



Universidad del Bío-Bío
Facultad de Ciencias Empresariales
Departamento de Computación y Tecnologías de la Información
Escuela de Ingeniería Civil En Informática

**Información al consumidor de los atributos de un producto de
supermercado mediante una aplicación móvil:
“FoodAdvice”**

Memoria para optar por el título de Ingeniero Civil en Informática.

César Núñez Arias.

Profesor Guía: Miguel Romero V.
Profesor informante: Gilberto Gutiérrez R.

Chillán, 2015

Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad de Bío-Bío en el proceso de titulación para a la carrera de Ingeniería Civil en Informática. El proyecto titulado “Información al consumidor de los atributos de un producto de supermercado mediante una aplicación móvil: FoodAdvice”, es un producto pensado para orientar a los usuarios sobre la calidad de diversos productos alimenticios.

El proyecto se enfoca en resolver el dilema de los consumidores sobre averiguar, además de su valor nutricional, cuál es la calidad real de un producto, cuál es su composición y qué tan beneficioso o perjudicial puede ser para la salud de una persona. Cabe destacar que estos últimos son datos que no están directamente al alcance de un consumidor cualquiera.

Dado que la gama de posibles productos a examinar es muy amplia, para esta fase del proyecto se tomó como muestra la investigación realizada por el cliente de la aplicación sobre un solo tipo de producto: aceites de oliva. Esta muestra sirve para realizar la prueba de concepto de la aplicación con el fin de validar la idea y así realizar un desarrollo futuro más amplio, complejo y elaborado.

El proyecto es una aplicación móvil creada para el sistema operativo Android, desarrollada con el IDE Android Studio, escrito en lenguaje Java, la cual permite la visualización de los atributos descriptivos de un producto a partir del escaneo de su código de barra desde el dispositivo móvil. Para almacenar la información que devuelve la aplicación, se dispone de una base de datos la cual permite la incorporación de los productos, distinguiendo aquí dos tipos de usuarios: el administrador de la base de datos quien tiene acceso total a la información y el usuario final que solo puede consultar la base de datos a través del dispositivo móvil mediante un servicio web integrado en el servidor de la base de datos.

El principal beneficio alcanzado con esta aplicación es que los usuarios finales podrán tener información adicional a la que da el etiquetado normal de los productos, y están mejor informados a la hora de elegir productos bajo el criterio calidad-precio.

La calidad de la información entregada dependerá, en todo caso, de la institución encargada de gestionar la base de datos la cual puede incluir información de mucho valor para el consumidor, como por ejemplo análisis de laboratorio, valoración de los consumidores, si es apto para celíacos, etc. Dichos datos de calidad, en conjunto con una institución que respalde la información existente, puede convertirse en una gran contribución a la comunidad, y es por eso que esta herramienta es un paso inicial hacia una iniciativa que empodere a los consumidores con información de los productos que actualmente no poseen.

Summary

This project is submitted to satisfy the requirements asked by the Universidad del Bío-Bío in the certification process for the Computing Civil Engineering career. The project named “Information to consumers about a supermarket product’s attributes through a mobile application: FoodAdvice”, is a product designed to guide users about the quality of various edible products.

The project focuses on solving the consumer’s issue about figuring out, in addition to nutritional information, how good the quality of an edible product is, what its composition is and how beneficial or harmful it could be for a person’s health. Note that the last information is not directly reachable for average consumers.

Since the products range to examine is too large, the application costumer’s investigation about olive oils shall be taken as sample for this stage of the project. This sample will be used to make a concept test of the application for the purpose of validating the idea and do a wider, more complex and more elaborated development in the future.

The project consists in a mobile application created for Android operative systems, developed with the Android Studio IDE, written in Java language, which allows the view of a product’s descriptive attributes from the scanning of its barcode from the mobile device. To store the information that the application returns, there is a database which allows adding products, telling apart two kind of users: the data base administrator who has total access to the information and the final user who only can consult the data base through a web service attached in the database server.

The main benefit reached with this application is that final users will have additional information in addition to the common product’s label gives, and they are better informed at the time of choosing product under the price-quality criteria.

The delivered information’s quality will depends, in any case, of the institution in charge of

managing the data base which can keep valuable information for the consumer, like laboratory analysis, customer's rating, if it's suitable for coeliacs, etc. That quality data, jointly with an institution that backs the existing information, can become a great contribution to the community, and that's why this tool is a first step to an initiative that empowers the consumers with information that product, nowadays, don't have.

Índice General

1	<u>INTRODUCCIÓN.....</u>	9
2	<u>DEFINICIÓN DE LA ENTIDAD INTERESADA.....</u>	11
2.1	DESCRIPCIÓN DEL CLIENTE	11
2.2	DESCRIPCIÓN DEL ÁREA DE ESTUDIO	11
2.3	DESCRIPCIÓN DE LA PROBLEMÁTICA.....	11
3	<u>DEFINICIÓN DEL PROYECTO</u>	13
3.1	OBJETIVOS DEL PROYECTO	13
3.1.1	OBJETIVO GENERAL:.....	13
3.1.2	OBJETIVOS ESPECÍFICOS:.....	13
3.1.3	AMBIENTE DE INGENIERÍA DE SOFTWARE.....	13
3.1.4	HERRAMIENTAS DE PROGRAMACIÓN.....	14
3.1.5	TECNOLOGÍAS.....	15
3.1.6	DEFINICIONES, SIGLAS Y ABREVIACIONES.....	16
4	<u>ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE.....</u>	17
4.1	ALCANCES	17
4.2	DESCRIPCIÓN GLOBAL DEL PRODUCTO	18
4.2.1	INTERFAZ DE USUARIO.....	18
4.2.2	INTERFAZ DE HARDWARE	19
4.2.3	INTERFAZ DE SOFTWARE.....	19
4.3	REQUERIMIENTOS ESPECÍFICOS.....	19
4.3.1	REQUERIMIENTOS FUNCIONALES DEL SISTEMA	19
4.3.2	DATOS DE ENTRADA.....	20
4.3.3	INTERFACES EXTERNAS DE SALIDA.....	20
4.3.4	ATRIBUTOS DE LA APLICACIÓN.....	20
5	<u>FACTIBILIDAD.....</u>	21
5.1	FACTIBILIDAD TÉCNICA.....	21
5.2	FACTIBILIDAD OPERATIVA.....	22
5.3	CONOCIMIENTOS PARA EL DESARROLLO.....	22
5.4	FACTIBILIDAD ECONÓMICA	23
5.5	CONCLUSIÓN DE LA FACTIBILIDAD.....	26
6	<u>ANÁLISIS.....</u>	27
6.1	DIAGRAMA DE CASOS DE USO.....	27
6.1.1	ACTORES.....	27
6.1.2	CASOS DE USO Y DESCRIPCIÓN.....	27
6.1.3	ESPECIFICACIÓN DE LOS CASOS DE USO.....	29
6.2	MODELAMIENTO DE DATOS	35

6.2.1	DESCRIPCIÓN DE LAS ENTIDADES	35
7	<u>DISEÑO</u>	<u>37</u>
7.1	DISEÑO DE FÍSICO DE LA BASE DE DATOS	37
7.1.1	CATEGORÍA	38
7.1.2	PRODUCTO	39
7.1.3	ENVASE	39
7.1.4	GRUPO ATRIBUTOS	40
7.1.5	ATRIBUTO	40
7.1.6	PRODUCTO_ATRIBUTO	41
7.2	DIAGRAMA ARQUITECTÓNICO	42
7.3	DIAGRAMA DE PAQUETES	44
7.4	DISEÑO INTERFAZ Y NAVEGACIÓN	48
7.5	DIAGRAMA DE SECUENCIA	52
8	<u>CONCLUSIONES</u>	<u>55</u>
9	<u>BIBLIOGRAFÍA</u>	<u>56</u>
10	<u>ANEXOS</u>	<u>59</u>
10.1	VALORES DE PLANES DE GIGAS.COM	59
10.2	DIAGRAMA DE CLASES JAVA	60

Índice de Tablas

Tabla 1: Requisitos funcionales de la aplicación.....	19
Tabla 2: Datos de entrada del sistema	20
Tabla 3: Datos de salida del sistema.....	20
Tabla 4: Cálculo de VAN.....	25
Tabla 5: Caso de uso Consultar producto	29
Tabla 6: Caso de uso Escanear producto	30
Tabla 7: Caso de uso Añadir producto	31
Tabla 8: Caso de uso Editar producto.....	32
Tabla 9: Caso de uso Eliminar producto	33
Tabla 10: Caso de uso Ingresar al mantenedor	34
Tabla 11: Descripción de la tabla Categorías.....	38
Tabla 12: Descripción de la tabla Productos.....	39
Tabla 13: Descripción de la tabla Envase	39
Tabla 14: Descripción de la tabla GrupoAtributos	40
Tabla 15: Descripción de la tabla Atributo	40
Tabla 16: Descripción de la tabla Producto_Atributo	41

Índice de Figuras

Figura 1: Prototipo de interfaz de FoodAdvice.....	18
Figura 2: Fórmula del Valor Actual Neto (VAN).....	25
Figura 3: Diagrama de casos de uso	28
Figura 4: Modelo entidad - relación.....	35
Figura 5: Modelo físico de la base de datos	37
Figura 6: Diagrama de arquitectura de la aplicación Android	42
Figura 7: Diagrama de paquetes de la aplicación móvil.	44
Figura 8: Pantalla de inicio de FoodAdvice.....	48
Figura 9: Pantalla de escaneado de códigos de barra.	49
Figura 10: Lista desplegable con los resultados de la consulta.....	50
Figura 11: Lista desplegable con el grupo “Información general” abierto.....	51
Figura 12: Lista con el grupo “otros” abierto.....	51
Figura 13: Diagrama de navegación.....	52
Figura 14: Diagrama de secuencia	53
Figura 15: Costos de planes de hosting en Gigas.com.....	59
Figura 16: Diagrama de clases.....	60

1 INTRODUCCIÓN

En Chile el etiquetado de productos alimenticios es obligatorio. No obstante, dicho etiquetado se limita a los atributos nutricionales, mientras hay más información que el cliente necesita conocer para tomar una buena decisión de compra. Tal es el caso del mercado de aceites de oliva, donde un factor determinante de su calidad es el porcentaje de acidez. Dicho atributo no se encuentra en la etiqueta.

Hoy en día las tecnologías móviles son de uso masivo en la población. Dispositivos como los teléfonos móviles permiten la incorporación de nuevas aplicaciones desarrolladas para un propósito específico, como obtener la información relacionada a productos de supermercado a través de códigos de barra.

Es en este contexto en el que se enmarca esta actividad de titulación llamada “Información al consumidor de los atributos de un producto de supermercado mediante una aplicación móvil”, donde se propone desarrollar una aplicación móvil que permita al usuario obtener información adicional a la entregada por el fabricante sobre algún producto.

Hay que destacar que, aunque se trata de una prueba de concepto donde la aplicación móvil es un programa sencillo, la base de datos se ha desarrollado de forma íntegra para concluir esa etapa del proyecto y además asegurar que la aplicación retornará la información adecuada sobre los productos incluidos en ella.

Las secciones de este documento comprenden detalles relativos a:

1. La entidad interesada en el desarrollo del proyecto.
2. Los objetivos fijados durante el desarrollo del proyecto.
3. Las características para el programa establecidas al inicio del desarrollo tales como funciones, interfaz y manejo de datos.
4. La viabilidad del proyecto en el contexto técnico y operativo.
5. La descripción del proceso de utilización del software y los casos de uso para cada ente involucrado en su utilización y funcionamiento.

6. Los diagramas del diseño del programa en ámbitos tales como base de datos y su composición de clases.
7. La presentación de la interfaz de usuario del programa.

2 DEFINICIÓN DE LA ENTIDAD INTERESADA

En esta sección se presenta al cliente, el actor interesado en que el proyecto se lleve a cabo. Se describen la ocupación del cliente, el área de estudio del proyecto y la problemática que se debe resolver con él.

2.1 Descripción del cliente

La idea inicial del proyecto ha sido planteada por el profesor Rodrigo Romo Muñoz [1], ingeniero comercial de la Universidad del Bío-Bío, campus Fernando May, doctor en economía agraria y jefe del departamento de investigación de dicha institución (DIUBB), en base a su investigación sobre la calidad real de los aceites de oliva presentes en el comercio chileno.

2.2 Descripción del área de estudio

No existe un área específica del lado del cliente que se pueda referenciar como área de estudio. Sin embargo es importante señalar que la idea es crear un producto software orientado, por una parte a los consumidores que posean un dispositivo móvil Android y por otra parte cualquier institución que esté interesada en convertirse en un referente en cuanto a proveer información adicional, relevante y de calidad a los consumidores chilenos, respecto de productos alimenticios.

2.3 Descripción de la problemática

Actualmente en Chile, por ley 20.606 sobre la composición nutricional de los alimentos y su publicidad [2], los productos que se comercializan en supermercados deben tener una etiqueta con información nutricional y los ingredientes utilizados en su elaboración. Aunque esta información es útil, en muchos casos es insuficiente para que el consumidor escoja informado qué producto es el que se ajusta más a sus necesidades, escogiendo generalmente de modo emocional o recurriendo a nutricionistas, lo que implica un costo adicional.

En este mismo sentido desde el 27 de junio de 2016 los alimentos que presentan en su composición nutricional contenidos elevados en calorías, grasas, azúcares, sal u otros

ingredientes, deberán llevar una leyenda de “Alto en...” según lo consigna en el Reglamento Sanitario de los Alimentos (BCN 2015).

A pesar de lo anterior, aún existe un problema que no se puede superar con una ley de esta naturaleza y que es el de la desconfianza de los consumidores frente a las empresas. Esta desconfianza se pudo constatar en los resultados del estudio sobre la Percepción acerca de la certificación y etiquetado de los alimentos [3]. En dicha encuesta frente a la pregunta: “En general, ¿cree usted en la información que dan las empresas sobre sus alimentos?”, sólo el 21.3% respondió que sí, mientras que el 38% no cree y el 40.8% A veces/depende.

Las principales razones de dicha desconfianza entre el 62% que responde que no cree o bien que a veces/depende son las siguientes:

- Las empresas generalmente mienten.
- Es difícil de entender / lenguaje muy técnico / poco claro.
- No son transparentes en los ingredientes / componentes.
- No le da confianza / desconfía / no hay credibilidad.

Por lo anterior se propone como solución una aplicación móvil que permita, mediante el escaneo de código de barra, obtener información adicional de cualquier producto de supermercado referente a su composición y sus atributos nutricionales y químicos rescatada desde una base de datos elaborada y mantenida por una institución de prestigio que dé a los consumidores la confianza que necesitan en cuanto a la información obtenida para dicho producto.

Es importante señalar que el alcance de este proyecto se limita en la construcción de la plataforma tecnológica y puntualmente al desarrollo de la aplicación Android de modo que en el futuro esta plataforma pueda ser implementada en la Universidad del Bío-Bío u otra institución que desee prestar este importante servicio de información a los consumidores chilenos.

3 DEFINICIÓN DEL PROYECTO

En este capítulo se describe el proyecto en cuanto a sus objetivos, tanto generales como específicos, así como el entorno de desarrollo, las herramientas y tecnologías utilizadas y los términos técnicos de uso común utilizados en este informe.

3.1 Objetivos del proyecto

3.1.1 Objetivo general:

El gran objetivo del proyecto es:

“Desarrollar una aplicación móvil que, mediante el escaneo del código de barra de un producto comestible, permita visualizar la información nutricional recuperada desde la base de datos junto con otros atributos descriptivos del producto los cuales deben estar agrupados por categorías”.

3.1.2 Objetivos específicos:

Para conseguir el objetivo general se han planteado los siguientes objetivos específicos

1. Determinación de los requisitos de la aplicación a desarrollar.
2. Diseñar e implementar la aplicación móvil que permita obtener los atributos descriptivos de un producto a partir del escaneo de su código de barra.
3. Diseñar e implementar una base de datos que permita registrar la información nutricional de los productos además de permitir la incorporación de atributos descriptivos de forma dinámica y asociar estos atributos a categorías.
4. Poblar la base de datos con algunos productos, sus atributos y categorías con la finalidad de validar la aplicación en el futuro.

3.1.3 Ambiente de Ingeniería de Software

Para desarrollar el software, se ha recurrido a la metodología de desarrollo iterativa incremental, ya que permite una actualización constante para el cliente sobre el estado actual

del desarrollo de la aplicación, además de obtener una retroalimentación por parte del cliente sobre las características deseadas en la aplicación.

Para crear esta aplicación se han utilizado recursos de programación para dispositivos móviles, gestor de bases de datos y programación web.

El programa principal -la aplicación para teléfono móvil- está programado en Java adaptado para funcionar bajo los parámetros del sistema operativo Android en su versión 4.2.2 (alias Jellybean).

Por otra parte, el módulo web que gestiona la base de datos, consiste en la creación de una aplicación en Yii para administrar de forma más fácil la base de datos programada con MySQL, el cual se configura mediante PHP MyAdmin.

3.1.4 Herramientas de programación.

- **IDE Android Studio [4]:** Basado en el IDE para programación en Java IntelliJ, Android Studio es una plataforma configurada para facilitar la programación de aplicaciones para sistemas operativos Android, ya que incorpora plantillas básicas para componentes de aplicaciones Android, la incorporación de un constructor de interfaces gráficas en formato XML y la posibilidad de importar bibliotecas externas para integrarlas al programa y utilizarlas en el desarrollo. Además provee herramientas de emulación de dispositivos móviles para probar la aplicación.
- **PHP MyAdmin [5]:** Gestor de base de datos web gratuito que permite manejar la administración de base de datos MySQL.
- **Yii [6]:** es un framework PHP de alto rendimiento basado en componentes que permite desarrollar aplicaciones Web de gran escala. El framework permite la reutilización de sus componentes en la programación web y puede acelerar el proceso de desarrollo. Se escogió para gestionar de forma más clara la base de datos a la que

accede la aplicación, esto gracias a que permite crear directamente sitios web muy completos que incorporan opciones de inserción, edición y eliminación de entradas en la base de datos.

- **yEd Graph Editor [7]:** Aplicación de escritorio gratuita para la generación de diagramas. Utilizado para diagrama de casos de uso y modelo entidad-relación.
- **draw.io [8]:** Herramienta de creación de diagramas online. Usada para la creación de modelos de datos, clases, secuencia y paquetes.

3.1.5 Tecnologías

- **Servicio web [9]:** Tecnología utilizada para permitir la comunicación y el intercambio de información entre aplicaciones por medio de internet. En este proyecto, se utiliza para conectar la aplicación móvil con la base de datos.
- **Volley [10]:** Librería HTTP que hace que las interacciones de red en Android sean más fáciles y rápidas, permitiendo el manejo concurrente de peticiones de conexión desde los procesos de una aplicación móvil.
- **Java [11]:** Lenguaje de programación orientado a objetos en el cual se basa Android.
- **XML [12]:** Por su significado Extensible Markup Language. Formato de archivo para publicación e intercambio de datos en Internet o cualquier otra aplicación. Con estos se construyen las interfaces gráficas en Android.
- **JSON [13]:** Por su significado JavaScript Object Notation, es un formato de intercambio de datos ligero, fácil de leer para programadores y fácil de interpretar y traducir mediante programación.

- **Zxing [14]:** Librería de procesamiento de códigos de barra programada en Java. La integración en Android hace que la lectura se realice mediante la cámara fotográfica del dispositivo.

3.1.6 Definiciones, Siglas y Abreviaciones

Framework: Por su traducción literal “marco de trabajo”. Patrón o esquema para orientar la programación de aplicaciones.

Aplicación móvil [15]: Todo programa que se ejecuta en algún dispositivo móvil.

Dispositivo móvil: Cualquier dispositivo electrónico suficientemente pequeño para ser transportado mientras se usa. Hoy se asocia este concepto a teléfonos móviles o tablets con sistema operativo Android, iOS o Windows Phone.

IDE: Integrated Development Environment o Entorno de desarrollo integrado. Programa compuesto de diversas herramientas dedicadas a uno o más lenguajes de programación como Java, C++ o PHP. Ej: Eclipse, Netbeans, IntelliJ o Android Studio.

4 ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

4.1 Alcances

Al tratarse de una prueba de concepto, para fines de evaluación sobre su funcionamiento y asegurar que la información ofrecida sea la correcta, solo se concentra en entregar información sobre un tipo de producto, que son aceites de oliva y limitada a un sub-conjunto de todos los aceites de oliva del mercado nacional.

La información a entregar será, por un lado, la misma de la etiqueta del producto y la información adicional que proporciona la aplicación que, entre otras cosas, indicará el porcentaje de acidez que tiene el aceite.

Las consultas por código de barra se realizan mediante una aplicación móvil, accediendo a una base de datos centralizada, para la cual se creará un servicio web para acceder a ella.

El programa también contará con un módulo web que permitirá a expertos administrar la base de datos de forma rápida. Esta fase del proyecto solo se centrará en hacer funcionar correctamente la aplicación móvil, dejando el módulo web implementado en forma básica, poniendo la base para su desarrollo futuro.

Se realizará una validación y verificación de la aplicación acotada al hecho de que el producto es una prueba de concepto. Y por lo tanto una validación final y rigurosa deberá ser llevada a cabo sobre el producto definitivo que se derive del presente proyecto.

A futuro, esta aplicación móvil permitirá obtener datos nutricionales y químicos de productos comestibles tales como aceites, condimentos, bebidas, golosinas, etc., con tal de orientar a cualquier consumidor de forma clara y concisa sobre su composición y ofrecer recomendaciones sobre su consumo.

4.2 Descripción Global del Producto

4.2.1 Interfaz de usuario

El usuario final del producto solo tendrá acceso a la aplicación para dispositivos móviles para interactuar con el sistema.

La aplicación móvil presentará al usuario una interfaz simple con un botón que, al pulsarlo, permite a éste acceder a la opción de escanear un producto mediante la cámara del dispositivo móvil. Luego de escanear el código, el programa presenta la información en una lista con los atributos del producto, incluyendo una imagen del producto en cuestión (ver Figura 1).

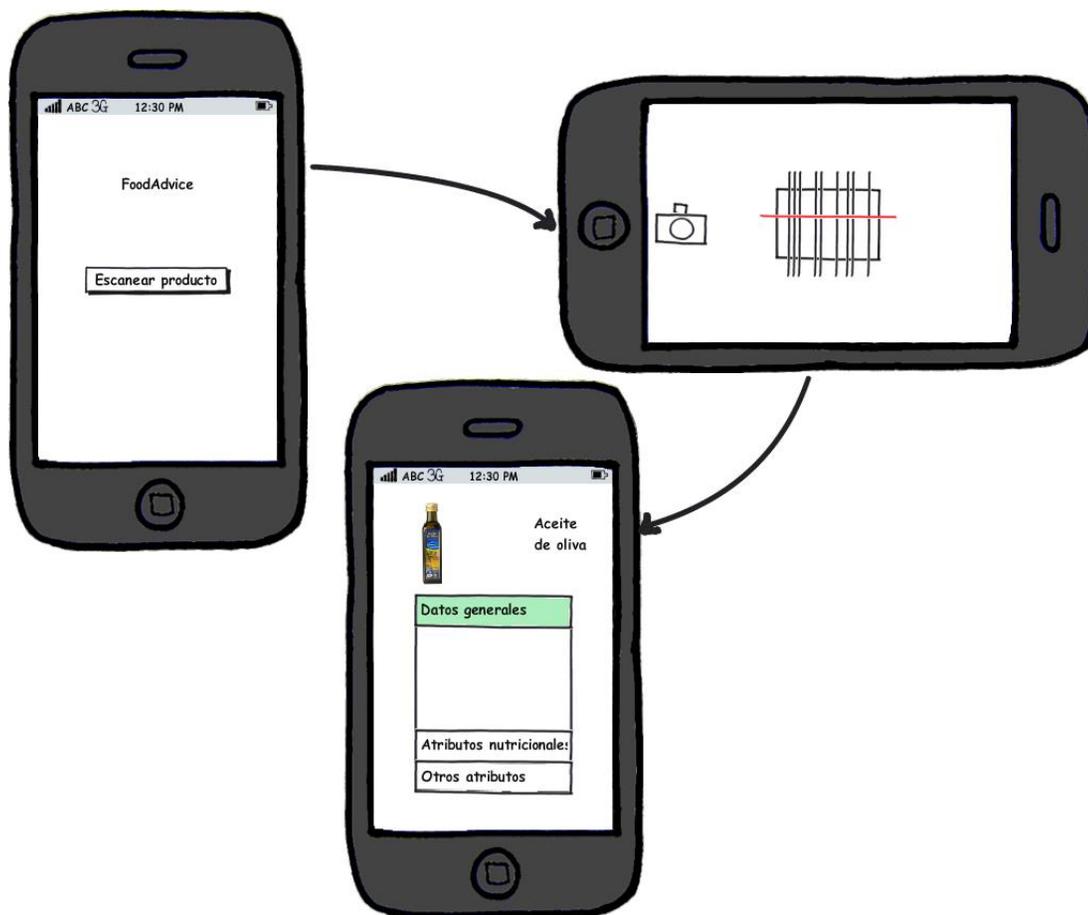


Figura 1: Prototipo de interfaz de FoodAdvice

El cliente requiere que, idealmente, la tipografía del programa sea una fuente de tamaño grande para las personas con problemas de visión.

4.2.2 Interfaz de Hardware

El hardware para el correcto funcionamiento del programa varía según el usuario, ya que es un programa que funciona en el marco de un sistema operativo preinstalado en dispositivos que pueden ser diferentes unos de otros en características técnicas.

Por lo anterior, el software funciona correctamente en cualquier dispositivo móvil que disponga del sistema operativo Android, en su versión 4.2.2 o superior.

4.2.3 Interfaz de Software

El programa consta de dos partes: La aplicación móvil y la base de datos alojada en la web. Para conectar a las dos partes del sistema, la aplicación en Android debe hacer uso de un servicio web que recoja la información desde la base de datos y la entregue en un formato que la aplicación móvil pueda reconocer. En este caso, el servicio web devuelve los datos en formato JSON que luego la aplicación móvil traduce para mostrar en pantalla.

4.3 Requerimientos Específicos

4.3.1 Requerimientos Funcionales del sistema

Id	Nombre	Descripción
RF-01	Escaneo de códigos de barra	El software debe ser capaz de captar la información de códigos de barra para posteriormente ser usada en la obtención de información de algún producto.
RF-02	Obtención de información	En base al RF-01, el software debe ser capaz de obtener información nutricional del producto.
RF-03	Recomendaciones	El sistema debe ser capaz de evaluar la calidad del producto escaneado en base a los atributos como la acidez, carbohidratos, grasas, etc.
RF-04	Sistema web	El sistema debe implementar una interfaz web para administrar información de productos del lado del administrador.

Tabla 1: Requisitos funcionales de la aplicación

4.3.2 Datos de entrada

Referente a la información que se ingresa al sistema para que el programa funcione correctamente.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem
DE_01	Código del producto	Código de barra único del producto. Es el dato que el programa tomará como entrada a la hora de buscar la información.
DE_02	Nombre del producto	El nombre de fantasía del producto
DE_03	Precio	El precio del producto, resultado del cálculo del promedio de precios en varios supermercados.
DE_04	Información nutricional	Energía, proteínas, grasa total, carbohidratos, sodio.

Tabla 2: Datos de entrada del sistema

4.3.3 Interfaces externas de Salida

Se especifica cada salida del sistema, indicando en cada caso el formato o medio de salida.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
IS_01	Imagen del producto	Una fotografía del código escaneado.	Pantalla de dispositivo móvil, formato jpg.
IS_02	Nombre del producto	El nombre de fantasía y marca del producto.	Pantalla de dispositivo móvil.
IS_03	Información nutricional	Energía, proteínas, grasa total, carbohidratos, sodio.	Pantalla de dispositivo móvil.
IS_04	Atributos adicionales	Atributos específicos que un producto pueda tener registrado en la base de datos	Pantalla de dispositivo móvil.

Tabla 3: Datos de salida del sistema

4.3.4 Atributos de la aplicación

- **Usabilidad y operatividad:** En caso de eventuales errores de conexión con la base de datos, el sistema deberá entregar una alerta clara diciendo que no se pudo lograr la conexión o que la base de datos no se encuentra disponible.
- **Comodidad:** La interfaz del producto debe ser diseñada con objetos claros y destacados de forma que el usuario distinga los controles de la aplicación.

5 FACTIBILIDAD

En esta sección se evalúan la factibilidad del proyecto según los recursos disponibles y necesarios para llevarlo a cabo. Se evaluará bajo criterios técnicos de hardware y software, operativos y de conocimientos en tecnologías de información.

5.1 Factibilidad técnica.

Este software es desarrollado en un equipo propio con las siguientes características:

- Equipo: MSI CX61.
- Procesador: Intel Core i5-4200M @2,5 GHz.
- RAM: 8GB DDR3.
- HDD: 750 GB.
- Gráficos: Nvidia GeForce GT 740M / 2GB DDR3.
- Sistema operativo: Windows 10.

Por el lado del software, la aplicación móvil está desarrollada íntegramente con el IDE de desarrollo Android Studio en su versión 1.0.1.

Para verificar su correcto funcionamiento, el producto se ejecuta en un smartphone Sony Xperia L, con sistema operativo Android 4.2.2 (Jellybean).

Por otra parte, el gestor de base de datos MySQL con sus funciones de manejo de información de los productos y el ordenamiento de ésta se encuentran implementadas en el servidor Parra del laboratorio de especialidad de la Universidad del Bío-Bío, campus Fernando May.

Los datos de los productos se organizan en una base de datos MySQL vinculada a un mantenedor de base de datos creado mediante el framework de programación web PHP Yii, en su versión 1.1.

5.2 Factibilidad operativa.

Dado que el programa es una prueba de concepto y no un producto final, la factibilidad operativa como tal no corresponde. Sin embargo se describe el análisis desde la perspectiva de la prueba de concepto.

Para funcionar, el programa necesitará:

- Acceso a internet
- Acceso a un servidor PHP conectado a base de datos
- Dispositivo Android versión 4.2.2 o superior.

Al terminar, la aplicación no requerirá de mantenimiento. Sin embargo, para que entregue la información correcta se necesitará de personal dispuesto a actualizar la base de datos de productos con información actualizada y fidedigna. En el caso de la prueba de concepto, no se realizarán mantenciones ya que quedarán datos de prueba digitados en la base de datos.

5.3 Conocimientos para el desarrollo.

Para desarrollar el proyecto se requieren conocimientos en las siguientes áreas:

1. Programación en Java: Es el lenguaje en el cual se basa Android.
2. Programación para dispositivos Android: Se debe conocer la estructura de programas para Android para poder programarlas.
3. Programación en MySQL: Para mantener la base de datos y resolver cualquier problema referente a ella.
4. Programación en PHP: Para crear el servicio web que conectará la aplicación con la base de datos.
5. Diseño de interfaces: Crucial para diseñar la presentación de la aplicación hacia el usuario.

5.4 Factibilidad económica

Para el estudio de factibilidad económica se considera la perspectiva de la aplicación final. En este sentido se presenta un estudio de costos que serán necesarios incurrir para la implantación y operación del sistema.

Los costos de la plataforma están del lado del servidor. Existen dos alternativas que se pueden adoptar a la hora de implantar este sistema que son:

- La compra de la infraestructura tecnológica necesaria y su respectiva operación y mantenimiento. Esta alternativa tiene una inversión inicial bastante elevada.
- La contratación de un servicio de plataforma en la nube, lo cual es muy atractivo porque no requiere una fuerte inversión inicial y es fácil de escalar en la medida que se requiera contar con una mayor capacidad de cómputo.

Para este estudio se considera la contratación de un servicio en la nube. En particular se ha optado por hacer el análisis basado en los costos del servicio de Gigas.com [16] a modo de referencia, pero sería importante hacer un estudio comparativo de costos para seleccionar al proveedor que más convenga en el caso de querer implantar la plataforma como tal en el futuro. Los costos de Gigas estimados por año para esta aplicación son de US\$342 equivalentes a \$243.846 en su plan de 50GB de almacenamiento con 4 GB de RAM (ver anexo 10.1).

Si bien los costos operacionales de la plataforma informática están cubiertos por la solución cloud, es necesario contar con una persona que administre la base de datos y que se preocupe de mantener la información actualizada. Para ello es necesario contar con un digitador cuyo costo anual varía desde los \$3.600.000 a los \$7.248.000[17]. Además se debe alquilar un lugar donde pueda trabajar y equipamiento para que pueda desempeñar sus funciones. Una computadora tiene un costo de alrededor de \$300.000, y una oficina tiene un costo de aproximadamente \$360.000 al año.

Junto con ello es necesario estimar el costo de producir información de calidad que sea de utilidad para los consumidores. Este costo podría ser absorbido por medio de alianzas

estratégicas con instituciones certificadoras, universidades y el gobierno a través de proyectos de investigación u otros fondos concursables.

Para poder cuantificar los beneficios de este sistema es necesario adoptar un modelo de negocio en particular que permita que la operación del sistema se autofinancie. Por ejemplo se podría adoptar el cobro directamente a todos los usuarios mediante el pago de una suscripción anual a los servicios. Otro modelo podría ser el contar con dos categorías de usuarios: los Free y los Premium. Los usuarios free pueden acceder a la información básica del producto que sería la información exigida por ley de las etiquetas, más uno o dos atributos adicionales, pero la información adicional de mayor valor estratégico estaría disponible para los usuarios Premium. Este último modelo se sustenta en la medida que los usuarios Premium sean lo suficientemente numerosos como para cubrir los costos de operación de todos los usuarios. Esta última alternativa es la usada en varios servicios móviles como Spotify y por lo tanto ha demostrado ser un modelo de negocio viable, y es un fuerte candidato a emplear si se quiere implantar el sistema. Otra alternativa que este servicio funcione con el patrocinio de alguna institución como puede ser la Universidad del Bío-Bío y que está absorba los costos operacionales en base a fondos propios o ganados en proyectos concursables.

Es difícil estimar cuantos son los potenciales usuarios sin antes haber hecho un estudio de mercado, por esta razón, el análisis de los posibles beneficios se hará teniendo en cuenta los siguientes supuestos:

1. El modelo de negocio adoptado es el de contar con dos tipos de usuarios uno free y el otro premium.
2. La relación entre usuarios premium y free al primer año es de 1/20 (5% premium y 95% free).
3. Que el total de usuarios es de 1000 (50 premium al primer año).
4. Que el costo por el servicio es de \$ 5.000.
5. Los usuarios premium aumentan a un 10% en el año 2 y siguen aumentando en razón de un 10%.

Para determinar si la futura puesta en marcha del proyecto sería rentable o no, se ha calculado el indicador de Valor Actual Neto o V.A.N. Este indicador permite calcular los futuros ingresos y egresos de un proyecto para luego determinar, luego de restar la inversión inicial, si el proyecto es viable [18]. Este se calcula con la fórmula de la Figura 2:

$$VAN = \sum_{t=1}^n \frac{V_t}{(1+k)^t} - I_0$$

Figura 2: Fórmula del Valor Actual Neto (VAN)

V_t representa los flujos de caja en cada periodo t , años en este caso.

I_0 es el valor del desembolso inicial de la inversión.

n es el número de períodos considerado.

K es la tasa de descuento anual.

Tomando en cuenta los datos y escenarios anteriormente expuestos, el resultado de calcular el VAN es como se muestra en la tabla 4.

Ingresos	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Suscripción clientes premium		3.000.000	6.000.000	6.600.000	7.260.000	7.986.000
Total ingresos		3.000.000	6.000.000	6.600.000	7.260.000	7.986.000
Gastos	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Hosting Gigas Chile		243.846	243.846	243.846	243.846	243.846
Personal		3.600.000	3.600.000	3.600.000	3.600.000	3.600.000
Conexión a internet		300.000	300.000	300.000	300.000	300.000
Equipo	300.000					
Oficina		360.000	360.000	360.000	360.000	360.000
Gastos comunes		240.000	240.000	240.000	240.000	240.000
Total gastos		4.743.846	4.743.846	4.743.846	4.743.846	4.743.846
Balance		-1.743.846	1.256.154	1.856.154	2.516.154	3.242.154
$V/(1-k)$		-1.683.249	1.170.371	1.669.301	2.184.229	2.716.657
Sumatoria $V/(1-k)$		6.057.309				
VAN		5.757.309				

Tabla 4: Cálculo de VAN

5.5 Conclusión de la factibilidad

El proyecto cuenta con todos los elementos para ser desarrollado, Además de impactar positivamente en la vida de los usuarios finales, dado que la aplicación apunta a facilitar la búsqueda de información de calidad sobre productos comestibles.

Económicamente, como muestra el cálculo del VAN, el proyecto al ser implantado sería rentable bajo el supuesto de tener una suscripción de \$5.000 y una proporción de 5% de usuarios Premium al primer año que aumente al 10% el segundo año y que mantenga la tasa de crecimiento en un 10%.

6 ANÁLISIS

Esta sección presenta el análisis de los requisitos, poniendo el énfasis en los actores involucrados en el uso de la aplicación y cómo interactúan con el sistema mediante la herramienta de análisis de casos de uso. También se hace un análisis respecto de la base de datos presentando un modelo entidad relación.

6.1 Diagrama de casos de uso

Se presenta el diagrama de casos de uso (ver figura 3), ilustrando la interacción de los distintos actores con el programa, tanto para obtener información como para asegurar que se entregue la información correcta.

6.1.1 Actores

Usuario:

- Rol: Usuario final del sistema.
- Conocimientos técnicos: Básicos.
- Privilegios: Solo consulta mediante la aplicación.

Administrador:

- Rol: Administrador de la base de datos.
- Conocimientos técnicos: Avanzados. Requiere conocimientos en gestión de bases de datos.
- Privilegios: Acceso completo a la modificación de la base de datos.

6.1.2 Casos de Uso y descripción

Aquí se presenta el diagrama de los casos de uso para mostrar la interacción entre los actores y los casos de uso.

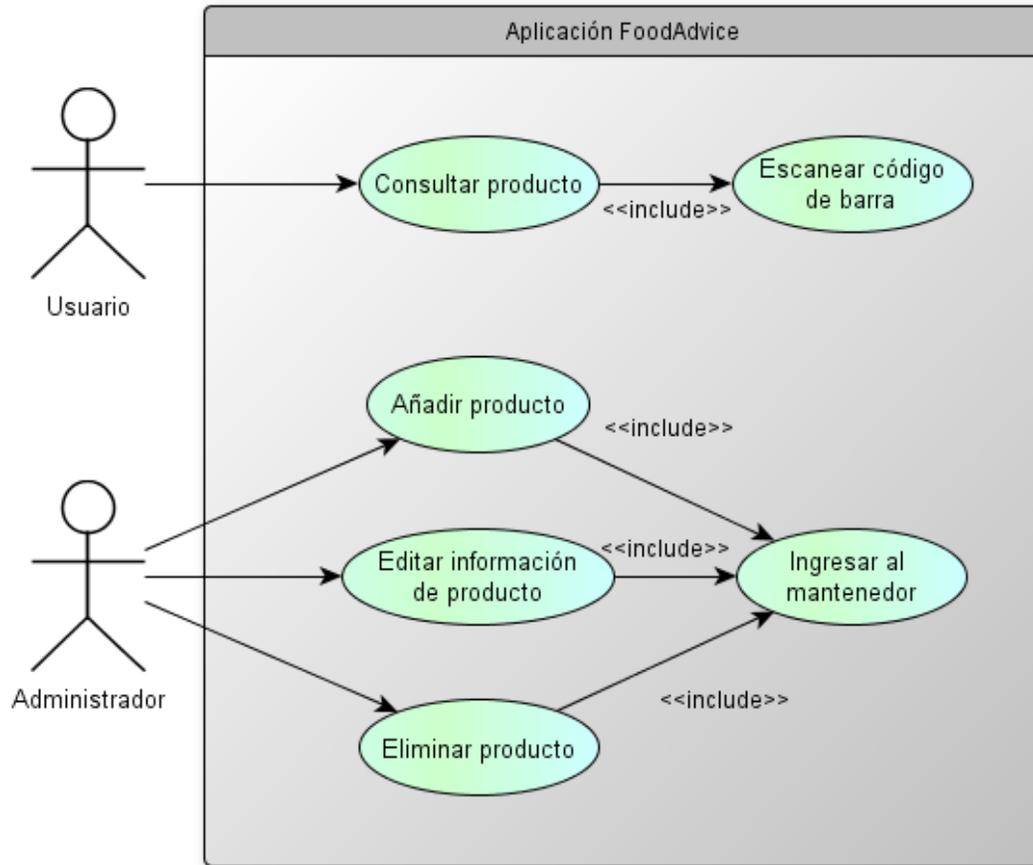


Figura 3: Diagrama de casos de uso

6.1.3 Especificación de los Casos de Uso

6.1.3.1 Caso de uso: Consultar producto	
ID:	CU 1
Descripción:	El usuario se dispone a consultar un producto
Actor principal:	Usuario
Actor secundario:	Ninguno
Precondición:	Ninguna
Flujo principal	
1-	El caso de uso comienza cuando el usuario pincha el botón “Escanear producto” en la pantalla principal.
2-	El sistema muestra la interfaz para escanear un producto.
3-	El usuario escanea un producto (incluye “escanear producto”).
4-	Si el sistema consigue retornar la información: <ul style="list-style-type: none"> a. El usuario obtiene la respuesta del sistema en su dispositivo en su pantalla.
5-	Si no: <ul style="list-style-type: none"> a. El sistema muestra la pantalla de error de conexión al usuario b. El usuario pincha en aceptar c. El sistema devuelve al usuario a la pantalla principal para empezar de nuevo.
Postcondición:	Los datos han sido devueltos por el sistema.
Flujo alternativo:	Ninguno

Tabla 5: Caso de uso Consultar producto

6.1.3.2 Caso de uso: Escanear producto <<include>>	
ID:	CU 2
Descripción:	El usuario escanea un producto con la cámara del teléfono
Actor principal:	Usuario
Actor secundario:	Ninguno
Precondición:	Ninguna.
Flujo principal	
1-	El caso de uso comienza cuando el usuario pincha el botón “Escanear producto” en la pantalla principal.
2-	El sistema muestra la interfaz para escanear un producto.
3-	El usuario enfoca el código de barra con la cámara del teléfono.
4-	El sistema escanea el código de barra.
Postcondición:	El sistema dirige al usuario a la pantalla de respuesta.
Flujo alternativo:	Ninguno.

Tabla 6: Caso de uso Escanear producto

6.1.3.3 Caso de uso: Añadir producto
ID: CU 3
Descripción: El administrador de base de datos añade un producto a la case de datos
Actor principal: Administrador
Actor secundario: Ninguno
Precondición: Ninguna.
Flujo principal
<ol style="list-style-type: none"> 1- El caso de uso comienza cuando el administrador entra al mantenedor de la base de datos (incluye “ingresar al mantenedor”). 2- El administrador pincha en la pestaña de productos. 3- El sistema muestra la lista de productos ingresados actualmente y las opciones “List” y “Create”. 4- El administrador pincha en la opción “Create” para crear un producto. 5- El sistema muestra la pantalla de creación de productos con campos de texto para llenar. 6- El administrador ingresa los datos del producto. 7- El administrador pulsa el botón “Create”.
Postcondición: Se ha creado un nuevo producto.
Flujo alternativo: Ninguno.

Tabla 7: Caso de uso Añadir producto

6.1.3.4 Caso de uso: Editar producto
ID: CU 4
Descripción: El administrador edita la información de un producto ya creado.
Actor principal: Administrador
Actor secundario: Ninguno
Precondición: Ninguna.
Flujo principal
<ol style="list-style-type: none"> 1- El caso de uso comienza cuando el administrador entra al mantenedor de la base de datos (incluye “ingresar al mantenedor”). 2- El administrador pincha en la pestaña de productos. 3- El sistema muestra la lista de productos ingresados actualmente con botones de edición (lápiz) y eliminación (equis) es sus respectivas casillas. 4- El administrador pincha en el lápiz para editar un producto. 5- El sistema muestra la pantalla de edición de productos con campos de texto editables con la información actual del producto y un botón de “Save” para guardar los cambios. 6- El administrador ingresa los nuevos datos del producto. 7- El administrador pulsa el botón “Save” para guardar los cambios.
Postcondición: Un producto tiene nueva información.
Flujo alternativo: Ninguno.

Tabla 8: Caso de uso Editar producto

6.1.3.5 Caso de uso: Eliminar producto	
ID:	CU 5
Descripción:	El administrador elimina un producto de la lista de productos.
Actor principal:	Administrador
Actor secundario:	Ninguno
Precondición:	Ninguna.
Flujo principal	
1-	El caso de uso comienza cuando el administrador entra al mantenedor de la base de datos (incluye “ingresar al mantenedor”).
2-	El administrador pincha en la pestaña de productos.
3-	El sistema muestra la lista de productos ingresados actualmente con botones de edición (lápiz) y eliminación (equis) es sus respectivas casillas.
4-	El administrador pincha en la equis para eliminar un producto.
5-	El sistema muestra en pantalla un mensaje emergente preguntando por la eliminación del producto a la espera de una confirmación.
6-	El administrador escoge “aceptar” en el mensaje.
Postcondición:	Un producto ha sido eliminado de la lista.
Flujo alternativo:	6.a- El administrador escoge “cancelar” en el mensaje.

Tabla 9: Caso de uso Eliminar producto

6.1.3.6 Caso de uso: Ingresar al mantenedor <<include>>	
ID:	CU 7
Descripción:	El administrador ingresa al mantenedor de la base de datos
Actor principal:	Administrador
Actor secundario:	Ninguno
Precondición:	Se debe ser administrador para ingresar al sistema.
Flujo principal	
1-	El caso de uso comienza cuando el administrador ingresa a la página mantenedora de base de datos.
2-	El administrador pincha en la pestaña "Login".
3-	El sistema muestra en la pantalla de login los campos para usuario y contraseña y el botón "Login".
4-	El administrador ingresa su nombre y contraseña.
5-	El administrador pincha el botón "Login".
6-	Si nombre y contraseña son correctos: <ul style="list-style-type: none"> a. El sistema se dirige a la ventana principal.
7-	Si no: <ul style="list-style-type: none"> a. El sistema muestra un mensaje de error junto a los campos de usuario y/o contraseña.
Postcondición:	El administrador ha ingresado al sistema.
Flujo alternativo:	Ninguno.

Tabla 10: Caso de uso Ingresar al mantenedor

6.2 Modelamiento de datos

La Figura 4 muestra el modelo de datos utilizado para el desarrollo de la base de datos de productos que utiliza la aplicación. Se ha utilizado un modelo entidad-relación.

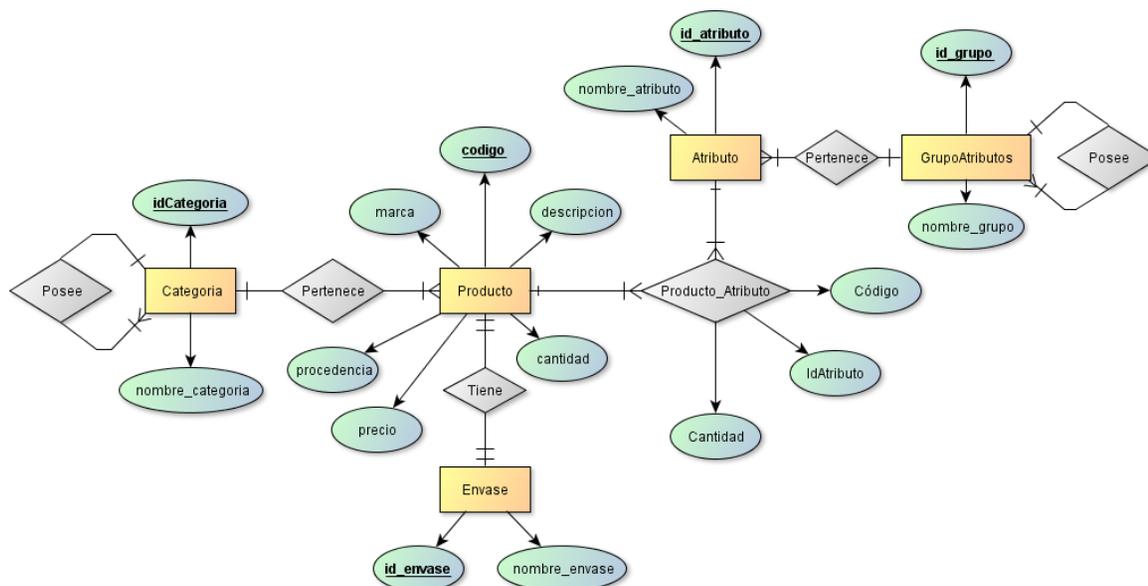


Figura 4: Modelo entidad - relación

6.2.1 Descripción de las entidades

1. **Producto:** Posee información básica sobre el producto examinado.
2. **Categoría:** Corresponde a la clasificación de un producto según su tipo. Una categoría para un producto podría ser Lácteos, Abarrotes, Congelados, entre otras. Estas pueden tener sub categorías como Yogurt, Quesos, Aceites, Galletas, Helados, Carnes, etc.
3. **Envase:** Contiene información sobre el envase en el que va el producto. Esta tabla se desprende de la tabla principal Producto por:
 - a. Referirse a elementos que se repetirán en futuras entradas en la base de datos
 - b. Evitar ambigüedades a la hora de introducir la información de un producto.
4. **GrupoAtributos:** Tabla que engloba a los grupos de diversos atributos que puedan tener los productos de la base de datos.
5. **Atributo:** Referente a características que pueden tener los productos de la base de datos,

desde la misma información nutricional hasta atributos químicos como la acidez.

6. **Producto_Atributo:** Tabla que relaciona a los productos con sus atributos asignando una cantidad según el contenido de un producto de determinado atributo.

7 DISEÑO

En esta sección se presentan el diseño de la aplicación expresados en los diagramas de base de datos, diagramas de clases y paquetes, los diagramas de navegación de la interfaz y un diagrama de secuencia.

7.1 Diseño de Físico de la Base de datos

Para el diseño de la base de datos (Figura 5) se ha tenido en cuenta la necesidad de contar con una descripción de los productos basada en atributos que son fijos por ley y luego que existen otro grupo de atributos que son deseables o adicionales. Estos últimos son variables y dinámicos, es decir que en el tiempo pueden aparecer o desaparecer atributos asociados a un producto. Además se consideró la necesidad de agrupar atributos en grupos de atributos de modo que al visualizar la información en el dispositivo móvil esta pudiera ser resumida o expandida en función de las necesidades de información de cada usuario.

A continuación se presenta el diseño físico de la base de datos.

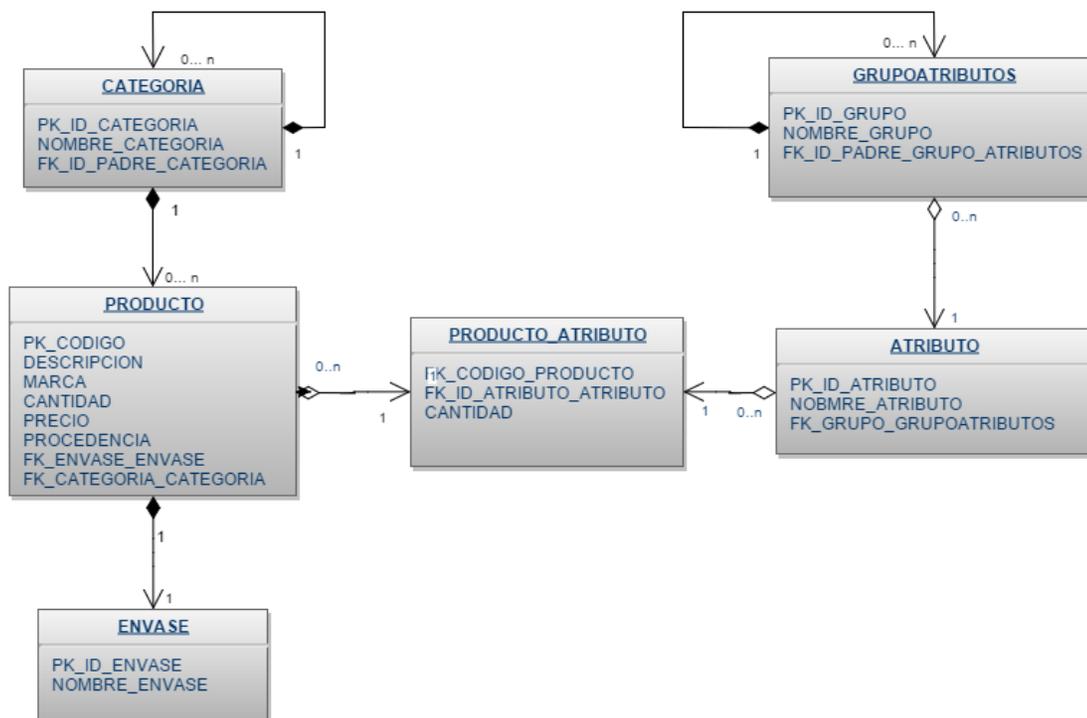


Figura 5: Modelo físico de la base de datos

A continuación se detallan las tablas y su significado:

7.1.1 Categoría	
Descripción:	<p>Corresponde a las categorías en las que se puede encasillar un producto. Se identifica por un número de id y posee referencia a sí misma, haciendo notar que puede haber subcategorías. Por ejemplo:</p> <ul style="list-style-type: none"> ❖ Abarrotes -> Aceite, ❖ Lacteos -> Quesos, ❖ Golosinas -> Chocolates.
Clave primaria:	id_categoria (INT).
Claves foráneas:	A. id_padre (Categoría.id_categoria. Se refiere a sí misma para incluir subcategorías).
Otras claves:	nombre_categoria.

Tabla 11: Descripción de la tabla Categorías

7.1.2 Producto	
Descripción:	Describe a un producto con sus datos más relevantes y generales. Estos incluyen su marca, su precio y su procedencia. Ésta posee el número del código con la que la aplicación móvil interactúa.
Clave primaria:	codigo (BIGINT).
Claves foráneas:	A. envase (Envase.id_envase. Describe la presentación del producto), B. categoría (Categoría.id_categoría. Describe la categoría a la que pertenece).
Otras claves:	descripción, marca, cantidad, precio, procedencia.

Tabla 12: Descripción de la tabla Productos

7.1.3 Envase	
Descripción:	Se refiere a los tipos de envase en los que puede presentarse un producto. Se desprende de la tabla principal de productos para evitar ambigüedades.
Clave primaria:	id_envase (INT).
Claves foráneas:	Ninguna.
Otras claves:	nombre_envase.

Tabla 13: Descripción de la tabla Envase

7.1.4 GrupoAtributos	
Descripción:	Categorías en los que se clasifican ciertos atributos de un producto.
Clave primaria:	id_grupo (INT).
Claves foráneas:	A. id_padre (GrupoAtributos.id_padre. Hace referencia a sí mismo para incluir subgrupos de atributos).
Otras claves:	nombre_grupo.

Tabla 14: Descripción de la tabla GrupoAtributos

7.1.5 Atributo	
Descripción:	Tabla donde se ordenan los atributos tanto nutricionales como químicos de un producto. Se organizan en grupos de atributos y se emparejan con los productos mediante una tabla de relación.
Clave primaria:	id_atributo (INT).
Claves foráneas:	A. grupo (GrupoAtributos.id_padre. Referencia al grupo al que pertenece.)
Otras claves:	nombre_atributo.

Tabla 15: Descripción de la tabla Atributo

7.1.6 Producto_Atributo	
Descripción:	Tabla de relación donde se emparejan productos con sus atributos, asignándole un valor que corresponde a la cantidad de cierto atributo que posee un producto.
Clave primaria:	id_combinacion (INT).
Claves foráneas:	A. codigo (Producto.codigo. Representa al producto al que se le asigna un valor de cierto atributo). B. id_atributo (Atributo.id_atributo. Representa al atributo cuya cantidad se asigna a un producto).
Otras claves:	cantidad.

Tabla 16: Descripción de la tabla Producto_Atributo

7.2 Diagrama arquitectónico

La Figura 6 muestra las partes que componen el software en su parte móvil.

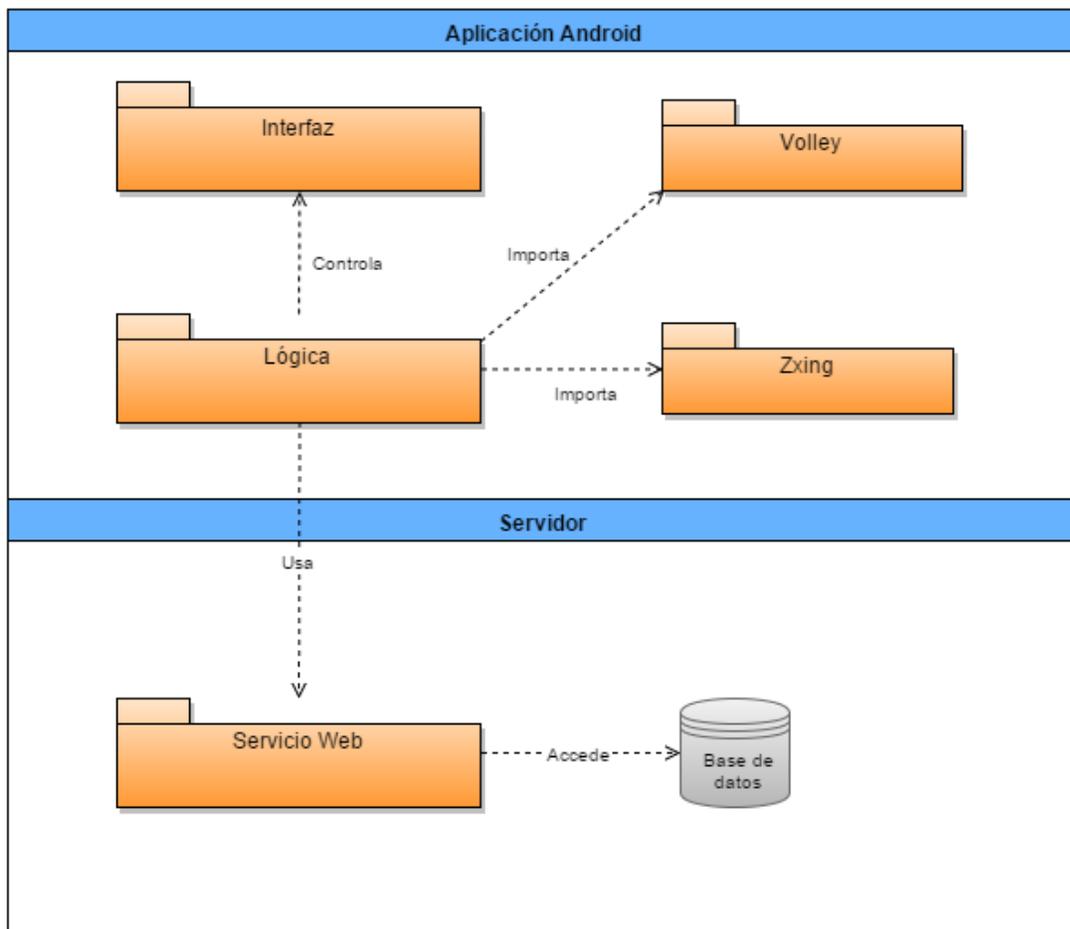


Figura 6: Diagrama de arquitectura de la aplicación Android

En la Figura 6 se puede ver que el software consta de dos partes principales: La aplicación Android que se ejecuta en el dispositivo móvil y el servidor donde se alojan el servicio web y la base de datos.

Aplicación Android: Sección del software que se ejecuta en el dispositivo Android. Está formado por las secciones de Lógica, Interfaz, ZXing y Volley. Los 2 últimos están vinculados a la parte lógica.

Lógica: Sección del programa formada por archivos Java encargados de hacer las operaciones de peticiones y ordenamiento de la información.

Interfaz: Sección que representa los archivos que constituyen al interfaz gráfica de usuario (GUI) del programa. Está formada por archivos .xml que están vinculados a los archivos Java de la sección Lógica de la aplicación.

ZXing: Librería externa que permite a la aplicación hacer uso de la cámara fotográfica del dispositivo móvil para escanear códigos de barra. Se explicará en detalle en la sección 7.3.

Volley: es una librería cuya función es facilitar las peticiones al servicio web desde toda la aplicación móvil sin necesidad de crear tareas asíncronas de forma manual. Se describe en detalle en la Sección 7.3.

Mediante Volley, la parte lógica del programa, luego de captar un código de barra, se encarga de hacer la petición al servicio web.

Servidor: Sección del software que representa el servidor que ejecuta el servicio web de la aplicación y aloja la base de datos.

Servicio web: Sección que representa los archivos encargados de conectar con la base de datos de productos. Los archivos se encuentran escritos en lenguaje PHP. El servicio web recibe el código de barras de la Aplicación Android para buscar en la base de datos para luego retornar la información en un archivo de formato JSON.

Base de datos: El lugar donde se encuentra la información que retorna la aplicación. Consta de una base de datos MySQL construida de la forma en que se ha mostrado en la parte 7.1 de este capítulo.

7.3 Diagrama de paquetes

La Figura 7 muestra los paquetes de archivos que conforman a la aplicación Android.

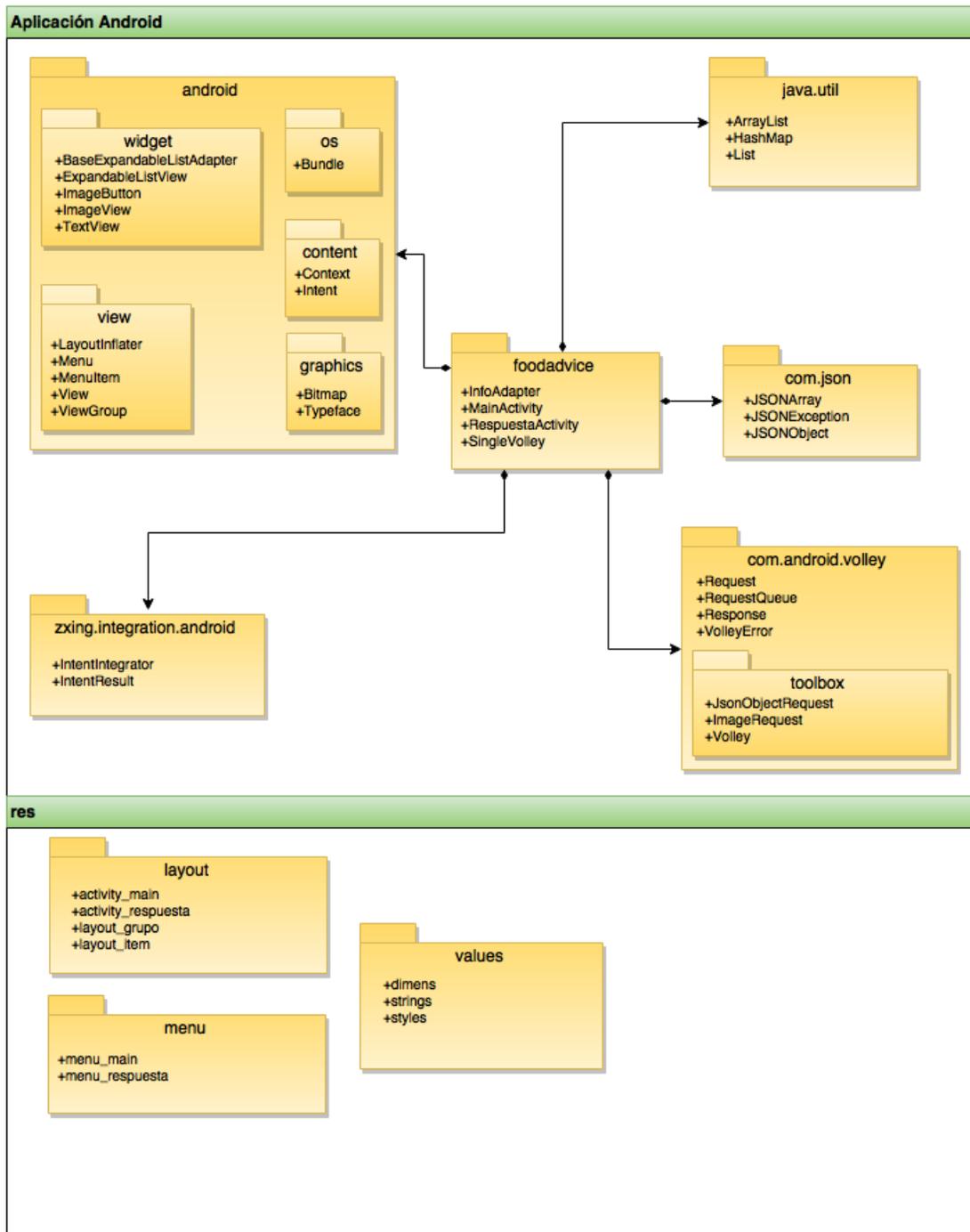


Figura 7: Diagrama de paquetes de la aplicación móvil.

Aplicación Android: En esta sección se encuentran los archivos Java que manejan el funcionamiento lógico del programa. Está conformado por los paquetes foodadvice, android, utilidades de Java, JSON, ZXing y Volley.

foodadvice: El nodo principal de la aplicación que contiene las actividades que conforman la aplicación que son MainActivity, la actividad de ejecución inicial; RespuestaActivity, encargada de renderizar la respuesta del servidor; InfoAdapter, encargada de renderizar la lista donde va la información; y SingleVolley, que se encarga de proporcionar una instancia de Volley para ser usada en toda la aplicación, es decir, usando el patrón de programación Singleton. Las clases se describen en detalle en el Anexo 10.2.

ZXing: Librería encargada de integrar las funciones de escáner de código de barras en la aplicación. Sus clases IntentIntegrator e IntentResult se encargan de iniciar la cámara y obtener el resultado según el formato del código de barras respectivamente.

Por un lado, IntentIntegrator al ser invocado se asegura de que el dispositivo tenga instalada la aplicación Barcode Scanner para poder obtener el código. En caso de no tenerla, se dirige al usuario a Google Play para descargar la aplicación. Por otro lado, IntentResult se encarga de, una vez escaneado el código de barras, obtener el número contenido en el código para devolverlo a MainActivity, que luego lo usa para hacer una petición mediante Volley al servicio web.

Volley: Librería externa encargada de simplificar el proceso de hacer peticiones a servicios web, haciendo que estas no tengan que hacerse de forma asíncrona. Sus clases principales son Request, RequestQueue, Response y VolleyError.

- **Request** es el objeto usado para representar las peticiones al servicio web, y se colocan en una cola de peticiones o RequestQueue.
- **RequestQueue** es la estructura utilizada para almacenar las peticiones (Request) al servicio web hechas desde toda la aplicación.
- **Response** es un objeto utilizado como escucha (Listener) ante los casos de obtener una respuesta positiva o negativa del servicio web al que Volley se conecta.
- **VolleyError** es un objeto de excepción utilizado para notificar de un error en la

ejecución de una petición.

Adicionalmente posee la carpeta toolbox con archivos que representan peticiones de tipos específicos además de Request, como objetos JSON e imágenes.

Android [19]: Gran librería que contiene todo lo necesario para que un programa para dispositivos android funcione como tal. En él se encuentran las siguientes sub librerías:

- **widget:** contiene clases para controlar y dar funciones a diversos objetos gráficos de la aplicación como botones, textos, imágenes, listas, entre muchos otros.
- **view:** contiene clases que controlan elementos básicos de interfaz de usuario como planos y contenedores de elementos.
- **graphics:** Librería de gráficos básica que contiene clases que controlan aspectos como filtros de color y objetos para construir imágenes desde secuencias binarias.
- **os:** Provee de funciones básicas del sistema operativo como paso de datos y comunicación entre procesos de algún dispositivo. Bundle, en este caso, se usa para empaquetar información y adjuntarlo a un Intent, del que se habla a continuación.
- **content:** Contiene clases para acceder y publicar datos en un dispositivo. Entre ellos se encuentra la clase Intent, utilizada para comunicar diferentes componentes de una aplicación.

Java.util [20]: Librería de Java que contiene colecciones de utilidades varias como, por ejemplo, estructuras de datos. En la aplicación se usan las estructuras de datos List, ArrayList y HashMap.

Com.JSON: Librería que implementa clases que permiten la creación y el manejo de objetos JSON.

Por otra parte tenemos los paquetes encargados de dibujar la interfaz.

Res: Sección que corresponde a los paquetes de archivos xml que dibujan la interfaz de la aplicación.

Layout: carpeta dedicada a los archivos de las interfaces gráficas del programa. Sus archivos son:

- **Activity_main:** Contiene los elementos gráficos de la pantalla de bienvenida (ver Figura 8). Es controlada por la clase MainActivity.
- **Activity_respuesta:** Contiene los elementos gráficos de la pantalla de lista desplegable. Es controlada por la clase RespuestaActivity (ver Figura 10).
- **Layout_grupo:** Corresponde al archivo gráfico de las barras desplegables de la lista de la figura 10.
- **Layout_item:** Corresponde al archivo gráfico con los elementos que dibujan los contenidos de la lista desplegable (ver Figura 11 y 12).

Values: carpeta con archivos que definen de cadenas de texto (strings.xml), las dimensiones en que se presentan los elementos en pantalla (dimens.xml) y los estilos de los artefactos gráficos usados en la aplicación (styles.xml).

Menú: Carpeta que guarda archivos que definen los elementos de los menús de cada pantalla de la aplicación.

7.4 Diseño interfaz y navegación

Se detalla a continuación las interfaces de la aplicación y su secuencia de navegación.



Figura 8: Pantalla de inicio de FoodAdvice.

La Figura 8 muestra la pantalla de bienvenida con el nombre de la aplicación y un botón que permite acceder a la pantalla de escaneo de productos.

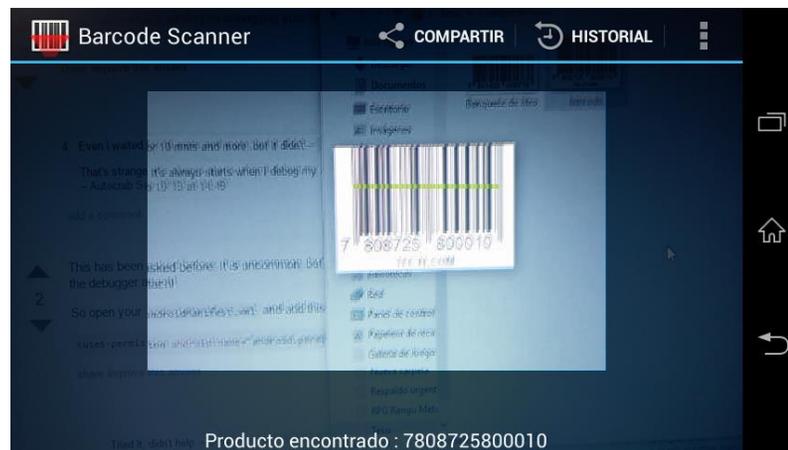


Figura 9: Pantalla de escaneo de códigos de barra.

La Figura 9 muestra la pantalla de escaneo a la que se accede desde la pantalla de bienvenida. Se muestra una barra horizontal roja por la que debe pasar el código de barra para ser reconocido. Cuando el código de barra se escanea, la barra se torna verde y se muestra en la parte inferior el número del código de barra que se usará para hacer la petición al servicio web. El código de barras utilizado en la figura corresponde a un producto real. Se ha copiado la numeración y se ha traspasado a digital mediante la página <http://barcode.tec-it.com/>.



Figura 10: Lista desplegable con los resultados de la consulta

La pantalla de respuesta muestra una lista desplegable que contiene toda la información relativa al producto cuyo código se ha escaneado (Figura 10). En el título se muestra la descripción general del producto, seguido de cuatro listas desplegables correspondientes a atributos del producto.

Al presionar sobre una de las listas, esta se abre y mostrará la información relacionada con el título de la lista desplegable (Figuras 11 y 12).



Figura 11: Lista desplegable con el grupo “Información general” abierto.



Figura 12: Lista con el grupo “otros” abierto.

Al pulsar “atrás” en el teléfono móvil estando en la pantalla de respuesta, se devuelve al usuario hasta la pantalla de inicio para realizar otro escaneo si así lo desea.

El siguiente diagrama de navegación (Figura 13) resume el proceso que sigue el usuario mientras la aplicación está en ejecución.

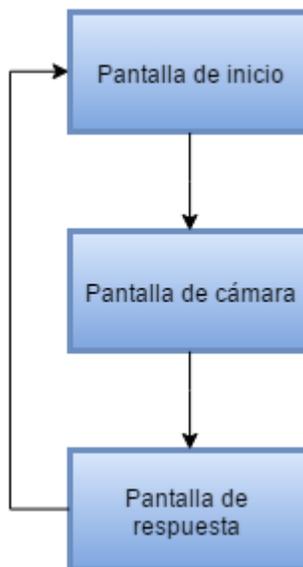


Figura 13: Diagrama de navegación

7.5 Diagrama de secuencia

La Figura 14 muestra un diagrama de secuencia que ilustra de mejor manera la interacción entre las clases de la aplicación con el servicio web y la base de datos.

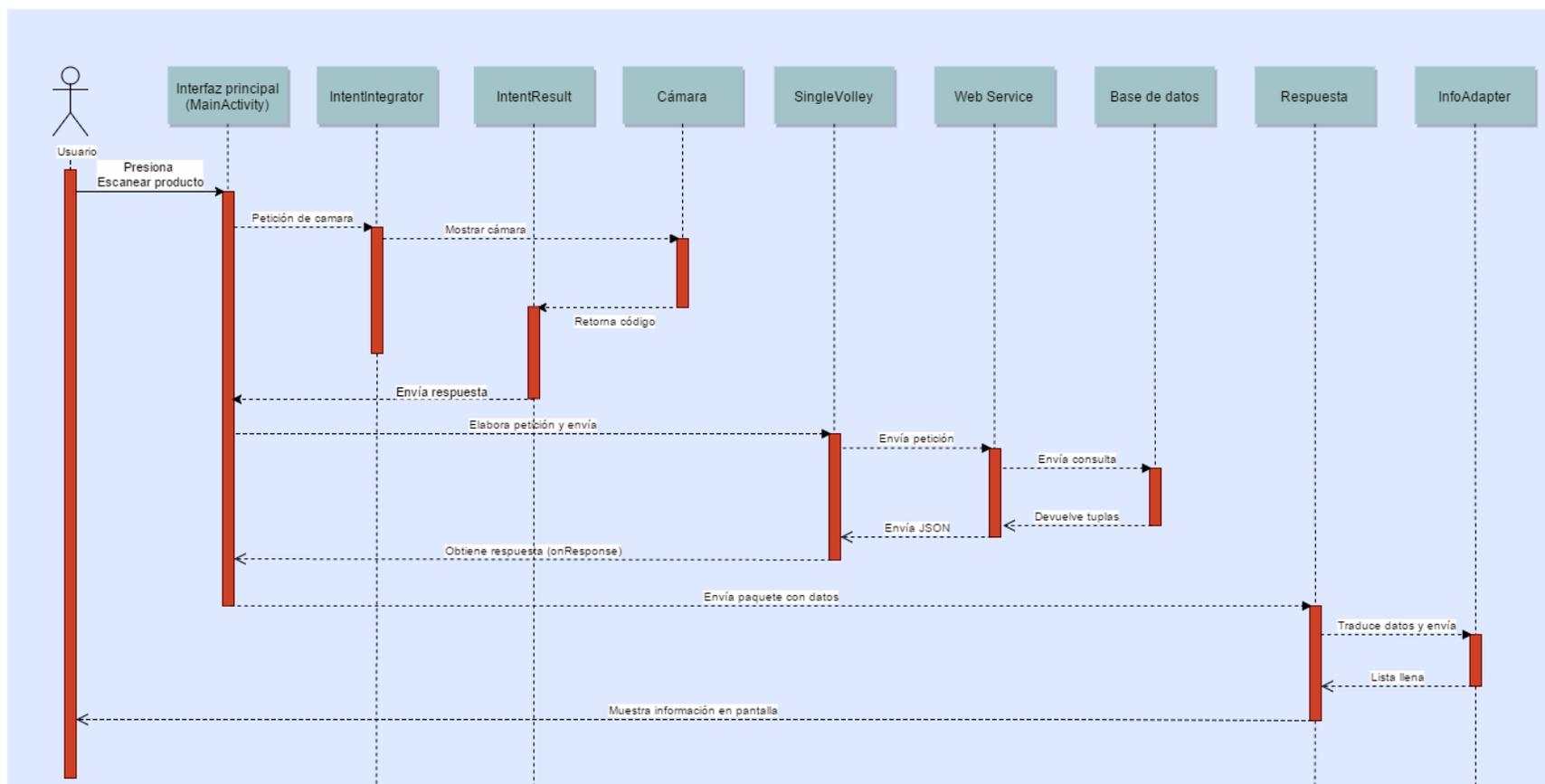


Figura 14: Diagrama de secuencia

Las interacciones comienzan con el usuario iniciando la aplicación entrando a la pantalla principal controlada por MainActivity. El usuario presiona el botón de escanear un producto lo que hace que MainActivity invoque a IntentIntegrator para hacer uso de la Cámara del dispositivo. La cámara capta el código e invoca a IntentResult para recuperar el número contenido en el código y pasárselo a MainActivity.

Con el código recibido, MainActivity envía una petición a Volley para recuperar la información mediante el servicio web. Volley contacta al servicio web que luego consulta a la base de datos por el producto correspondiente al código escaneado anteriormente. La base de datos devuelve el producto, el servicio web lo convierte en un objeto JSON y se lo devuelve a MainActivity. Ésta última convierte el JSON a String y lo manda a RespuestaActivity para traducirlo.

RespuestaActivity tiene la misión de poblar la lista final con la información recibida. Para eso descompone el objeto JSON en partes que añade a una lista que luego pasa a InfoAdapter, la clase encargada de poblar la lista desplegable con la información recibida. Ésta llena la lista y RespuestaActivity puede finalmente mostrar al usuario el resultado de la búsqueda en forma ordenada.

8 CONCLUSIONES

Luego de desarrollar el proyecto, cabe destacar que al ser una prueba de concepto de un proyecto con gran potencial, el software cumple la meta de escanear códigos de barra correctamente y, en base a este, obtener la información requerida en la definición del proyecto.

En relación a las tecnologías y herramientas utilizadas en el desarrollo del proyecto, se destaca el hecho que el uso de Android Studio y el uso de servicios web mediante la librería Volley para obtener la información de la base de datos, así como el previo conocimiento de programación de aplicaciones Android y el manejo en el lenguaje de programación Java y PHP, han ayudado a cumplir con lo planeado en esta fase del proyecto.

Académicamente, el desarrollo de esta aplicación ha servido para aplicar las habilidades de programación y planificación aprendidas durante el transcurso de la carrera de Ingeniería Civil en Informática.

En el ámbito personal, el proyecto ha servido para desarrollar más las habilidades técnicas, de investigación y de planificación personal, al contar con un tiempo acotado para el desarrollo del proyecto y al tener que usar tecnologías emergentes en el campo de la programación en Android.

9 BIBLIOGRAFÍA

- [1] Directores DGI de la Universidad del Bío-Bío. Última visita el 22 de noviembre de 2015:
http://www.ubiobio.cl/w/#Directores_DGI
- [2] BCN (2015). Guía legal sobre etiquetado de los alimentos. Última visita el 24 de noviembre de 2015: <http://www.bcn.cl/leyfacil/recurso/etiquetado-de-alimentos>.
- [3] GfK Adimark (2013). Percepción a cerca de la certificación y etiquetado de los alimentos. Estudio Imagen y posicionamiento INTA 2013. Última visita el 24 de noviembre de 2015: http://www.uchile.cl/documentos/percepcion-acerca-de-la-certificacion-y-etiquetado-de-los-alimentos-gfk-adimark_94480_0_3449.pdf
- [4] Android Studio Overview, Android Developers. Última visita el 24 de noviembre de 2015: <http://developer.android.com/intl/es/tools/studio/index.html>
- [5] PHPMyAdmin. Última visita el 21 de diciembre de 2015:
<https://www.phpmyadmin.net/>
- [6] Introducción: ¿Qué es Yii? Última Visita el 21 de diciembre de 2015:
<http://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii>
- [7] yEd – Graph Editor. Última visita el 21 de diciembre de 2015:
<https://www.yworks.com/products/yed>
- [8] draw.io. Definición propia. Última visita el 21 de diciembre de 2015:
<https://www.draw.io/>
- [9] W3C (2002). Web Services @ W3C. Última visita el 21 de diciembre de 2015:
<http://www.w3.org/2002/ws/>

- [10] Google (2015). Transmitting Network Data Using Volley. Última visita el 24 de noviembre de 2015:
<http://developer.android.com/intl/es/training/volley/index.html>
- [11] Lenguajes de Programación (2009). Programación Java. Última visita el 21 de diciembre de 2015: <http://www.lenguajes-de-programacion.com/programacion-java.shtml>
- [12] W3C. XML. Última visita el 21 de diciembre de 2015.
<http://www.w3.org/XML/>
- [13] JSON. Última visita el 21 de diciembre de 2015: <http://www.json.org/>
- [14] Zxing (2015). Zxing Readme. Última visita el 21 de diciembre de 2015:
<https://github.com/zxing/zxing/blob/master/README.md>
- [15] Fernandez, JM (2006). Tipos de dispositivos móviles. Recuperado el 26 de agosto de 2015: http://leo.ugr.es/l2ME/INTRO/intro_4.htm
- [16] Gigas.com. Cloud VPS Hosting: <https://gigas.com/cloud-vps?gclid=CLmr8pO9tMkCFVgHkQodSu8IYA>
- [17] Universia (2012). Las carreras técnicas con mejores sueldos y mayor empleabilidad. Última visita el 25 de noviembre de 2015:
<http://noticias.universia.cl/en-portada/noticia/2012/06/28/946481/carreras-tecnicas-mejores-sueldos-empleabilidad.html>
- [18] Crecenegocios.com (2014). El VAN y el TIR. Última visita el 25 de noviembre de 2015. <http://www.crecenegocios.com/el-van-y-el-tir/>
- [19] Google (2015). Información sobre librerías Android. Última visita el 24 de noviembre de 2015:

<http://developer.android.com/intl/es/reference/android/package-summary.html>

[20] Oracle (2015). Package java.util. Última visita el 26 de noviembre de 2015:

<http://docs.oracle.com/javase/7/docs/api/java/util/package-summary.html>

10 ANEXOS

10.1 Valores de planes de Gigas.com

	nova	pro	élite	summum
Mensual	USD 24.9	USD 38	USD 64	USD 90
Semestral	USD 129	USD 95 <small>(antes: USD 190)</small> BLACK FRIDAY 50%	USD 160 <small>(antes: USD 320)</small> BLACK FRIDAY 50%	USD 225 <small>(antes: USD 450)</small> BLACK FRIDAY 50%
Anual	USD 249 <small>(USD 20.8 / mes)</small>	USD 171 <small>(antes: USD 342)</small> BLACK FRIDAY 50%	USD 288 <small>(antes: USD 576)</small> BLACK FRIDAY 50%	USD 405 <small>(antes: USD 810)</small> BLACK FRIDAY 50%
	comprar	comprar	comprar	comprar

Figura 15: Costos de planes de hosting en Gigas.com

10.2 Diagrama de clases Java

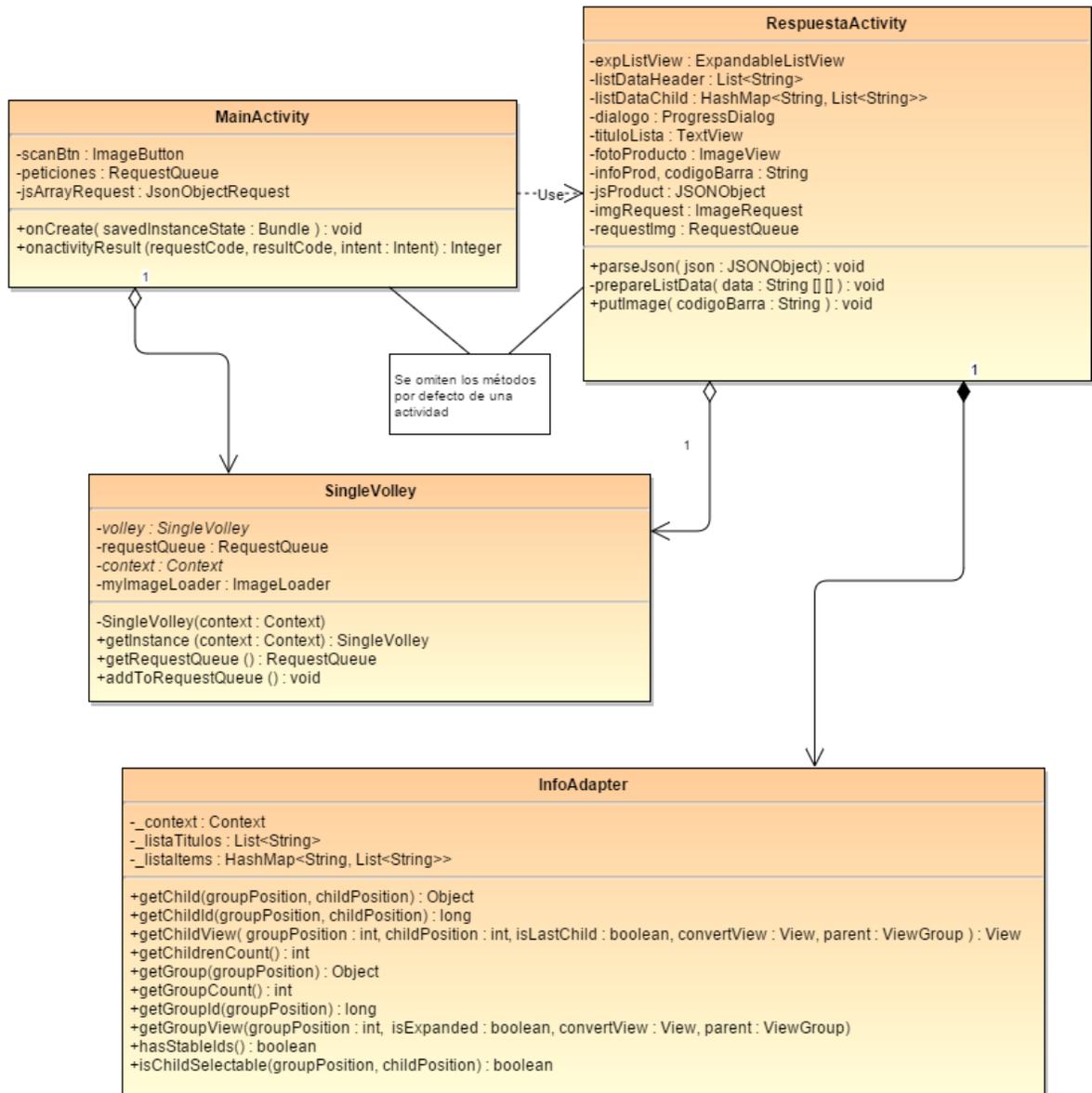


Figura 16: Diagrama de clases

La figura 16 muestra el diagrama de clases java de la aplicación en Android. El núcleo de la aplicación se conforma de 4 clases que se describen a continuación:

MainActivity: Es la actividad con la que la aplicación inicia. Usa la variable scanBtn para controlar el botón del archivo activity_main.xml.

En el método onCreate es donde se asignan sus variables y se hace la llamada al escáner de códigos de barra al pulsar el botón de escanear producto para luego, en el método onActivityResult, hacer la petición a Volley del producto y enviar el resultado JSON a la actividad RespuestaActivity mediante Intent.

RespuestaActivity: Actividad donde se traduce el archivo JSON recibido de MainActivity y se renderiza la pantalla controlada por activity_respuesta.xml. Posee una ExpandableListView para controlar la lista desplegable del archivo activity_respuesta.xml. Posee listas que hacen referencia a los títulos y contenido de cada ítem de la lista desplegable. Añade también un objeto de InfoAdapter para poblar la lista desplegable.

Sus métodos se encargan de descomponer y organizar el objeto JSON recibido de MainActivity.

InfoAdapter: Objeto encargado de poblar la lista desplegable, puesto que ResuestaActivity no puede hacerlo por si sola. Sus atributos hacen referencia a los títulos de cada ítem desplegable de la lista y a sus contenidos.

Sus métodos son para modificar el contenido de los grupos de la lista y sus contenidos.

SingleVolley: Clase encargada de entregar una instancia única de colas de peticiones Volley, de forma que no se creen colas de peticiones en cada clase de la aplicación ocupando memoria de manera innecesaria. Esto corresponde al patrón de programación Singleton.