera, mediante el campo “language” del intent, se sabe el idioma en el que debe estar la aplicación y todos los textos de la interfaz de la aplicación se escriben en ese idioma. En la Figura 34 se muestra la aplicación en español.

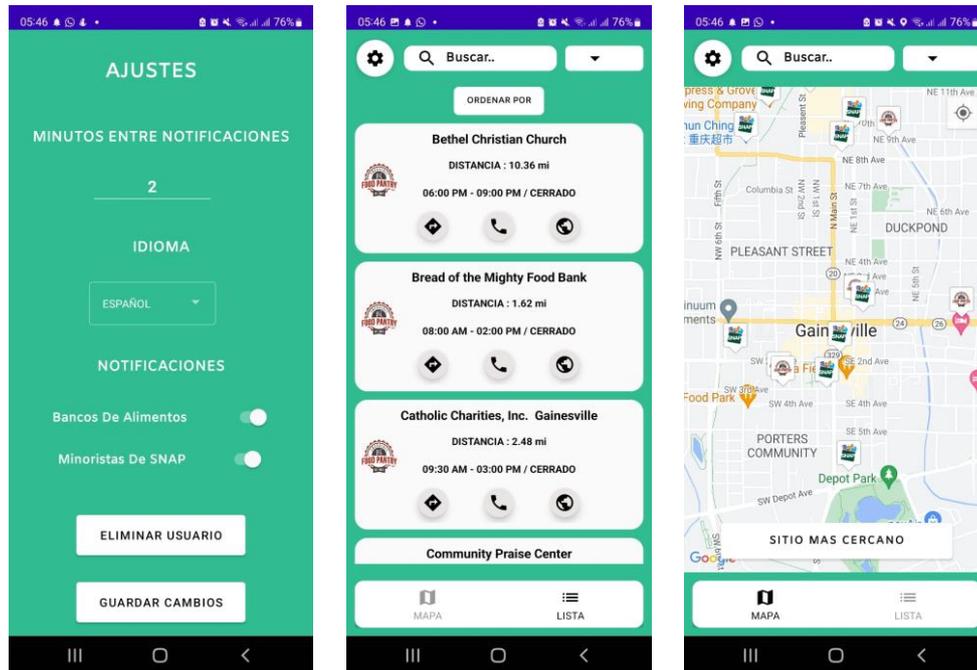
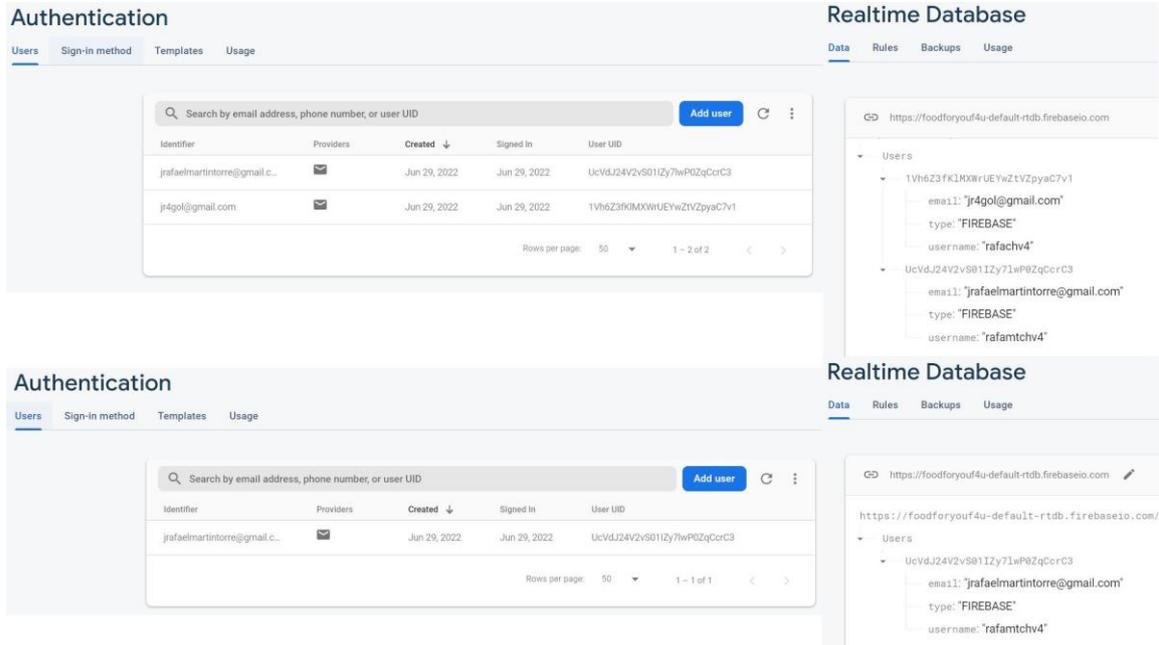


Figura 34. Aplicación en Español

- El usuario también es capaz de eliminar su cuenta de la aplicación. Esto provoca la eliminación del usuario de la base de datos de FIREBASE y de la plataforma de autenticación. Además, tras eliminar su cuenta, el usuario aparece de nuevo en la pantalla de Log In. En la Figura 35 se muestra el cambio de la real-time database de FIREBASE y de la plataforma de autenticación una vez eliminado el usuario asociado al correo electrónico jr4gol@gmail.com.



The image displays two screenshots of the Firebase console interface. The top screenshot shows the 'Authentication' section with a table of users and the 'Realtime Database' section showing a JSON tree with two user entries. The bottom screenshot shows the 'Authentication' section with only one user entry and the 'Realtime Database' section showing only one user entry, demonstrating the deletion of a user.

Figura 35. Plataforma FIREBASE Delete User

Cabe mencionar, que para que la configuración se guarde, se ha de pinchar en el botón “Save Changes”. Tras pinchar en el botón, la aplicación lleva a cabo dos tareas.

- Actualiza los valores de determinadas variables que se pasan de “activity” en “activty” a través de los intents. Esas variables son las que determinan el idioma de la aplicación, el tiempo que tiene que transcurrir notificaciones, que tipo de notificaciones quiere recibir el usuario...
- Guarda esa configuración del usuario en la memoria caché del dispositivo móvil. Se hace esto para que cuando el usuario cierre la aplicación y posteriormente vuelva a abrirla, se mantengan sus configuraciones. Para esto se hace uso de las “Shared Preferences”, que permiten tanto cargar variables almacenadas en la memoria caché, como actualizar y guardar nuevos valores.

A continuación se muestra la parte de código fuente más importante con respecto a la pantalla de Configuración:

USO DE LAS SHARED PREFERENCES Y LOS INTENTS PARA GUARDAR LA CONFIGURACIÓN DEL USUARIO

Para cargar la configuración del usuario cuando se inicia la aplicación se deben obtener las “Shared Preferences”, mediante las cuales, en este caso, se obtienen de la memoria caché las variables relativas a los minutos entre las notificaciones que envía el sistema, el idioma de la aplicación, y los valores de los switches con los que se indica que tipo de notificaciones desea recibir el usuario. Para guardar la información de configuración cuando el usuario desea se deben obtener las “Shared Preferences”, mediante las cuales, en este caso, se guardan en la memoria caché las variables relativas a los minutos entre las notificaciones que envía el sistema, el idioma de la aplicación, y los valores de los switches con los que se indica que tipo de notificaciones desea recibir el usuario.

```
private void loadDataUser() {
    SharedPreferences sharedPreferences =
getSharedPreferences(SHARED_PREF, MODE_PRIVATE);
    minutes = sharedPreferences.getInt(minutesPref, 2);
    language = sharedPreferences.getBoolean(languagePref, true);
    notFP = sharedPreferences.getBoolean(switchFPPref, true);
    notSNAP = sharedPreferences.getBoolean(switchSNAPPref, true);
}

private void save() {

    SharedPreferences sharedPreferences =
getSharedPreferences(SHARED_PREF, MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPreferences.edit();

    editor.putInt(minutesPref, minutes);
    editor.putBoolean(languagePref, language);
    editor.putBoolean(switchFPPref, notfp);
    editor.putBoolean(switchSNAPPref, notsnap);

    editor.apply();
}
```

Mediante el uso de un intent se permite pasar de una “activity” a otra información de determinadas variables. En este caso, información acerca de las notificaciones que envía el sistema, el idioma de la aplicación, los valores de los switches con los que se indica que tipo de notificaciones desea recibir el usuario, la localización del usuario...

```
Intent intent = new Intent(getApplicationContext(), MapPpal.class);
intent.putExtra("pantrieVsSnap", pantrieVsSnap);
intent.putExtra("allInfo", allInfo);
intent.putExtra("time", minutes);
intent.putExtra("firstTime", firstLoc);
intent.putExtra("foodPantryLocated", fbIs);
intent.putExtra("snapLocated", snapIs);
intent.putExtra("language", language);
intent.putExtra("notiFP", notfp);
intent.putExtra("notiSNAP", notsnap);
intent.putExtra("longitude", longitude);
intent.putExtra("latitude", latitude);
if (foodPantrySearch == null) {
    intent.putExtra("foodPantryCurrent", 0);
} else {
    intent.putExtra("foodPantryCurrent", foodPantrySearch.getIndex());
}

if (snapSearch == null) {
    intent.putExtra("snapCurrent", 0);
} else {
    intent.putExtra("snapCurrent", snapSearch.getIndex());
}
startActivity(intent);
overridePendingTransition(10, 10);
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.ECLAIR) {
    overridePendingTransition(0, 0);
}
```

NOTIFICACIÓN DEL SISTEMA

Para el uso de notificaciones generadas automáticamente por el sistema se necesita crear un canal para la notificación. Ese canal debe tener un ID que se utilizara para guiar la notificación hacia el mismo.

```
private void createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = "Notification Channel";
        String description = "Notification for near food banks";
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
    }
}
```

```
NotificationChannel channel = new
NotificationChannel("notifyF4U",name,importance);
channel.setDescription(description);
NotificationManager notManager =
getSystemService(NotificationManager.class);
notManager.createNotificationChannel(channel);
}
```

Para que la notificación se muestre al usuario se le debe asociar al ID del canal que se va a utilizar y se le debe dar prioridad máxima.

```
NotificationCompat.Builder builder = new
NotificationCompat.Builder(context,"notifyF4U")
.setSmallIcon(R.drawable.ic_baseline_notifications_24)
.setContentTitle("LOOKING FOR FOOD?")
.setContentText("YOU HAVE NEAR A FOOD PANTRY OR A SNAP RETAILER")
.setPriority(NotificationCompat.PRIORITY_MAX);
NotificationManagerCompat notMan = NotificationManagerCompat.from(context);
notMan.notify(200,builder.build());
```

ELIMINACIÓN DE UN USUARIO DE FIREBASE

Para eliminar un usuario de la base de datos, se debe identificar al usuario que se quiere dar de baja de la aplicación. Para ello se crea un objeto “FirebaseUser” que tiene la información acerca del usuario que ha iniciado sesión desde el dispositivo móvil. Ese objeto permite eliminar al usuario de la real-time database de FIREBASE así como de la plataforma de autenticación.

```
FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
user.delete().addOnCompleteListener(new OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {
if (task.isSuccessful()) {
FirebaseDatabase.getInstance().getReference("Users")
.child(user.getUid())
.removeValue();
Intent intent = new Intent(getApplicationContext(),
loginActivity.class);
startActivity(intent);
overridePendingTransition(10, 10);
}
```

COMPROBACIÓN DEL IDIOMA Y CAMBIO DE LOS CAMPOS

Tras haber obtenido el idioma de la aplicación a través del “intent”, se lleva a cabo la comprobación de si el idioma de la aplicación es español o inglés. En función del idioma se establece el texto en los diferentes elementos de la interfaz de la aplicación en un idioma u otro.

```
TextView mins = findViewById(R.id.infoTime);
MaterialButton butt = findViewById(R.id.buttChangeSettings);
MaterialButton buttP = findViewById(R.id.buttProfile);
EditText text = findViewById(R.id.Time);
TextView lang = findViewById(R.id.infoLanguage);
TextView not = findViewById(R.id.infoNotifications);
TextView fpant = findViewById(R.id.foodPantries);
TextView snapret = findViewById(R.id.snapRetailers);
TextView settings = findViewById(R.id.textView);

if (language) {
    dropDown.setText("ENGLISH");
    mins.setText("MINUTES BETWEEN NOTIFICATIONS");
    butt.setText("SAVE CHANGES");
    buttP.setText("DELETE USER");
    lang.setText("LANGUAGE");
    not.setText("NOTIFICATIONS");
    fpant.setText("Food Pantries");
    snapret.setText("Snap Retailers");
    settings.setText("SETTINGS");
    adapterItems = new ArrayAdapter<>(this, R.layout.dropdownitem, languages);
} else {
    dropDown.setText("ESPAÑOL");
    mins.setText("MINUTOS ENTRE NOTIFICACIONES");
    butt.setText("GUARDAR CAMBIOS");
    buttP.setText("ELIMINAR USUARIO");
    lang.setText("IDIOMA");
    not.setText("NOTIFICACIONES");
    fpant.setText("Bancos De Alimentos");
    settings.setText("AJUSTES");
    snapret.setText("Minoristas De SNAP");
    adapterItems = new ArrayAdapter<>(this, R.layout.dropdownitem, idiomas);
}
```

Capítulo 6. ANÁLISIS DE RESULTADOS

El resultado del proyecto ha sido una aplicación para móvil (ANDROID) totalmente funcional y sencilla de utilizar que cumple con todos los requisitos definidos por los promotores de la idea.

Esta aplicación para móvil se encuentra subida a la plataforma GOOGLE PLAY CONSOLE. Por medio de esta plataforma se puede llevar a cabo la distribución de la aplicación de tal manera que personas la pueden descargar desde el Google Play Store. Actualmente, solo determinadas personas previamente autorizadas pueden descargarse la aplicación, ya que se encuentra en proceso final de testing.

A continuación, se van a exponer algunos de los aspectos más destacados de la aplicación desarrollada.

6.1 VISTA DE LA APLICACIÓN

La validación de la interfaz gráfica de la aplicación de móvil y el correcto funcionamiento de la misma se llevó a cabo a través de la plataforma de ANDROID STUDIO y de un dispositivo móvil.

- ANDROID STUDIO. Esta plataforma permitió comprobar el correcto funcionamiento de la aplicación de móvil gracias a su herramienta de “run/debug”. A través de esta herramienta se llevó a cabo la compilación del proyecto y la instalación de la aplicación en el dispositivo móvil.
- Dispositivo móvil. El dispositivo se modificó para que tuviera opciones de desarrollador y fuese posible correr la aplicación en el mismo. Gracias a esto se pudo comprobar el correcto funcionamiento la aplicación de móvil.

Esto permitió, a lo largo del desarrollo de la aplicación de móvil, comprobar que las pantallas y los elementos de las mismas tenían el aspecto deseado, y comprobar el correcto funcionamiento de la aplicación de móvil. De esta forma, se pudieron corregir tanto los errores relacionados con el diseño de la interfaz, como los errores relacionados con las funcionalidades de la aplicación de móvil.

A continuación se presentan las pantallas de la aplicación.

- En la Figura 36 se puede observar la pantalla de la aplicación que muestra el mapa. En ella se pueden apreciar las funcionalidades anteriormente mencionadas.

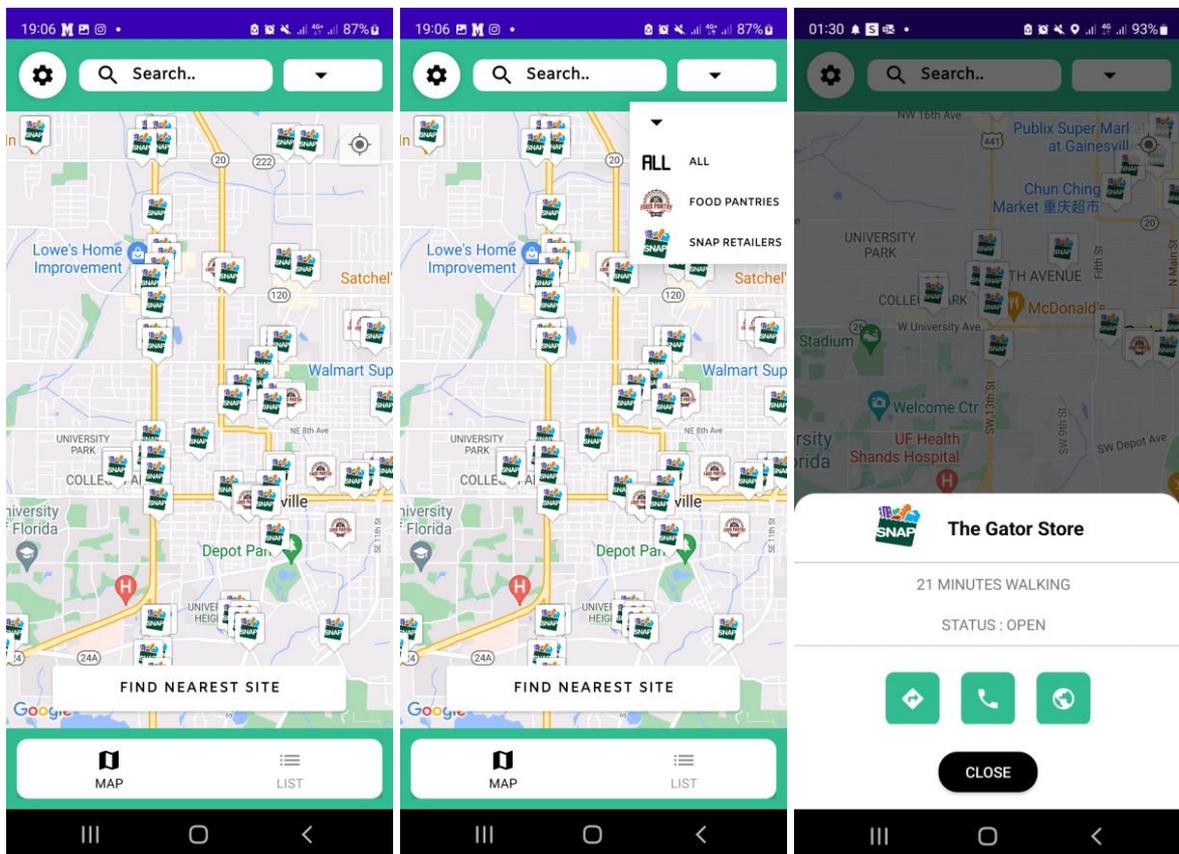


Figura 36. Pantalla del Mapa

- En la Figura 37 se puede observar la pantalla de la aplicación que muestra la lista. En ella se pueden apreciar las funcionalidades anteriormente mencionadas.

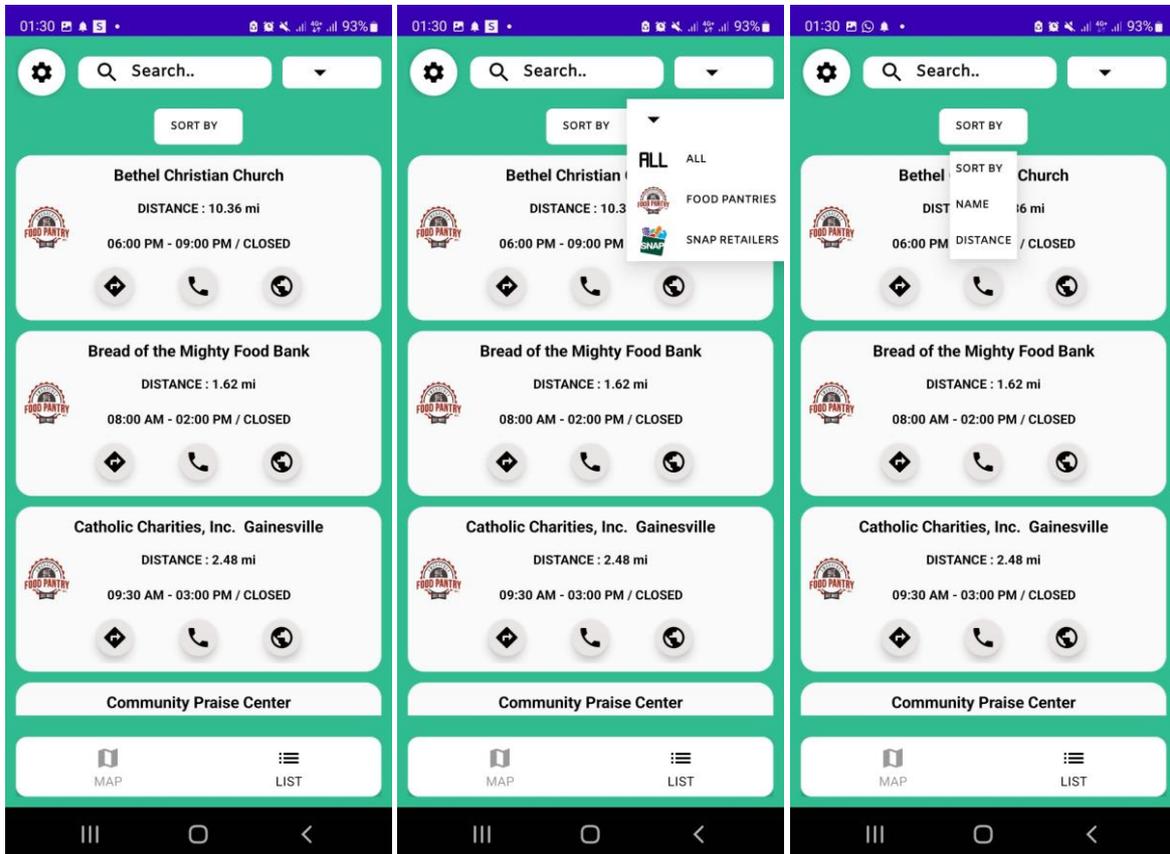


Figura 37. Pantalla de la Lista

- En la Figura 38 se puede observar la pantalla de la aplicación que muestra la configuración. En ella se pueden apreciar las funcionalidades anteriormente mencionadas.

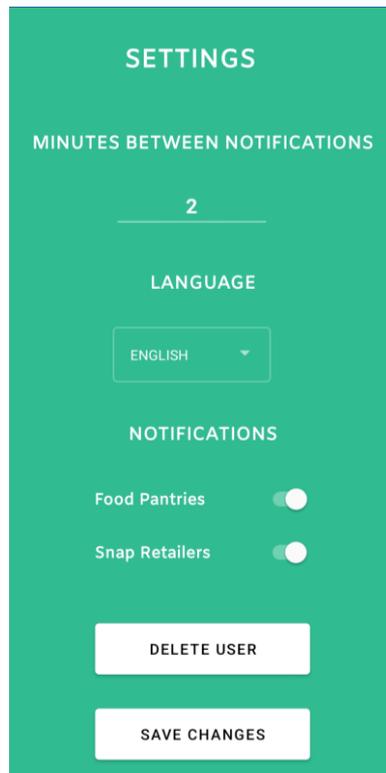


Figura 38. Pantalla de Configuración

- En la Figura 39 se pueden observar las pantallas de la aplicación que muestran el inicio de sesión, el registro de usuarios y la recuperación de contraseña. En ella se pueden apreciar las funcionalidades anteriormente mencionadas.

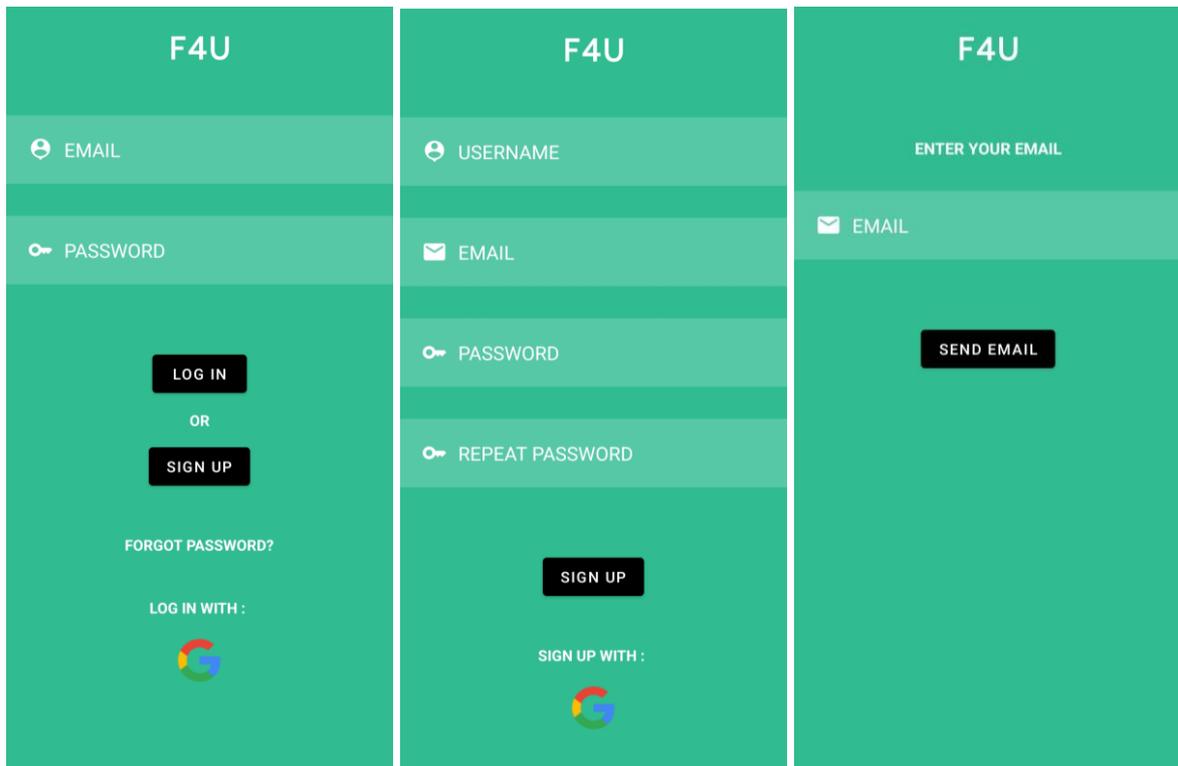


Figura 39. Pantallas de Inicio de Sesión, Registro y Recuperación de Contraseña

6.2 PERSISTENCIA

El uso de la plataforma FIREBASE ha sido esencial para poder llevar a cabo el proyecto. Gracias a esta, se han podido utilizar sus servicios de almacenamiento en la nube para poder guardar la información acerca de los usuarios que usan la aplicación de móvil.

Mediante el uso de la real-time database y la plataforma de autenticación de las que FIREBASE provee, se han podido desarrollar aquellas funcionalidades relacionadas con el almacenamiento y la autenticación de los usuarios que usan la aplicación de móvil.

Entre esas funcionalidades, de las que se ha comprobado su correcto funcionamiento, se encuentran:

- Registro de un usuario mediante su correo electrónico, su nombre de usuario y contraseña. Esta funcionalidad permite añadir un nuevo usuario tanto a la real-time database como a la plataforma de autenticación. Como se mencionó anteriormente, ese usuario no está autenticado hasta llevar a cabo la verificación de su correo electrónico.
- Autenticación de los usuarios. Mediante su correo electrónico, los usuarios son capaces de llevar a cabo este proceso de autenticación mediante la verificación de su email.
- Verificación de las credenciales de los usuarios. Esta funcionalidad permite comprobar que los usuarios se encuentran en la real-time database cuando intentan iniciar sesión. Si es así, se les permite el acceso a la aplicación de móvil.
- Eliminación de una cuenta. Los usuarios son capaces de eliminar su cuenta de la aplicación de móvil. De este modo, se elimina su información de la real-time database y de la plataforma de autenticación.

Cabe mencionar que FIREBASE tiene una herramienta de análisis muy completa que proporciona información muy interesante de la aplicación de móvil. Con ella se pueden generar los siguientes gráficos:

- Actividad de los usuarios.
- Usuarios en función de la versión de la aplicación.
- Actividad de los usuarios por categoría temporal.
- Número de visitas que se han realizado a cada una de las pantallas de la aplicación.
- Número de eventos por tipo de evento de la aplicación.
- Usuarios en función del modelo del dispositivo móvil.
- Usuarios en función del país.

En las Figuras 40 y 41 se pueden observar los gráficos anteriormente mencionados.

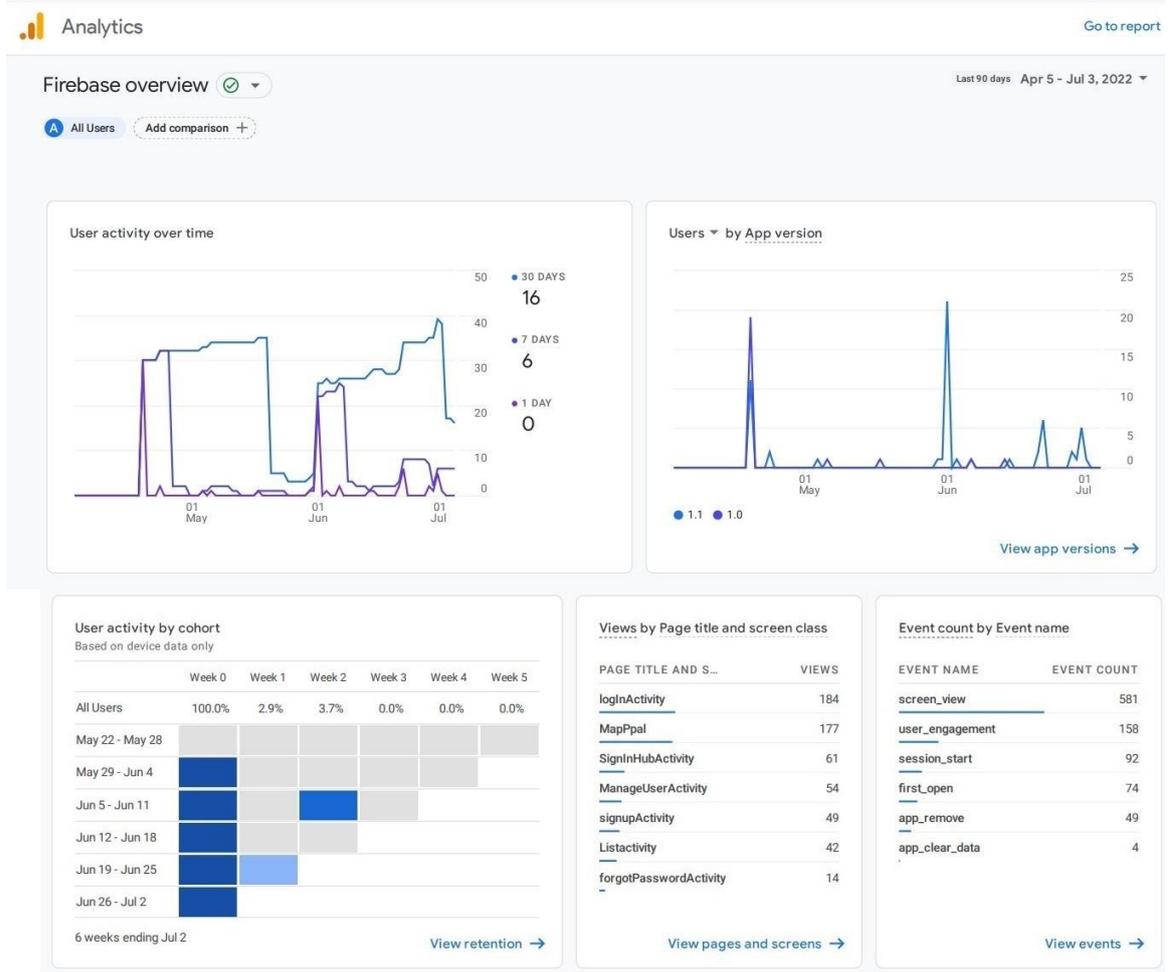


Figura 40. Plataforma de Análisis de FIREBASE 1

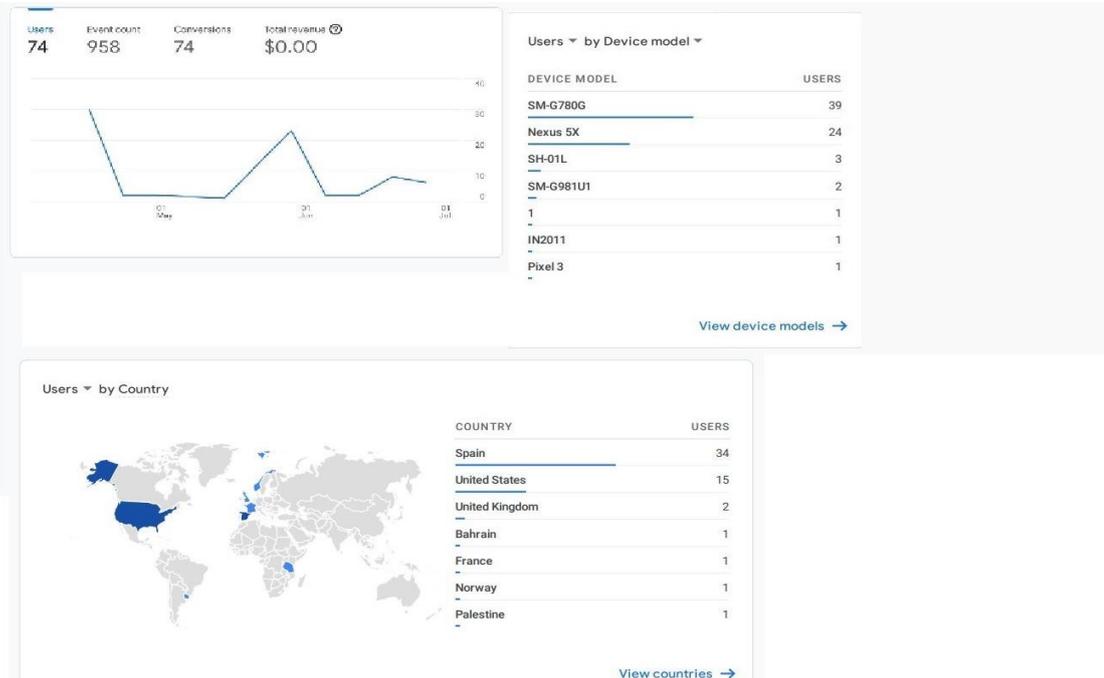


Figura 41. Plataforma de Análisis de FIREBASE 2

6.3 SOLICITUDES DE CARGA DE MAPAS

La plataforma de GOOGLE MAPS PLATFORM permite visualizar el tráfico por código de respuesta asociado a las solicitudes de carga de mapas. En él se pueden apreciar el número de solicitudes de carga de mapas que se realizaron desde los distintos dispositivos móviles que accedieron a la aplicación. En la Figura 42 se pueden observar las solicitudes de carga de mapas de los últimos 30 días.



Figura 42. Tráfico por Código de Respuesta

6.4 USUARIOS DE PRUEBA Y USUARIOS REALES

Como se mencionó anteriormente, mediante el proceso de “Friends & Family” un ámbito reducido de personas (usuarios de prueba), entre los que se encontraban tanto compañeros de intercambio, como compañeros de la universidad, como familiares, se descargaron la aplicación en un formato .APK en sus dispositivos móviles para comprobar su funcionamiento e identificar errores.

Después del proceso de testing de “Friends & Family” se inició el proceso de “User Testing” en el que se generó un archivo bundle de la aplicación y se subió a la plataforma GOOGLE PLAY CONSOLE. Por medio de esta plataforma se pudo llevar a cabo la distribución de la aplicación de tal manera que solo algunas personas (usuarios reales) mediante una autorización previa pueden descargarse la aplicación desde el Google Play Store. La aplicación todavía se encuentra en este proceso de testing.

Los resultados de esas descargas y el uso de la aplicación se pueden observar mediante las gráficas que proporciona la plataforma de análisis de FIREBASE anteriormente mencionada.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

7.1 CONCLUSIONES

Para poder sacar conclusiones, se deben comparar los objetivos principales definidos al principio del proyecto y su cumplimiento tras el desarrollo del mismo.

Los objetivos principales fijados para este proyecto eran los siguientes:

- Desarrollar una aplicación para móvil sencilla y “friendly” que los usuarios puedan usar.
- Implementar la SDK de Google Maps para poder situar un mapa en la aplicación.
- Permitir que la aplicación envíe notificaciones a los usuarios.
- Desarrollar una aplicación en los idiomas de español e inglés.
- Diseñar la base de datos para el almacenamiento de la información de los usuarios que accedan a la aplicación.
- Diseñar el sistema de autenticación y autorización de los usuarios que accedan a la aplicación.
- Llevar a cabo una fase final de testing de la aplicación con usuarios de prueba y usuarios reales.

Tras todo el trabajado realizado, se puede afirmar que se han completado la gran mayoría de los objetivos principales.

- Se ha conseguido desarrollar una aplicación de móvil que los usuarios pueden usar.
- Se ha utilizado la SDK de Google Maps y se ha situado un mapa en la aplicación.
- La aplicación es capaz de enviar notificaciones propias a los usuarios.
- La aplicación ofrece la posibilidad de cambiar el idioma de la misma entre español e inglés.

- Se ha desarrollado la funcionalidad necesaria para poder llevar a cabo el almacenamiento de usuarios en una base de datos así como su autenticación y autorización.
- Se ha llevado a cabo la fase de testing de “Friends & Family” con usuarios de prueba.

Es decir, se ha conseguido desarrollar una herramienta de ayuda para las personas con inseguridad alimentaria y problemas económicos, que les permite localizar y obtener información acerca de los establecimientos dispuestos en la aplicación para obtener alimentos y otros productos de primera necesidad de una manera rápida y sencilla

7.2 TRABAJOS FUTUROS

Como trabajos futuros se pueden destacar fundamentalmente dos:

- Finalizar la fase de testing de la aplicación con usuarios reales censados por la Universidad de Florida. Como se mencionó anteriormente, el proceso de “User Testing”, en el que se generó un archivo bundle de la aplicación y se subió a la plataforma GOOGLE PLAY CONSOLE para poder llevar a cabo la distribución de la aplicación, se inició pero no se finalizó completamente.

Las principales tareas restantes de esta fase de testing serían las siguientes:

- La primera tarea será la de organizar una reunión con la Doctorada en Ciencias de la Alimentación y Nutrición Humana, Karla Shelnut, y el grupo de personas del programa ampliado de educación alimentaria y nutricional de la Universidad de Florida. Los objetivos de esta reunión serían, el de definir una lista de personas (usuarios finales) a las que permitir la posibilidad de descargar la aplicación y el de definir la forma en la que estas personas puedan reportar los posibles errores que puedan detectar mediante el uso de la aplicación de móvil.

- Hacer el seguimiento del uso de la aplicación de estas personas y valorar su opinión sobre la misma, mediante la realización de una encuesta de usabilidad (SUS – System Usability Scale) compuesta por 10 preguntas. Esta encuesta permitirá evaluar la usabilidad de la aplicación de móvil desarrollada. [12]
- Tras esto, se organizará otra reunión con el grupo de trabajo previamente mencionado, para analizar los resultados y decidir si sacar la aplicación de móvil al Google Play Store o llevar a cabo algunos cambios en esta antes de ello.
- Crear una base de datos en FIREBASE para el almacenamiento de la información de los establecimientos. Otra tarea por realizar es la de llevar a cabo la incorporación de toda la información de los establecimientos a la plataforma de Cloud Storage que FIREBASE provee. El objetivo de esto es poder llevar a cabo un mantenimiento de esa información mucho más eficiente.
- Desarrollo de una plataforma de mantenimiento de los datos de los establecimientos (Alta, baja, modificación) para facilitar la actualización de los datos sin necesidad de intervención de personal técnico. Otra tarea por realizar es la de facilitar a determinadas personas (administradores) la posibilidad de llevar a cabo el mantenimiento de la base de datos que contiene la información de los establecimientos sin la necesidad de depender de personal técnico. Se busca otorgarles la posibilidad, de mediante una plataforma, ser capaces de añadir, eliminar y modificar información relativa a los establecimientos de la aplicación.

Estos tres grupos de tareas se van a realizar a partir del mes de septiembre conjuntamente con la Universidad de Florida.

Capítulo 8. BIBLIOGRAFÍA

- [1] IPES-FOOD, “El COVID-19 y la crisis en los sistemas alimentarios: Síntomas, causas y posibles soluciones”. <https://www.biodiversidadla.org/Recomendamos/El-COVID-19-y-la-crisis-en-los-sistemas-alimentarios-Si-ntomas-causas-y-posibles-soluciones>
- [2] Univisión Noticias, “Cómo ha aumentado la inseguridad alimentaria en EEUU: inflación y el final de los cheques de ayuda son algunas razones”. <https://www.univision.com/noticias/estados-unidos/inseguridad-alimentaria-aumenta-eeuu-inflacion-cheques-de-ayuda>
- [3] Alisha Coleman-Jensen y Laura Hales, U.S Department of Agriculture. <https://www.ers.usda.gov/topics/food-nutrition-assistance/food-security-in-the-u-s/interactive-charts-and-highlights/>
- [4] Bill Loeffler, “Cloud Computing: What is Infrastructure as a Service”. [https://docs.microsoft.com/en-us/previous-versions/technet-magazine/hh509051\(v=msdn.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/technet-magazine/hh509051(v=msdn.10)?redirectedfrom=MSDN)
- [5] Golap Gunjan Barman, “What is Firebase? All you need to know about Firebase”. <https://androidapps-development-blogs.medium.com/what-is-firebase-all-you-need-to-know-about-firebase-68f8a8a363d0>
- [6] Acens, “Bases de datos NoSQL. Qué son y tipos que nos podemos encontrar”. <https://www.acens.com/comunicacion/white-papers/bases-de-datos-nosql/>
- [7] Alisha Coleman-Jensen y Matthew P.Rabbitt, U.S Department of Agriculture. <https://www.ers.usda.gov/amber-waves/2021/november/food-pantry-use-increased-in-2020-for-most-types-of-u-s-households/#:~:text=Data%20from%20the%20Current%20Population,from%204.4%20percent%20in%202019>
- [8] Ed Bolen y Elisabeth Wolkomir, “SNAP Boosts Retailers and Local Economies”. <https://www.cbpp.org/research/food-assistance/snap-boosts-retailers-and-local-economies>

- [9] Google Maps Platform Website. <https://mapsplatform.google.com/pricing/>
- [10] Firebase Website. <https://firebase.google.com/pricing>
- [11] United States Census Bureau.
<https://www.census.gov/quickfacts/fact/table/gainesvillecityflorida/PST045221>
- [12] Nathan Thomas, “How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website”. <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

En el año 2015, los líderes mundiales, a través de la ONU, aprobaron una nueva agenda sobre el Desarrollo Sostenible, con el objetivo de erradicar la pobreza, proteger al planeta y asegurar la prosperidad de este. En ella se definieron un conjunto de 17 objetivos, con unas metas específicas que deben alcanzarse en el año 2030. Esos 17 objetivos se pueden apreciar en la Figura 43.



Figura 43. 17 Objetivos de Desarrollo Sostenible

De esos 17 objetivos de desarrollo sostenible, este proyecto está ligado principalmente a:

- Objetivo de dimensión social 1. Fin de la Pobreza.
- Objetivo de dimensión social 2. Hambre Cero.
- Objetivo de dimensión social 3. Salud y Bienestar.
- Objetivo de dimensión económica 12. Producción y Consumo Responsables.

Con respecto a los objetivos de dimensión social, la ONU define los objetivos de la siguiente manera:

- Objetivo 1. Poner fin a la pobreza en todas sus formas en todo el mundo.
- Objetivo 2. Poner fin al hambre.
- Objetivo 3. Garantizar una vida sana y promover el bienestar para todos en todas las edades.

El proyecto que se ha desarrollado está ligado principalmente con el objetivo 2 ya que se busca como objetivo principal poner fin al hambre, tratando el tema de la inseguridad alimentaria. De esta manera, se busca ayudar a esas personas con pocos recursos económicos a gastar lo mínimo posible o incluso nada, y poder obtener alimentos de primera necesidad. Las consecuencias de ayudar a esas personas con problemas económicos e inseguridad alimentaria son las que están ligadas con los otros dos objetivos, ya que al ayudar a las personas a gastar lo mínimo posible o incluso nada, se está ayudando a poner fin a la pobreza, y al ayudar a esas personas a conseguir alimentos de primera necesidad, se está ayudando a garantizar una vida sana y a promover el bienestar de todas las personas, ya que con esos alimentos van a poder ser capaces de llevar una vida y una dieta equilibrada.

Con respecto al objetivo de dimensión económica, la ONU define el objetivo 12 como Garantizar modalidades de consumo y producción sostenibles.

Con este proyecto se podrían buscar vías de difusión de tal manera que no se quedase solo en una aplicación que ayuda a personas a conseguir alimentos, sino que también sea un medio difusor y sirva como ejemplo para esas personas que desperdician alimentos o esas compañías que tiran alimentos en buen estado por haber tenido una sobre producción.

ANEXO II: CÓDIGO DE BÚSQUEDA DEL ESTABLECIMIENTO MÁS CERCANO

```
private void findNearest(int i) {

    int avatar1 = R.drawable.clipart298892;
    int avatar2 = R.drawable.fpantries_2;

    if(i==1){

        try{

            FoodPantryDistance fpMinDist = findNearestFoodPantry();
            FoodPantryLocation fpLocMin = new FoodPantryLocation();
            Iterator it = mFoodPantryLocation.iterator();
            while(it.hasNext()){
                Object o = it.next();
                if(o instanceof FoodPantryLocation){
                    FoodPantryLocation fp = (FoodPantryLocation) o;
                    if(fp.getFoodPantry().equalsTo(fpMinDist.getFoodPantry())
){
                        fpLocMin = fp;
                    }
                }
            }

            ClusterMarker newCluster = new ClusterMarker(new LatLng(
                fpLocMin.getGeo_point().getLatitude(),
                fpLocMin.getGeo_point().getLongitude()),
                fpLocMin.getFoodPantry().getFoddBankName(), fpMinDist.getDistance(), avatar2, fpLocMin.getFoodPantry());
            bottomSheetDialog(newCluster);

        }catch (Exception e){
            AlertDialog alrt;
            if(language){
                alrt = new AlertDialog.Builder(MapPpal.this)
                    .setTitle("ENABLE LOCATION")
                    .setMessage("PLEASE ENABLE LOCATION")
                    .setNegativeButton("OK", new
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int
which) {
                            dialog.dismiss();
                        }
                    })
            }
        }
    }
}
```

```

        .create();
    }else{
        alrt = new AlertDialog.Builder(MapPpal.this)
            .setTitle("HABILITA LA LOCALIZACION")
            .setMessage("PORFAVOR HABILITA LA LOCALIZACION")
            .setNegativeButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int
which) {
                    dialog.dismiss();
                }
            })
            .create();
    }
    alrt.show();
}

}else if(i==2){

    try{

        SNAPDistance spMinDist = findNearestSNAP();
        SNAPLocation spLocMin = new SNAPLocation();
        Iterator it = mSnapLocation.iterator();
        while(it.hasNext()){
            Object o = it.next();
            if(o instanceof SNAPLocation){
                SNAPLocation sp = (SNAPLocation) o;
                if(sp.getSNAP().equalsTo(spMinDist.getSnap())){
                    spLocMin = sp;
                }
            }
        }

        ClusterMarker newCluster = new ClusterMarker(new LatLng(
            spLocMin.getGeo_point().getLatitude(),
            spLocMin.getGeo_point().getLongitude()),
            spLocMin.getSNAP().getSNAPname(), spMinDist.getDistance(),
            avatar1, spLocMin.getSNAP());
        bottomSheetDialog(newCluster);

    }catch (Exception e){
        AlertDialog alrt;
        if(language){
            alrt = new AlertDialog.Builder(MapPpal.this)
                .setTitle("ENABLE LOCATION")
                .setMessage("PLEASE ENABLE LOCATION")
                .setNegativeButton("OK", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
which) {

```

```

        dialog.dismiss();
    }
    })
    .create();
}
}else{
    alrt = new AlertDialog.Builder(MapPpal.this)
        .setTitle("HABILITA LA LOCALIZACION")
        .setMessage("PORFAVOR HABILITA LA LOCALIZACION")
        .setNegativeButton("OK", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int
which) {
        dialog.dismiss();
    }
    })
    .create();
}
alrt.show();
}
}
}else if(i==3){
    try{
        SNAPDistance spMinDist = findNearestSNAP();
        SNAPLocation spLocMin = new SNAPLocation();
        Iterator it = mSnapLocation.iterator();
        while(it.hasNext()){
            Object o = it.next();
            if(o instanceof SNAPLocation){
                SNAPLocation sp = (SNAPLocation) o;
                if(sp.getSNAP().equalsTo(spMinDist.getSnap())){
                    spLocMin = sp;
                }
            }
        }
        FoodPantryDistance fpMinDist = findNearestFoodPantry();
        FoodPantryLocation fpLocMin = new FoodPantryLocation();
        Iterator it2 = mFoodPantryLocation.iterator();
        while(it2.hasNext()){
            Object o = it2.next();
            if(o instanceof FoodPantryLocation){
                FoodPantryLocation fp = (FoodPantryLocation) o;
                if(fp.getFoodPantry().equalsTo(fpMinDist.getFoodPantr
y())){
                    fpLocMin = fp;
                }
            }
        }
        ClusterMarker newCluster;
        if(spMinDist.getDistance() < fpMinDist.getDistance()) {
            newCluster = new ClusterMarker(new LatLng(

```

```

        spLocMin.getGeo_point().getLatitude(),
spLocMin.getGeo_point().getLongitude()),
        spLocMin.getSNAP().getSNAPname(),
spMinDist.getDistance(), avatar1, spLocMin.getSNAP());
    }else{
        newCluster = new ClusterMarker(new LatLng(
            fpLocMin.getGeo_point().getLatitude(),
fpLocMin.getGeo_point().getLongitude()),
            fpLocMin.getFoodPantry().getFoddBankName(),
fpMinDist.getDistance(), avatar2, fpLocMin.getFoodPantry());
    }
    bottomSheetDialog(newCluster);

}catch (Exception e){
    AlertDialog alrt;
    if(language){
        alrt = new AlertDialog.Builder(MapPpal.this)
            .setTitle("ENABLE LOCATION")
            .setMessage("PLEASE ENABLE LOCATION")
            .setNegativeButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog,
int which) {
                    dialog.dismiss();
                }
            })
            .create();
    }else{
        alrt = new AlertDialog.Builder(MapPpal.this)
            .setTitle("HABILITA LA LOCALIZACION")
            .setMessage("PORFAVOR HABILITA LA LOCALIZACION")
            .setNegativeButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog,
int which) {
                    dialog.dismiss();
                }
            })
            .create();
    }
    alrt.show();
}
}
}

Object nearest = null;
double distanceMin = 0.0;
double distance;

if (sitesWithTheSequence.size() > 0) {
    for(Object o: sitesWithTheSequence){

        if(o instanceof SNAPLocation){

```

```

        SNAPLocation snapLoc = (SNAPLocation) o;
        if(distanceMin == 0){
            try{
                distance =
distFrom(snapLoc.getGeo_point().getLatitude(),snapLoc.getGeo_point().getLongitude
(),current.getLatitude(),current.getLongitude());
                distanceMin = distance;
                nearest = snapLoc.getSNAP();
            }catch(Exception e){
                Log.d(TAG, "findNearestSNAP: error =
"+e.getMessage());
            }
        }else{
            distance =
distFrom(snapLoc.getGeo_point().getLatitude(),snapLoc.getGeo_point().getLongitude
(),current.getLatitude(),current.getLongitude());
            if(distance<distanceMin){
                distanceMin = distance;
                nearest = snapLoc.getSNAP();
            }
        }
    }else{
        FoodPantryLocation fpLoc = (FoodPantryLocation) o;
        if(distanceMin == 0){
            try{
                distance =
distFrom(fpLoc.getGeo_point().getLatitude(),fpLoc.getGeo_point().getLongitude(),c
urrent.getLatitude(),current.getLongitude());
                distanceMin = distance;
                nearest = fpLoc.getFoodPantry();
            }catch(Exception e){
                Log.d(TAG, "findNearestSNAP: error =
"+e.getMessage());
            }
        }else{
            distance =
distFrom(fpLoc.getGeo_point().getLatitude(),fpLoc.getGeo_point().getLongitude(),c
urrent.getLatitude(),current.getLongitude());
            if(distance<distanceMin){
                distanceMin = distance;
                nearest = fpLoc.getFoodPantry();
            }
        }
    }
}

distanceMin = ((double)((int)(distanceMin *100.0)))/100.0;

try{
    if(nearest instanceof SNAP){
        SNAP spMinDist = (SNAP) nearest;
        SNAPLocation spLocMin = new SNAPLocation();
    }
}

```

```

        Iterator it = mSnapLocation.iterator();
        while(it.hasNext()){
            Object o = it.next();
            if(o instanceof SNAPLocation){
                SNAPLocation sp = (SNAPLocation) o;
                if(sp.getSNAP().equalsTo(spMinDist)){
                    spLocMin = sp;
                }
            }
        }

        ClusterMarker newCluster = new ClusterMarker(new LatLng(
            spLocMin.getGeo_point().getLatitude(),
            spLocMin.getGeo_point().getLongitude()),
            spLocMin.getSNAP().getSNAPname(), distanceMin,
            avatar1, spLocMin.getSNAP());
        bottomSheetDialog(newCluster);

    }else{
        FoodPantry fpMinDist = (FoodPantry) nearest;
        FoodPantryLocation fpLocMin = new FoodPantryLocation();
        Iterator it2 = mFoodPantryLocation.iterator();
        while(it2.hasNext()){
            Object o = it2.next();
            if(o instanceof FoodPantryLocation){
                FoodPantryLocation fp = (FoodPantryLocation) o;
                if(fp.getFoodPantry().equalsTo(fpMinDist)){
                    fpLocMin = fp;
                }
            }
        }
        ClusterMarker newCluster = new ClusterMarker(new LatLng(
            fpLocMin.getGeo_point().getLatitude(),
            fpLocMin.getGeo_point().getLongitude()),
            fpLocMin.getFoodPantry().getFoddBankName(), distan
            ceMin, avatar2, fpLocMin.getFoodPantry());
        bottomSheetDialog(newCluster);

    }
}catch (Exception e){
    AlertDialog alrt;
    if(language){
        alrt = new AlertDialog.Builder(MapPpal.this)
            .setTitle("ENABLE LOCATION")
            .setMessage("PLEASE ENABLE LOCATION")
            .setNegativeButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog,
int which) {
                    dialog.dismiss();
                }
            })
        .create();
    }
}

```


ANEXO III: CÓDIGO DE ORDENACIÓN POR DISTANCIA Y ORDEN ALFABÉTICO

```
private void sortByDist() {

    double longFb = 0;
    double latFb = 0;
    double longSp = 0;
    double latSP = 0;
    double distance = 0;

    double longDevice = loc.getLongitude();
    double latDevice = loc.getLatitude();

    Iterator it = mFoodPantryLocation.iterator();
    while(it.hasNext()){
        Object o = it.next();
        if(o instanceof FoodPantryLocation){
            FoodPantryLocation fp = (FoodPantryLocation) o;
            longFb = fp.getGeo_point().getLongitude();
            latFb = fp.getGeo_point().getLatitude();
            distance = distFrom(latDevice, longDevice, latFb, longFb);
            distance = ((double) ((int) (distance * 100.0)))/100.0;
            foodPantriesDistance.add(new FoodPantryDistanceLoc(fp, distance));
        }
    }

    Iterator it2 = mSnapLocation.iterator();
    while(it2.hasNext()){
        Object o = it2.next();
        if(o instanceof SNAPLocation){
            SNAPLocation sp = (SNAPLocation) o;
            longSp = sp.getGeo_point().getLongitude();
            latSP = sp.getGeo_point().getLatitude();
            distance = distFrom(latDevice, longDevice, latSP, longSp);
            distance = ((double) ((int) (distance * 100.0)))/100.0;
            snapsDistance.add(new SNAPDistanceLoc(sp, distance));
        }
    }

    FoodPantryDistanceLoc fp;
    SNAPDistanceLoc sp;

    for(int i = 0; i < snapsDistance.size(); i++){
        for(int j = i+1; j < snapsDistance.size(); j++){
            if(snapsDistance.get(i).getDistance() >
snapsDistance.get(j).getDistance()){
```

```

        sp=snapsDistance.get(i);
        snapsDistance.set(i, snapsDistance.get(j));
        snapsDistance.set(j,sp);
    }
}

for(int k = 0;k<foodPantriesDistance.size();k++){
    for(int l = k+1;l<foodPantriesDistance.size();l++){
        if(foodPantriesDistance.get(k).getDistance() >
foodPantriesDistance.get(l).getDistance()){
            fp=foodPantriesDistance.get(k);
            foodPantriesDistance.set(k, foodPantriesDistance.get(l));
            foodPantriesDistance.set(l,fp);
        }
    }
}

if(allInfoArr.size()>0){

    Iterator it3 = allInfoArr.iterator();
    while(it3.hasNext()){
        Object o = it3.next();
        if(o instanceof FoodPantryLocation){
            FoodPantryLocation fp2 = (FoodPantryLocation) o;
            longFb = fp2.getGeo_point().getLongitude();
            latFb = fp2.getGeo_point().getLatitude();
            distance = distFrom(latDevice,longDevice,latFb,longFb);
            distance = ((double)((int)(distance *100.0)))/100.0;
            allInfoArrDistance.add(new FoodPantryDistanceLoc(fp2,distance));
        }else{
            SNAPLocation sp2 = (SNAPLocation) o;
            longFb = sp2.getGeo_point().getLongitude();
            latFb = sp2.getGeo_point().getLatitude();
            distance = distFrom(latDevice,longDevice,latFb,longFb);
            distance = ((double)((int)(distance *100.0)))/100.0;
            allInfoArrDistance.add(new SNAPDistanceLoc(sp2,distance));
        }
    }

    Place p;

    for(int i = 0;i<allInfoArrDistance.size();i++){
        for(int j = i+1;j<allInfoArrDistance.size();j++) {

            if(allInfoArrDistance.get(i).getDistance()>allInfoArrDistance.get
(j).getDistance()){
                p = allInfoArrDistance.get(i);
                allInfoArrDistance.set(i,allInfoArrDistance.get(j));
                allInfoArrDistance.set(j,p);
            }
        }
    }
}

```

```
private void sortByNameAll() {

    FoodPantryLocation fp;
    SNAPLocation sp;

    for(int i = 0;i<allInfoArr.size();i++){
        for(int j = i+1;j<allInfoArr.size();j++){
            if(allInfoArr.get(i) instanceof FoodPantryLocation){
                FoodPantryLocation fploci = (FoodPantryLocation)
allInfoArr.get(i);
                if(allInfoArr.get(j) instanceof FoodPantryLocation){
                    FoodPantryLocation fplocj = (FoodPantryLocation)
allInfoArr.get(j);
                    if(fploci.getFoodPantry().getFoddBankName().compareTo(fplocj.
getFoodPantry().getFoddBankName())>0){
                        fp=fploci;
                        allInfoArr.set(i,fplocj);
                        allInfoArr.set(j,fp);
                    }
                }else{
                    SNAPLocation splocj = (SNAPLocation) allInfoArr.get(j);
                    if(fploci.getFoodPantry().getFoddBankName().compareTo(splocj.
getSNAP().getSNAPname())>0){
                        fp=fploci;
                        allInfoArr.set(i, splocj);
                        allInfoArr.set(j,fp);
                    }
                }
            }else{
                SNAPLocation sploci = (SNAPLocation) allInfoArr.get(i);
                if(allInfoArr.get(j) instanceof FoodPantryLocation){
                    FoodPantryLocation fplocj = (FoodPantryLocation)
allInfoArr.get(j);
                    if(sploci.getSNAP().getSNAPname().compareTo(fplocj.getFoodPan
try().getFoddBankName())>0){
                        sp=sploci;
                        allInfoArr.set(i, fplocj);
                        allInfoArr.set(j,sp);
                    }
                }else{
                    SNAPLocation splocj = (SNAPLocation) allInfoArr.get(j);
                    if(sploci.getSNAP().getSNAPname().compareTo(splocj.getSNAP().
getSNAPname())>0){
                        sp=sploci;
                        allInfoArr.set(i, splocj);
                        allInfoArr.set(j,sp);
                    }
                }
            }
        }
    }
}
```