**UNIVERSIDAD AUTÓNOMA GABRIEL RÉNE MORENO**

**FACULTAD DE INGENIERIA EN**

**DE LA COMPUTACION Y TELECOMUNICACIONES**

****

**Grupo # 10**

**Food Shazam aplicación móvil de reconocimiento visual de comidas a través de machine learning**

**INTEGRANTES:** Arhuata Pacosillo Rubi Guiselle 217101909

                                Aguilar Araya Matias                      217160204

  Caceres Cabezas Liz Yesenia 217059031

Loma Saldias Alejandro 214044130

Rejas Guerra Jose Matias 213112523

Terceros Arandia Jose Luis 214056497

**MATERIA:** Ingeniería de Software I

**DOCENTE:** Ing. Martínez Canedo Rolando Antonio

**GRUPO:** SC

**Santa Cruz – Bolivia**

Contenido

[1. Introducción 3](#_Toc92804696)

[2. Propósito del Proyecto 4](#_Toc92804697)

[3. Descripción del problema 4](#_Toc92804698)

[4. Estimaciones del software 5](#_Toc92804699)

[**4.1 Puntos del proyecto** 5](#_Toc92804700)

[4.1.1 Tamaño 5](#_Toc92804701)

[4.1.2 Complejidad 5](#_Toc92804702)

[**4.2 Ámbito del proyecto** 5](#_Toc92804703)

[4.2.1 Objetivos del proyecto 5](#_Toc92804704)

[4.2.1.1 Objetivo General 5](#_Toc92804705)

[4.2.1.2 Objetivo especifico 5](#_Toc92804706)

[4.2.2 Requerimientos principales 6](#_Toc92804707)

[4.2.3 Rendimiento 6](#_Toc92804708)

[4.2.4 Fiabilidad 6](#_Toc92804709)

[4.2.5 Restricciones 6](#_Toc92804710)

[5. Métricas del software 7](#_Toc92804711)

[5.1 Métricas MOT 7](#_Toc92804712)

[5 7](#_Toc92804713)

[5.2 Métricas MOF 7](#_Toc92804714)

[5.2 Métricas orientadas a la función 8](#_Toc92804715)

[6. Análisis de Riesgo 11](#_Toc92804716)

[7. Metodología de Desarrollo 13](#_Toc92804717)

[7.1 Flujo de trabajo: Captura de requisitos 13](#_Toc92804718)

[7.1.1 Identificar Actores y Casos de Uso 13](#_Toc92804719)

[7.1.2 Priorizar Caso de Uso 14](#_Toc92804720)

[7.1.3 Detallar Caso de Uso 14](#_Toc92804721)

[7.1.4 Estructura General de Caso de Uso 18](#_Toc92804722)

[7.2 Flujo de Trabajo: Análisis 19](#_Toc92804723)

[7.2.1 Análisis de Arquitectura 19](#_Toc92804724)

[7.2.2 Relación entre paquete y caso de uso 20](#_Toc92804725)

[7.2.3 Análisis de Casos de uso 21](#_Toc92804726)

[7.2.4 Análisis de Paquete 22](#_Toc92804727)

[7.3 Flujo de Trabajo: Diseño 22](#_Toc92804728)

[7.3.1 Arquitectura Lógica 22](#_Toc92804729)

[7.3.2 Arquitectura Física 23](#_Toc92804730)

[4.3.3 Diagrama de procesos 24](#_Toc92804731)

[**7.4** **Flujo de Trabajo: Implementación** 25](#_Toc92804732)

[7.4.1 Herramientas de desarrollo 25](#_Toc92804733)

[**7.5** **Flujo de Trabajo: Prueba** 26](#_Toc92804734)

[7.5.1 Planificar Pruebas 26](#_Toc92804735)

[7.5.2 Diseñar Pruebas 27](#_Toc92804736)

[8 Factores de Calidad 28](#_Toc92804737)

[9 Manual Bibliografía 30](#_Toc92804738)

[Anexos 30](#_Toc92804739)

# Introducción

Siempre se busca realizar labores de forma eficientes y eficaces, es así que las personas encontraron la manera de almacenar información, ya fuera por considerarla como una simple actividad o por tener la importancia que tiene. En los últimos tiempos hemos visto cómo la tecnología e innovación avanzaron considerablemente y esto lleva a que las personas tiendan a manejar su información de una manera óptima, así también cubrir las exigencias de los clientes, porque existe una ventaja con la utilización de las tecnologías de hoy en día resaltan y que a la vez facilitan la interacción entre cualquier dispositivo electrónico ya sea computadoras, celulares, etc.

En la actualidad los avances tecnológicos son cada vez mejores y permiten que las personas tengan acceso a varias aplicaciones en sus teléfonos móviles que permitan de alguna manera contribuir y cubrir ciertas necesidades que requieran en cualquier ámbito; y uno de ellos que debemos de lograr que sea parte fundamental de sus necesidades diarias es llevar un control de nuestra alimentación o de lo que comemos.

El proyecto a realizar refiere a la creación de un diseño para la implementación de una aplicación móvil de restaurante de comidas rápidas tipo shaman, le permite detectar información sobre una imagen donde se pretenderá demostrar que mediante un api de reconocimiento de imágenes todo tipo de platos de comidas podrá ser descrito desde de los ingredientes y las recetas, partiendo de un origen con buenas bases, y en donde se evidencie su futuro a nivel comercial.

# Propósito del Proyecto

El propósito del proyecto es desarrollar una aplicación móvil que permita extraer información de las etiquetas de los alimentos mediante reconocimiento visual de ingredientes y comidas a través de machine learning nuestro objetivo es garantizar que la información almacenada en la aplicación sea de ayuda para las personas en sus comidas diarias de esta manera el usuario mantendrá un control estricto de los alimentos que consuman, la aplicación será íntegra, veraz y con un tiempo de respuesta eficiente para alcanzar la confiabilidad de los usuarios

****

# Descripción del problema

La creciente población en el país ha generado una fuerte demanda que no puede ser cubierta por las empresas grandes, medianas y pequeñas únicamente a través de los canales de ventas tradicionales ya que no das abasto a todos los posibles clientes. Lo que impulsa a estas empresas a incurrir en las tecnologías del **machine learning** entre otros.

# Estimaciones del software

## **4.1 Puntos del proyecto**

### 4.1.1 Tamaño

El tamaño del proyecto es mediano ya que la cantidad de funciones que debe realizar la aplicación no es masiva.

### 4.1.2 Complejidad

La complejidad del proyecto es mediana debido a que el grado de desconocimiento de los componentes, herramientas y tecnologías que se emplearan en el desarrollo de la aplicación no es significativo, debido a que contamos con un equipo con experiencia en el desarrollo de aplicaciones web.

## **4.2 Ámbito del proyecto**

## 4.2.1 Objetivos del proyecto

### 4.2.1.1 Objetivo General

Desarrollar una aplicación móvil de reconocimiento visual de comidas a través de machine learning.

### 4.2.1.2 Objetivo especifico

* Definir el contexto y alcance del proyecto, identificar los participantes e intentar definir áreas de incertidumbre.
* Definir los requerimientos funcionales y no funcionales
* Identificar claramente las tecnologías para el desarrollo de la aplicación móvil
* Describir a detalle los procesos y funcionalidades que presenta la aplicación móvil
* Realizar labores de análisis y diseño adaptados de la ingeniería del software, desarrollando análisis y modelos de diseño que definan los requisitos.
* Implementar el software de acuerdo al diseño realizado.
* Realizar las pruebas para garantizar que el software desarrollado cumpla con los requerimientos del usuario.

## 4.2.2 Requerimientos principales

* Debe permitir registrar, modificar la información de los clientes.
* Debe ser posible registrar, modificar la información de los usuarios y también bloquear a estos en caso de que sea necesario.
* El software debe registrar los productos y listar estos.
* Hacer una búsqueda de productos por categoría y por nombre.
* Debe ser capaz de mostrar sucursales o lugares.

## 4.2.3 Rendimiento

Debido a que el proyecto se trata de una aplicación móvil que permita extraer información de las etiquetas de los alimentos mediante reconocimiento visual de ingredientes y comidas a través de machine learning el rendimiento de la aplicación debe ser alta puesto que es necesario un tiempo de respuesta ágil por la naturaleza de los datos que manejara.

## 4.2.4 Fiabilidad

El software debe ser lo más confiable posible para hacer su trabajo correctamente bajo cualquier contexto de operación debido a que el software trabajara bajo la necesidad y exigencia de productos de una empresa.

## 4.2.5 Restricciones

**Restricciones técnicas:**

* Se utilizará como metodología el Proceso Unificado de Desarrollo de software.
* Para el desarrollo de la agenda se utilizará el lenguaje de programación Node Js para el lado del servidor y React para el lado del cliente.
* Como IDE de desarrollo se tomará Visual Studio Code
* Para la aplicación se utilizarán servidores con sistema operativo Windows.
* Para la documentación del desarrollo se dará uso de una herramienta CASE Enterprise Architect.

**Restricciones de recursos:**

* El tiempo dado para el desarrollo parcial del proyecto es de 4 meses, lo que implica mayor esfuerzo por parte de los desarrolladores.

**Restricciones legales:**

* No existe alguna disposición legal que regula el funcionamiento de este software.

# Métricas del software

## 5.1 Métricas MOT

Tiempo **≈** 3 meses

Gente. - 5 personas.

Errores ≈ 50.

Defectos ≈ 10

KLDC ≈ 1088

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto | KLDC | Costo\*1000 | Tiempo  (mes) | Errores | Defectos | Esfuerzo  (PM) | Gente | Pág.  documentadas |
| FOOD SHAZAM | 1088 | 5 | 3 | 50 | 10 | 10 | 5 | 45 |

|  |  |  |
| --- | --- | --- |
| Proyecto | Calidad | Productividad |
| FOOD SHAZAM | = |  |

Conclusión.- Se obtuvo `` fallas por KLDC’’ y `` LDC escribió cada persona al mes’’

## 5.2 Métricas MOF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Factor de Peso** | | |  |
| **Parámetros de Medición** | **Cuenta** | **Simple** | **Medio** | **Complejo** | **Total** |
| # de entradas de usuario | **12** | **0** | **10** | **6** | **120** |
| # de salidas de usuario | **8** | **5** | **20** | **2** | **160** |
| # de peticiones de usuario | **15** | **3** | **4** | **6** | **60** |
| # de archivos | **27** | **7** | **10** | **15** | **270** |
| # de interfaces externas | **1** | **5** | **7** | **10** | **10** |
|  |  |  |  |  | **620** |

## 5.2 Métricas orientadas a la función

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | No influencia | Incidental | Moderado | Medio | Significativo | Esencial | Subtotal |
| Factores | 0 | 1 | 2 | 3 | 4 | 5 |
| 1.- ¿Requiere el sistema copias de seguridad y de recuperación fiables? |  |  |  | SI |  |  | 3 |
| 2.- ¿Se requiere comunicación de datos? |  |  |  |  |  | SI | 5 |
| 3.- ¿Existen funciones de procesamiento distribuido? |  |  |  | SI |  |  | 3 |
| 4.- ¿Es crítico el rendimiento? |  |  |  |  | SI |  | 4 |
| 5.- ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado? |  |  |  | SI |  |  | 3 |
| 6.- ¿Requiere el sistema entrada de datos interactiva? |  |  |  |  | SI |  | 4 |
| 7.- ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones? |  |  |  |  | SI |  | 4 |
| 8.- ¿Se actualizan los archivos maestros de forma interactiva? |  |  |  | SI |  |  | 3 |
| 9.- ¿Son complejas las entradas, las salidas, los archivos o las peticiones? |  |  | SI |  |  |  | 2 |
| 10.- ¿Es complejo el procesamiento interno? |  |  |  | SI |  |  | 3 |
| 11.- ¿Se ha diseñado el código para ser reutilizable? |  |  |  | SI |  |  | 3 |
| 12.- ¿Están incluidas en el diseño la conversión y la instalación? |  |  |  |  | SI |  | 4 |
| 13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones? |  | SI |  |  |  |  | 1 |
| 14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario? |  |  |  |  | SI |  | 4 |
| Total |  |  |  |  |  |  | 46 |

**K.L.D.C**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proyecto** | **KLDC** | | | |
| **Optimista** | **Pesimista** | **Más Probable** | **Esperados** |
| **Food Shazam** | 6 | 7.5 | 9.2 | ***8*** K.L.D.C |

***VE = [****Optimista + 4\*(Más probable) + Pesimista] / 6*

***VE= 6 + 4 \*(9.2) + 7.5 ] / 6 = 8.384***

Se estima que el proyecto puede tener aproximadamente ***8.384 ≈ 8*** K.L.D.C.

**COCOMO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tipo objeto** | **Cuenta** | **Factor de peso** | | | **Total** |
| **Simple** | **Medio** | **Avanzado** |
| **Interfaces** | 14 | **1** | **2** | **3** | 28 |
| **Reportes** | 1 | **2** | **5** | **8** | 5 |
| **Componentes 4GL** | 4 | **\_** | **\_** | **9** | 36 |
|  |  |  |  | **P.O.:** | **69** |

El proyecto presenta 69 Puntos objetos. Se estima que se tendrá un 25 % de reutilización de los objetos empleados en el proyecto

**PON =**

**PON =**

**PON = 51.75**

Se obtuvo **51.75** nuevos **puntos de objetos** a desarrollar para el proyecto.

Con **Cocomo** estimamos la cantidad de interfaces, componentes y reportes a desarrollar.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Proporciones de Productividad | Muy Baja | Baja | Normal | Alta | Muy Ata |
| Capacidad y Madurez del Desarrollador |  |  |  |  |  |
| Capacidad y Estructuración del Entorno |  |  |  |  |  |
| Total | 7 | 7 | 13 | 25 | 50 |

**PROD = 13 + 7**

**PROD = 20**

E =69/ 20 = 3.45 ≈ 3

Se estima que se tendrá un esfuerzo de personas-mes.

**Estimación de Tiempo**

Tmin= 8.14\*(LDC/P)^0.43 en meses para tmin> 6

Tmin= 8.14\*(20000/2000)^0.43 = 6.63 meses

# Análisis de Riesgo

|  |  |  |
| --- | --- | --- |
| **Riesgo** | **Probabilidad (%)** | **Impacto** |
| R1. Incumplimiento con la fecha de entrega. | 20 | Critico |
| R2. El cliente no estará conforme con el software. | 40 | Critico |
| R3. Abandono de personal. | 10 | Moderado |
| R4. Al programador le tomara tiempo dominar el lenguaje. | 20 | Critico |
| R5. Problemas de comunicación en el equipo. | 15 | Moderado |
| R6. Nuevos integrantes programadores sin experiencia en el tipo software. | 30 | Significativo |
| R7. Mala planificación en el diseño | 30 | Significativo |
| R8. Incumplimiento con el horario asignado a los programadores | 35 | Significativo |
| R9.-Incumplimiento con las fechas intermedias | 25 | Significativo |

**Elaborar plan de aversión**

* **No influye (NI)**
* **Moderado (MO)**
* **Significativo (SG)**
* **Critico (CR)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgo** | **Probabilidad** | **Impacto** | **Plan de aversión** | |
|  | **(%)** |  | **Reducción** | **Supervisión** |
| R1: Integrante del equipo de desarrollo se retira del proyecto. | 50 | SG | -Firmar Contrato de trabajo con los integrantes del equipo.  -Motivar a los integrantes del equipo a través de dinámicas de trabajo.  -Tener una buena comunicación entre los integrantes del equipo. | -No saturar de manera desmedida el trabajo. |
| R2: Pérdida de información, ya sea por motivos técnicos, infección de virus, etc. | 20 | CR | -Realizar periódicamente copias de seguridad de la información.  -Documentar lo avanzado en backups.  -Contar con antivirus actualizados en todos los equipos. | - Usar diferentes dispositivos de almacenamiento (CD, Pen Drive, etc) que sean seguros, de buena calidad, etc.  -Revisar la documentación realizada. |
| R3: Estrategia de desarrollo desconocido | 20 | MO | -Trabajar con estrategias empleadas previamente | -Trabajar de manera organizada y minuciosamente, empleando toda la documentación posible acerca de la estrategia a utilizar. |
| R4: Herramienta de desarrollo desconocida | 25 | SG | -Trabajar con herramientas empleadas previamente. | -Buscar información y ayuda en el manejo de dicha herramienta. |
| R5: Fallas de hardware o algún equipo. | 30 | SG | -Realizar mantenimiento de equipos periódicamente. | -Adquirir equipos y/o Accesorios Informáticos con garantía. |
| R6: Mala estimación de tiempo debido a no contemplar actividades ajenas al proyecto. | 45 | CR | -En el momento de realizar la planificación de tiempos y actividades, realizarla siguiendo un calendario en el cual contemple posibles eventualidades y tiempos reales de trabajo de los integrantes del equipo. | -Contemplar dentro de la planificación de tiempo un tiempo de demora asumiendo cualquier tipo de eventualidad. |

# Metodología de Desarrollo

# Flujo de trabajo: Captura de requisitos

### 7.1.1 Identificar Actores y Casos de Uso

#### Aplicación Usuario



### Priorizar Caso de Uso

#### Aplicación Usuario

|  |  |
| --- | --- |
|  | **PRIORIODAD** |
| Obtener Ingrediente | Significativo |
| Obtener Comida | Critico |
| Visualizar Ubicación | Normal |
| Visualizar Información del Restaurant | Normal |

### Detallar Caso de Uso

#### Aplicación Arrendatario

##### Obtener Ingrediente



|  |  |
| --- | --- |
| Nombre de Caso de Uso | CU1 OBTENER INGREDIENTE |
| Propósito | Obtiene Resultado de la comida |
| Descripción |  |
| Actores | usuario |
| Actor Iniciador | usuario |
| Flujo Principal | 1. Presiona el icono de la cámara    1. Captura la comida    2. Devuelve un listado de ingrediente de la comida |
| Excepciones | 1.2 Error de tipos de datos  1.2 No tiene buena resolución la camara |

##### Obtener Comida



|  |  |
| --- | --- |
| Nombre de Caso de Uso | CU2 OBTENER COMIDA |
|  |  |
| Propósito | Sacar foto a la comida y devuelve los ingredientes |
| Descripción |  |
| Actores | usuario |
| Actor Iniciador | usuario |
| Post Condición |  |
| Flujo Principal | 1. Presiona el icono de la cámara    1. Captura la comida    2. Devuelve un listado de ingrediente de la comida |
| Excepciones |  |

##### Visualizar Ubicación



|  |  |
| --- | --- |
| Nombre de Caso de Uso | CU3 VISUALIZAR UBICACIONES |
|  |  |
| Propósito | El Arrendatario puede visualizar los precios de las casas, como de los contratos como también de los contratos de alquiler. |
| Descripción |  |
| Actores | Usuario |
| Actor Iniciador | Usuario |
| Post Condición |  |
| Flujo Principal | 1. Presiona un lado del modal 2. Y saldrá el mapa de los restaurant |
| Excepciones | 1.2 Falta rellenar algún filtro  2.3 Error de filtrado no pudo conectarse al Servidor |

##### Visualizar Información del Restaurant

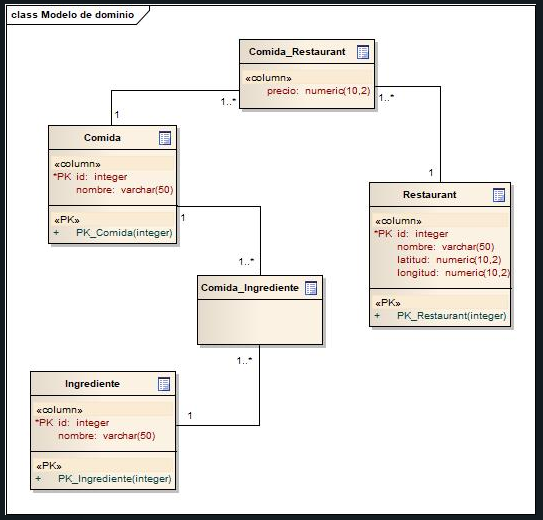
****

|  |  |
| --- | --- |
| Nombre de Caso de Uso | Cu.4 GESTIONAR TIPO DE INMUEBLE |
|  |  |
| Propósito | Visualizar cada uno de los restaurant. |
| Descripción | Se podrá visualizar cada uno de los restaurant que están a la disposición para el cliente. |
| Actores | Usuario |
| Actor Iniciador | Usuario |
| Pre Condición | Ninguna |
| Post Condición | Ninguna |
| Flujo Principal | 1. Presiona un lado del modal    1. Y saldrá el mapa de los restaurant |
| Excepciones | 2.3 Error de tipos de datos   * 1. El restaurant no existe |

### Estructura General de Caso de Uso



# Diagrama de clases



## Script

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Base de datos: `bloomingproyecto`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `comida`

--

CREATE TABLE `comida` (

`id` int(11) NOT NULL,

`nombre` varchar(191) COLLATE utf8mb4\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `comida`

--

INSERT INTO `comida` (`id`, `nombre`) VALUES

(2, 'pizza'),

(3, 'sushi');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `comidasrestaurants`

--

CREATE TABLE `comidasrestaurants` (

`comidaId` int(11) NOT NULL,

`restaurantId` int(11) NOT NULL,

`Precio` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `comidasrestaurants`

--

INSERT INTO `comidasrestaurants` (`comidaId`, `restaurantId`, `Precio`) VALUES

(2, 1, 25),

(3, 1, 25);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `ingrediente`

--

CREATE TABLE `ingrediente` (

`id` int(11) NOT NULL,

`nombre` varchar(191) COLLATE utf8mb4\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `ingrediente`

--

INSERT INTO `ingrediente` (`id`, `nombre`) VALUES

(8, 'salsa de tomate'),

(9, 'mozarela'),

(10, 'orégano'),

(11, 'sal'),

(12, 'aceite de oliva'),

(13, 'harina'),

(14, 'levadura'),

(15, 'arroz de sushi'),

(16, 'azúcar'),

(17, 'vinagre de arroz'),

(18, 'surimi'),

(19, 'salmón'),

(20, 'nori'),

(21, 'semillas de sésamo'),

(22, 'pepino'),

(23, 'aguacate'),

(24, 'salsa de soja');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `ingredientescomidas`

--

CREATE TABLE `ingredientescomidas` (

`ingredienteId` int(11) NOT NULL,

`comidaId` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `ingredientescomidas`

--

INSERT INTO `ingredientescomidas` (`ingredienteId`, `comidaId`) VALUES

(8, 2),

(9, 2),

(10, 2),

(11, 2),

(12, 2),

(13, 2),

(14, 2),

(15, 3),

(16, 3),

(17, 3),

(18, 3),

(19, 3),

(20, 3),

(21, 3),

(22, 3),

(23, 3),

(24, 3);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `restaurant`

--

CREATE TABLE `restaurant` (

`id` int(11) NOT NULL,

`nombre` varchar(191) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`latitud` double NOT NULL,

`longitud` double NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `restaurant`

--

INSERT INTO `restaurant` (`id`, `nombre`, `latitud`, `longitud`) VALUES

(1, 'Safari', -17.778467, -63.199437),

(2, 'Casa del cerdo', -17.803716, -63.193358),

(3, 'ABDCASD', -17.772604, -63.1602);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `\_prisma\_migrations`

--

CREATE TABLE `\_prisma\_migrations` (

`id` varchar(36) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`checksum` varchar(64) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`finished\_at` datetime(3) DEFAULT NULL,

`migration\_name` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`logs` text COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`rolled\_back\_at` datetime(3) DEFAULT NULL,

`started\_at` datetime(3) NOT NULL DEFAULT current\_timestamp(3),

`applied\_steps\_count` int(10) UNSIGNED NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `\_prisma\_migrations`

--

INSERT INTO `\_prisma\_migrations` (`id`, `checksum`, `finished\_at`, `migration\_name`, `logs`, `rolled\_back\_at`, `started\_at`, `applied\_steps\_count`) VALUES

('60a7efe9-00e0-4013-a62c-92b66f70d23d', '16102d5dc500fe12f238aa6768a9e39c9deb6b87a71e6be96442b47a5ea2f9bc', '2022-01-10 03:03:11.572', '20220104150739\_init', NULL, NULL, '2022-01-10 03:03:11.374', 1);

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `comida`

--

ALTER TABLE `comida`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `comidasrestaurants`

--

ALTER TABLE `comidasrestaurants`

ADD PRIMARY KEY (`comidaId`,`restaurantId`),

ADD KEY `ComidasRestaurants\_restaurantId\_fkey` (`restaurantId`);

--

-- Indices de la tabla `ingrediente`

--

ALTER TABLE `ingrediente`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `ingredientescomidas`

--

ALTER TABLE `ingredientescomidas`

ADD PRIMARY KEY (`ingredienteId`,`comidaId`),

ADD KEY `IngredientesComidas\_comidaId\_fkey` (`comidaId`);

--

-- Indices de la tabla `restaurant`

--

ALTER TABLE `restaurant`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `\_prisma\_migrations`

--

ALTER TABLE `\_prisma\_migrations`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT de las tablas volcadas

--

--

-- AUTO\_INCREMENT de la tabla `comida`

--

ALTER TABLE `comida`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

--

-- AUTO\_INCREMENT de la tabla `ingrediente`

--

ALTER TABLE `ingrediente`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=25;

--

-- AUTO\_INCREMENT de la tabla `restaurant`

--

ALTER TABLE `restaurant`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

--

-- Restricciones para tablas volcadas

--

--

-- Filtros para la tabla `comidasrestaurants`

--

ALTER TABLE `comidasrestaurants`

ADD CONSTRAINT `ComidasRestaurants\_comidaId\_fkey` FOREIGN KEY (`comidaId`) REFERENCES `comida` (`id`) ON UPDATE CASCADE,

ADD CONSTRAINT `ComidasRestaurants\_restaurantId\_fkey` FOREIGN KEY (`restaurantId`) REFERENCES `restaurant` (`id`) ON UPDATE CASCADE;

--

-- Filtros para la tabla `ingredientescomidas`

--

ALTER TABLE `ingredientescomidas`

ADD CONSTRAINT `IngredientesComidas\_comidaId\_fkey` FOREIGN KEY (`comidaId`) REFERENCES `comida` (`id`) ON UPDATE CASCADE,

ADD CONSTRAINT `IngredientesComidas\_ingredienteId\_fkey` FOREIGN KEY (`ingredienteId`) REFERENCES `ingrediente` (`id`) ON UPDATE CASCADE;

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

## Flujo de Trabajo: Análisis

### Análisis de Arquitectura



* **Paquete Mapas.-** este paquete contiene las funcionalidades del usuario en la aplicación móvil
* **Paquete Comida.-** el paquete contiene las funcionalidades de los ingredientes de la comida y buscar la comida

### Relación entre paquete y caso de uso

Mapas y Comida



### Análisis de Casos de uso

CU Obtener Comida



CU Visualizar Informacion del Restaurant



CU Visualizar Ubicacion



### Análisis de Paquete



## Flujo de Trabajo: Diseño

### Arquitectura Lógica



### Arquitectura Física



### 4.3.3 Diagrama de procesos

Diagrama de Secuencia

Visualizar Ubicación y Visualizar Restaurant

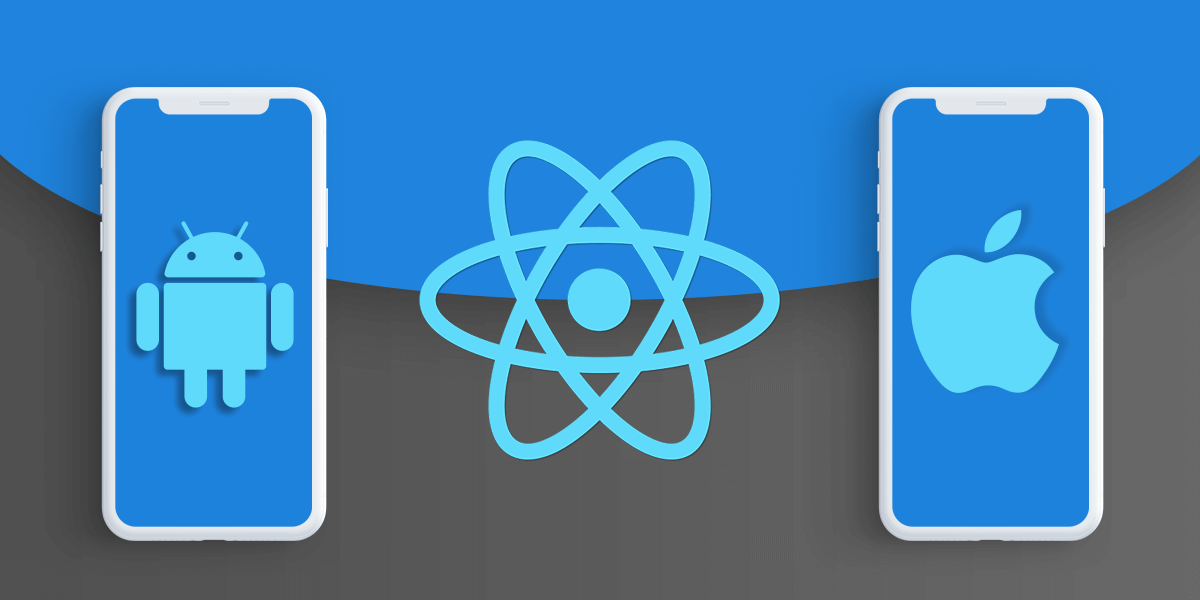




## **Flujo de Trabajo: Implementación**

### Herramientas de desarrollo

**Framework:**



**React-Native:** React Native es un Framework JavaScript para crear aplicaciones reales nativas para iOS y Android, basado en la librearía de JavaScript React para la creación de componentes visuales, cambiando el propósito de los mismos para, en lugar de ser ejecutados en navegador, correr directamente sobre las plataformas móviles nativas, en este caso iOS y Android

**Sistemas Operativos**

**Android:** en un Sistema Operativo además de una plataforma de Software basada en el núcleo de Linux. Diseñada en un principio para dispositivos móviles, Android permite controlar dispositivos por medio de bibliotecas desarrolladlas o adaptados por Google mediante el lenguaje de programación Java

**IOS:** es un sistema operativo propiedad de Apple orientado a sus dispositivos móviles táctiles como el iPhone, el iPod touch el iPad. Cuenta con actualizaciones periódicas que están disponibles para su descarga y actualización a través de iTunes, que es el software gratuito e indispensable para manipular y sincronizar toda clase da archivos en estos dispositivos.

Tanto Android como IOS nos permitieron hacer pruebas al momento de desarrollar la aplicación.

**Otras Herramientas**



**Enterprise Architect:** es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad

## **Flujo de Trabajo: Prueba**

### Planificar Pruebas

Con el propósito de asegurar el correcto funcionamiento, se ha desarrollado un plan de pruebas para la aplicación, estableciendo el alcance, diseño del software, objetivos, roles y recurrencia.

Para la realización de las pruebas a la aplicación se utilizará el método de prueba “caja negra”.

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisará por completo todos los requerimientos funcionales para un programa.

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

* Funciones incorrectas o faltantes.
* Errores de interfaz.
* Errores de la estructura de datos o en el acceso de la base de datos.
* Errores de comportamiento o rendimiento.
* Errores de inicialización y terminación.

Los casos de uso sobre los cuales se realizarán las respectivas pruebas son:

* CU Obtener Comida
* CU Obtener Ingredientes
* CU Visualizar Ubicaciones
* CU Visualizar Información del Restaurant

### Diseñar Pruebas

**Visualizar Ubicación**

El Usuario podrá visualizar las zonas del restaurant

**Procedimiento de prueba para Visualizar Ubicacion**

* Presionara fuera del modal de ingrediente
* Y se abre google map y podrá visualizar el mapa del restaurant cercanos

**Obtener Comida**

El usuario podrá tomar una foto y le reconocerá la comida captura por la cámara del celular del usuario.

**Procedimiento de prueba para Obtener La comida**

* Presionara el icono de cámara de la app
* Y podrá captura la comida que desea saber que ingredientes tiene o en qué lugar se encuentra.

# Factores de Calidad

**1 Software funcional:**

El software cumple con los requerimientos funcionales expresados por las

necesidades de los usuarios de manera que cumpla con la seguridad e

integridad, protegiendo los datos las de entrada como como de salida.

**2 Software de Confiabilidad:**

El software cuenta con variables globales de control para evitar fallas por errores

del sistema y respalda los datos del usuario manejando una base de datos en la

nube.

**3 Software de eficiencia:**

El software es capaz de hacer un buen uso de los recursos disponibles del

hardware relacionado con el software para brindar buenos tiempos de respuestas

con respecto a las funcionalidades.

**4 Usabilidad del software:**

El software está relacionado con términos y conceptos del ámbito de la salud

entendible para los usuarios, tanto así que cuenta con interfaces intuitivas y

comprensibles que facilitan la interacción con la aplicación móvil.

**5 Mantenibilidad del software:**

El software estará disponible en una tienda de aplicaciones móviles y tendrá la

posibilidad de ser evaluado por los usuarios y así mejorar el software ofreciendo

actualizaciones y mejoras tanto en IOS como en Android. Gracias a la

arquitectura del software nos permitirá hacer cambios factibles bajo riesgos de

efectos inesperados (estabilidad).

**6 Facilidad de pruebas del software:**

El software cuenta con una arquitectura que define la estructura basada en

componentes que nos facilitara aplicar estrategia de prueba para la detección de

errores

**7 Portabilidad del software:**

El software está desarrollado para soportar las diferentes versiones del sistema

operativo Android e IOS al ser desarrollado en una herramienta multiplataforma;

React Native.

# Manual de usuario

**Aplicación Arrendatario**

**introducción general**

Este manual de Usuario le ayudará a poder utilizar la aplicación “Busca Comida”, sin

mayores costos donde encontrará las funcionalidades principales de la misma que garantizará

una óptima utilización.

**Características**

Esta aplicación fue diseñada para poder facilitar una búsqueda de comida que no son conocidas o lugar del restaurant.

**sistema operativo**

para poder hacer uso de la aplicación debe tener una tablet o un celular android o ios

no obstante, es también es compatible con las diferentes versiones de android y ios y es

recomendable tener la última actualización

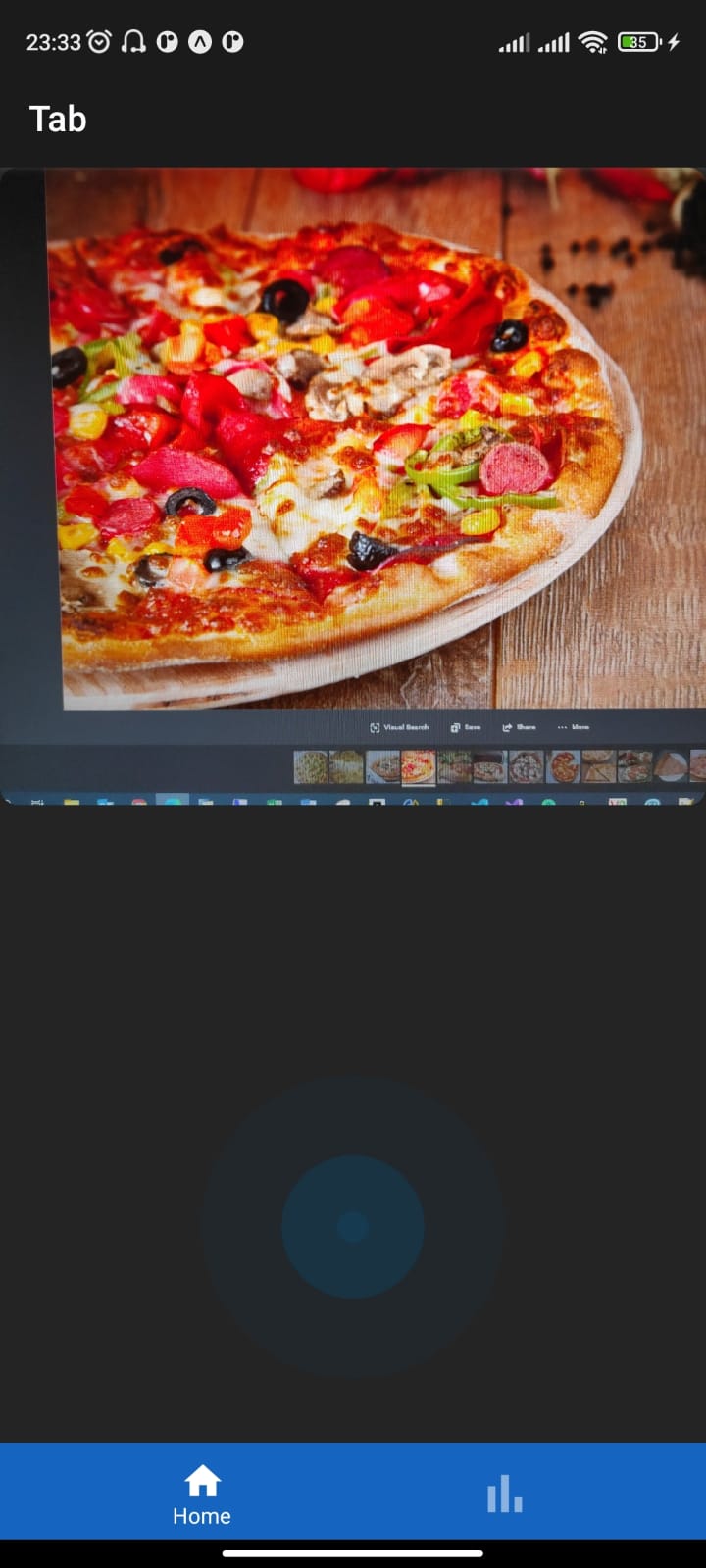
ahí tenemos la vista principal de “Busca Comida**”** donde se ingresa el usuario para

poder realizar la búsqueda de la comida o restaurant.

**funciones principales**

**Buscar Comida**

En esta función podrá sacar una foto de la comida y la aplicación le responderá con el lugar e ingredientes de la comida.



# Bibliografía

**Libros**

* Jacobson, Booch, Rumbaugh -“PUDS” El proceso Unificado de Desarrollo de Software
  + Software
* Roger S. Pressman- Ingeniería del Software

**Paginas de Internet:**

* <https://facebook.github.io/react-native/>
* <https://stackoverflow.com/>
* <https://victortisnado.com/blog/comunidad-react-native-espanol/>

# Anexos

**A1**

**Pantalla: 5.5", 1080 x 1920 pixels**

**Procesador: Exynos 7870 Octa**

**RAM: 3GB**

**Almacenamiento: 16GB**

**Expansión: microSD**

**Cámara: 13MP**

**Batería: 3300 mAh**

**OS: Android 6.0.1**

**Perfil: 8 mm**

**Peso: 167 g**

**A2**

****

● Intel Core i5-4210U 1.70GHz (c/TB 2.70 GHz)

● Pantalla LED 15.6" HD (1600x900)

● 4GB RAM / Disco Duro 500GB

● Video NVideo GeForce 820M con 2GB DDR3

● DVD+RW, Windows 8.1