



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona



TRABAJO DE FIN DE MÁSTER

Máster Universitario en Ingeniería Informática

Aplicación Móvil para Detección de Potenciales Alérgenos en Recetas y Textos

Autor:

Ing. Pamela Castro Guerrero

Tutor:

Dr. Pere Pau Vázquez Alcocer

Barcelona, 30 de abril del 2020



Dedicatoria

A Dios por ser mi guía y el pilar de mi vida. A mis amados padres y hermanos por haberme acompañado durante este largo camino siempre brindándome su soporte y aliento en momentos de dificultad. A mi novio Erick por su paciencia, amor y ánimo constante. A mis amigos que me han apoyado en mi formación profesional y personal. A mis profesores que han sido una parte fundamental en este proceso de aprendizaje. A todos ellos mi más grande cariño y dedicatoria de este trabajo.



“La educación es el arma más poderosa que puedes usar para cambiar el mundo”.

Nelson Mandela

Resumen

Hoy en día, vivimos en un mundo en donde el uso de los *smartphones* impera cada vez más, sin distinguir edad ni ubicación geográfica, debido a las facilidades y servicios que ellos proporcionan en la vida diaria de las personas. Android es el Sistema Operativo para dispositivos móviles que ocupa la mayor cuota del mercado, su innovación constante crea mayores oportunidades para los desarrolladores de aplicaciones, en donde es fundamental proporcionar un servicio eficiente que satisfaga una necesidad.

Las enfermedades alérgicas afectan a millones de personas y en los últimos años es cada vez más común escuchar de personas con alergias a un grupo de alimentos. Para un alérgico, el momento de preparar alimentos dentro de casa, salir a comer a un restaurante, comprar productos en un supermercado, puede volverse un gran suplicio, ya que el comprobar la presencia de cada uno de los alérgenos que afectan a su salud puede ser bastante molesto y difícil, sobre todo si se encuentra en un lugar cuyo idioma sea desconocido.

En el ámbito sanitario, la tecnología es un gran aliado, por ello el presente proyecto pretende cubrir un nicho de mercado que aún no ha sido explotado, al brindar al usuario la facilidad de buscar recetas de acuerdo a sus necesidades, detectar los alérgenos que causan afecciones por medio de escaneos, detección de texto y traducción de imágenes en tiempo real o fotografías, además del análisis, basado en palabras clave o nombre del plato, del contenido de un conjunto de recetas, de manera que sirva como una herramienta de soporte, previo al consumo de alimentos que puedan afectar la salud de la persona.

En definitiva, esta aplicación pretende ser un instrumento que ayude tanto en la preparación de comida en casa, como en el consumo de productos fuera de ella, con la finalidad de servir como prevención de la ingesta de alimentos que puedan producir reacciones alérgicas y con ello mejorar la calidad de vida de aquellas personas que sufren de estas afecciones.

Abstract

Today, in a world where smartphones use is reigning more and more without make out age or location because they provide services and facilities for daily life.

Android is the Operative System for smartphones with greater amount of users around the world. The permanent innovation creates more opportunities for mobile developers where is fundamental to offer an efficient service for satisfying all needs.

Allergic diseases affect million people and the last years is more common find people suffering allergies for a specific food group. For allergic people, many moments are complicated such as cooking, go to a restaurant or buy products in a supermarket, because verify if one of the allergens which are bad for their health are present is and annoying and difficult task, especially when people travel to places where another language is spoken.

Technology is a great ally for health field, for this reason the present project aims to exploit a new market niche, giving to final users the possibility of looking recipes according their personal needs, detecting allergens through scanning methods, detecting text and translating images in real time. Additionally, the app can analyses the content of each recipe based on keywords or food name, becoming a prevention tool before people consume harmful nourishment for their health.

Finally, this app wants to become a helping instrument when people eat inside and outside home for improving quality life of people who suffers this condition.

Tabla de Contenidos

| | |
|---|----|
| 1. Introducción | 12 |
| 2. Descripción | 13 |
| 3. Motivación | 13 |
| 4. Objetivos | 14 |
| 4.1 Objetivo Principal | 14 |
| 4.2 Objetivos Específicos | 14 |
| 5. Tecnologías y Herramientas de Desarrollo | 15 |
| 5.1 Android Studio | 15 |
| 5.2 RapidAPI | 16 |
| 5.3 Firebase | 17 |
| 5.4 Google Cloud | 22 |
| 5.4.1 Google Cloud Vision API | 23 |
| 5.4.2 Google Cloud Translate API | 24 |
| 6. Arquitectura de la aplicación | 24 |
| 7. Implementación y Desarrollo | 26 |
| 7.1 Reconocimiento de Texto con Google Cloud | 27 |
| 7.2 Configuración de Android Studio con Firebase | 29 |
| 7.2.1 Visualización de la configuración del proyecto FireBase | 31 |
| 7.3 Almacenamiento de información en la Base de Datos de Firebase | 33 |
| 7.4 Búsqueda de información de la Base de Datos de Firebase | 37 |
| 7.5 Reconocimiento de Texto con Firebase | 37 |
| 7.6 Uso de RapidAPI para consulta de recetas | 41 |
| 7.7 Traducción de textos | 46 |
| 8. Interfaz de Usuario | 53 |
| 8.1 Árbol de Interfaces | 54 |
| 8.2 Descripción de Interfaces | 55 |
| 8.2.1 Splash | 55 |
| 8.2.2 Login | 56 |
| 8.2.3 User Register | 57 |
| 8.2.4 Dashbloard | 60 |
| 8.2.5 User Profile | 61 |



| | | |
|--------|---|-----|
| 8.2.6 | Recipes Search | 63 |
| 8.2.7 | Recipes..... | 65 |
| 8.2.8 | Recipes Analysis | 67 |
| 8.2.9 | Scan..... | 69 |
| 8.2.10 | Upload | 72 |
| 9. | Casos de Uso | 75 |
| 9.1 | Búsqueda de Recetas (Caso de uso 1)..... | 76 |
| 9.2 | Análisis del contenido de un plato (Caso de uso 2)..... | 81 |
| 9.3 | Análisis de un texto (Caso de uso 3)..... | 86 |
| 10. | Análisis de la competencia | 92 |
| 11. | Extensiones..... | 94 |
| 12. | Explotación Comercial | 95 |
| 13. | Conclusiones | 96 |
| 14. | Referencias | 97 |
| 15. | Anexos..... | 99 |
| | Anexo 1. Reconocimiento de texto en tiempo real | 99 |
| | Anexo 2. Conexión a Firebase para almacenar datos..... | 99 |
| | Anexo 3. Configuración para el almacenamiento de datos | 100 |
| | Anexo 4. Almacenamiento de datos del usuario | 100 |
| | Anexo 5. Configuración para reconocimiento de texto con Firebase..... | 101 |
| | Anexo 6. Reconocimiento de texto con Firebase..... | 101 |
| | Anexo 7. Consulta de Recetas (Rapid API)..... | 102 |
| | Anexo 8. Consulta de Recetas (Android Studio)..... | 102 |
| | Anexo 9. Flujo para la Consulta de Recetas | 102 |
| | Anexo 10. Traducción de textos (Google Cloud) | 103 |
| | Anexo 11. Traducción de textos (Android Studio) | 103 |

Tabla de Ilustraciones

| | |
|---|----|
| Ilustración 1. Interfaz de desarrollo de Android Studio | 15 |
| Ilustración 2. Interfaz de RapidAPI | 16 |
| Ilustración 3. Interfaz de Firebase Database | 17 |
| Ilustración 4. Base de Datos en Tiempo Real de Firebase | 18 |
| Ilustración 5. Alojamiento Web de Firebase | 18 |
| Ilustración 6. Almacenamiento con Firebase | 19 |
| Ilustración 7. Autenticación con Firebase | 19 |
| Ilustración 8. Monitoreo de Rendimiento con Firebase | 20 |
| Ilustración 9. Laboratorio de Test con Firebase..... | 20 |
| Ilustración 10. Machile Learning Kit con Firebase | 21 |
| Ilustración 11. Consola de Google Cloud | 22 |
| Ilustración 12. Consola de Google Cloud Vision API | 23 |
| Ilustración 13. Consola de Google Cloud Translate API | 24 |
| Ilustración 14. Arquitectura de la Aplicación | 25 |
| Ilustración 15. Opciones escogidas en SDK Plataforms | 26 |
| Ilustración 16. Opciones escogidas en SDK Tools | 27 |
| Ilustración 17. Dependencia para google play services vision | 27 |
| Ilustración 18. Permisos y metadatos para usar OCR de Google Cloud..... | 28 |
| Ilustración 19. Componente SurfaceView usado para escanear..... | 28 |
| Ilustración 20. Declaración del objeto CameraSource | 28 |
| Ilustración 21. Declaración del objeto TextRecognizer | 28 |
| Ilustración 22. Relación de TextRecognizer con CameraSource | 28 |
| Ilustración 23. Obtener el texto reconocido con setProcessor | 29 |
| Ilustración 24. Conectar Android Studio con Google | 29 |
| Ilustración 25. Relacionar Android Studio con Realtime Database | 30 |
| Ilustración 26. Crear nuevo proyecto Firebase desde Android Studio | 30 |
| Ilustración 27. Consola de Firebase | 31 |
| Ilustración 28. Ubicación del archivo google-services.json | 32 |
| Ilustración 29. Nombre del paquete y huellas SHA en la consola..... | 32 |
| Ilustración 30. SHA1 del proyecto en Android Studio | 33 |
| Ilustración 31. Crear Base de Datos en Firebase | 34 |
| Ilustración 32. Reglas de Seguridad con Firebase | 34 |
| Ilustración 33. Implementaciones a agregar para uso de Firebase Database..... | 35 |
| Ilustración 34. Clase usada para almacenar información en Firebase..... | 35 |
| Ilustración 35. Asignación de valores a un objeto antes de almacenarlo | 36 |
| Ilustración 36. Almacenamiento de alérgenos en un Map | 36 |
| Ilustración 37. Almacenamiento de un usuario en Firebase | 36 |
| Ilustración 38. Estructura de usuarios en la Base de Datos de Firebase | 36 |
| Ilustración 39. Referencia a la Base de Datos en Firebase | 37 |
| Ilustración 40. Consultar a Base de Datos en Firebase | 37 |



| | |
|---|----|
| Ilustración 41. Dependencia para uso de OCR con Firebase | 38 |
| Ilustración 42. Metadatos para uso de OCR con Firebase | 38 |
| Ilustración 43. Conexión de Android Studio con Google..... | 38 |
| Ilustración 44. Usar Machile Learning Kit desde Android Studio | 39 |
| Ilustración 45. Dependencia para usar Firebase Vision..... | 39 |
| Ilustración 46. Código para abrir la galería fotográfica | 39 |
| Ilustración 47. Dependencias para usar Glide | 40 |
| Ilustración 48. Carga de imágenes desde galeria usando Glide | 40 |
| Ilustración 49. Carga de imagen a un ImageView | 40 |
| Ilustración 50. Relación de Firebase Vision con una imagen cargada..... | 40 |
| Ilustración 51. Declaración del objeto FirebaseVisionTextRecognizer..... | 41 |
| Ilustración 52. Obtención de texto reconocido con Firebase | 41 |
| Ilustración 53. Suscripción a API de Spoonacular para extracción de recetas..... | 42 |
| Ilustración 54. Consulta de métodos de prueba con RapidAPI..... | 42 |
| Ilustración 55. Estructura JSON de recetas con Spoonacular de RapidAPI | 43 |
| Ilustración 56. Consulta de código de peticiones en distintos lenguajes usando RapidAPI . | 43 |
| Ilustración 57. Código para consulta de recetas usando OkHttp | 44 |
| Ilustración 58. Dependencia para uso de OkHttp | 44 |
| Ilustración 59. Estructura de la URL para consulta de recetas | 44 |
| Ilustración 60. Código del Request con OkHttp para obtener recetas | 45 |
| Ilustración 61. Obtener resultado de la petición de recetas | 45 |
| Ilustración 62. Descomposición del objeto JSON de recetas obtenidas | 45 |
| Ilustración 63. Extracción de cada receta del objeto JSON | 46 |
| Ilustración 64. Selección de un proyecto en la Plataforma de Google Cloud..... | 47 |
| Ilustración 65. Habilitación de APIs y servicios de Google Cloud | 47 |
| Ilustración 66. Búsqueda de la API de traducción en Google Cloud..... | 47 |
| Ilustración 67. Habilitar API de traducción..... | 48 |
| Ilustración 68. Habilitar la facturación de la API de traducción | 48 |
| Ilustración 69. Crear Cuenta de Facturación | 48 |
| Ilustración 70. Aceptar términos y condiciones de Google Cloud | 49 |
| Ilustración 71. Administrar la API de traducción de Google Cloud..... | 49 |
| Ilustración 72. Administrar las Credenciales de Google Cloud | 50 |
| Ilustración 73. Despliegue de opciones adicionales de credenciales..... | 50 |
| Ilustración 74. Selección de credenciales para la API de traducción | 51 |
| Ilustración 75. Creación de carpeta para almacenar credenciales..... | 51 |
| Ilustración 76. Permisos necesarios para realizar una traducción | 52 |
| Ilustración 77. Dependencias necesarias para uso de Google Cloud Translate | 52 |
| Ilustración 78. Exclusión de propiedades | 52 |
| Ilustración 79. Uso de las credenciales de Google..... | 53 |
| Ilustración 80. Código para realizar una traducción con Google Cloud | 53 |
| Ilustración 81. Árbol de interfaces de la aplicación | 54 |
| Ilustración 82. Ícono de la aplicación..... | 55 |
| Ilustración 83. Pantalla Splash | 55 |
| Ilustración 84. Pantalla Login | 56 |
| Ilustración 85. Dialogo informativo de uso de datos públicos | 57 |
| Ilustración 86. Pantalla User Register | 58 |
| Ilustración 87. Botones de la pantalla User Register | 58 |
| Ilustración 88. Listado de alérgenos por defecto | 59 |



| | |
|---|----|
| Ilustración 89. Cuadro para ingreso de nuevo alérgeno | 59 |
| Ilustración 90. Presentación de errores al ingresar un usuario | 60 |
| Ilustración 91. Pantalla Dashboard..... | 61 |
| Ilustración 92. Pantalla User Profile | 62 |
| Ilustración 93. Botones de la pantalla User Profile | 62 |
| Ilustración 94. Pantalla Recipe Search..... | 63 |
| Ilustración 95. Botones de la pantalla Recipe Search..... | 63 |
| Ilustración 96. Cuadro de diálogo de listado de alérgenos | 64 |
| Ilustración 97. Cuadro de diálogo de lugares de origen de recetas | 65 |
| Ilustración 98. Pantalla Recipes | 66 |
| Ilustración 99. Pantalla Recipe Detail..... | 66 |
| Ilustración 100. Pantalla Recipes Analysis..... | 67 |
| Ilustración 101. Alerta de precaución en análisis de recetas | 68 |
| Ilustración 102. Mensaje de no detección de alérgenos | 68 |
| Ilustración 103. Pantalla Scan..... | 69 |
| Ilustración 104. Botones de la pantalla Scan..... | 69 |
| Ilustración 105. Escaneo y reconocimiento de texto..... | 70 |
| Ilustración 106. Especificación de idiomas origen y destino en escaneo | 70 |
| Ilustración 107. Mensaje de Precaución en la pantalla Scan..... | 71 |
| Ilustración 108. Mensaje de no detección de alergenosen Scan | 72 |
| Ilustración 109. Pantalla Upload..... | 72 |
| Ilustración 110. Botones de la pantalla Upload..... | 73 |
| Ilustración 111. Especificación de idiomas de traducción en carga de imágenes | 73 |
| Ilustración 112. Traducción de texto..... | 74 |
| Ilustración 113. Mensaje de precaución en pantalla Upload | 75 |
| Ilustración 114. Mensaje de no detección en pantalla Upload | 75 |
| Ilustración 115. Registro de Usuario | 76 |
| Ilustración 116. Ingreso de nuevos alergenosen. Caso de uso 1 | 77 |
| Ilustración 117. Registro de usuario con nuevos alérgenos..... | 77 |
| Ilustración 118. Inicio de sesión, Caso de uso 1 | 78 |
| Ilustración 119. Mensaje informativo | 78 |
| Ilustración 120. Selección de búsqueda de recetas, Caso de uso 1 | 79 |
| Ilustración 121. Búsqueda de receta, Caso de uso 1 | 80 |
| Ilustración 122. Listado de recetas, Caso de uso 1 | 80 |
| Ilustración 123. Detalle de la receta, Caso de uso 1..... | 81 |
| Ilustración 124. Registrar usuario, Caso de uso 2 | 82 |
| Ilustración 125. Inserción de nuevos alérgenos, Caso de uso 2 | 82 |
| Ilustración 126. Registro de nuevo usuario y alérgenos, Caso de uso 2..... | 83 |
| Ilustración 127. Inicio de sesión, Caso de uso 2..... | 83 |
| Ilustración 128. Mensaje informativo, Caso de uso 2 | 84 |
| Ilustración 129. Selección de Búsqueda de recetas, Caso de uso 2 | 84 |
| Ilustración 130. Ingreso de valores previos a análisis de receta | 85 |
| Ilustración 131. Análisis de receta, Caso de uso 2 | 85 |
| Ilustración 132. Detalle de análisis de receta, Caso de uso 2..... | 86 |
| Ilustración 133. Registrar usuario, Caso de uso 3 | 87 |
| Ilustración 134. Registro de usuario y alergenosen, Caso de uso 3 | 87 |
| Ilustración 135. Inicio de sesión, Caso de uso 3..... | 88 |
| Ilustración 136. Selección de carga de imágenes | 89 |



| | |
|--|----|
| Ilustración 137. Captura fotográfica, Caso de uso 3 | 89 |
| Ilustración 138. Carga de una imagen, Caso de uso 3 | 90 |
| Ilustración 139. Selección de imagen de la galería | 90 |
| Ilustración 140. Traducción de texto, Caso de uso 3 | 91 |
| Ilustración 141. Mensaje de precaución, Caso de uso 3 | 91 |
| Ilustración 142. Resumen de bondades y carencias de la competencia | 94 |

1. Introducción

Una aplicación móvil es un software creado para funcionar en dispositivos móviles como teléfonos inteligentes, *tablets* entre otros. Según las tendencias globales, el uso de este tipo de dispositivos se ha intensificado de tal manera que ya forman parte indispensable en la vida diaria de sus consumidores.

Se afirma que “En España, los usuarios de Internet han aumentado en 2 millones, lo que supone ya el 85% de la población” (JUSTE, 2018), y esto ha ido incrementándose hasta la actualidad. Además, se asevera un evidente crecimiento del uso de dispositivos móviles, según el informe “un 9% más de usuarios emplean los smartphones para navegar de forma habitual por Internet, mientras que la cuota en ordenadores baja un 6%” (JUSTE, 2018), esto corresponde a un aproximado de 37,27 millones de usuarios móviles.

Gracias a la capacidad que tienen los teléfonos móviles de capturar, almacenar, alertar y transmitir información de una forma ágil y sencilla, el crecimiento de su uso en diversos sectores es inevitable. Una de esas áreas es la de alerta y prevención para mantener e incluso mejorar la calidad de vida de personas que sufren alguna afección de salud.

Las Alergias Alimentarias son “una respuesta exagerada del sistema inmunitario” (ConSalud.es, 2018) ante el consumo de un determinado alimento, que de ahora en adelante lo denominaremos “**Alérgeno**”.

Según una publicación del diario El Español “En los últimos años se ha registrado un aumento en la incidencia en las alergias alimentarias. En España, se estima que cerca de dos millones de personas son alérgicas a algún alimento” (D'Amelio, 2019). Las personas que las padecen deben prestar una especial atención a los alimentos que consumen, ya que un pequeño descuido puede provocar brotes en la piel, así como también problemas en el tracto digestivo o respiratorio. La gravedad de las reacciones depende de las características del alérgeno, la respuesta inmunológica de la persona afectada, la reactividad del órgano afectado.

La principal recomendación, para evitar las reacciones que puede producir un alérgeno en una persona, es evitar la ingesta del mismo, por lo cual se ha visto la necesidad de facilitar herramientas para la detección de estos productos de una manera ágil, rápida e intuitiva.

2. Descripción

Se propone como Trabajo de Fin de Master (TFM) el diseño e implementación de una aplicación para dispositivos móviles que usen el sistema operativo Android, que contribuya a identificar alérgenos en productos alimenticios y, a su vez, alertar para evitar el consumo de los mismos.

El nombre que se le ha otorgado a este proyecto es **Food Allergy Help**, y plantea una aplicación que brinda a los usuarios la facilidad de un escaneo de textos o imágenes, traducción y alerta ante la detección de presencia de potenciales alérgenos de acuerdo a los previamente configurados por cada usuario. Por otro lado, esta aplicación también permite buscar un amplio listado de recetas que pueden ser filtradas por sus características e ingredientes. Adicionalmente, es posible obtener una estadística de la presencia, en un determinado plato, de alguno de los alérgenos preestablecidos, junto con el despliegue de un detalle de las secciones en donde se ha identificado el alérgeno.

El idioma base de la aplicación es el inglés, por lo tanto, las recetas, ingredientes y su proceso de preparación se mostrarán en este idioma. De igual manera, los alérgenos preestablecidos por el usuario deben ser ingresados en inglés.

En concreto, gracias a esta aplicación, el comer en un restaurante, comprar en el supermercado, cocinar y viajar a otras zonas con idiomas diferentes ya no resultaría un problema, debido a que la detección del texto en otros idiomas y su traducción es inmediata, sin olvidar los análisis que realiza. Todo esto facilita mucho la prevención de ingesta de alimentos inadecuados que afectan la salud y bienestar de una persona.

3. Motivación

Personalmente, a lo largo de mi vida he podido palpar el inconveniente que representa el tener que estar constantemente con el temor de llegar a consumir algún producto que contenga un alérgeno que cause efectos adversos en mi cuerpo, ya que esas reacciones pueden ser leves, tales como un pequeño brote cutáneo, pero también pueden llegar a ser tan graves como un ataque de asma que obligue a ir a Emergencias de un Hospital.

El número de personas con alergias alimentarias alrededor del mundo crece constantemente. En el caso de alergias con reacciones severas, aunque existen algunos métodos que pueden ayudar a sobrellevarlas, tales como vacunas y medicamentos, cuyo camino más seguro para enfrentarlas es evitar el consumo del alérgeno. Pero existen individuos que tienen alergias a un conjunto amplio de alimentos, por lo que el ir a restaurantes y preguntar por platos que no contengan ninguno de los alérgenos resulta bastante tedioso, al igual que ir a un supermercado y revisar en las etiquetas los ingredientes de los productos. Además, cada vez son más las personas que viajan alrededor del mundo, por lo que el idioma también puede ser un gran inconveniente para identificar alérgenos.

Una alergia alimentaria puede ocasionar un estrés constante que afecta diversas áreas de la vida diaria como el trabajo, hogar, ocio, escuela y entre otras. **Food Allergy Help** brinda al usuario, en pocos segundos, la facilidad de detectar alérgenos en platos, etiquetas de productos y textos en general con el objetivo de ayudar a prevenir las afecciones que implican el consumo de un alérgeno.

4. Objetivos

4.1 Objetivo Principal

El objetivo principal de este trabajo es ofrecer una aplicación que permita ser una ayuda para las personas con distintas restricciones alimentarias, de manera que puedan preparar sus propias recetas o identificar productos cuyos ingredientes sean perjudiciales para su salud y bienestar.

Para llevar a cabo el desarrollo de la aplicación se han definido las siguientes funcionalidades:

4.2 Objetivos Específicos

- Registrar a nuevos usuarios y almacenar los ingredientes a los cuales tenga alergia.
- Presentar un listado de recetas de acuerdo a un conjunto de filtros tales como: palabras clave, tipo de plato, origen, ingredientes excluidos, ingredientes a los que se tiene intolerancia, tipo de dieta (vegetariana, vegana, entre otros).



- Detallar cada uno de los ingredientes necesarios y el proceso respectivo para preparar una receta.
- Escanear y leer textos en diversos idiomas, y traducirlos al idioma que el usuario necesite.
- Identificar posibles alérgenos a partir de las imágenes escaneadas o capturadas con el móvil.
- Alertar al usuario en caso de reconocer alérgenos.
- Evaluar los ingredientes de un conjunto de recetas en base a un nombre proporcionado y presentar un pequeño análisis sobre la presencia de elementos que causen reacciones alérgicas de acuerdo a la información del usuario.

5. Tecnologías y Herramientas de Desarrollo

5.1 Android Studio

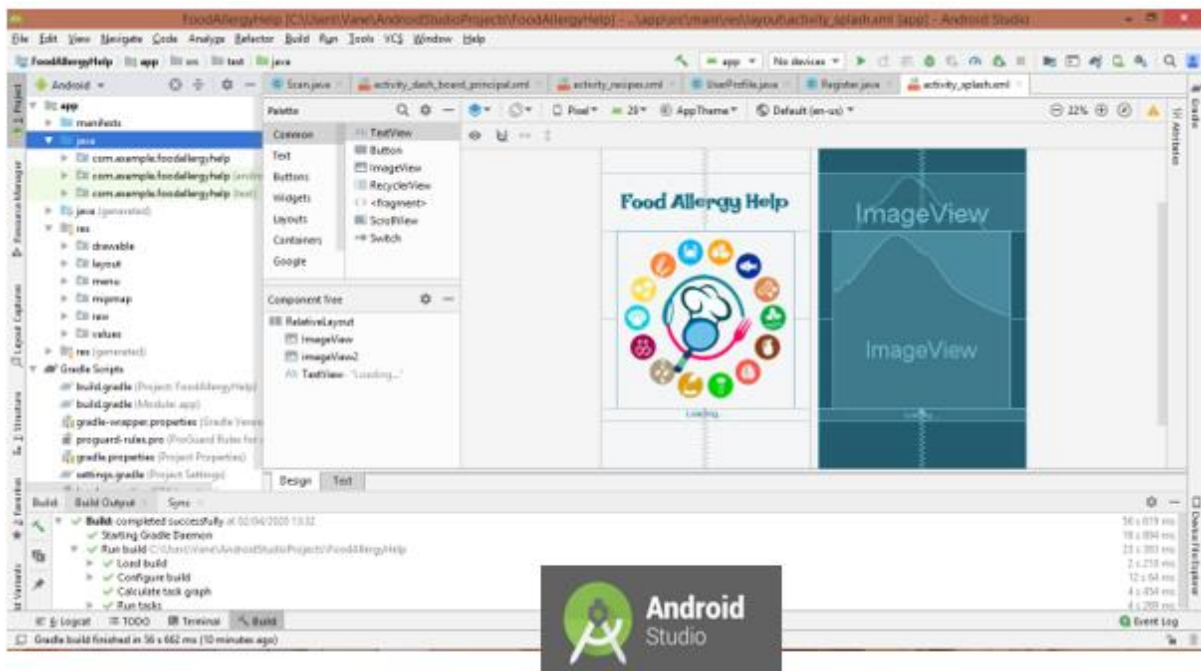
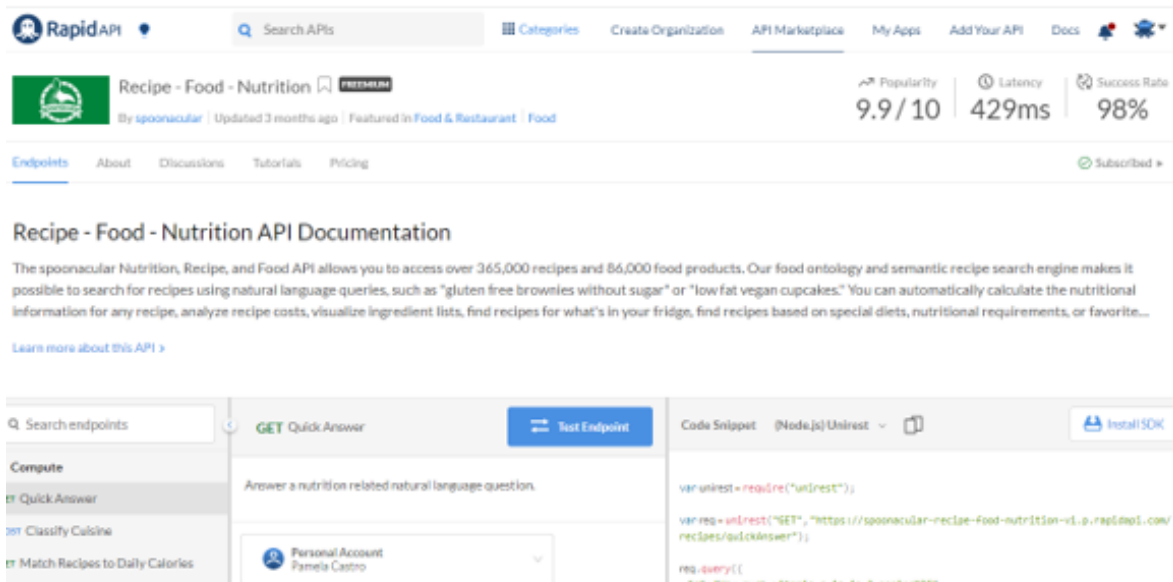


Ilustración 1. Interfaz de desarrollo de Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android, cuya interfaz la podemos visualizar en la Ilustración 1. Este está basado en IntelliJ IDEA. Con este IDE es mucho más fácil realizar cualquier tipo de aplicación para Android, desde la más sencilla hasta aquella mucho más compleja, y ha superado con creces a su predecesor Eclipse. Entre sus beneficios y características están:

- Posee un sistema de compilación flexible basado en Gradle
- Cuenta con un emulador rápido
- Tiene un entorno de desarrollo unificado para todos los dispositivos Android
- Permite la integración con GitHub y plantillas de código para compilar funciones de apps comunes.
- Proporciona un entorno visual de la interfaz.
- Posee una variedad de marcos de trabajo y herramientas de prueba
- Es compatible con C++, NDK y Google Cloud Platform

5.2 RapidAPI



The screenshot shows the RapidAPI website interface. At the top, there's a search bar and navigation links like 'Categories', 'Create Organization', 'API Marketplace', 'My Apps', 'Add Your API', and 'Docs'. Below this, the 'Recipe - Food - Nutrition' API is featured, with a popularity score of 9.9/10, a latency of 429ms, and a success rate of 98%. The main content area displays the API documentation for the 'GET Quick Answer' endpoint, including a description, a code snippet for using the API, and a 'Test Endpoint' button. The code snippet shows how to use the API with Node.js and Unirest.



Ilustración 2. Interfaz de RapidAPI

Es una *Startup* que permite la integración de APIs en cualquier plataforma, por medio de un Marketplace de APIs públicas, liberadas por empresas como Google, Microsoft, entre otras.

Esta permite a los desarrolladores alrededor del mundo consultar las APIs que libera una empresa y conectarlas directamente con su aplicación.

La interfaz de RapidAPI se visualiza en la Ilustración 2 y permite buscar fácilmente APIs relacionadas a diversos temas o áreas, y a su vez, realizar evaluaciones de las funcionalidades de cada API mediante el ingreso de parámetros para solicitudes de prueba online y visualización de las respuestas directamente desde el navegador. Adicionalmente, proporciona la facilidad de agregar APIs propias al mercado de RapidAPI para que sean consumidas por otras aplicaciones. Usando RapidAPI, puede consumir cualquier API usando un formato REST unificado que es sencillo de entender e incrustar en una aplicación

5.3 Firebase

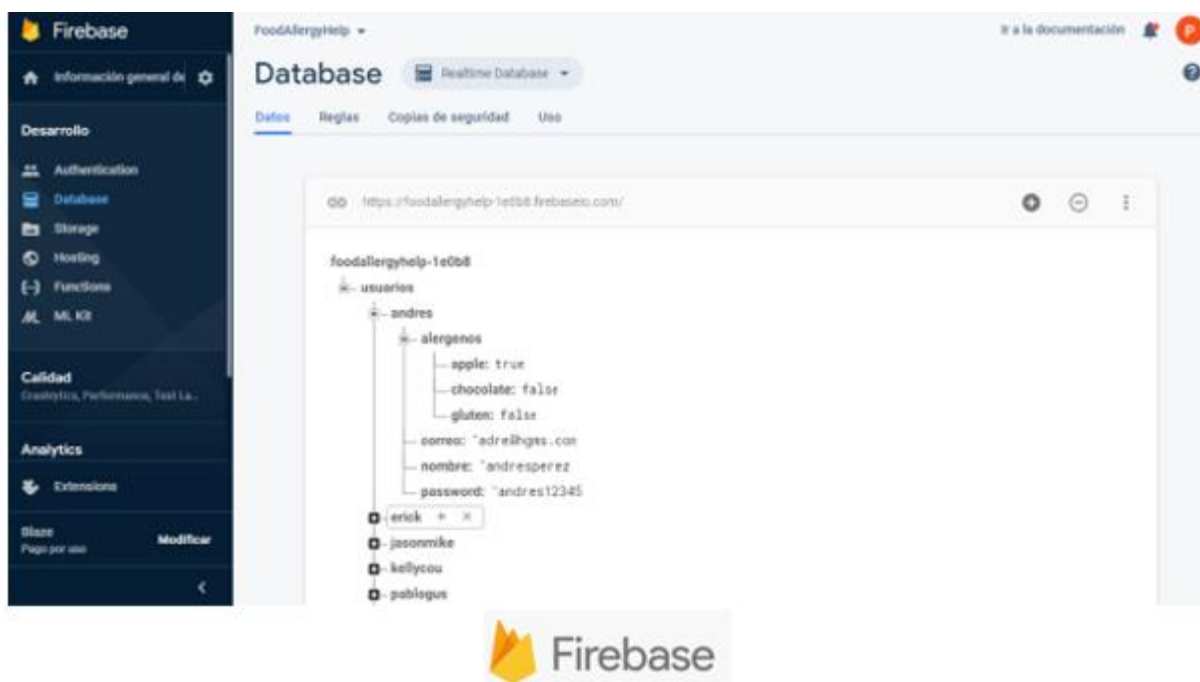


Ilustración 3. Interfaz de Firebase Database

Firebase es una plataforma en la nube, constituida por un grupo de herramientas y aplicaciones que hacen más fácil el desarrollo de aplicaciones. Esta es propiedad de Google, es usada para el desarrollo móvil y está disponible para diferentes plataformas (Android, iOS, web), por lo que es una gran alternativa para el ahorro de tiempo en el desarrollo y móvil.

Firebase es compatible con JavaScript (tanto en el Frontend como con Node), iOS (Swift y Objective C), Android (Java), Python, C++ y Unity.

Dispone de planes: gratuito, fijo y según servicios consumidos. Las facilidades más importantes que proporciona se describen a continuación:

5.3.1 Base de datos en tiempo real

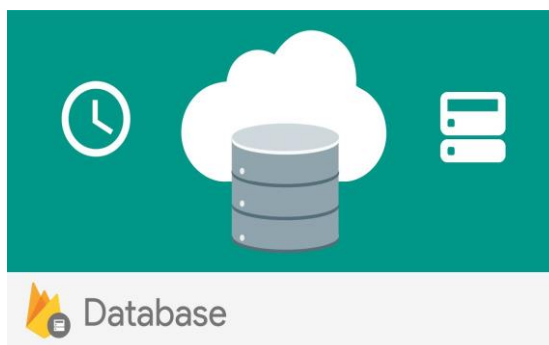


Ilustración 4. Base de Datos en Tiempo Real de Firebase

Firestore Realtime Database, cuya interfaz se muestra en la Ilustración 3, es una base de datos NoSQL alojada en la nube que permite guardar los datos que requiera una aplicación. Los datos son almacenados en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Proporciona un lenguaje flexible de reglas de seguridad basadas en expresiones y está diseñada para permitir solo operaciones que se puedan ejecutar rápidamente.

5.3.2 Alojamiento web



Ilustración 5. Alojamiento Web de Firebase

Permite el almacenamiento seguro y rápido de aplicaciones web (Ilustración 5). Al subir una aplicación, automáticamente, esta contará con SSL y HTTP2, y formará parte del CDN (Content Delivery Network) de Firebase, por lo tanto, se replicará en servidores a lo largo de todo el mundo, disminuyendo el tiempo de transferencia desde estos a el ordenador del visitante.

5.3.3 Almacenamiento

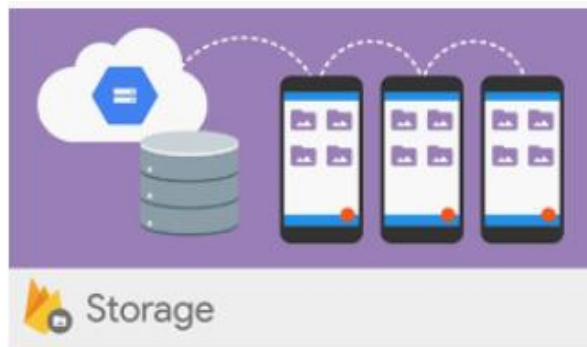


Ilustración 6. Almacenamiento con Firebase

Utilizado por programadores para guardar archivos generados por el usuario como videos, audios o imágenes (Ilustración 6). Al igual que otros servicios tiene un conjunto de reglas que se pueden configurar. Proporciona la seguridad de Google a las operaciones de carga y descarga de archivos sin importar la calidad de la red.

5.3.4 Autenticación



Ilustración 7. Autenticación con Firebase

Simplifica el inicio de sesión y su gestión dentro de la aplicación, ya sea web o móvil. Provee servicios de procesamiento detrás de la web, kits de desarrollo de software fáciles de usar y bibliotecas de interfaz de usuario ya elaboradas para autenticar a los usuarios en una app. Es fácil de configurar ya que aprovecha los estándares OAuth 2.0 y OpenID Connect. Permite una autenticación más allá del correo, contraseña o número de teléfono que son los más comúnmente utilizados, ya que puede hacerlo a través de otros proveedores de identidad populares como Twitter, Github, Facebook, Google, entre otros (Ilustración 7).

5.3.5 Monitoreo de Rendimiento



Ilustración 8. Monitoreo de Rendimiento con Firebase

Ayuda a medir el tiempo que transcurre mientras los usuarios están en los diferentes procesos de una aplicación, con la finalidad de detectar problemas en el tiempo de carga y poder efectuar las acciones necesarias para prevenirlos antes de que causen cualquier otro tipo de problema (Ilustración 8).

5.3.6 Laboratorio de Test



Ilustración 9. Laboratorio de Test con Firebase

Es una infraestructura de prueba de apps basada en la nube que es muy útil para los desarrolladores de Android ya que ofrece la posibilidad de probar una aplicación móvil en distintos dispositivos y entornos en donde se la configure.

5.3.7 ML Kit

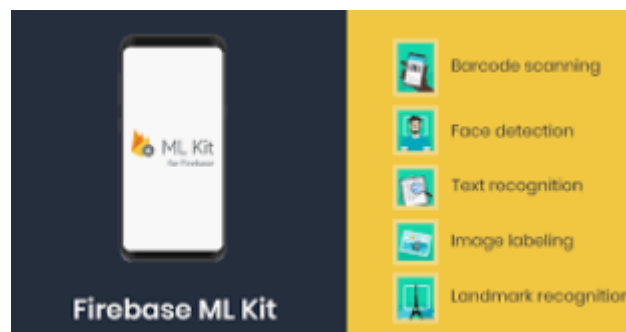


Ilustración 10. Machine Learning Kit con Firebase

Con pocas líneas de código brinda las bondades del aprendizaje automático a las aplicaciones de Android y iOS, aún sin tener un conocimiento profundo en modelos de optimización o en redes neuronales. Para aquellos que poseen mayores conocimientos este Kit facilita APIs para usar modelos propios de Tensor Flow en aplicaciones móviles.

Entre las funciones disponibles están: detección de rostro, escaneo códigos de barras, traducción, detección y seguimiento de objetos, identificación de idiomas, inferencia de modelos y respuesta inteligente (Ilustración 10).

Además, esta API puede reconocer texto en cualquier idioma latino. Esto incluye, pero no se limita a:

- Catalán
- Danés
- Holandés
- Inglés
- Finlandés
- Francés
- Alemán
- Húngaro
- Italiano
- Latín
- Noruego
- Polaco
- Portugués
- Rumano
- Español
- Sueco

- Tagalo
- Turco

5.4 Google Cloud

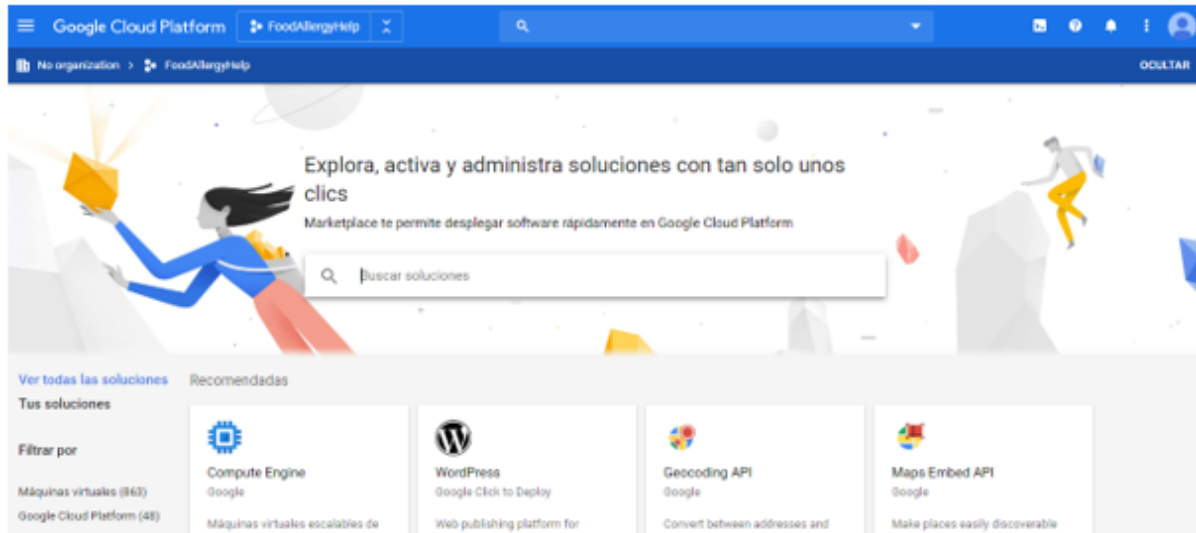


Ilustración 11. Consola de Google Cloud

Google Cloud es una plataforma cuyo objetivo principal es reunir a todas las aplicaciones de desarrollo web que en tiempos anteriores Google estaba ofreciendo por separado. Google Cloud aprovecha la tecnología almacenada en la nube para proporcionar un entorno virtual (Ilustración 11) que facilita la creación de nuevas aplicaciones que saquen provecho la escalabilidad de su infraestructura, rapidez, la gestión y almacenamiento de datos.

A continuación, se describen 2 herramientas de Google Cloud utilizadas en este proyecto:

5.4.1 Google Cloud Vision API

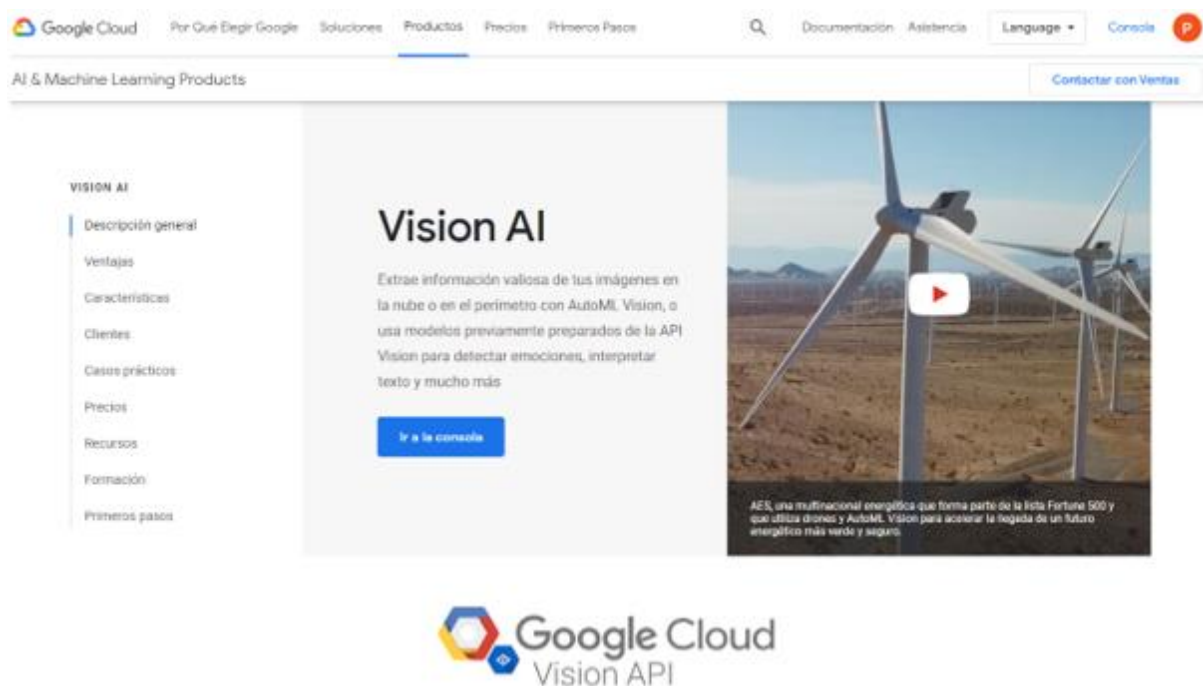


Ilustración 12. Consola de Google Cloud Vision API

Google Cloud Vision API (Ilustración 12) ofrece modelos de aprendizaje automático muy potentes a través de las API REST y RPC. Esta API asigna etiquetas a imágenes y las clasifica ágilmente en millones de categorías predefinidas, con ello detecta objetos y caras, lee texto impreso y manuscrito, y obtiene metadatos de amplio valor para un catálogo de imágenes.

5.4.2 Google Cloud Translate API

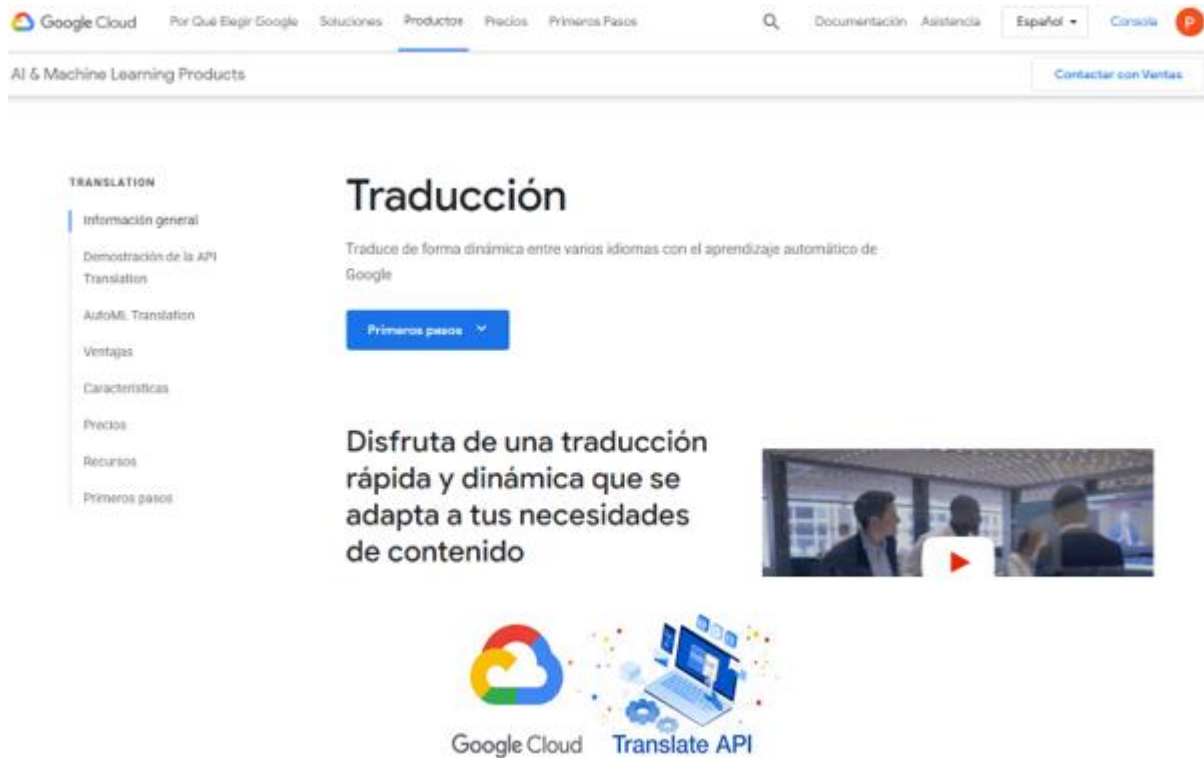


Ilustración 13. Consola de Google Cloud Translate API

Proporciona una consola intuitiva (Ilustración 13) que permite realizar las configuraciones necesarias para hacer uso de sus funciones de traducción dinámica de texto de un idioma a otro. Para ello, utiliza modelos de aprendizaje automático de Google preparados previamente o modelos personalizados.

Se ofrecen varias ediciones de traducción (básico y avanzado) con sus propios planes de precios. La edición básica permite traducir de forma instantánea textos a más de cien idiomas. Por otro lado, la edición avanzada admite traducción por lotes, glosarios personalizados, modelos de AutoML y etiquetas.

6. Arquitectura de la aplicación

En este apartado se visualiza en la Ilustración 14, de una manera gráfica, el flujo de información existente entre las distintas tecnologías utilizadas y la aplicación desarrollada, con el objetivo de lograr un mayor entendimiento técnico global de todo el proyecto.

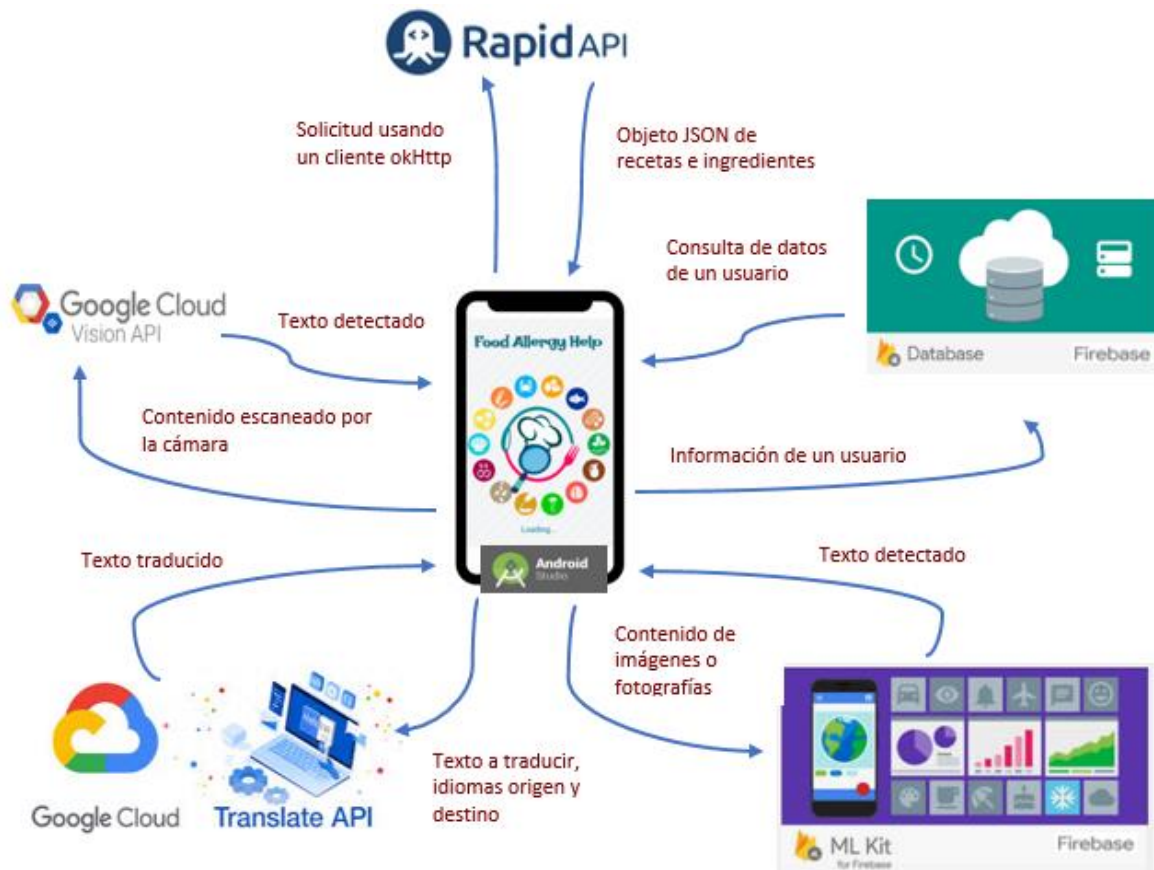


Ilustración 14. Arquitectura de la Aplicación

El diagrama expuesto (Ilustración 14) constituye la infraestructura necesaria para poder implementar el proyecto, en este se distinguen 2 partes principales:

- **Front-end:** Corresponde a la interfaz de usuario. Para implementarla se utiliza **Android Studio**, el cual realiza el diseño de las interfaces por medio de archivos XML asociados a cada Actividad o formulario, y usa Java para proporcionarles funcionalidad.
- **Back-end:** La aplicación necesita una conexión a internet para efectuar las consultas necesarias con el objetivo de obtener información proporcionada por algunos servicios en la nube de Google y RapidAPI. Los servicios utilizados se detallan a continuación:
 - **RapidAPI:** Utiliza OkHttp para efectuar las consultas de recetas alimenticias, por lo que se envía desde la aplicación los filtros correspondientes en base a las necesidades del usuario y RapidAPI devuelve un objeto JSON con la información detallada de las recetas consultadas.
 - **Google Cloud Vision API:** Es útil para la detección de texto en tiempo real. La aplicación envía el contenido escaneado por la cámara del móvil y esta API dará como respuesta el texto identificado.



- **Google Cloud Translate API:** Ayuda a traducir un texto desde un idioma origen a un idioma destino, previamente configurados por el usuario. La aplicación envía el texto, los idiomas origen y destino, y esta API devuelve el texto traducido.
- **ML Kit de Firebase:** Contribuye a detectar el texto existente en una imagen cargada por el usuario a la aplicación. La aplicación envía el contenido de una imagen o fotografía y Firebase devolverá el texto identificado en la misma.
- **DataBase de Firebase:** Permite guardar y consultar la información de cada uno de los usuarios que hacen uso de la aplicación.

7. Implementación y Desarrollo

Food Allergy Help ha sido implementada utilizando el IDE Android Studio 3.5.3 y Java Development Kit 1.8.0_241.

Una vez instalado Android Studio, es fundamental tener actualizado el *SDKManager*, por lo que hay que ir a Tools > SDKManager y marcar al menos las opciones indicadas en la Ilustración 15 y la Ilustración 16.

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

| Name | API Level | Revision | Status |
|---|-----------|----------|---------------|
| <input type="checkbox"/> Android R Preview | R | 1 | Not installed |
| <input checked="" type="checkbox"/> Android 10.0 (Q) | 29 | 4 | Installed |
| <input type="checkbox"/> Android 9.0 (Pie) | 28 | 6 | Not installed |
| <input type="checkbox"/> Android 8.1 (Oreo) | 27 | 3 | Not installed |
| <input checked="" type="checkbox"/> Android 8.0 (Oreo) | 26 | 2 | Installed |
| <input checked="" type="checkbox"/> Android 7.1.1 (Nougat) | 25 | 3 | Installed |
| <input checked="" type="checkbox"/> Android 7.0 (Nougat) | 24 | 2 | Installed |
| <input checked="" type="checkbox"/> Android 6.0 (Marshmallow) | 23 | 3 | Installed |
| <input type="checkbox"/> Android 5.1 (Lollipop) | 22 | 2 | Not installed |
| <input type="checkbox"/> Android 5.0 (Lollipop) | 21 | 2 | Not installed |
| <input type="checkbox"/> Android 4.4W (KitKat Wear) | 20 | 2 | Not installed |
| <input type="checkbox"/> Android 4.4 (KitKat) | 19 | 4 | Not installed |
| <input type="checkbox"/> Android 4.3 (Jelly Bean) | 18 | 3 | Not installed |
| <input type="checkbox"/> Android 4.2 (Jelly Bean) | 17 | 3 | Not installed |

Ilustración 15. Opciones escogidas en SDK Plataforms

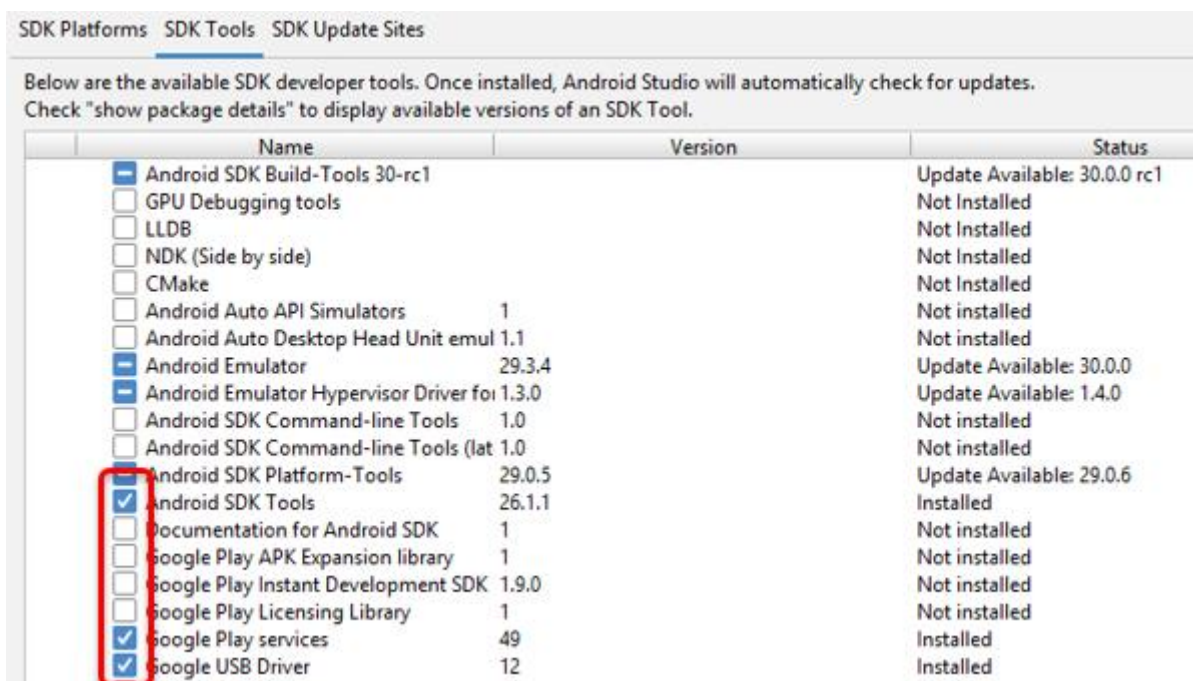


Ilustración 16. Opciones escogidas en SDK Tools

7.1 Reconocimiento de Texto con Google Cloud

El reconocimiento de texto o, normalmente conocido como Reconocimiento Óptico de Caracteres (OCR), es aquel que permite identificar símbolos o caracteres, pertenecientes a un alfabeto, a partir de imágenes o fotografías. En nuestra aplicación necesitamos esta detección de texto para poder identificar alérgenos en tiempo real en etiquetas de productos procesados, recetas, cartas en restaurantes o cualquier otro texto que el usuario requiera analizar. El resumen de este proceso se muestra en [Anexo 1. Reconocimiento de texto en tiempo real](#)

A continuación, se describe un resumen de los pasos principales necesarios para efectuar el escaneo:

- a. En el proyecto de Android Studio, en *Build gradle (Module)* agregar la dependencia de *google play services vision* como indica la Ilustración 17.

```
implementation 'com.google.android.gms:play-services-vision:9.8.0'
```

Ilustración 17. Dependencia para google play services vision

- b. En *AndroidManifest.xml* agregar a nivel de la etiqueta “manifest” agregar los metadatos y permisos respectivos, mostrados en la Ilustración 18.



```
<uses-permission android:name="android.permission.INTERNET" />

<meta-data
    android:name="com.google.android.gms.vision.DEPENDENCIES"
    android:value="ocr" />
```

Ilustración 18. Permisos y metadatos para usar OCR de Google Cloud

- c. Agregar una nueva Actividad llamada “**Scan**”.
- d. En el diseño visual de la actividad **Scan** agregar un componente *SurfaceView*, que será el área en donde se visualizará la imagen a escanear.

```
<SurfaceView
    android:id="@+id/sfvCamara"
    android:layout_width="match_parent"
    android:layout_height="260dp" />
```

Ilustración 19. Componente *SurfaceView* usado para escanear

- e. Declarar un objeto *CameraSource* dentro de la actividad *Scan.java*

```
CameraSource cameraSource;
```

Ilustración 20. Declaración del objeto *CameraSource*

- f. Crear un objeto *TextRecognizer*, éste procesa las imágenes y determina qué texto aparece dentro de ellas.

```
TextRecognizer textRecognizer = new TextRecognizer.Builder(getApplicationContext()).build();
```

Ilustración 21. Declaración del objeto *TextRecognizer*

- g. Relacionar el objeto *TextRecognizer* con *CameraSource* para que pueda reconocer el texto que visualice con la cámara del móvil.

```
if (textRecognizer.isOperational()) {
    cameraSource = new CameraSource.Builder(getApplicationContext(), textRecognizer)
        .setFacing(CameraSource.CAMERA_FACING_BACK)
        .setRequestedPreviewSize(640, 480)
        .setAutoFocusEnabled(true)
        .build();
}
```

Ilustración 22. Relación de *TextRecognizer* con *CameraSource*



- h. Para la detección del texto es necesario invocar al método “**setProcessor**”, tal como se indica en la Ilustración 23, el cual recibe los elementos reconocidos en un *array*. En este caso el texto final reconocido se lo mostrará en un componente visual *TextView* llamado “**tvCamara**”.

```
textRecognizer.setProcessor(new Detector.Processor<TextBlock>() {  
    @Override  
    public void release() {  
    }  
  
    @Override  
    public void receiveDetections(Detector.Detections<TextBlock> detections) {  
        final SparseArray<TextBlock> items = detections.getDetectedItems();  
        if(items.size() != 0){  
            tvCamara.post(() -> {  
                StringBuilder stringBuilder = new StringBuilder();  
                for(int i = 0; i<items.size();i++){  
                    TextBlock item = items.valueAt(i);  
                    stringBuilder.append(item.getValue());  
                    stringBuilder.append("\n");  
                }  
                tvCamara.setText(stringBuilder.toString());  
            });  
        }  
    }  
});
```

Ilustración 23. Obtener el texto reconocido con *setProcessor*

7.2 Configuración de Android Studio con Firebase

Antes de empezar a utilizar las bondades que brinda Firebase es necesario realizar la configuración previa en Android Studio, sin olvidar el asegurarse de tener instalada al menos la versión 26 del Repositorio de Google. El resumen de los pasos a seguir se visualiza en [Anexo 2. Conexión a Firebase para almacenar datos](#). El proceso a realizar es el siguiente:

- Abrir el proyecto en Android Studio.
- Seleccionar la opción de conectar a *gmail* presionando el último botón de la barra de tareas.

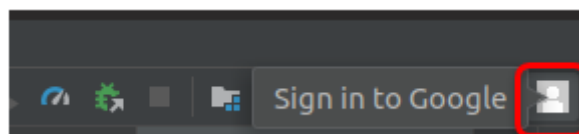


Ilustración 24. Conectar Android Studio con Google



- c. Iniciar sesión con una cuenta de *gmail*. Esto permite, en los siguientes pasos, conectarse directamente con Firebase.
- d. Ir a Tools > Firebase y en la parte derecha del IDE se abrirá el asistente de Firebase para Android.
- e. Ir a “**Realtime Database**” y hacer clic en el link “**Save and Retrieve Data**”.

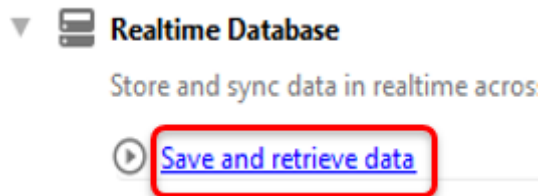


Ilustración 25. Relacionar Android Studio con Realtime Database

- f. Desplegar la opción de FireStore > Leer y escribir información > Connect to Firebase
- g. Ingresar el nombre del nuevo proyecto Firebase (o seleccionar uno ya existente) y presionar el botón “**Connect to Firebase**”. Una vez conectado se agregará un archivo denominado “**google-services.json**” al proyecto, el cual contiene la información referente a la conexión entre Android Studio y Firebase.

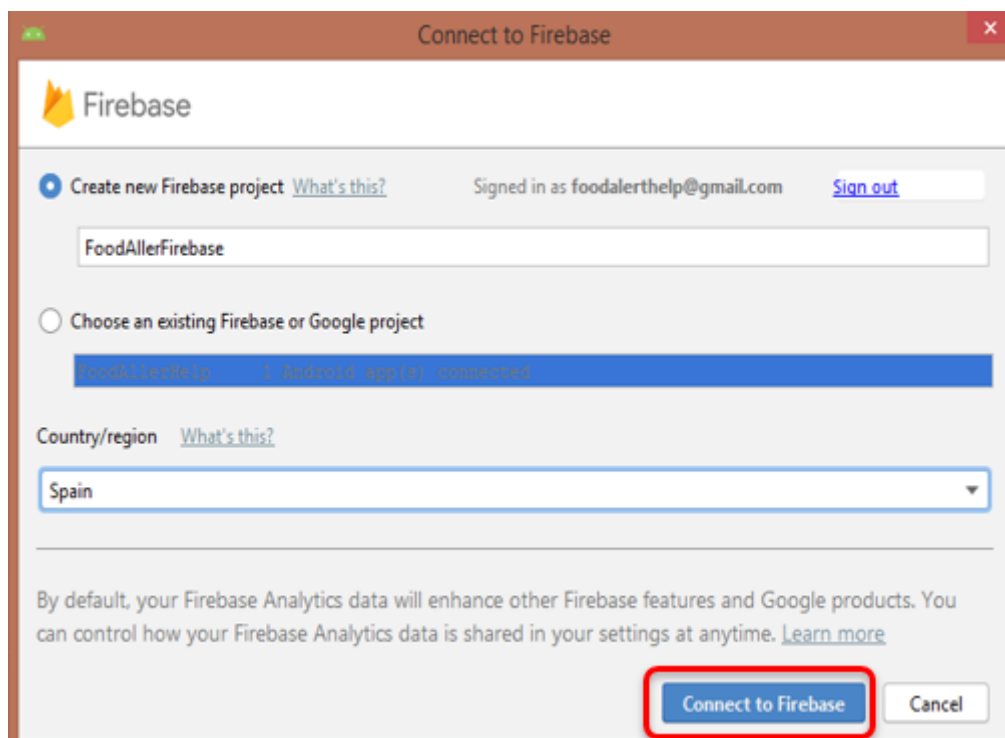


Ilustración 26. Crear nuevo proyecto Firebase desde Android Studio

7.2.1 Visualización de la configuración del proyecto FireBase

Como se verá más adelante, la aplicación permite utilizar Firebase para algunos aspectos tales como el almacenamiento en tiempo real de los datos del usuario y la detección de texto. En caso de algún momento se requiera visualizar la configuración realizada con Firebase se requerirán los siguientes pasos:

- a. Ir a la consola en la siguiente dirección <https://console.firebase.google.com/>. Aquí debe aparecer el proyecto anteriormente creado. Seleccionar este proyecto.



Ilustración 27. Consola de Firebase

- b. Al ir a Project Overview > Configuración del Proyecto se pueden ver los datos tales como: *Id del proyecto*, *Nombre del paquete*, *Huellas digitales de certificado SHA*, *Clave de API de la Web*, entre otros.

En el árbol del proyecto del IDE se puede verificar que se ha creado un archivo llamado “**google-services.json**”, el cual contiene toda la información necesaria para realizar la conexión con Firebase. En la Ilustración 28 se muestra el *path* de localización del archivo dentro del proyecto creado con Android Studio.

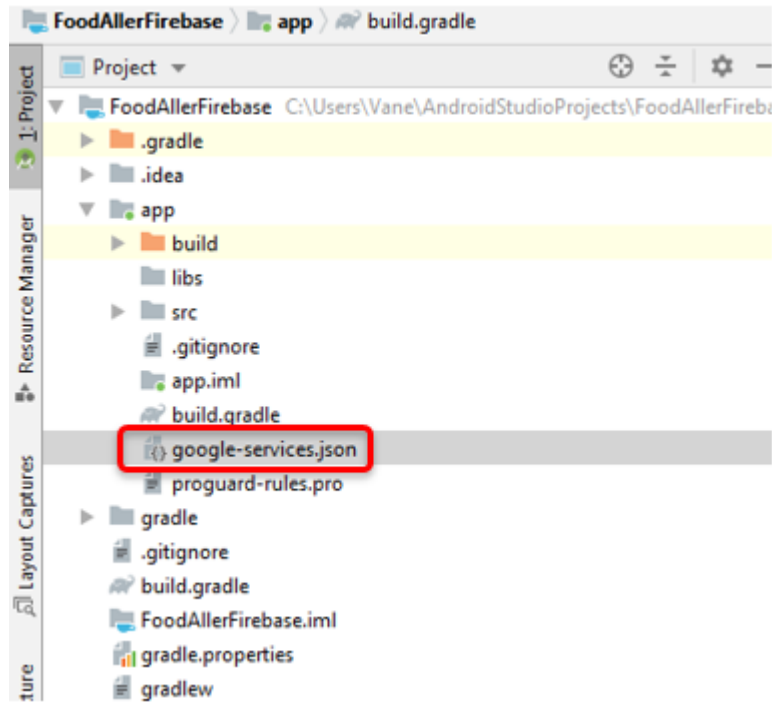


Ilustración 28. Ubicación del archivo google-services.json

La clave **SHA1** (Ilustración 29) es la que da acceso al usuario conectado para escribir y leer, tomando en cuenta la aplicación indicada en “**Nombre del paquete**”. Este último debe ser el que se encuentra en el archivo **AndroidManifest.xml** del proyecto de Android Studio.



Ilustración 29. Nombre del paquete y huellas SHA en la consola

Para saber el nombre del **SHA1** del proyecto se puede ir a Gradle > app > android > signing Report. En parte inferior de la ventana se ejecuta un comando que muestra algunos valores incluidos el **SHA1**.

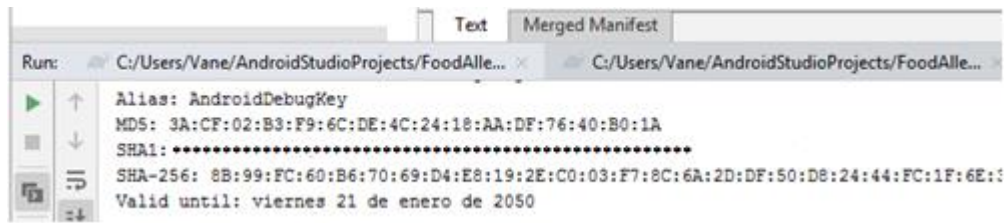


Ilustración 30. SHA1 del proyecto en Android Studio

Cabe recalcar que los valores dados para el *SHA1* y *Nombre del paquete* en el archivo *.json* ya son dados automáticamente al realizar los pasos anteriores.

7.3 Almacenamiento de información en la Base de Datos de Firebase

Para poder guardar los datos de los usuarios y sus restricciones alimenticias en la nube aprovechamos las bondades que brinda **Realtime Database** de Firebase. La síntesis del proceso se halla en [Anexo 3. Configuración para el almacenamiento de datos](#) y [Anexo 4. Almacenamiento de datos del usuario](#). Su configuración y uso se realiza de la siguiente manera:

- Ir a la consola de Firebase <https://console.firebase.google.com/> e iniciar sesión con una cuenta de *gmail*
- Seleccionar el proyecto anteriormente creado.
- Ir a Desarrollo > Database y presionar el botón “**Crear base de datos**”. Para este caso escogeremos la opción “**Empezar en el modo de prueba**” y la ubicación de *Cloud Firestore* más cercana a nuestra localidad. Se demorará algunos hasta que se termine de crear todos los componentes necesarios.

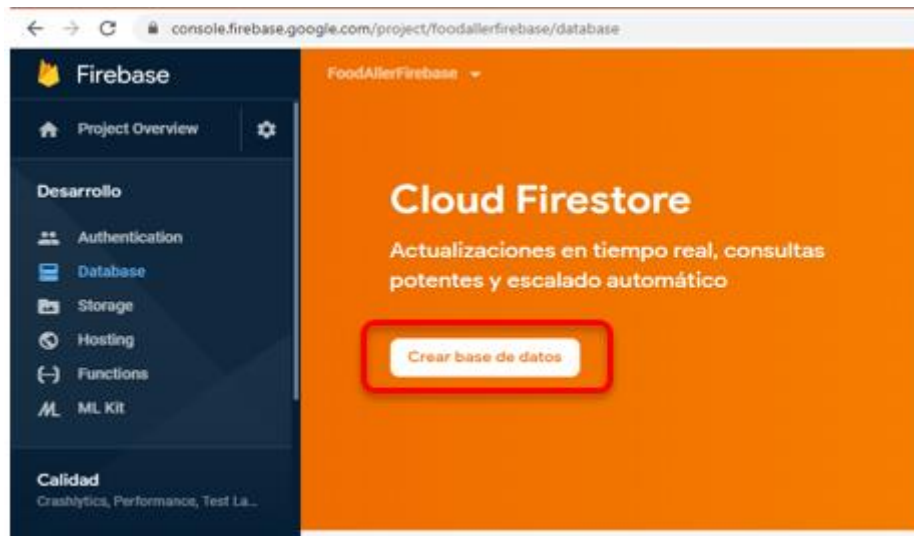


Ilustración 31. Crear Base de Datos en Firebase

- d. Una vez creada la base de Datos, aparecerán 2 opciones posibles para trabajar con la base de datos en tiempo real: *Cloud Firestore* y *Realtime database*. En nuestro caso escogeremos la última opción.
- e. En la pestaña **“Reglas”** cambiar las condiciones para que se pueda leer y escribir en Base de Datos, es decir, **“read”** y **“write”** con el valor de **“true”** y hacer clic en el botón **“Publicar”**.



Ilustración 32. Reglas de Seguridad con Firebase

- f. Para que nuestro proyecto en Android Studio trabaje con la base de datos de Firebase hay que ir a **Tools > Firebase** y presionar el paso 2 llamado **“Add the Realtime Database to your app”**. Esto agrega las implementaciones necesarias al *Gradle* de nuestro proyecto por lo que hay terminar presionando el botón **“Accept Changes”**. En la Ilustración 33 se detallan las dependencias que se agregan.

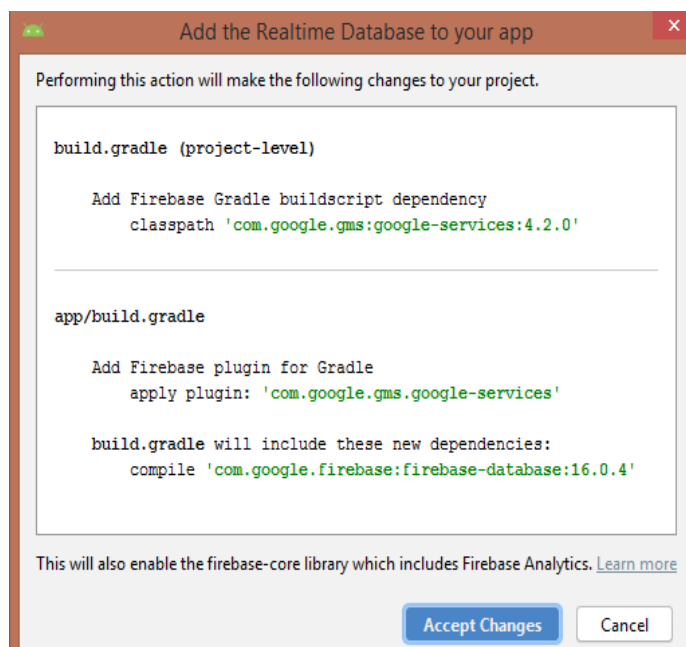


Ilustración 33. Implementaciones a agregar para uso de Firebase Database

- g. Hay que recalcar que Firebase maneja una base de datos en tiempo real NoSQL en formato JSON, por lo que la forma de administrarla es distinta. En el caso de nuestro proyecto se maneja un nodo principal llamado **“usuarios”** que es el que contendrá toda la información referente a los usuarios que utilicen la aplicación. Para ingresar un nuevo registro es importante, primeramente, crear una clase llamada **“UsuarioObject”**, cuyos atributos correspondan a los datos del usuario que se requieran guardar. Adicionar el constructor y los *getters* y *setters* de todos los atributos. En la Ilustración 34 se muestran los atributos del usuario, que definen la estructura de lo que posteriormente se almacenarán en Firebase.

```
public class UsuarioObject {

    private String nombre;
    private String password;
    private String correo;
    private Map<String, Boolean> alergenos;
```

Ilustración 34. Clase usada para almacenar información en Firebase

- h. Crear objeto tipo usuario que sea una instancia de la clase *UsuarioObject* y asignarle los valores dados por el usuario tales como: Nombre, Password y Correo.



```

UsuarioObject user = new UsuarioObject();
user.setNombre(etUser.getText().toString());
user.setPassword(etPassword.getText().toString());
user.setCorreo(etEmail.getText().toString());

```

Ilustración 35. Asignación de valores a un objeto antes de almacenarlo

- i. Crear un objeto *Map*, como se describe en la Ilustración 36, para ir añadiendo cada uno de los alérgenos que afecten a ese usuario.

```

// ingredientes
Map<String, Boolean> alergenos = new HashMap<>();
for (int i = 0; i < UsuarioAlergiasAdapter.items.size(); i++){
    //Insertamos un nuevo elemento en base de datos
    if (UsuarioAlergiasAdapter.items.get(i).isSeleccionado())
        alergenos.put (UsuarioAlergiasAdapter.items.get(i).getNombre(),
            UsuarioAlergiasAdapter.items.get(i).isAgregado());
}
user.setAlergenos (alergenos);

```

Ilustración 36. Almacenamiento de alérgenos en un Map

- j. Para guardar el registro completo del usuario es necesario crear una referencia a la Base de Datos de Firebase, específicamente al nodo **“usuarios”**, y añadir el objeto tipo *UsuarioObject* que contiene los datos.

```

databaseReference.child("usuarios").child(user.getNombre()).setValue(user);

```

Ilustración 37. Almacenamiento de un usuario en Firebase

En la consola de Firebase se puede visualizar una estructura similar a la mostrada en la Ilustración 38.

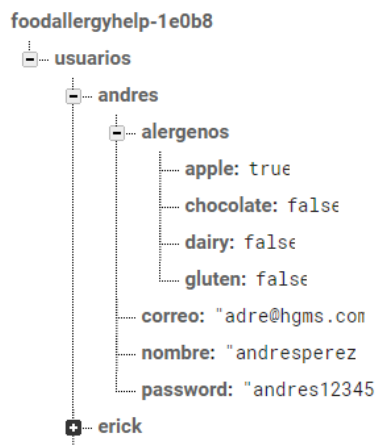


Ilustración 38. Estructura de usuarios en la Base de Datos de Firebase

7.4 Búsqueda de información de la Base de Datos de Firebase

En pasos anteriores se mostró como se almacena información en la Base de Datos en tiempo real utilizando una clase definida en Android Studio como “UsuarioObject”. Para extraer información de Firebase se utiliza esa misma clase. El proceso se detalla a continuación:

- Crear una referencia al nodo “**usuarios**” de la base de datos de Firebase como se ve en la Ilustración 39.

```
// Read from the database
DatabaseReference dbr = FirebaseDatabase.getInstance().getReference( path: "usuarios");
```

Ilustración 39. Referencia a la Base de Datos en Firebase

- Añadir en el evento `onDataChange`, de la referencia creada, el código necesario para extraer la información del usuario consultado y guardarlo en un objeto del tipo “**UsuarioObject**” de manera que pueda ser manipulado más fácilmente.

```
dbr.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        UsuarioObject usuario = new UsuarioObject();
        String pw = "";
        for (DataSnapshot postSnapshot : dataSnapshot.getChildren()) {
            usuario = (UsuarioObject) postSnapshot.getValue(UsuarioObject.class);

            if (usuario.getNombre().equals(nombre)) {
                pw = usuario.getPassword();
                break;
            }
        }
    }
})
```

Ilustración 40. Consultar a Base de Datos en Firebase

7.5 Reconocimiento de Texto con Firebase

El presente proyecto permite buscar ingredientes peligrosos a partir de un texto, que puede ser por ejemplo una receta, composición de un alimento o el listado de ingredientes de un producto procesado. Con la API de reconocimiento de Firebase se puede reconocer texto, en cualquier idioma que provenga del latín, a partir de una imagen o fotografía tomada por el usuario. El resumen de los pasos a seguir se muestra en [Anexo 5. Configuración para](#)

[reconocimiento de texto con Firebase](#) y [Anexo 6. Reconocimiento de texto con Firebase](#)

Para efectuar el reconocimiento de texto es necesario efectuar el siguiente proceso:

- a. Agregar dependencia en el proyecto de Android Studio, en *build.gradle*.

```
// para deteccion de textos  
implementation 'com.google.android.gms:play-services-vision:9.8.0'
```

Ilustración 41. Dependencia para uso de OCR con Firebase

- b. En el archivo AndroidManifest.xml, agregar dentro de la etiqueta “application” el metadato de **ocr**.

```
<meta-data  
    android:name="com.google.firebase.ml.vision.DEPENDENCIES"  
    android:value="ocr" />
```

Ilustración 42. Metadatos para uso de OCR con Firebase

- c. Asegurarse de mantener conectado el proyecto de Android Studio con su cuenta de *gmail*. Esto se puede constatar al ir al último icono de la barra de herramientas (Ilustración 43) y al hacer clic se puede visualizar los datos de la cuenta conectada, caso contrario es necesario ingresar los datos de la cuenta de *gmail* con la que se va a efectuar el reconocimiento de texto.

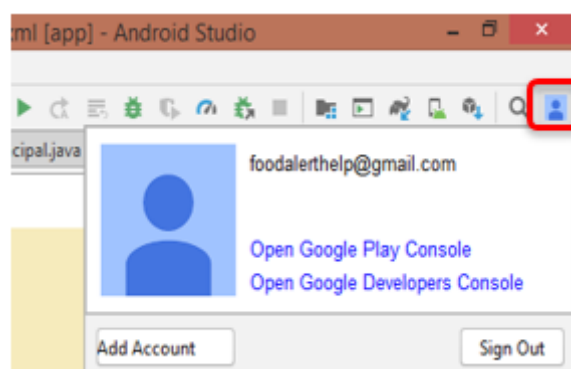


Ilustración 43. Conexión de Android Studio con Google

- d. Ir a Tools > Firebase, con lo que se desplegará una ventana que nos permite facilitar el uso de las distintas bondades que proporciona Firebase.



- e. Ir a *ML Kit* y desplegar la opción “**Use ML Kit to recognize text in images**”.

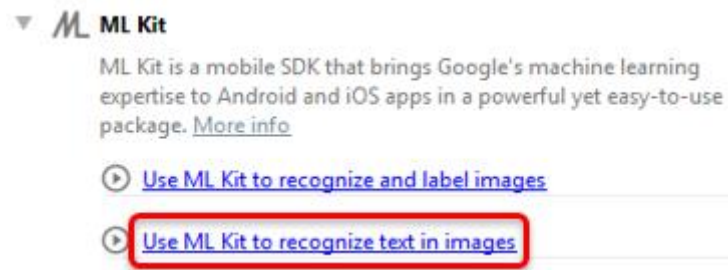


Ilustración 44. Usar Machine Learning Kit desde Android Studio

- f. En pasos previos ya hemos conectado nuestro proyecto a Firebase, por lo que ahora es únicamente necesario añadir *ML* (Machine Learning) a nuestro proyecto. Seleccionar “**Add ML to your app**”, esto añade la dependencia necesaria a nuestro *build.gradle* (*Module*).

```
build.gradle will include these new dependencies:  
compile 'com.google.firebase:firebase-ml-vision:18.0.1'
```

Ilustración 45. Dependencia para usar Firebase Vision

- g. En el *layout* de la Actividad denominada “**Upload**” agregar un *ImageView* que es donde se visualizará la imagen a cargar.
- h. Para abrir la galería fotográfica cuando se presione un botón se debe agregar un *Intent ACTION PICK* descrito en la Ilustración 46. En caso de querer tomar una fotografía se debe utilizar un *Intent IMAGE CAPTURE*.

```
btnUpload.setOnClickListener((v) -> {  
    Intent galleryInt = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.INTERNAL_CONTENT_URI);  
    startActivityForResult(galleryInt, RESULT_LOAD_IMAGE);  
});
```

Ilustración 46. Código para abrir la galería fotográfica

- i. Para mostrar la imagen escogida de la galería en el objeto *ImageView* se debe hacer uso de la librería **Glide**, por lo que hay que añadir las dependencias en el *build.gradle* (*Module*) del proyecto.



```
implementation 'com.github.bumptech.glide:glide:4.11.0'  
annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'
```

Ilustración 47. Dependencias para usar Glide

- j. En la *Activity* de carga de la imagen añadir el método para realizar esa acción. Es permite que la imagen no se deforme al colocarla dentro del objeto *ImageView*. El código sería similar al mostrado en la Ilustración 48.

```
private void useImage(Uri uri)  
{  
    Glide.with(getApplicationContext())  
        .load(uri)  
        .into(imageUP);  
}
```

Ilustración 48. Carga de imágenes desde galería usando Glide

- k. Agregar el método *OnActivityResult* (Ilustración 49) para que cargue la imagen en el *ImageView* cuando haya terminado de seleccionar la imagen de la galería.

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
  
    if(requestCode == RESULT_LOAD_IMAGE && resultCode == RESULT_OK && data != null) {  
        final Uri uri = data.getData();  
  
        useImage(uri);  
    }  
}
```

Ilustración 49. Carga de imagen a un *ImageView*

- l. Agregar el código para relacionar la imagen cargada con el objeto *FirebaseVisionImage* que se encarga del reconocimiento del texto.

```
Bitmap bm=((BitmapDrawable)imageUP.getDrawable()).getBitmap();  
FirebaseVisionImage image = FirebaseVisionImage.fromBitmap(bm);
```

Ilustración 50. Relación de *Firebase Vision* con una imagen cargada

- m. Crear un objeto *FirebaseVisionTextRecognizer* que es el que permite el reconocimiento a partir de una imagen


```
FirebaseVisionTextRecognizer textRecognizer = FirebaseVision.getInstance()
    .getOnDeviceTextRecognizer();
```

Ilustración 51. Declaración del objeto FirebaseVisionTextRecognizer

- n. Procesar el objeto *FirebaseVisionTextRecognizer* creado para que procese la imagen cargada y convertida en bitmap. El texto reconocido se guardará en una variable tipo *String* que puede en un futuro mostrarse de forma visual al usuario.

```
textRecognizer.processImage(image)
    .addOnSuccessListener(new OnSuccessListener<FirebaseVisionText>() {
        @Override
        public void onSuccess(FirebaseVisionText result) {

            // Task completed successfully
            String resultText = result.getText();
```

Ilustración 52. Obtención de texto reconocido con Firebase

7.6 Uso de RapidAPI para consulta de recetas

Dentro de la aplicación, para consultar recetas alimenticias que cumplan ciertos criterios, se hace uso de la API de **Spoonacular** proporcionada por RapidAPI. Para conectar una API de RapidAPI a un proyecto o aplicación se debe tener una clave para autenticar la solicitud. Al crear una aplicación dentro de RapidAPI, se genera una clave API (X-RapidAPI-Key) específica para esa aplicación y, además, se puede ver análisis basados en las llamadas API que realiza con esta clave de aplicación. El compendio del proceso necesario se presenta en [Anexo 7. Consulta de Recetas \(Rapid API\)](#), [Anexo 8. Consulta de Recetas \(Android Studio\)](#) y [Anexo 9. Flujo para la Consulta de Recetas](#).

Para este caso específico de consulta de datos de recetas, los pasos necesarios son los siguientes:

- Ir a la página de RapidAPI <https://rapidapi.com/> y crear una cuenta o iniciar sesión si ya dispone de una.
- Buscar la API que necesite de acuerdo a sus necesidades. En nuestro caso buscamos API's de recetas y escogimos a la **API de Nutrición, Recetas y Alimentos Spoonacular** que permite acceder a más de 365,000 recetas y 86,000 productos alimenticios.
- Presionar el botón "**Suscribe to Test**" y RapidAPI proveerá una *X-RapidAPI-Key* que es la que permite acceder a los servicios de la API.

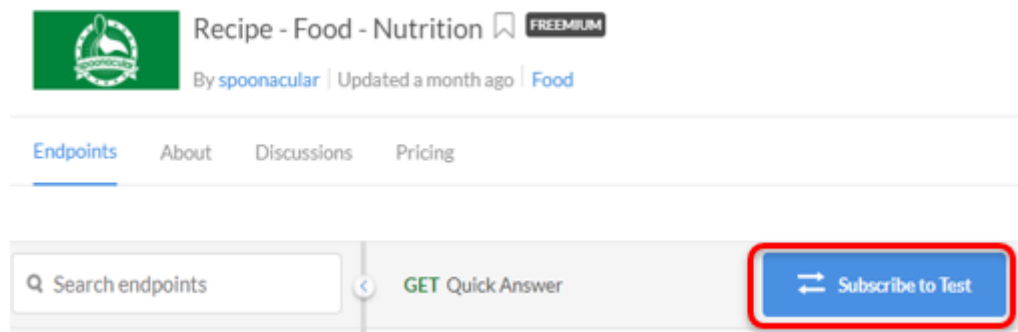


Ilustración 53. Suscripción a API de Spoonacular para extracción de recetas

- d. Dependiendo la API escogida, existen algunos planes de suscripción para su uso (tanto gratuitos como de pago en base al número de peticiones que se hagan por día o mes). Muchas API tienen un plan **BASIC** gratuito, lo que significa que puede usar ese plan para probar la API de forma gratuita siempre que se mantenga dentro de los límites de la cuota establecida. Se debe escoger alguno de los planes presentados de acuerdo a las necesidades del proyecto, en nuestro caso se ha escogido el plan **Basic Gratuito**.
- e. Una vez suscrito, se puede probar las distintas consultas que se pueden hacer. En la parte izquierda, como se muestra en la Ilustración 54, se localizan los nombres de los métodos disponibles, al hacer clic en uno de ellos se muestra en la parte central de la página un ejemplo de ese método con los parámetros de entrada respectivos.

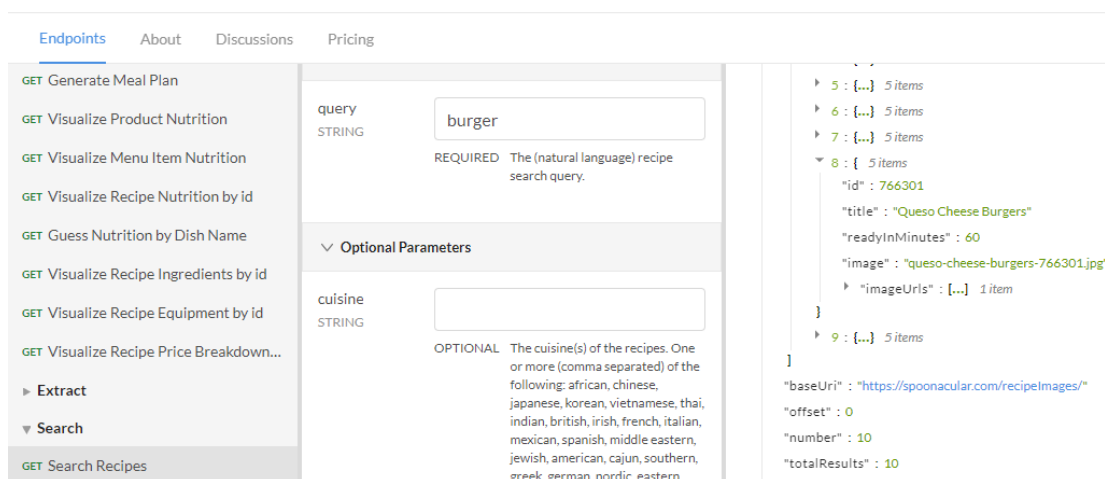


Ilustración 54. Consulta de métodos de prueba con RapidAPI

En la parte derecha de esa misma página, en la pestaña **“Test Result”** se muestran los resultados de la consulta realizada en formato JSON. Esto permite visualizar todos los atributos y valores que devuelve la consulta efectuada.



Ilustración 55. Estructura JSON de recetas con Spoonacular de RapidAPI

- f. Junto a la pestaña **“Test Result”** se encuentra la pestaña **“Code Snippet”** (Ilustración 56), que ofrece el código necesario para realizar la petición en distintos lenguajes de programación.

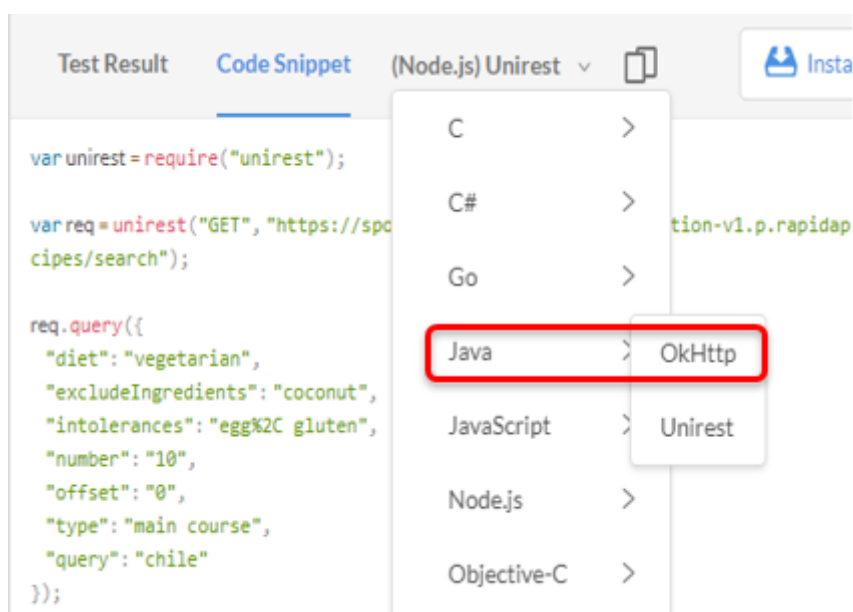


Ilustración 56. Consulta de código de peticiones en distintos lenguajes usando RapidAPI



En nuestro caso hemos escogido el lenguaje **Java** con **OkHttp**, por lo que se muestra el código de la petición con el host de la API (Ilustración 57) y su respectiva *API key*, que fue otorgada cuando nos suscribimos. Este nos servirá para hacer las peticiones Http desde el proyecto de Android Studio.

```
Test Result Code Snippet (Java) OkHttp Install SDK

OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
    .url("https://spoonacular-recipe-food-nutrition-v1.p.rapidapi.com/recipes/search?diet=vegetarian&excludeIngredients=coconut&intolerances=egg%252C%20gluten&number=10&offset=0&type=main%20course&query=chile")
    .get()
    .addHeader("x-rapidapi-host", "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com")
    .addHeader("x-rapidapi-key", "aquí debe colocar su API Key")
    .build();

Response response = client.newCall(request).execute();
```

Ilustración 57. Código para consulta de recetas usando OkHttp

- g. **OkHttp** es una librería de código abierto que permite realizar peticiones HTTP de una forma mucho más sencilla. En el proyecto de Android Studio se debe agregar la dependencia necesaria en el *Build.Gradle (module)* para poder realizar consultas utilizando esta librería.

```
// para peticiones a RestAPI
implementation 'com.squareup.okhttp3:okhttp:3.4.1'
```

Ilustración 58. Dependencia para uso de OkHttp

- h. Construir la URL de la petición en base a los datos otorgados por el usuario. En la Ilustración 59 se ve un ejemplo de cómo se estructura la URL junto con los valores que se asignan a cada atributo, que en este caso son variables.

```
////////// construye la petición ////////////
urlPetición = "https://spoonacular-recipe-food-nutrition-v1.p.rapidapi.com/recipes/search?" +
    "cuisine=" + cocina + "&" +
    "diet=" + dieta + "&" +
    "excludeIngredients=" + ingEx + "&" +
    "intolerances=" + ingInt + "&" +
    "number=" + numero + "&offset=0&" +
    "type=" + tipo + "&" +
    "query=" + etKeyWords.getText();
```

Ilustración 59. Estructura de la URL para consulta de recetas



- i. En la Actividad donde se desea hacer la consulta, crear un objeto **OkHttpClient** y construir la petición **Request** con los parámetros proporcionados por RapidAPI.

```
OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
    .url(urlPetición)
    .get()

    .addHeader( name: "x-rapidapi-host", value: "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com")
    .addHeader( name: "x-rapidapi-key", value: "Put you key here")
    .build();
```

Ilustración 60. Código del Request con OkHttp para obtener recetas

- j. Ejecutar la petición y obtener el resultado en un objeto tipo **Response**. En este caso lo objetemos en una variable tipo String, como se ve en la Ilustración 61.

```
Response response = client.newCall(request).execute();
String resultado = response.body().string();
```

Ilustración 61. Obtener resultado de la petición de recetas

- k. El resultado proporcionado por RapidAPI es en formato JSON, razón por la cual es necesario descomponerlo para extraer los valores, para ello se utiliza un objeto **JSONObject** que se relaciona con un **JSONArray** que es el que contiene todo el listado de recetas fruto de la consulta efectuada.

```
JSONObject json = new JSONObject(resultado);
JSONArray resultsArray = json.getJSONArray( name: "results");
```

Ilustración 62. Descomposición del objeto JSON de recetas obtenidas

- l. Por último, se recorre todo el objeto **JSONArray** con el objetivo de ir extrayendo cada registro y añadiéndolo a una lista para poder presentarla al usuario.

En la Ilustración 63 se visualiza como a partir de un JSONObject, correspondiente a cada receta, se extrae cada uno de sus atributos tales como: Id, título, imagen, duración y número de porciones.



```
for (int indice = 0; indice < resultsArray.length(); indice++) {
    // Obtener objeto a través del índice
    JSONObject receta = resultsArray.getJSONObject(indice);

    id = receta.getString( name: "id");
    titulo = receta.getString( name: "title");
    if(receta.has( name: "image"))
        imagen = receta.getString( name: "image");
    else
        imagen = "none";
    duracion = receta.getString( name: "readyInMinutes");
    porciones = receta.getString( name: "servings");

    listRecetas.add(new RecetaObject(id,titulo,imagen,duracion,porciones));
}
```

Ilustración 63. Extracción de cada receta del objeto JSON

7.7 Traducción de textos

Con el objetivo de identificar alérgenos en recetas, etiquetas de productos y cualquier otro texto que esté escrito en otro idioma, se ha visto la necesidad de efectuar una traducción. La traducción del texto identificado, mediante el escaneo en tiempo real o la carga de imágenes, se la realiza mediante a API de traducción de Google, la cual usa modelos de aprendizaje automático. El resumen del proceso necesario se ve en [Anexo 10. Traducción de textos \(Google Cloud\)](#) y [Anexo 11. Traducción de textos \(Android Studio\)](#). Los pasos necesarios para realizar esta traducción son los siguientes:

- a. Ir a *Google API Console* en <https://console.cloud.google.com/apis/> e iniciar sesión con una cuenta de *gmail*.
- b. En la pestaña “**Google Cloud Platform**” se debe seleccionar un proyecto existente o crear uno nuevo. En la Ilustración 54 se muestra un componente de lista cuyo encabezado dice “Selecciona un proyecto”, al desplegar este se puede escoger el proyecto creado con anterioridad.

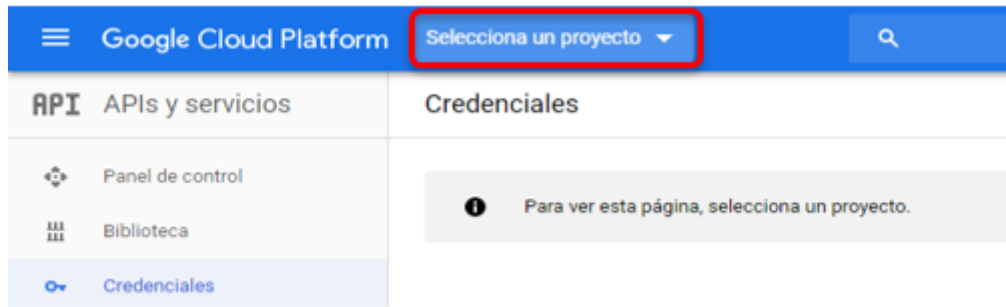


Ilustración 64. Selección de un proyecto en la Plataforma de Google Cloud

- c. Hacer clic en **“APIs y Servicios”** en la opción de navegación del lado izquierdo y luego en el botón **“+ HABILITAR APIS Y SERVICIOS”**

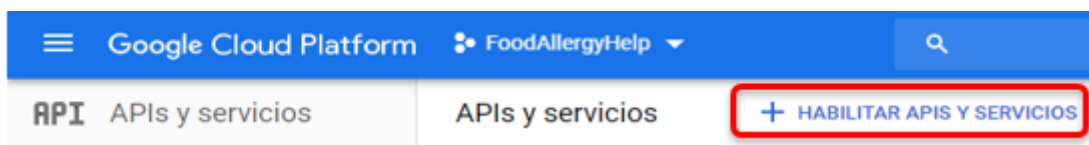


Ilustración 65. Habilitación de APIs y servicios de Google Cloud

- d. Se presentará un buscador de APIs y servicios, en este caso se busca la API de traducción como se ve en la Ilustración 66.

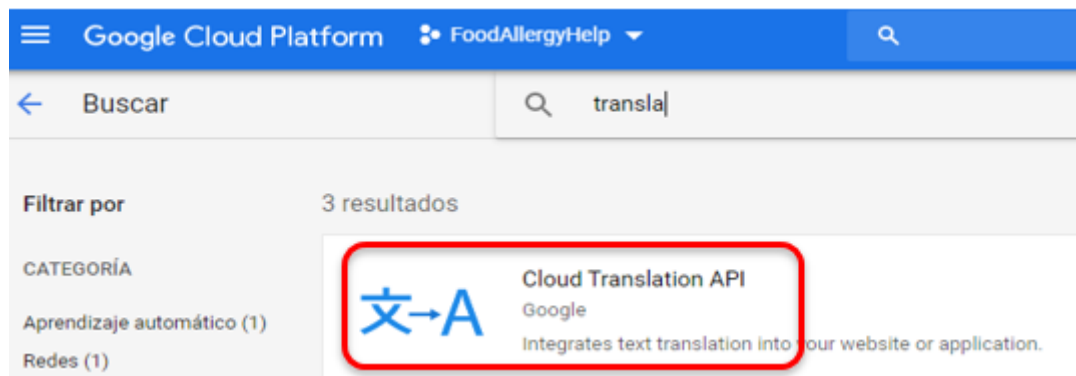


Ilustración 66. Búsqueda de la API de traducción en Google Cloud

- e. Habilitar el API de Traducción al presionar el botón **“Habilitar”**.



Ilustración 67. Habilitar API de traducción

- f. Hacer clic en el botón **“Habilitar la Facturación”**.

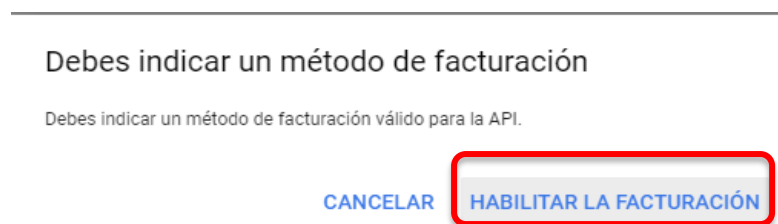


Ilustración 68. Habilitar la facturación de la API de traducción

- g. Presionar el botón **“Crear Cuenta de Facturación”**.

Habilita la facturación del proyecto "FoodAllergyHelp"

No administras ninguna cuenta de facturación. Para habilitar la facturación en este proyecto, crea una cuenta de facturación o ponte en contacto con tu administrador de cuentas de facturación para que la habilite por ti. [Más información](#)

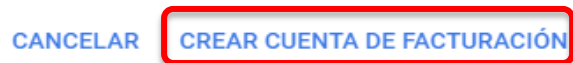


Ilustración 69. Crear Cuenta de Facturación

- h. Escoger el país y aceptar los Términos y Condiciones. Presionar el botón **“Continuar”**.



Paso 1 de 2

País

España

Condiciones del servicio

He leído y acepto las [condiciones del servicio de la versión de prueba gratuita de Google Cloud Platform](#).

Debes marcar esta casilla para continuar

CONTINUAR

Ilustración 70. Aceptar términos y condiciones de Google Cloud

- i. Ingresar la información personal solicitada.
- j. Iniciar la versión de prueba o de pago.
- k. Hacer clic en el botón **“ADMINISTRAR”** (Ilustración 71) para empezar configurar esta API para poder utilizarla.



Ilustración 71. Administrar la API de traducción de Google Cloud

- l. Ir a la opción de menú llamada **“Credenciales”**, que en la Ilustración 72 es la última opción de todo el listado presentado.

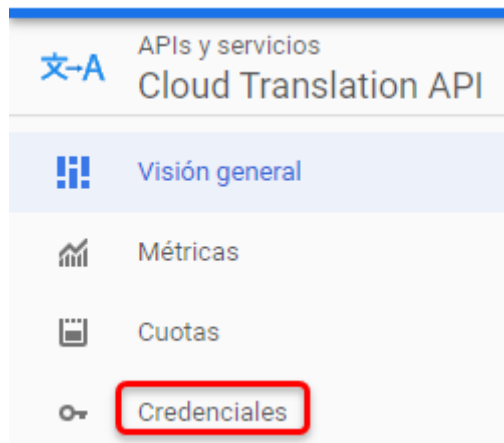


Ilustración 72. Administrar las Credenciales de Google Cloud

- m. Al presionar el botón “+ **Crear Credenciales**” se extenderá un grupo de opciones de las cuales se debe escoger la última, denominada “**Ayudadme a elegir**”.

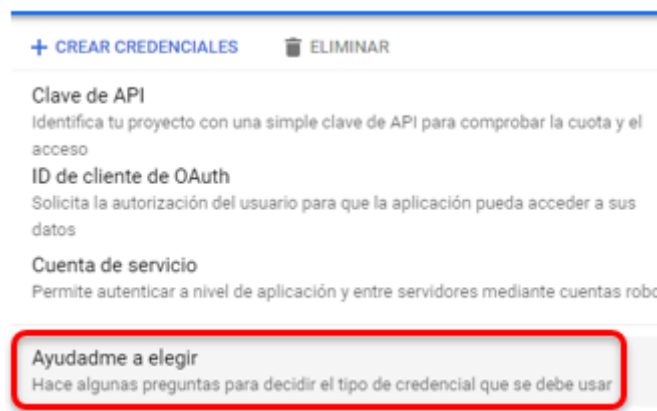


Ilustración 73. Despliegue de opciones adicionales de credenciales

- n. Escoger la API “**Cloud Translation API**”. En la parte inferior de la misma ventana preguntan si se tiene pensado utilizar esta API en **App Engine o Compute Engine** (Ilustración 74), ante lo cual se debe escoger la opción “**No, no las estoy usando**” y presionar el botón “**¿Qué credenciales necesito?**”



Añadir credenciales al proyecto

1 Averigua qué tipo de credenciales necesitas

Te ayudaremos a configurar las credenciales adecuadas. Puedes saltarte este paso y crear una clave de API, un ID de cliente o una cuenta de servicio.

¿Qué API estás utilizando?

Las API utilizan diversas plataformas de autorización, y se pueden restringir algunas credenciales para que solo llamen a ciertas API.

Cloud Translation API

A esta API de Google Cloud Platform se suele acceder desde un servidor mediante una cuenta de servicio. Si, por el contrario, quieres acceder en nombre de otro usuario (lo cual es habitual para algunas tareas administrativas), crea un cliente de OAuth.

¿Tienes pensado utilizar esta API en App Engine o en Compute Engine?

Puedes usar las credenciales de aplicación predeterminadas en las aplicaciones que se estén ejecutando en GCE y GAE. En otras palabras: no es necesario que crees otras.

No, no las estoy usando

¿Qué credenciales necesito?

Ilustración 74. Selección de credenciales para la API de traducción

- o. Se ingresa el nombre de la cuenta de servicio y se escoge que el tipo de clave sea **JSON**. Se generará un archivo de extensión **json**, que se descargará automáticamente dentro del equipo.
- p. Para evitar errores futuros, se recomienda renombrar el archivo descargado previamente a “**credentials.json**”.
- q. En Android Studio, ir a la carpeta **res** que se encuentra en el navegador del proyecto, hacer clic derecho y escoger New > Android Resource Directory. El tipo de recurso debe ser **raw**.

Directory name: raw

Resource type: raw

Source set: main

Ilustración 75. Creación de carpeta para almacenar credenciales

- r. Pegar el archivo “**credentials.json**” que se descargó y renombró en pasos previos, dentro de la carpeta raw.



- s. Agregar los permisos necesarios en el archivo AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Ilustración 76. Permisos necesarios para realizar una traducción

- t. Adicionar la dependencia de Google Cloud a nivel de módulo de *build.gradle*.

```
// para traducción de textos
implementation('com.google.cloud:google-cloud-translate:1.12.0') {
    exclude group: 'org.apache.httpcomponents'
    exclude group: 'org.json', module: 'json'
}
annotationProcessor 'com.google.cloud:google-cloud-translate:1.12.0'
```

Ilustración 77. Dependencias necesarias para uso de Google Cloud Translate

- u. Para evitar conflictos de dependencias, en el mismo archivo *build.gradle* de módulo añadir, dentro del nodo **android** (Ilustración 78), el código **exclude 'project.properties'**.

```
android {
    ...
    packagingOptions {
        exclude 'project.properties'
    }
}
```

Ilustración 78. Exclusión de propiedades

- v. Iniciar el servicio de traducción usando las credenciales descargadas y almacenadas dentro de la carpeta **raw** del proyecto. En la Ilustración 79 se muestra el código necesario para acceder a los recursos del proyecto, leer las credenciales guardadas en el archivo “credenciales.json” y utilizarlas para crear el servicio de traducción.



```
StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);

try (InputStream is = getResources().openRawResource(R.raw.credentialsgoogle)) {

    //Get credentials:
    final GoogleCredentials myCredentials = GoogleCredentials.fromStream(is);

    //Set credentials and get translate service:
    TranslateOptions translateOptions = TranslateOptions.newBuilder().setCredentials(myCredentials).build();
    translate = translateOptions.getService();

} catch (IOException ioe) {
    ioe.printStackTrace();
}
```

Ilustración 79. Uso de las credenciales de Google

- w. Realizar la traducción, del idioma de origen al idioma destino que escoja el usuario, utilizando el objeto **Translation** de Google Cloud.

El modelo que se va a utilizar en este caso es el “**base**”, este se indica como tercer parámetro en el método de traducción de la Ilustración 80.

```
Translation translation;

translation = translate.translate(originalText,
    Translate.TranslateOption.sourceLanguage(idiomaOrigen),
    Translate.TranslateOption.targetLanguage(idiomaDestino),
    Translate.TranslateOption.model("base"));

textoTraducido = translation.getTranslatedText();
```

Ilustración 80. Código para realizar una traducción con Google Cloud

8. Interfaz de Usuario

Una interfaz es el recurso mediante el cual el usuario realiza la interacción con la aplicación que está usando, por ello es de suma importancia que sea muy intuitiva. Atractiva y que proporcione la información de una forma clara y precisa. En esta sección se explicarán cada una de las interfaces de la aplicación desarrollada.

8.1 Árbol de Interfaces

Previo al desarrollo de la aplicación se estableció el árbol de interfaces (Ilustración 81) para poder conocer la navegación que tiene que realizar el usuario para efectuar una determinada acción.

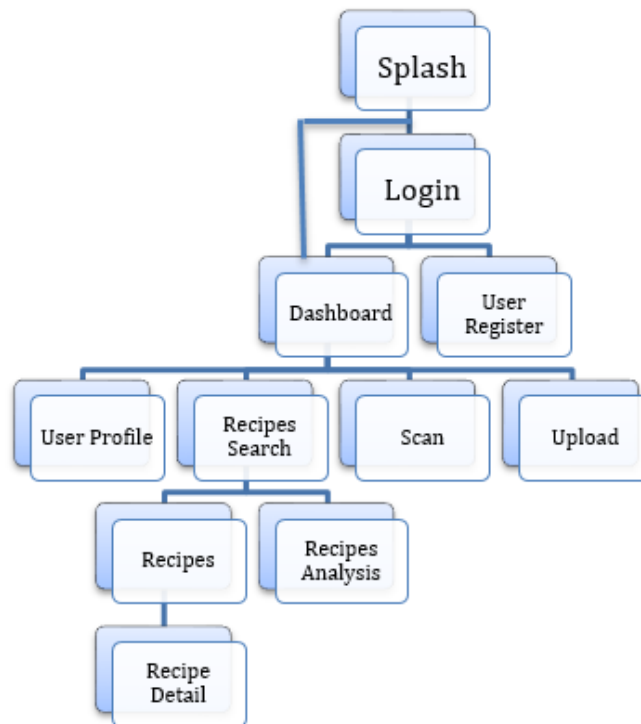


Ilustración 81. Árbol de interfaces de la aplicación

Al abrir la aplicación, la primera pantalla que se visualizará será “Splash”, la cual da una especie de bienvenida y dentro de un par de segundos, dependiendo si el usuario ha iniciado una sesión previamente, se redireccionará a la pantalla “Login” si no hay una sesión iniciada o a “Dashboard” si ya hay una sesión. Desde la ventana “Login” se puede ir a “User Register” para registrar un nuevo usuario o a “Dashboard” comenzar a hacer uso de las funcionalidades propias de la aplicación o modificar los datos del usuario. En el siguiente apartado de describen a mayor detalle la funcionalidad cada una de las pantallas del Árbol de Interfaces.

8.2 Descripción de Interfaces

8.2.1 Splash

Para ingresar a la aplicación se debe hacer clic en el icono de **Food Allergy** (Ilustración 82) desplegado en la lista de aplicaciones del móvil.

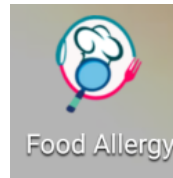


Ilustración 82. Ícono de la aplicación

La pantalla inicial llamada “**Splash**” (Ilustración 83) es la que se muestra por un par de segundos mientras se carga la aplicación.

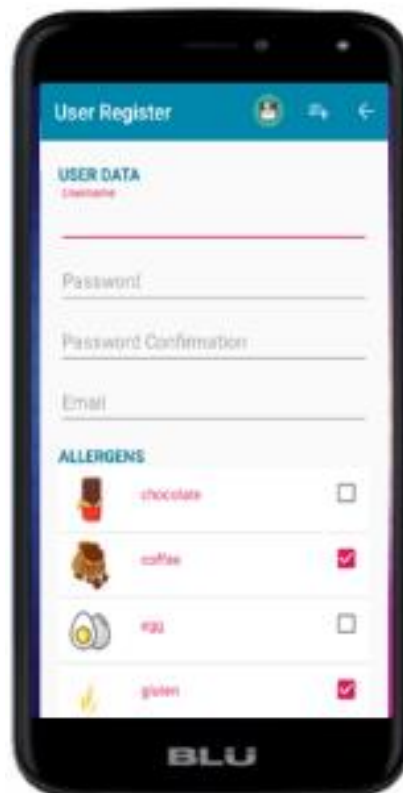


Ilustración 83. Pantalla Splash

En caso de que haya una sesión de la aplicación abierta, esta pantalla direccionará directamente a la ventana “**Dashboard**”, caso contrario se mostrará la ventana “**Login**” para que se inicie una sesión o se registre un nuevo usuario.

8.2.2 Login

Esta pantalla (Ilustración 84) permite el Inicio de sesión o de Registro de un nuevo usuario.

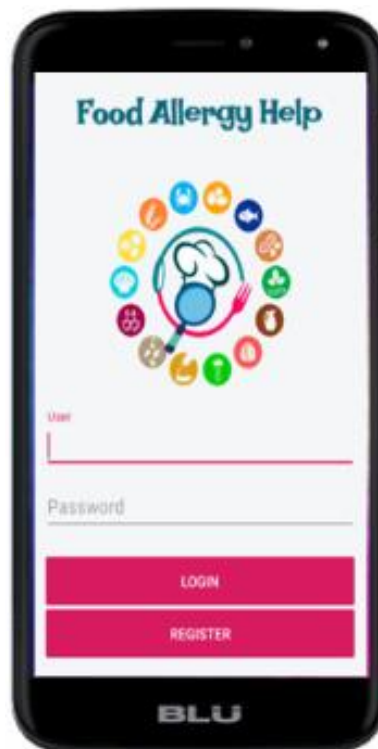


Ilustración 84. Pantalla Login

Si el usuario ya ha sido registrado, basta con ingresar el nombre de usuario y contraseña en los campos “**User**” y “**Password**” respectivamente, y presionar el botón “**LOGIN**”. Si se desea registrar un nuevo usuario hay que presionar el botón “**REGISTER**”.

Desde el menú de todas las ventanas de la aplicación se puede cerrar sesión, con lo cual se borrará la sesión actual y se direccionará la aplicación a esta ventana.

Cabe recalcar que la información proporcionada por esta app está basada en datos públicos consultados en la web por lo que sólo sirven de ayuda o guía, sin embargo, no ofrecen una garantía médica. Este aviso informativo (Ilustración 85) se presentará cuando el usuario ingrese correctamente su usuario y contraseña.

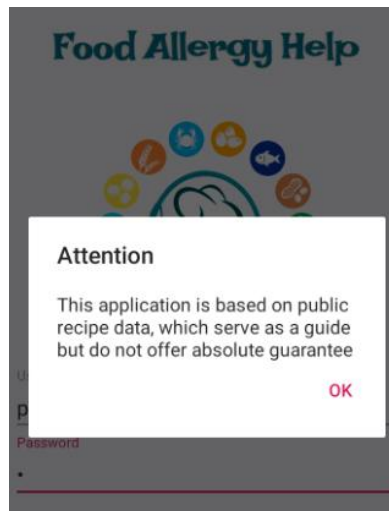


Ilustración 85. Dialogo informativo de uso de datos públicos

Por el momento, la contraseña en Firebase no ha sido encriptada, pero en caso de querer comercializar esta aplicación será necesario realizar el proceso de encriptación para proteger la información.

8.2.3 User Register

Este formulario (Ilustración 86) permite ingresar los datos de un nuevo usuario y el listado de sus alérgenos.

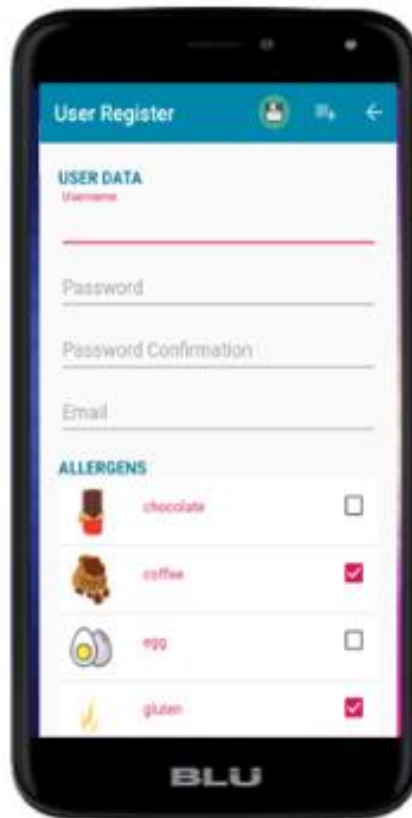


Ilustración 86. Pantalla User Register




| Botón | Nombre | Función |
|---|--------------|---|
|  | Save user | Guarda los datos del Usuario Registrado |
|  | Add allergen | Añade un nuevo alérgeno al listado |
|  | Back | Retrocede a la pantalla anterior. |

Ilustración 87. Botones de la pantalla User Register

La Ilustración 87 muestra los botones de este formulario junto con las funcionalidades de cada uno.

Existe un conjunto de alérgenos por defecto (Ilustración 88) que pueden ser escogidos, estos son los que afectan más usualmente a las personas.















| | | |
|---|-----------|--------------------------|
|  | chocolate | <input type="checkbox"/> |
|  | coffee | <input type="checkbox"/> |
|  | egg | <input type="checkbox"/> |
|  | gluten | <input type="checkbox"/> |
|  | milk | <input type="checkbox"/> |
|  | peanut | <input type="checkbox"/> |
|  | sesame | <input type="checkbox"/> |
|  | fish | <input type="checkbox"/> |
|  | soy | <input type="checkbox"/> |
|  | sulfite | <input type="checkbox"/> |
|  | tree nut | <input type="checkbox"/> |
|  | wheat | <input type="checkbox"/> |

Ilustración 88. Listado de alérgenos por defecto

Existe la posibilidad de añadir un nuevo alérgeno al presionar el segundo botón de la barra de tareas “**Add allergen**”, con lo que se despliega un cuadro de diálogo (Ilustración 89) que facilita el ingreso del nuevo alérgeno que se añadirá al final de la lista de alérgenos.

New Allergy

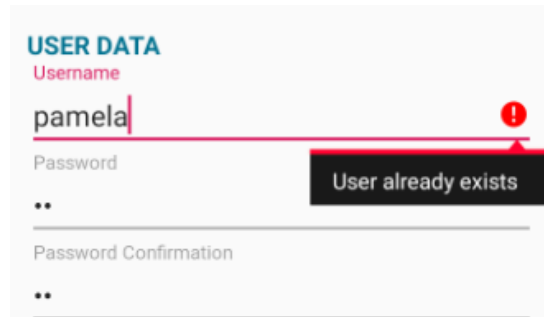
Enter the ingredient

CANCEL OK

Ilustración 89. Cuadro para ingreso de nuevo alérgeno

Para guardar los datos del nuevo usuario debe seleccionar el botón “**Save user**”.

En caso de que ya exista el usuario o que las contraseña y su confirmación no coincidan se presentará un aviso de error (Ilustración 90) en el propio cuadro de texto de ingreso de los datos.



The screenshot shows a form titled "USER DATA" with three input fields: "Username", "Password", and "Password Confirmation". The "Username" field contains the text "pamela" and has a red exclamation mark icon to its right. A red line connects this icon to a black error message box that says "User already exists". The "Password" and "Password Confirmation" fields are currently empty and masked with dots.

Ilustración 90. Presentación de errores al ingresar un usuario

8.2.4 Dashboard

Una vez se inicie sesión, se direccionará la aplicación a la ventana principal (Ilustración 91) desde la cual se puede acceder a las principales funcionalidades de la aplicación:

- **User Profile:** Consulta o modificación de los datos del usuario y sus alérgenos.
- **Recipes Search:** Consulta y análisis de recetas alimenticias.
- **Scan:** Escaneo, traducción y análisis de texto.
- **Upload:** Carga, traducción y análisis de imágenes.
- **Sign out:** Cierra la sesión del usuario actual.

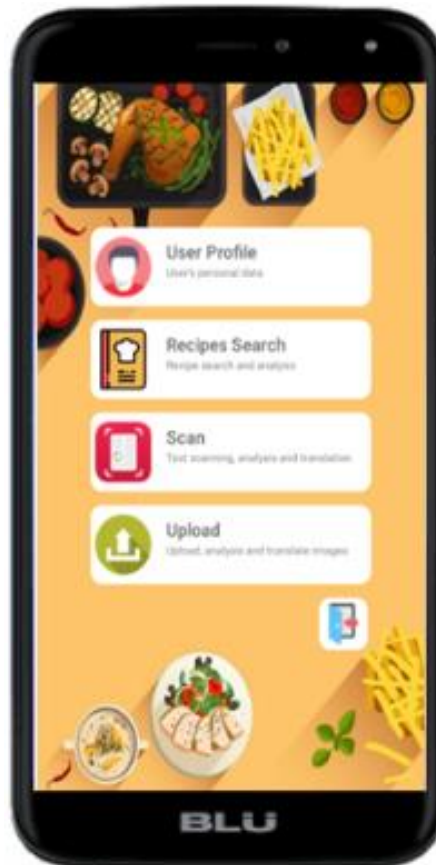


Ilustración 91. Pantalla Dashboard

8.2.5 User Profile

Es muy similar a la ventana “**User Register**”, con la diferencia de que en esta se mostrarán los datos del usuario que ha iniciado sesión en la aplicación y los alérgenos que podrían provocar reacciones en él. En este formulario (Ilustración 92) se pueden modificar los datos del usuario y añadir o quitar alérgenos de su listado.

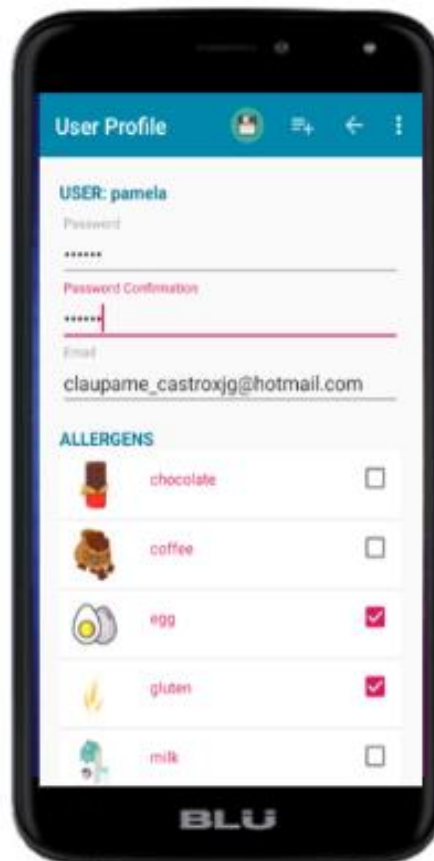


Ilustración 92. Pantalla User Profile




| Botón | Nombre | Función |
|---|--------------|---|
|  | Save User | Guarda los datos del Usuario Registrado |
|  | Add Allergen | Añade un nuevo alérgeno |
|  | Back | Retrocede a la pantalla anterior. |

Ilustración 93. Botones de la pantalla User Profile

Se utiliza el botón **“Add allergen”** para añadir un nuevo alérgeno al usuario actual y para guardar los cambios realizados se debe presionar el botón **“Save user”**.

8.2.6 Recipes Search

Permite realizar una búsqueda de recetas en base a los filtros (Ilustración 94) que el usuario especifique y, a su vez, realizar un análisis de detección de alérgenos por nombres de platos o palabras clave.

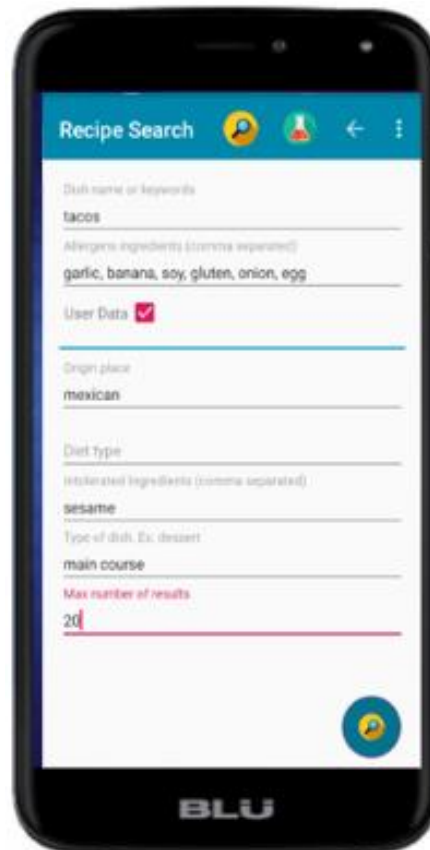


Ilustración 94. Pantalla Recipe Search




| Botón | Nombre | Función |
|---|-----------------|---|
|  | Search recipes | Busca un grupo de recetas en base a los filtros ingresados |
|  | Analyze recipes | Realiza un análisis de los ingredientes de un grupo de recetas cuyo nombre o <i>Keyword</i> sea el indicado por el usuario. |
|  | Back | Retrocede a la pantalla anterior |

Ilustración 95. Botones de la pantalla Recipe Search

Los filtros para realizar una búsqueda de recetas pueden ser los siguientes:

- **Dish name o keywords:** Palabras clave o el nombre de un plato.
- **Allergens ingredients:** Lista separada por comas de alérgenos del usuario. Por defecto se cargan los ingredientes alérgenos del usuario actual, aunque existe la posibilidad de eliminar aquellos o agregar unos nuevos. Debe hacer doble clic sobre este campo para que se despliegue el cuadro de diálogo (Ilustración 96) con el listado de alérgenos posibles. Cuando la casilla de selección **User Data** es seleccionada se cargan los alérgenos del usuario y si se quita la selección se borrarán todos los alérgenos para que el usuario ingrese unos específicos de la búsqueda o análisis que vaya a realizar.

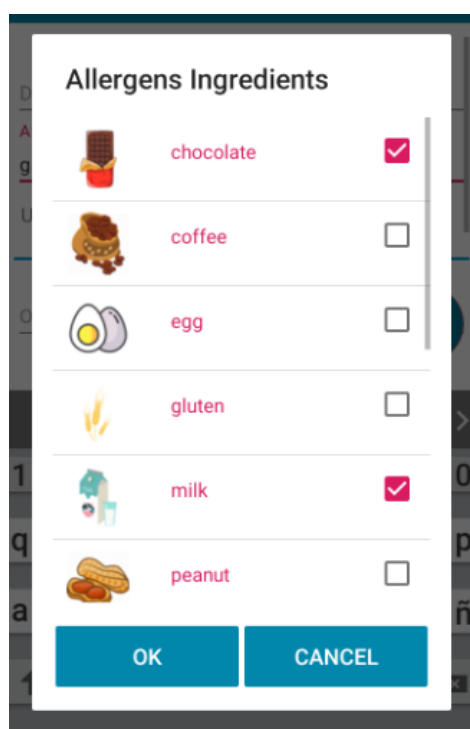


Ilustración 96. Cuadro de diálogo de listado de alérgenos

- **Origin place:** Permite establecer como filtro el lugar de origen del plato que se quiere consultar, puede ser: african, american, british, cajun, caribbean, chinese, entre otros. Al igual que en los ingredientes alérgenos, también se puede abrir un cuadro de diálogo (Ilustración 97) que muestre las distintas opciones a escoger al hacer doble clic en el cuadro de texto.



Origin Place

| | |
|--------------------------|----------|
| <input type="checkbox"/> | none |
| <input type="checkbox"/> | african |
| <input type="checkbox"/> | american |
| <input type="checkbox"/> | british |
| <input type="checkbox"/> | cajun |

Ilustración 97. Cuadro de diálogo de lugares de origen de recetas

- **Diet type:** Tipo de dieta en particular. Puede ser: vegan, vegetarian, pescetarian, lacto vegetarian y ovo vegetarian.
- **Intolerated ingredients:** Lista separada por comas de ingredientes a los que se tenga intolerancia.
- **Type of dish:** Tipo de plato a consultar. Puede ser: appetizer, beverage, bread, breakfast, dessert, drink, main course, salad, sauce, side dish o soup.
- **Max number of results:** Número máximo de recetas a consultar.

Para buscar recetas que cumplan con los filtros indicados hay que presionar el botón “**Search recipes**” de la barra de tareas o aquel que se encuentra flotando en la parte inferior derecha de la ventana. En este tipo de búsqueda el único dato obligatorio es “**Dish name o keywords**”.

Para hacer un análisis de los ingredientes que contiene un plato en particular, se debe ingresar su nombre en el campo “**Dish name or keywords**” y los alérgenos en “**Allergens ingredients**”, y presionar el botón “**Analyze recipes**”

En los siguientes apartados se detallas estas 2 opciones.

8.2.7 Recipes

Despliega un listado de recetas (Ilustración 98) que cumplan con los filtros previamente indicados por el usuario. Cada receta mostrará su fotografía, nombre, duración de preparación (en minutos) y las porciones.

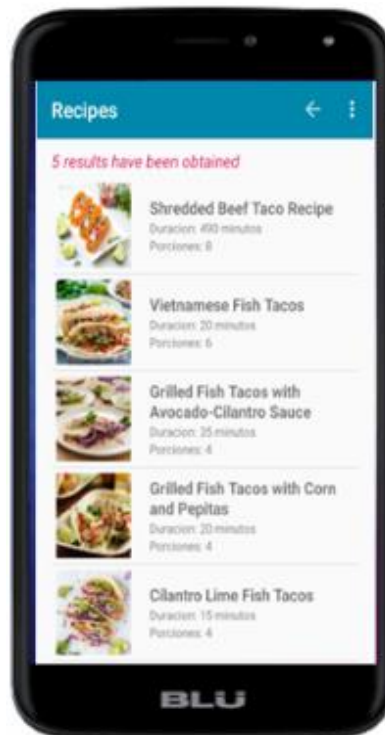


Ilustración 98. Pantalla Recipes

Para tener un mayor detalle de una de las recetas de la lista hay que hacer clic sobre ella y se abrirá una nueva ventana con mayor información (Ilustración 99), indicando la cantidad e ingredientes que necesita, así como también el proceso paso a paso para prepararla.



Ilustración 99. Pantalla Recipe Detail

8.2.8 Recipes Analysis

Realiza un análisis de los ingredientes de un grupo de recetas que tengan el nombre que el usuario indique en el campo **“Dish name or keywords”**, tomando en cuenta los alérgenos establecidos, de manera que pueda informar si se han detectado los alérgenos del usuario como ingredientes de las recetas.

El número de recetas a analizar se indica en el campo **“Max number of results”** de la ventana **“Recipe Search”**, en caso de no indicar ningún número en este atributo el valor por defecto son 5 recetas a analizar.

Si los ingredientes de las recetas indican que contienen alguno de los alérgenos, se mostrará una ventana (Ilustración 100) alertando detalladamente las razones en base a un análisis de cada alérgeno.



Ilustración 100. Pantalla Recipes Analysis

En la Ilustración 100, se muestra un ejemplo que indica que se han analizado 5 recetas cuyo nombre contenga la palabra **“tacos”**. Vemos como resultado que el **80%** de ellas contienen el alérgeno **“onion”** (cebolla), el **40%** tienen **“garlic”** (ajo), por lo cual hay que tener precaución si se desea consumir ese plato.

En caso de querer conocer a mayor detalle el reconocimiento de un alérgeno en las recetas analizadas, basta con hacer clic sobre su nombre y se desplegará una nueva ventana (Ilustración 101) con los nombres similares encontrados como ingredientes en las recetas consultadas y el número de recetas en las que aparecen.

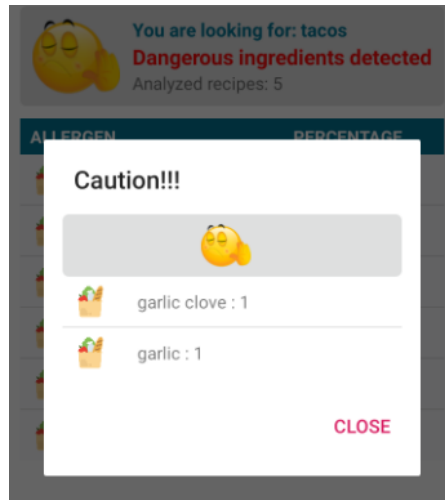


Ilustración 101. Alerta de precaución en análisis de recetas

En caso de que las recetas analizadas no contengan ninguno de los alérgenos se presentará un mensaje indicando aquello, como se puede ver en la Ilustración 102.



Ilustración 102. Mensaje de no detección de alérgenos



8.2.9 Scan

Permite realizar un escaneo, en tiempo real, de una etiqueta, receta o cualquier otro texto, con el objetivo de realizar un reconocimiento del texto contenido y efectuar una traducción o análisis para identificar posibles alérgenos (Ilustración 103).



Ilustración 103. Pantalla Scan

| Botón | Nombre | Función |
|-------|----------------|---|
| | Scan | Inicia el escaneo del texto. En la parte inferior de la ventana se puede visualizar el texto detectado. |
| | Translate text | Detiene el escaneo y traduce el texto detectado al idioma que se especifique. |
| | Analyze text | Detiene el escaneo y realiza un análisis de los ingredientes encontrados en el texto detectado. |
| | Back | Retrocede a la pantalla anterior. |

Ilustración 104. Botones de la pantalla Scan

Para iniciar el escaneo se debe presionar el botón **“Scan”**, el cual permite enfocar el móvil sobre un texto y que su contenido sea reconocido y mostrado en la parte inferior de la ventana de la aplicación. Esto se puede observar en la Ilustración 105.

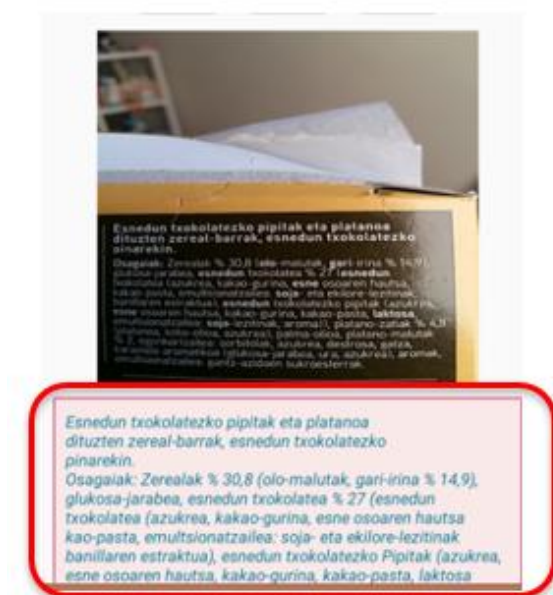


Ilustración 105. Escaneo y reconocimiento de texto

El idioma origen para la traducción puede ser auto detectado por la aplicación o escogido por el usuario. El idioma al que se desea se traduzca el texto debe ser obligatoriamente especificado. Estos 2 atributos pueden ser configurados en la sección **“Translation”** mostrada en la Ilustración 106.

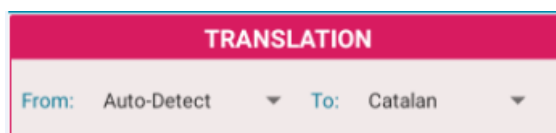


Ilustración 106. Especificación de idiomas origen y destino en escaneo

Los idiomas de origen y destino pueden ser:

- Catalán
- Inglés
- Francés
- Alemán
- Italiano
- Japonés

- Coreano
- Ruso
- Español

Una vez escogidos los idiomas origen y destino de la traducción, se debe presionar el botón **“Translate text”** para parar el escaneo y poder verificar si el texto reconocido es el adecuado. Si el texto reconocido es correcto, basta con volver a presionar el botón **“Translate text”** para efectuar la traducción respectiva.

Si se desea efectuar un análisis del contenido del texto escaneado, no es necesario especificar los idiomas origen y destino, ya que internamente se realizará todo este proceso.

Al presionar el botón **“Analyze text”** se detendrá el proceso de escaneo y se analizará todo el texto detectado con el objetivo de identificar posibles alérgenos. En caso de encontrar alérgenos en el texto, la aplicación alertará (Ilustración 107) al usuario detallando las secciones o bloques en donde se localicen.

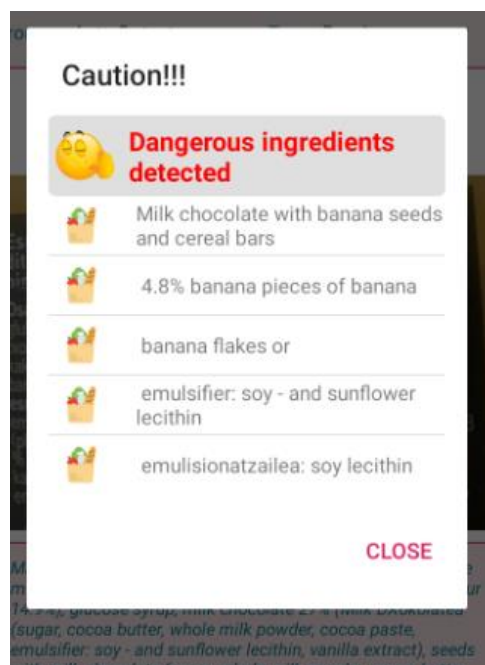


Ilustración 107. Mensaje de Precaución en la pantalla Scan

En caso de no encontrar alérgenos en el texto, también se informará de ello mediante un cuadro de diálogo (Ilustración 108).

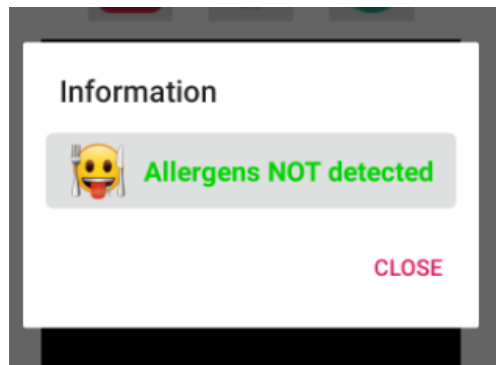


Ilustración 108. Mensaje de no detección de alérgenos en Scan

8.2.10 Upload

Permite realizar fotografías, cargarlas desde la galería, traducir el texto detectado en las imágenes tomadas o descargadas y realizar un análisis para identificación de alérgenos. Este formulario es similar al de “Scan” (Ilustración 103), con la diferencia de que en lugar de escanear en tiempo real un texto, carga una imagen desde la galería (Ilustración 109).



Ilustración 109. Pantalla Upload








| Botón | Nombre | Función |
|---|----------------|---|
|  | Upload image | Abre la ventana de Galería del móvil para poder selección la imagen a analizar. |
|  | Take a picture | Enciende la cámara del móvil para poder tomar una fotografía. |
|  | Translate text | Detiene el escaneo y traduce el texto detectado |
|  | Analyze text | Realiza un análisis de los ingredientes encontrados en el texto detectado. |
|  | Back | Retrocede a la pantalla anterior. |

Ilustración 110. Botones de la pantalla Upload

Para realizar la carga de una imagen desde a galería se debe presionar el botón **“Upload image”**, que abrirá la galería del móvil para permitir que el usuario pueda escoger la fotografía o imagen que desee se traduzca o analice.

En caso de aun no tener la imagen en galería y se desee realizar una fotografía de un texto, basta con presionar el botón **“Take a picture”**, el cual enciende la cámara fotográfica del móvil para poder realizar la captura y, posteriormente, cargar esa imagen a la aplicación.

En la sección **“Translation”** de la ventana (Ilustración 111) se muestran los filtros de traducción, en donde **“From”** indica el idioma origen de la traducción y **“To”** establece el idioma destino.

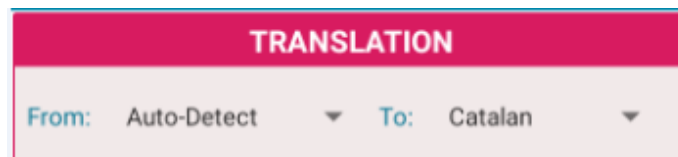


Ilustración 111. Especificación de idiomas de traducción en carga de imágenes

Para traducir el texto identificado en la imagen se debe presionar el botón **“Translate text”** y en la parte inferior de la ventana se mostrará el texto traducido al idioma que establezca el usuario, tal y como visualizamos en la Ilustración 112.



Ilustración 112. Traducción de texto

Los idiomas de origen y destino pueden ser:

- Catalán
- Inglés
- Francés
- Alemán
- Italiano
- Japonés
- Coreano
- Ruso
- Español

Si se desea efectuar un análisis del contenido del texto detectado con el objetivo de identificar posibles alérgenos, se debe presionar el botón **“Analyze text”**. En caso de encontrar alérgenos en el texto, la aplicación alertará al usuario (Ilustración 113) detallando las secciones o bloques en donde se encuentren.

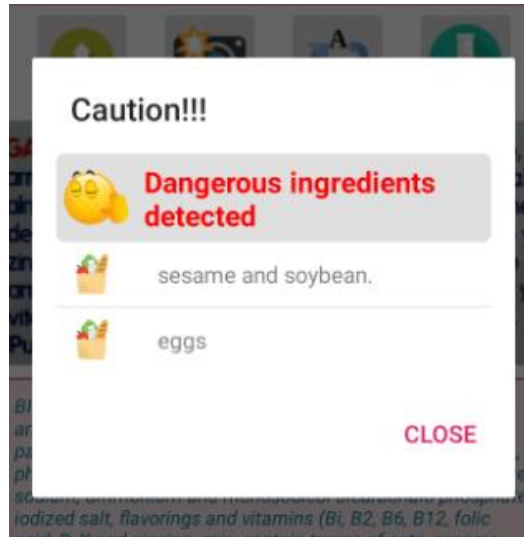


Ilustración 113. Mensaje de precaución en pantalla Upload

En caso de no encontrar alérgenos en el texto identificado, también se informará de aquello (Ilustración 114).

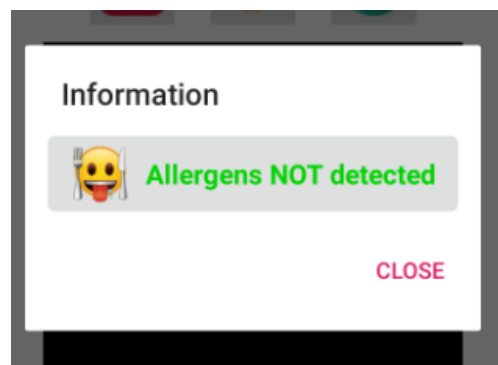


Ilustración 114. Mensaje de no detección en pantalla Upload

9. Casos de Uso

Con el objetivo de proporcionar un mayor entendimiento de las circunstancias en las cuales la aplicación **Food Allergy Help** sería de gran uso, se ha propuesto algunos ejemplos.


9.1 Búsqueda de Recetas (Caso de uso 1)

Juan tiene alergia al ajo, mostaza y al vino. Además, tiene intolerancia al gluten. Se encuentra en su hogar y desea preparar un plato llamado “Filet Mignon” para una cena especial con su familia. Desea una receta que no incluya los alérgenos e intolerancias que tiene, por ello procede a ejecutar una búsqueda de recetas con **Food Allergy Help** realizando los siguientes pasos:

- a. Abrir la aplicación y en la ventana **“Login”** (Ilustración 115) y presionar el botón **“REGISTER”**



Ilustración 115. Registro de Usuario

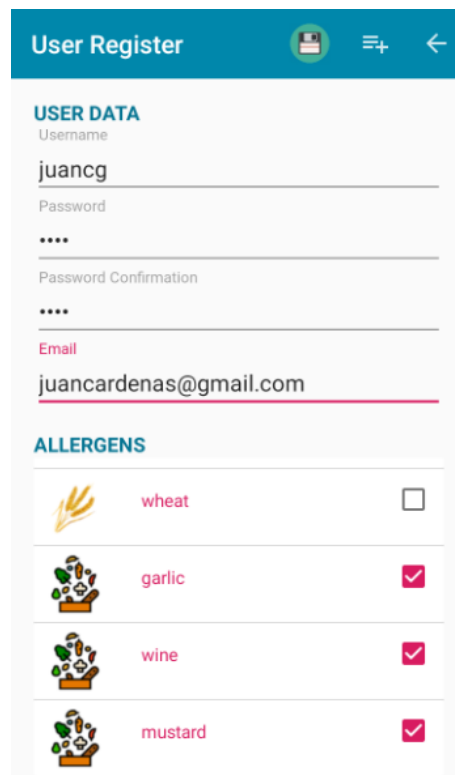
- b. En la ventana de registro de usuario ingresar los datos del usuario. Por defecto existen algunos alérgenos definidos, pero ninguno es el que afecta a Juan, por esta razón debe presionar el botón **“Add ingredient”**  e ingresar cada uno de los alérgenos. En este caso, los alérgenos peligrosos son ajo, mostaza y vino. La aplicación trabaja con alérgenos del usuario descritos en inglés, por lo que los valores a ingresar son *garlic*, *wine* y *mustard*, como vemos en la Ilustración 116.



The image shows three separate dialog boxes, each titled "New Allergy". The first dialog has the text "garlic" in a pink underline, with "CANCEL" and "OK" buttons below it. The second dialog has the text "wine" in a pink underline, with "CANCEL" and "OK" buttons below it. The third dialog has the text "mustard" in a pink underline, with "CANCEL" and "OK" buttons below it.

Ilustración 116. Ingreso de nuevos alérgenos. Caso de uso 1

El formulario, previo al almacenamiento de los datos, quedaría similar a lo que se muestra en la Ilustración 117.



The image shows a "User Register" form. The title bar is blue with a back arrow, a menu icon, and a user icon. The form is divided into two sections: "USER DATA" and "ALLERGENS".

USER DATA

- Username: juancg
- Password: ****
- Password Confirmation: ****
- Email: juancardenas@gmail.com

ALLERGENS





| | | |
|---|---------|-------------------------------------|
|  | wheat | <input type="checkbox"/> |
|  | garlic | <input checked="" type="checkbox"/> |
|  | wine | <input checked="" type="checkbox"/> |
|  | mustard | <input checked="" type="checkbox"/> |

Ilustración 117. Registro de usuario con nuevos alérgenos

- c. Ingresar los campos **“User”** y **“Password”** que corresponden con el usuario y contraseña del usuario registrado en el paso previo. Presionar el botón **“LOGIN”** como se ve en la Ilustración 118.



Ilustración 118. Inicio de sesión, Caso de uso 1

- d. Recibirá un aviso informando que la aplicación está basada en datos públicos de recetas, se debe presionar el botón “OK”

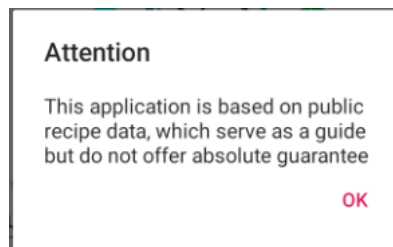


Ilustración 119. Mensaje informativo

- e. De la lista de opciones desplegada en la Ilustración 120, presionar la segunda opción que es “**Recipes Search**”.

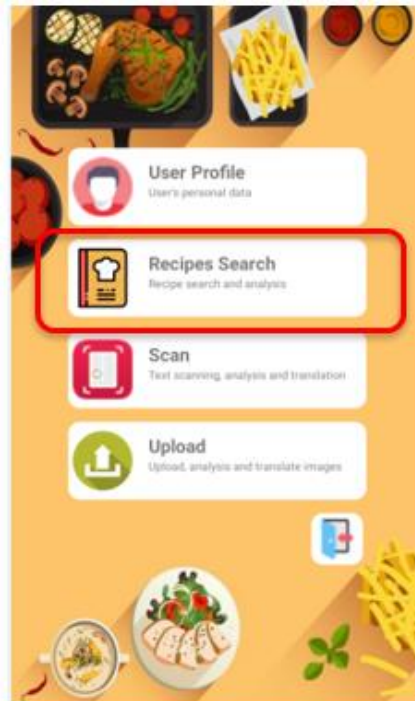


Ilustración 120. Selección de búsqueda de recetas, Caso de uso 1

- f. Ingresar los filtros necesarios. El campo **“Allergens Ingredients”** se cargará automáticamente en base a los alérgenos del usuario. En este caso se consultará como máximo 10 recetas, y esto se indica en el atributo **“Max number o results”**. Presionar el botón flotante **“Search Recipes”** mostrado en la Ilustración 121, este efectuará la búsqueda de recetas de acuerdo a los atributos indicados en sus campos.

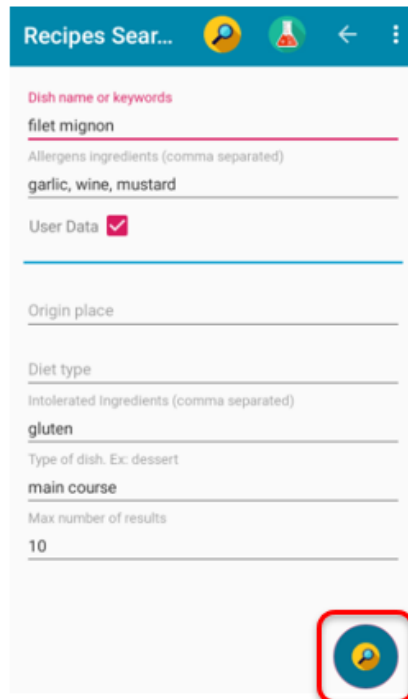


Ilustración 121. Búsqueda de receta, Caso de uso 1

- g. Se desplegará un listado de recetas (Ilustración 122) que cumplen con los filtros establecidos. Al hacer clic sobre una de ellas se mostrará más detalladamente los ingredientes y el proceso para realizar la receta (Ilustración 123).

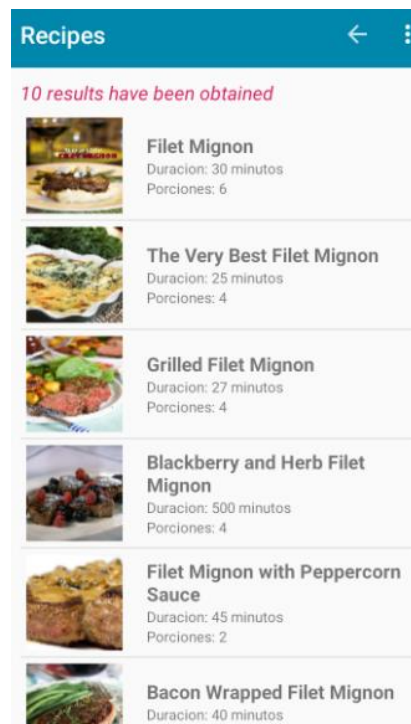


Ilustración 122. Listado de recetas, Caso de uso 1



Ilustración 123. Detalle de la receta, Caso de uso 1

9.2 Análisis del contenido de un plato (Caso de uso 2)

Kelly es una ciudadana estadounidense que está de vacaciones en Perú. Le han recomendado un plato típico llamado “Ceviche”. Ella tiene alergia a la cebolla, tomate, gambas, café, y kiwi. Desea analizar el plato que le han recomendado con el objetivo de realizar una detección de los alérgenos que le causan daño. Los pasos que debe llevar a cabo son los descritos a continuación:

- a. Abrir la aplicación y en la ventana “**Login**” presionar el botón “**REGISTER**”, como lo vemos en la Ilustración 124.

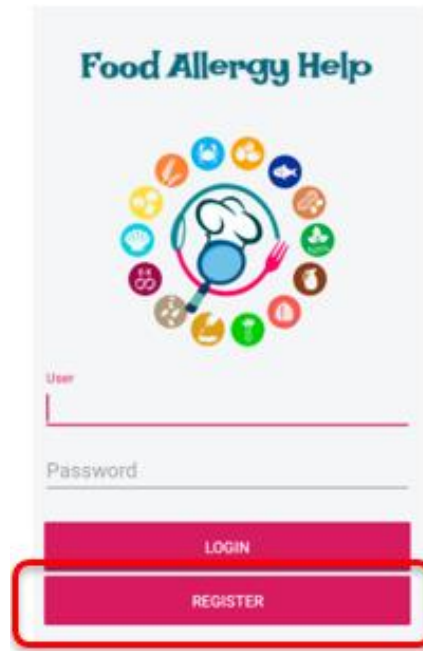


Ilustración 124. Registrar usuario, Caso de uso 2


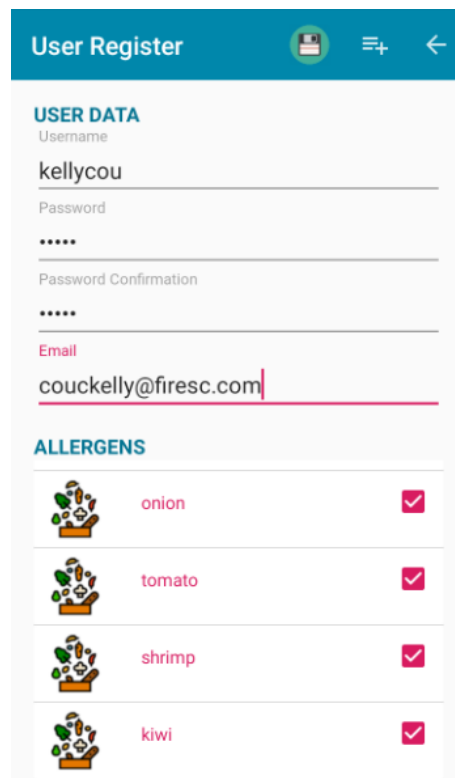
- b. En la ventana de registro de usuario, se debe ingresar los datos del usuario. Por defecto existen algunos alérgenos definidos como por ejemplo el café, los otros alérgenos que afectan a Kelly deben ser agregados, por esta razón se debe presionar el botón “**Add ingredient**”  e ingresar cada uno de los alérgenos.



Ilustración 125. Inserción de nuevos alérgenos, Caso de uso 2

El formulario quedaría similar al siguiente:



| USER DATA | | |
|-----------------------|----------------------|--|
| Username | kellycou | |
| Password | ***** | |
| Password Confirmation | ***** | |
| Email | couckelly@firesc.com | |





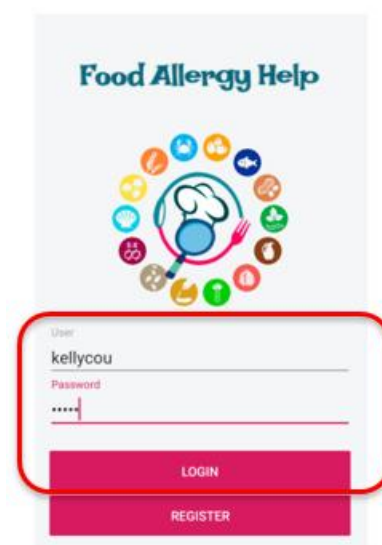
| ALLERGENS | | |
|--|--------|-------------------------------------|
|  | onion | <input checked="" type="checkbox"/> |
|  | tomato | <input checked="" type="checkbox"/> |
|  | shrimp | <input checked="" type="checkbox"/> |
|  | kiwi | <input checked="" type="checkbox"/> |

Ilustración 126. Registro de nuevo usuario y alérgenos, Caso de uso 2

- c. Ingresar los campos “User” y “Password” que corresponden con el nombre de usuario y contraseña registrados en el paso previo. Presionar el botón “LOGIN”.



Food Allergy Help

User
kellycou

Password

LOGIN

REGISTER

Ilustración 127. Inicio de sesión, Caso de uso 2



- d. Recibirá un aviso (Ilustración 128) informando que la aplicación está basada en datos públicos de recetas, se debe presionar el botón “OK”

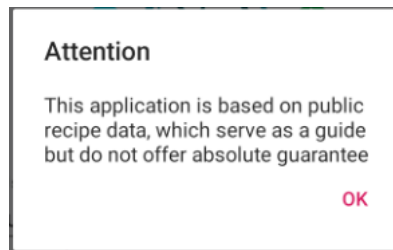


Ilustración 128. Mensaje informativo, Caso de uso 2

- e. En el *Dashboard* desplegado (Ilustración 129), presionar la segunda opción que es “**Recipes Search**”.

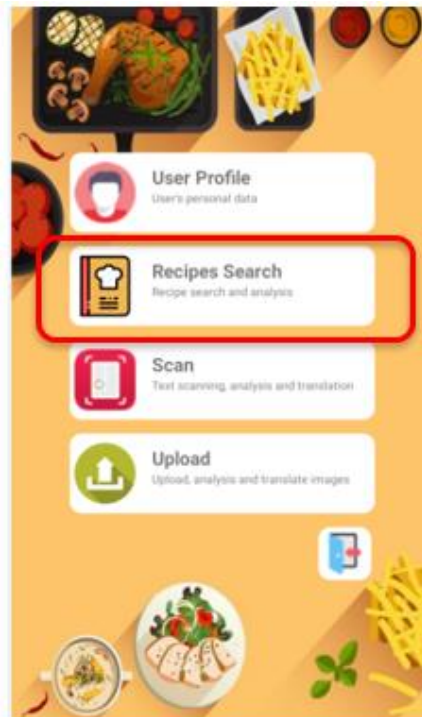


Ilustración 129. Selección de Búsqueda de recetas, Caso de uso 2

- f. El campo “**Allergens Ingredients**” se cargará automáticamente con los alérgenos del usuario. En el campo “**Dish name or keyword**” (Ilustración 130) ingresar el nombre del plato a consultar. En este caso se analizará como máximo 10 recetas, y esto se indica en el atributo “**Max number o results**”. Presionar el botón flotante “**Analyze Recipes**”.

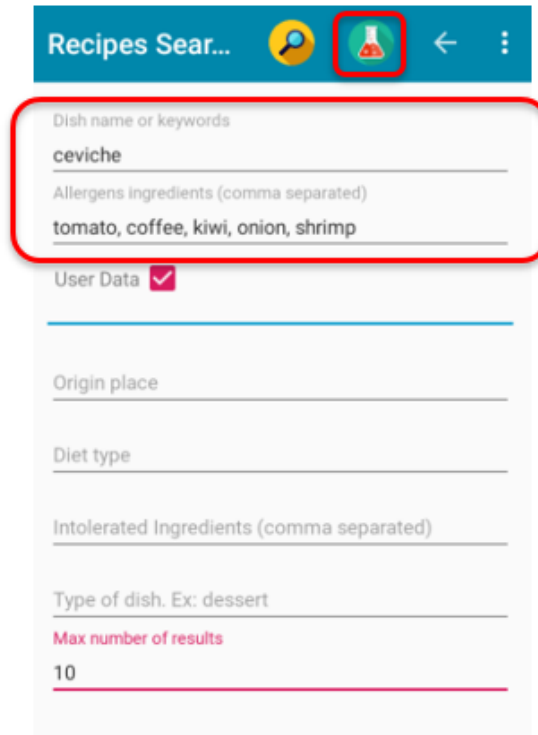


Ilustración 130. Ingreso de valores previos a análisis de receta

- g. Si se detectan alérgenos en las recetas consultadas, se abrirá una ventana que alerta al usuario (Ilustración 131). En este caso se visualiza que solo hay 8 recetas cuyo nombre o palabra clave es “ceviche”. En análisis de los ingredientes se basan en las 8 recetas encontradas.



| ALLERGEN | PERCENTAGE |
|--|------------|
|  tomato | 62.0% |
|  coffee | 0.0% |
|  kiwi | 0.0% |
|  onion | 100.0% |
|  shrimp | 50.0% |

Ilustración 131. Análisis de receta, Caso de uso 2

Aquellos alérgenos que aparecen en color rojo son los que, según el análisis de la aplicación, representan un riesgo para el usuario.

Si se desea ver a más detalle el nombre de los ingredientes encontrados que son similares a uno de los alérgenos, se debe hacer clic sobre el nombre del alérgeno y se desplegará una ventana (Ilustración 132) en donde se muestran los hallazgos junto con el número de recetas en las que aparece.

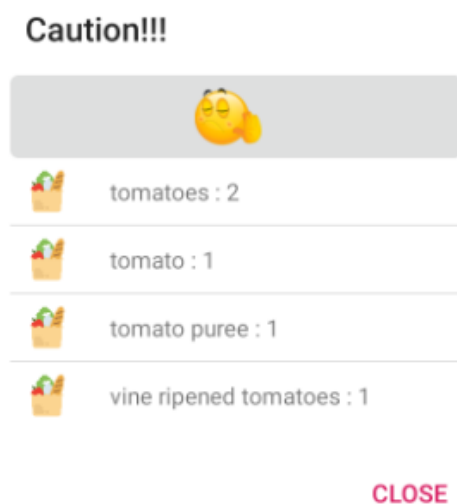


Ilustración 132. Detalle de análisis de receta, Caso de uso 2

9.3 Análisis de un texto (Caso de uso 3)

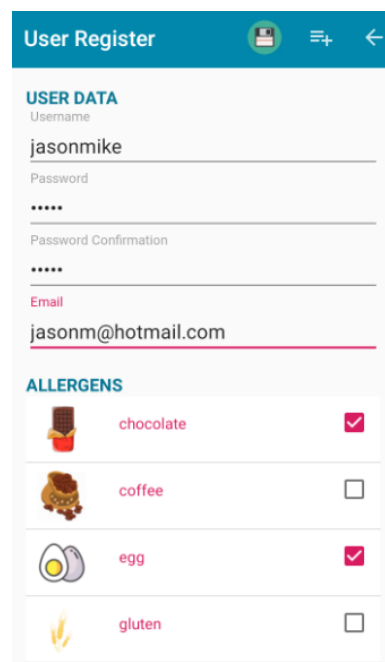
Jason es un filipino - japonés que tiene alergia al chocolate, huevo, leche y soja. Él se encuentra fuera de su país, en un supermercado en Italia y desea comprar unas galletas. Además, quiere ver la traducción a inglés de las etiquetas de algunos empaques y analizar los ingredientes para detectar la presencia de alérgenos en los mismos. Para ello debe hacer lo siguiente:

- a. Abrir la aplicación. En la ventana “**Login**” (Ilustración 133) presionar el botón “**REGISTER**”



Ilustración 133. Registrar usuario, Caso de uso 3

- b. En la ventana de registro, debe ingresar los datos del usuario. Por defecto existen algunos alérgenos definidos, entre ellos están los que afectan a Jason (chocolate, egg, milk, soy) por lo que es necesario seleccionarlos (Ilustración 134).



| USER DATA | | |
|-----------------------|--------------------|--|
| Username | jasonmike | |
| Password | **** | |
| Password Confirmation | **** | |
| Email | jasonm@hotmail.com | |





| ALLERGENS | | |
|---|-----------|-------------------------------------|
|  | chocolate | <input checked="" type="checkbox"/> |
|  | coffee | <input type="checkbox"/> |
|  | egg | <input checked="" type="checkbox"/> |
|  | gluten | <input type="checkbox"/> |

Ilustración 134. Registro de usuario y alérgenos, Caso de uso 3



- c. Ingresar los valores de los campos “**User**” y “**Password**” que corresponden con el usuario y contraseña del usuario registrado en el paso previo. Presionar el botón “**LOGIN**”.

The image shows a login interface for 'Food Allergy Help'. At the top, the title 'Food Allergy Help' is displayed in blue. Below it is a circular logo with various food-related icons. The main form contains two input fields: 'User' with the value 'jasonmike' and 'Password' with masked characters. A red rounded rectangle highlights these two fields. Below the input fields are two buttons: 'LOGIN' and 'REGISTER'.

Ilustración 135. Inicio de sesión, Caso de uso 3

- d. Recibirá un aviso (Ilustración 119) informando que la aplicación está basada en datos públicos de recetas, se debe presionar el botón “**OK**” para ir a la ventana principal de la aplicación.
- e. En el *Dashboard* desplegado (Ilustración 136), presionar la última opción que es “**Upload**” para poder cargar imágenes.



Ilustración 136. Selección de carga de imágenes

- f. Se despliega la ventana de carga de imágenes (Ilustración 137). Para tomar la fotografía de un empaque debe presionar el botón **“Take a picture”**. Con esto abrirá la cámara fotográfica para capturar la imagen.

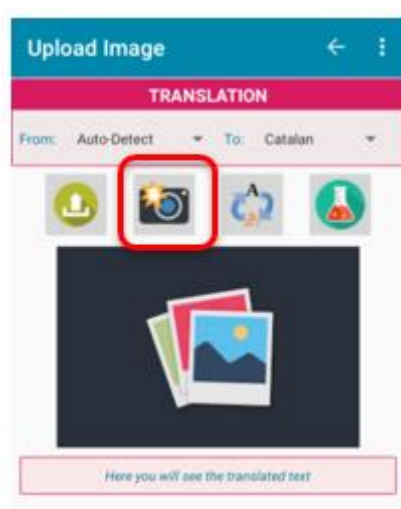


Ilustración 137. Captura fotográfica, Caso de uso 3

- g. Una vez tomada la fotografía, para cargarla a la aplicación debe presionar el botón **“Upload image”** (Ilustración 138), que abrirá la galería del móvil (Ilustración 139) para que el usuario pueda escoger la imagen tomada anteriormente.

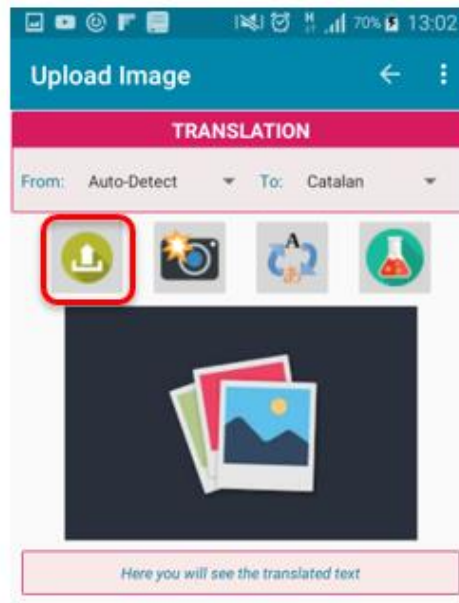


Ilustración 138. Carga de una imagen, Caso de uso 3



Ilustración 139. Selección de imagen de la galería

- h. Para traducir a algún idioma debe escoger el idioma destino del apartado “TRANSLATION” y presionar el botón “Translate text”. En la parte inferior de la ventana (Ilustración 140) visualizará el texto traducido a japonés.



Ilustración 140. Traducción de texto, Caso de uso 3

- i. Para realizar el análisis del producto debe hacer clic en el botón **“Analyze text”** y en caso de detectar posibles alérgenos, mostrará los bloques de texto en donde se los ha identificado. En la Ilustración 141 se puede visualizar lo importante que es que se muestren los bloques donde aparecen los posibles alérgenos ya que como podemos ver el producto **NO CONTIENE HUEVO**, pero si contiene los otros alérgenos.

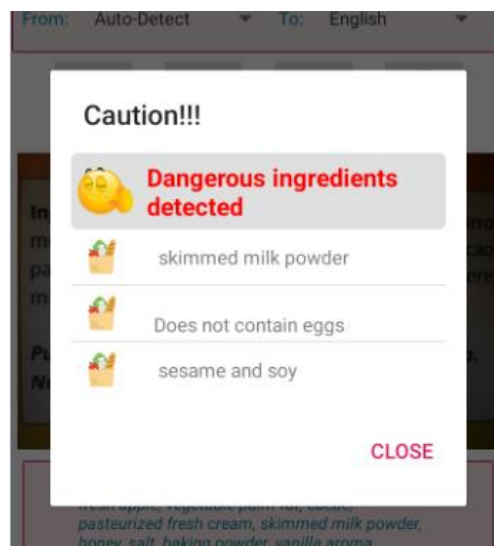


Ilustración 141. Mensaje de precaución, Caso de uso 3

10. Análisis de la competencia

Por competencia se entiende a aquellas aplicaciones que brindan iguales o similares funcionalidades que las que **Food Allergy Help** ofrece, es por ello que se han seleccionado un grupo de aplicaciones que, aunque no realizan exactamente las mismas funciones, realizan al menos una parecida a ellas. A continuación, se definen algunas:

- **¿Qué puedo comer?**



Trabaja con una base de datos de alrededor de 20.000 productos. Permite analizar los ingredientes de un determinado producto de acuerdo a su código de barras y, en base a los alérgenos que indique el usuario, se indica si el producto es apto para el consumo, pero no permite un escaneo de texto ni la traducción a distintos lenguajes. Adicionalmente, proporciona información nutricional del producto analizado. Como algo extra, esta aplicación tiene un apartado para acceder a recetas de cocina y promociones, pero direccionando hacia una página web, pero no realiza análisis del nombre de un plato en un conjunto de recetas para poder identificar la presencia de alérgenos.

- **Bueno para mi**

En base al código de barras de un producto comestible, muestra el listado de sus ingredientes, descripción de cada ingrediente, aditivos y función de cada aditivo. Guarda también un histórico de los alimentos consultados, facilita la búsqueda ingresando el nombre del producto a consultar y puede manejar varios perfiles de usuario con distintas restricciones. Pero no permite una búsqueda de alérgenos en varios idiomas, ni la detección de ellos de en diferentes formatos de texto, solamente lo hace a partir del código de barras.





- **Yum**



Proporciona más de 2 millones de recetas que pueden ser filtradas en base a gustos y preferencias dietéticas. Estas recetas pueden ser guardadas y organizadas, ofrece recomendaciones y también permite agregar los ingredientes de las recetas seleccionadas a una lista de compras. Pero no ofrece el gran beneficio de **Food Allergy Help** de analizar los platos buscando una estadística de la presencia de alérgenos en los mismos, y mucho menos la capacidad de escanear texto o cargar imágenes para detección de alérgenos en distintos idiomas.

- **Food Ingredients, Additives & E Numbers Scanner**

Por medio de un escaneo de la etiqueta que contiene los ingredientes de un producto, alerta sobre la presencia de aquellos que sean potencialmente dañinos, todo esto mediante una lista de diferentes colores ilustrativos. Ofrece detección de texto en varios idiomas con un costo adicional.

Pero no permite, por medio de la aplicación, realizar una toma fotográfica y carga de imágenes de la galería. Tampoco es posible consultar recetas en base a restricciones del usuario, ni realizar un análisis de las mismas.



En síntesis, existen algunas aplicaciones que realizan tareas similares a las anteriormente mencionadas, pero ninguna proporciona todas las facilidades que “FoodAllergyHelp” brinda. Un cuadro ilustrativo de resumen de las bondades y carencias de cada aplicación se muestra en la Ilustración 142.



| | Que puedo comer | Bueno para mi | Yum | Food ingredients |
|--------------------------------------|-----------------|---------------|-----|------------------|
| Consulta de recetas | ✓ | x | ✓ | x |
| Escaneo de códigos de barra | ✓ | ✓ | x | x |
| Escaneo de texto en fotografías | x | x | x | x |
| Escaneo de texto en tiempo real | x | x | x | ✓ |
| Análisis de ingredientes por escaneo | ✓ | ✓ | x | ✓ |
| Almacenamiento de datos del usuario | ✓ | ✓ | ✓ | ✓ |
| Traducción de texto | x | x | x | ✓ |
| Análisis de recetas | x | x | x | x |

Ilustración 142. Resumen de bondades y carencias de la competencia

11. Extensiones

Esta aplicación posee un gran potencial de expansión e incremento de sus funcionalidades. Una de las grandes fortalezas de **Food Allergy Help** es la posibilidad de evolucionar, de manera que el número de usuarios que puedan consumirla se amplíe y que la información que ella proporcione sea lo más precisa posible. Para ello se tiene pensado añadir la capacidad de que al momento de analizar recetas o textos buscando los alérgenos, previamente indicados, se busquen también sus sinónimos, ya que existen ciertos alimentos que en otros países o localidades que, aunque hablen el mismo idioma, se conocen con nombres distintos. También se podría ayudar al usuario al momento de escoger sus alérgenos, permitiéndole la posibilidad de seleccionar un grupo de alimentos o sus derivados, como es el caso de los lácteos y mariscos, que son los que más comúnmente causan alergias en las personas. Se considera que sería de gran ayuda el hecho de que el idioma base de uso de la aplicación pueda ser elegido por el usuario, lo cual ampliaría en gran manera el número final de consumidores que podrían hacer uso de la aplicación.

12. Explotación Comercial

Si se decide dar un uso comercial a la aplicación se podría aprovechar la capacidad de autenticación de Firebase para utilizar proveedores de identidad populares tales como Gmail, Twitter y Facebook para el inicio de sesión. Si el usuario escoge un inicio de sesión convencional, es decir, sin usar otros proveedores de identidad, sería necesario encriptar la contraseña del usuario para proporcionar mayor seguridad de la información.

En cuanto a costos de las plataformas utilizadas, la Base de Datos en tiempo real de Firebase basa su facturación en 2 aspectos: los datos que se almacenan en la base de datos y el tráfico de red saliente en la capa de sesión (capa 5) del modelo OSI. El costo del almacenamiento es de USD 5 por GB al mes. En lo que concierne al tráfico saliente, incluye el gasto de conexión y encriptación de todas las operaciones de la base de datos y los datos descargados a través de lecturas de la base de datos. Tanto la lectura como la escritura de la base de datos pueden generar costos de conexión en la factura.

Respecto a la detección de texto por medio de API de Cloud Vision de Firebase, si se requiere un plan para uso comercial, el costo es de USD 1.50 por cada 1000 consultas.

El costo de la traducción de texto de Google Cloud es de 20 USD por millón de caracteres. Por otro lado, en la detección de texto de Google, considera una imagen como una unidad facturable; la detección de las primeras 1000 unidades al mes son gratuitas, De 1001 a 5.000.000 unidades tienen un costo de USD 1.50 al mes por cada 1000 unidades, y de 5.000.001 a 20.000.000 unidades tiene un costo de USD 0.60 al mes por cada 1000 unidades.

Para consultar las recetas, RapidAPI con la API proporcionada por Spoonacular, ofrece algunos tipos de planes, desde el Pro (con un costo de 26 euros al mes, permite 500 peticiones al día, con 0,00360 euros por cada consulta adicional) hasta el Mega (con un valor de 899 euros al mes, 30000 peticiones al día, con un costo 0.00090 euros por cada consulta adicional). RapidAPI brinda un gran conjunto de APIs de recetas, que también podrían adaptarse a las necesidades de **Food Allergy Help**, y sus costos varían en cada una de ellas.

En un futuro se considera la posibilidad de añadir las características mencionadas en el apartado de **Extensiones** con el propósito de poder subir la aplicación a Play Store y empezar a comercializarla.

13. Conclusiones

Este Trabajo de Fin de Máster ha sido desarrollado en base a necesidades y vivencias propias, cuyo propósito principal es proporcionar al usuario una herramienta que permita detectar o alertar de una forma rápida, intuitiva y sencilla sobre la presencia de alérgenos en los alimentos que desea consumir, y con ello convertirse en un instrumento que mejore su calidad de vida.

Este proyecto cumple con todos los objetivos inicialmente planteados y ha permitido integrar distintas plataformas como Firebase y Google Cloud, que utilizan Inteligencia Artificial para proporcionar información de gran ayuda, como es el caso del reconocimiento de texto a partir de imágenes de la galería o en tiempo real. Las bondades que brindan estas plataformas son muy amplias y podría ser muy útiles para crear nuevos proyectos en un futuro.

Sin duda, la realización de este trabajo me ha dado grandes satisfacciones y ha establecido un mayor compromiso personal de crear y desarrollar nuevos proyectos para mejorar la vida de aquellas personas que sufren algún problema de salud; sin olvidar que, adicionalmente, me ha permitido profundizar y adquirir nuevos conocimientos y competencias muy importantes para mi profesión, llevándome a actualizarme acerca de las herramientas que están en auge y que contribuyen a mejorar la experiencia de desarrollo de software.

14. Referencias

ANDROID STUDIO. Aprende a desarrollar aplicaciones

José Dimas Luján Castillo, 2017

Desarrolle una aplicación Android. Programación en Java con Android Studio

Sylvain Hebuterne, 2018

Introducción a Android Studio

<https://developer.android.com/studio/intro?hl=es>

RapidAPI o cómo revolucionar la App Economy

<https://www.muypymes.com/2016/11/22/rapidapi-revolucionar-la-app-economy>

Documentación de Firebase

<https://firebase.google.com/docs>

Soluciones de Google Cloud

<https://cloud.google.com/solutions>

Google Cloud Vision API

<https://cloud.google.com/vision/?hl=es#section-1>

Visión general de OkHttp

<https://square.github.io/okhttp/>

Tutorial de Firebase RealtimeDatabase

<https://blog.mindorks.com/firebase-realtime-database-android-tutorial>

Detect Text from Image in Android with Google Mobile Vision API

<https://medium.com/teachmind/detect-text-from-image-in-android-with-google-mobile-vision-api-112a527e556d>



Firestore Text Recognition with Android

<https://medium.com/@abeythilakeudara3/firebase-text-recognition-with-android-e46814c49dfe>

CRUD en Firebase

<https://www.youtube.com/watch?v=35AY9YHfiSw&feature=youtu.be>

Glide

<https://github.com/bumptech/glide>

ML Kit for Firebase

<https://codelabs.developers.google.com/codelabs/mlkit-android/#0>

CITAS

ConSalud.es. (2018, 02 07). *ConSalud*. Retrieved from El Clínico San Carlos analiza las alergias más prevalentes: https://www.consalud.es/rss-portada/el-clinico-san-carlos-analiza-las-alergias-mas-prevalentes_46797_102.html

D'Amelio, M. J. (20 de 04 de 2019). *El Español*. Obtenido de El misterio de las alergias alimentarias: https://www.lespanol.com/ciencia/nutricion/20190420/misterio-alergias-alimentarias-vez-casos/391211714_0.html

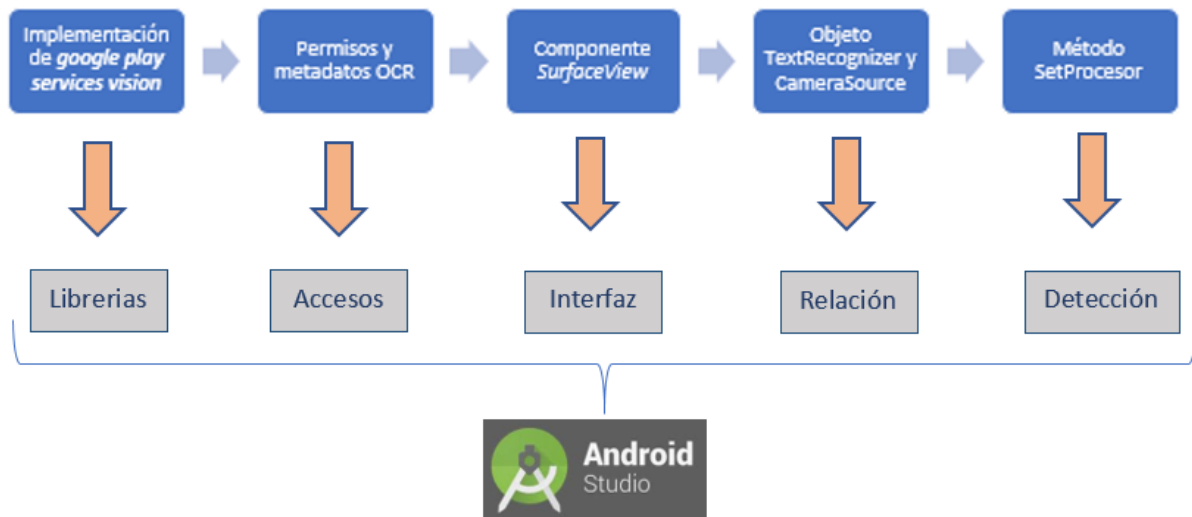
Expansión, Economía Digital. (1 de 2 de 2018). Obtenido de <https://www.expansion.com/economia-digital/innovacion/2018/02/01/5a72e73a22601db2288b4658.html>

JUSTE, M. (2018, 02 01). *Expansión, Economía Digital*. Retrieved from Las cifras de Internet en España: <https://www.expansion.com/economia-digital/innovacion/2018/02/01/5a72e73a22601db2288b4658.html>

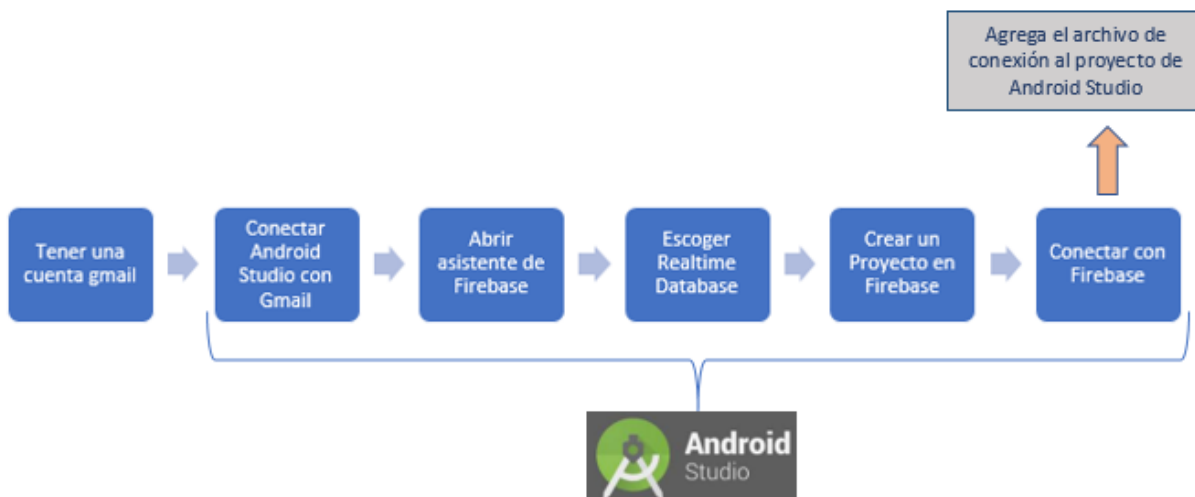
Perez, P. (2018). El uso de los móviles. *Revista la Vanguardia*, 3.

15. Anexos

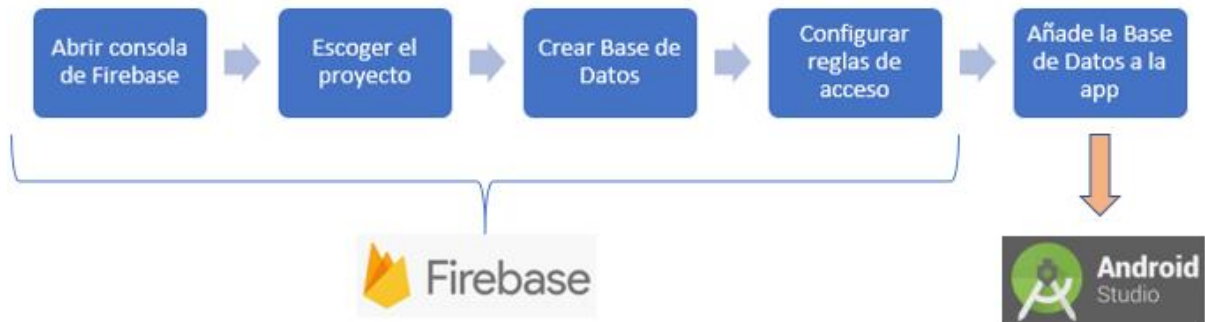
Anexo 1. Reconocimiento de texto en tiempo real



Anexo 2. Conexión a Firebase para almacenar datos



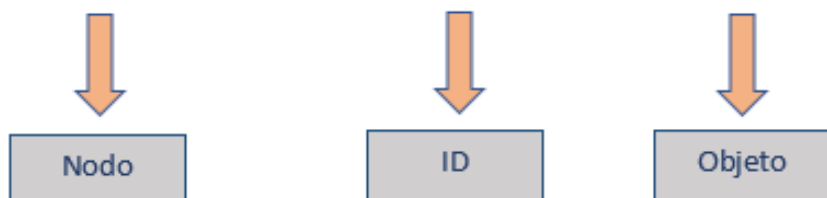
Anexo 3. Configuración para el almacenamiento de datos



Anexo 4. Almacenamiento de datos del usuario



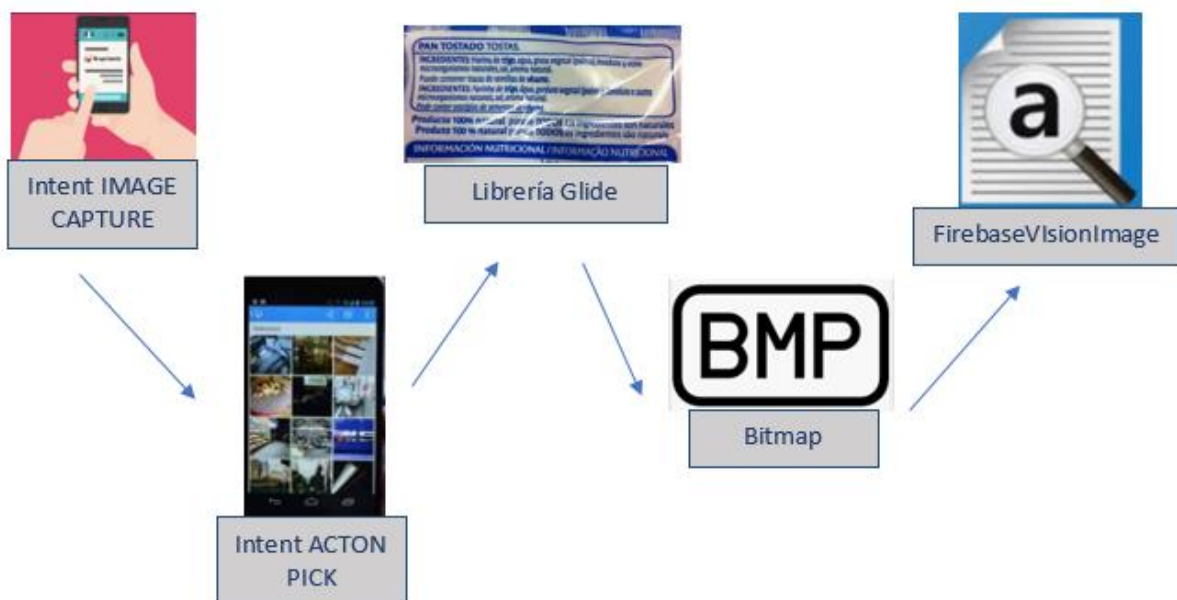
```
databaseReference.child("usuarios").child(user.getNombre()).setValue(user) ;
```



Anexo 5. Configuración para reconocimiento de texto con Firebase



Anexo 6. Reconocimiento de texto con Firebase



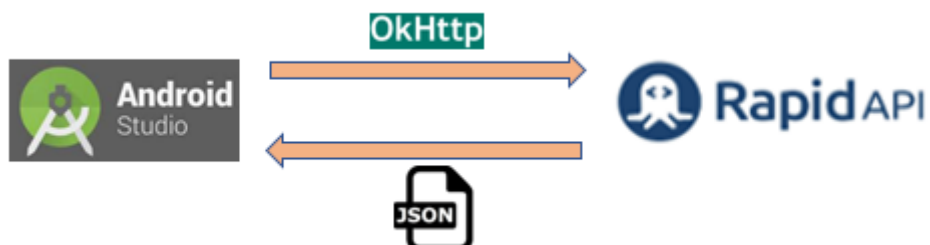
Anexo 7. Consulta de Recetas (Rapid API)



Anexo 8. Consulta de Recetas (Android Studio)



Anexo 9. Flujo para la Consulta de Recetas



Anexo 10. Traducción de textos (Google Cloud)



Anexo 11. Traducción de textos (Android Studio)

