

Análisis del proyecto

WarnMarket



12 de junio de 2022

José Manuel Marín Domínguez

Contenido

[Introducción 3](#_Toc105956342)

[Propósito 3](#_Toc105956343)

[Objetivos 3](#_Toc105956344)

[Viabilidad técnica y económica del proyecto y Temporalización 3](#_Toc105956345)

[Análisis 4](#_Toc105956346)

[Documentación relevante 4](#_Toc105956347)

[Definiciones 4](#_Toc105956348)

[Requisitos funcionales y no funcionales 4](#_Toc105956349)

[Requisitos funcionales 4](#_Toc105956350)

[Requisitos No Funcionales 5](#_Toc105956351)

[Casos de Uso. Diagramas UML de casos de Uso 7](#_Toc105956352)

[Modelado E/R. Diagramas E/R. 8](#_Toc105956353)

[Diseño 8](#_Toc105956354)

[Arquitectura hardware y software de la solución 8](#_Toc105956355)

[Modelado funcional de la solución. Diagramas de clase 8](#_Toc105956356)

[Modelado de datos. Modelo relacional. Diccionario de Datos 9](#_Toc105956357)

[Implementación 9](#_Toc105956358)

[Requisitos de instalación y ejecución 9](#_Toc105956359)

[Android 9](#_Toc105956360)

[Escritorio 9](#_Toc105956361)

[Servidor 10](#_Toc105956362)

[Implementación funcional. Clases 10](#_Toc105956363)

[Implementación del modelo de datos. Tablas 10](#_Toc105956364)

[Pruebas 10](#_Toc105956365)

[Pruebas de módulos 10](#_Toc105956366)

[Pruebas de integración 10](#_Toc105956367)

[Pruebas del sistema 10](#_Toc105956368)

[Pruebas de instalación 10](#_Toc105956369)

[Conclusiones 10](#_Toc105956370)

[Grado de consecución de los objetivos inicialmente planteados 10](#_Toc105956371)

[Dificultades encontradas 10](#_Toc105956372)

[Propuestas de mejora y posibles ampliaciones 11](#_Toc105956373)

[Bibliografía y recursos on-line 11](#_Toc105956374)

[Glosario de términos 11](#_Toc105956375)

[Informáticos 11](#_Toc105956376)

[Problema 11](#_Toc105956377)

# Introducción

## Propósito

Este proyecto consiste en desarrollar una aplicación que tratará de ayudar a las personas con sus compras del día a día. Los usuarios subirán ofertas de los supermercados que frecuentan con el fin de que los demás puedan comparar precios en los supermercados de la zona y así comprar de la manera más barata posible.

La aplicación se diseñará y programará tanto para escritorio como para Android, teniendo ambas diferentes funcionalidades.

Para ello se desarrollará utilizando el lenguaje de programación JAVA, haciendo uso del entorno de desarrollo integrado NetBeans IDE, y diseñándose una base de datos (gestionada con MySQL) que recoja la información de las ofertas, recetas, supermercados.

Con la aplicación, el usuario podrá subir, editar y borrar tanto las ofertas como las recetas. Obtendrá puntos por hacer estas acciones y más, los cuales podrán ser canjeados por cupones de descuento en los principales supermercados.

Para el futuro se espera que, si la aplicación se hace eco entre las empresas de supermercados, se pueda llegar a algún acuerdo con ellas para traer más promociones a ésta, por lo que habrá que rediseñar la base de datos y parte de la aplicación.

## Objetivos

Las principales metas de este proyecto son:

* Ayudar al usuario medio con sus compras
* Llegar al mayor número de usuarios
* Llegar a las grandes empresas para que podamos llegar a diferentes acuerdos de promociones
* Obtener beneficios de esas grandes empresas y/o con anuncios.

## Viabilidad técnica y económica del proyecto y Temporalización

La realización de este proyecto apenas requiere de financiación, pues lo único que se necesita es un ordenador y conexión a Internet. Se podría contratar a una serie de personal para llevar a cabo diferentes modalidades, como es el diseño de la aplicación y más programadores con el fin de compartir la carga de trabajo.

Con un equipo de trabajo de 3 personas, un diseñador y dos programadores, se podrían estimar unos 22.000 euros en total:

* 18.000€ como sueldo para los trabajadores
* 3.000€ para material como los ordenadores
* 1.000€ para Software

El lugar de trabajo, en principio, podría ser en nuestras propias casas. Esto se podría replantear en un futuro si todo sale bien; movernos a una oficina y contratar más personal.

En cuanto al Software, se utilizan aplicaciones gratuitas como el IDE Netbeans y Android Studio. No obstante, también debemos usar aplicaciones de pago como Photoshop o Justinmind para hacer el diseño de la aplicación, los logos, etc.

El proyecto, con un equipo de 3 personas, podría realizarse en 3 meses.

# Análisis

## Documentación relevante

Para llevar a cabo el proyecto se han consultado, sobre todo, páginas web como docs.oracle y [developer.android](https://developer.android) para consultar la documentación de java y stackoverflow y YouTube para problemas más puntuales.

## Definiciones

**IDE**: (Integrated Development Environment) Entorno de desarrollo integrado. Es un programa informático compuesto por un conjunto de herramientas de programación. Para el proyecto se ha usado Netbeans y Android Studio.

**JAVA**: Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems.

**JRE**: Java Runtime Environment es un conjunto de utilidades que permite la ejecución de programas Java. Para el proyecto se ha usado JRE versión 8 y 15.

**MYSQL**: Es un sistema de gestión de bases de datos relacional, multihilo y Multiusuario.

**XAMPP**: XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL.

## Requisitos funcionales y no funcionales

### Requisitos funcionales

1. Los usuarios se podrán registrar para poder iniciar sesión en la app.
2. Los usuarios podrán iniciar sesión en la aplicación para acceder a más funciones.
3. Los usuarios podrán cambiar sus contraseñas.

* Si se inicia sesión en un dispositivo móvil:
  1. Se podrá publicar una nueva oferta añadiendo una imagen, su precio, su precio/unidad (kg, litros, etc.), el supermercado donde se encuentra, el lugar donde se encuentra y una serie de etiquetas que lo describirán.
  2. Un usuario podrá modificar una oferta que ha subido con anterioridad.
  3. Se podrá buscar una oferta usando las etiquetas o el supermercado. Estas saldrán de forma ordenada por la distancia a la que se encuentran de ti.
  4. Se podrá denunciar una oferta si éste no es un producto de un supermercado, su información es falsa, etc.
  5. Se puede ver si una oferta es verídica o no si esta cuenta con un pulgar verde apuntando hacia arriba. Esto significará que la misma oferta la han subido dos o más personas.
  6. Se podrá publicar una nueva receta existente subiendo una imagen del plato terminado, el nombre, número de comensales, los pasos a seguir, el tiempo necesario, los ingredientes y los utensilios necesarios.
  7. Un usuario podrá modificar una receta que ha subido con anterioridad.
  8. Se podrá denunciar una receta.
  9. Se podrá guardar una o varias recetas como favoritas.
  10. El usuario podrá buscar recetas usando ingredientes como filtro para encontrar todas las recetas disponibles con esos ingredientes.
  11. Si el usuario, a la hora de buscar recetas, no usa ningún alimento como filtro, podrá ver todas las recetas subidas en ese momento.
  12. Un usuario obtendrá puntos al recibir me gustas de recetas subidas o cuando suba nuevas ofertas o recetas.
  13. Un usuario podrá ver su perfil para ver cuántos puntos lleva acumulados, ver sus recetas, sus ofertas, o las recetas guardadas.
  14. Un usuario podrá canjear sus puntos por recompensas.
* Si se inicia sesión en la aplicación de escritorio:

1. Se podrá buscar una oferta filtrando por Comunidad Autónoma, localidad o provincia.
2. Si el usuario, a la hora de buscar una oferta no elige la zona, se le mostrarán todos los productos.
3. Se podrá denunciar una oferta si éste no es un producto de un supermercado, su información es falsa, etc.
4. Se puede ver si una oferta es verídica o no si esta cuenta con un pulgar verde apuntando hacia arriba. Esto significará que la misma oferta la han subido dos o más personas
5. Se podrá guardar una o varias recetas como favoritas.
6. El usuario podrá buscar recetas usando ingredientes como filtro para encontrar todas las recetas disponibles con esos ingredientes.
7. Si el usuario, a la hora de buscar recetas, no usa ningún ingrediente como filtro, podrá ver todas las recetas subidas en ese momento.
8. Un usuario obtendrá puntos al recibir me gustas de recetas subidas o cuando suba nuevas ofertas o recetas.
9. Un usuario podrá ver su perfil para ver cuántos puntos lleva acumulados, ver sus recetas, sus ofertas, o las recetas guardadas.
10. Podrá ver una serie de estadísticas sobre las recetas.

* Si no se inicia sesión tanto en escritorio como en móvil:

1. Se podrá buscar una oferta.
2. Se podrá buscar recetas usando ingredientes como filtro para encontrar todas las recetas disponibles con esos ingredientes.

### Requisitos No Funcionales

* Escritorio:

1. Se necesitará Windows 7 como mínimo.
2. Se necesitará Internet para poder usarse.

* Móvil:

1. Se necesitará Android 8.0 como mínimo.
2. Se necesitará Internet para usarse.
3. Se necesitará el acceso a la cámara.
4. Se necesitarán permisos para acceder a la localización.

## Casos de Uso. Diagramas UML de casos de Uso

Diagrama

Descripción generada automáticamente

## Modelado E/R. Diagramas E/R.

Diagrama

Descripción generada automáticamente

# Diseño

## Arquitectura hardware y software de la solución

* Servidor:
* Windows 7
* Java 8
* Gestor de bases de datos XAMPP
* 1 GB de memoria interna
* 2 GB de RAM
* Intel Celeron o AMD Athlon
* Escritorio:
  + Windows 7
  + Java 15
* MySQL
* 2 GB de memoria interna
* 2 GB de RAM
  + Intel Celeron o AMD Athlon
* Android:
  + Android 8.0
  + 6 MB de memoria interna
  + 2 GB de RAM
  + Qualcomm Snapdragon 750G

## Modelado funcional de la solución. Diagramas de clase

* Aplicación de escritorio:

Imagen que contiene Texto

Descripción generada automáticamente

* Servidor:

Diagrama

Descripción generada automáticamente

## Modelado de datos. Modelo relacional. Diccionario de Datos

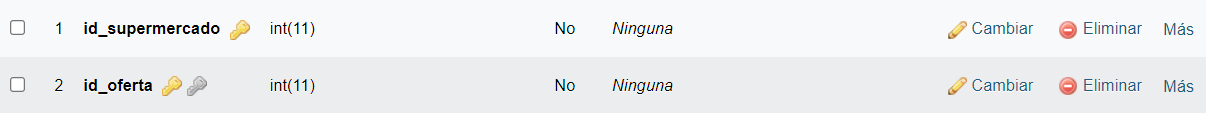
* Tabla ‘denunciasofertas’:



* Tabla ‘denunciasrecetas’:



* Tabla ‘estar’:

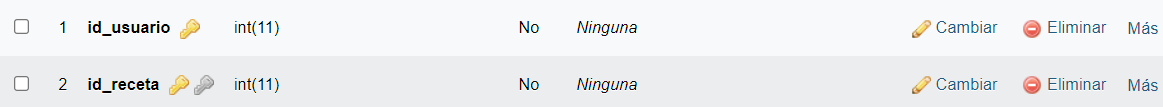


* Tabla ‘etiqueta’:

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

* Tabla ‘favoritosrecetas’:



* Tabla ‘oferta’:

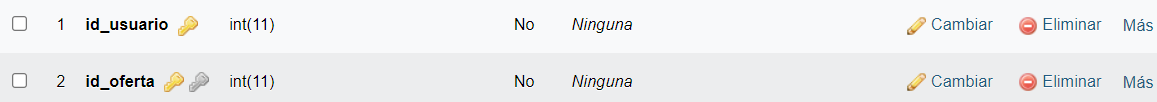
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Tabla ‘producto’:



* Tabla ‘publicar’:



* Tabla ‘receta’:

Interfaz de usuario gráfica

Descripción generada automáticamente

* Tabla ‘relacionetiqueta’:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto, Sitio web

Descripción generada automáticamente

* Tabla ‘supermercado’:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Tabla ‘tener’:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

* Tabla ‘tenerproducto’:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Tabla ‘usuario’:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

# Implementación

## Requisitos de instalación y ejecución

### Android

|  |  |  |
| --- | --- | --- |
|  | **Mínimo** | **Recomendado** |
| **Sistema Operativo** | Android 8 | Android 10 |
| **Espacio en el disco** | 6 MB | 30 MB |
| **Memoria RAM** | 2 GB | 4GB |
| **Procesador** | Qualcomm Snapdragon 750G | Qualcomm Snapdragon 888 |

### Escritorio

|  |  |  |
| --- | --- | --- |
|  | **Mínimo** | **Recomendado** |
| **Sistema Operativo** | Windows 7 | Windows 10 |
| **Espacio en el disco** | 1 GB | 2 GB |
| **Memoria RAM** | 2 GB | 4GB |
| **Procesador** | Intel Celeron o AMD Athlon | Intel i3 6100 |

### Servidor

|  |  |  |
| --- | --- | --- |
|  | **Mínimo** | **Recomendado** |
| **Sistema Operativo** | Windows 7 | Windows 10 |
| **Espacio en el disco** | 2 GB | 4 GB |
| **Memoria RAM** | 2 GB | 4GB |
| **Procesador** | Intel Celeron o AMD Athlon | Intel i3 6100 |

## Implementación funcional. Clases

* Servidor:

Texto

Descripción generada automáticamente con confianza mediaImagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

* Aplicación de escritorio:

Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamente

* Android:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteTexto

Descripción generada automáticamente

## Implementación del modelo de datos. Tablas

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

# Pruebas

## Pruebas funcionales

A la hora de **crear una receta**, a ésta se le debe añadir un **tiempo con el formato HH:MM** (Horas:minutos).

#### Clase de equivalencia

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Módulo a probar** | **Objetivo** | **Datos de entrada** | **Condición de ejecución** | **Resultado esperado** | **Resultado Obtenido** | **Estado** |
| 1 | Registrar receta | Comprobar que te deja registrar una receta insertando un formato de tiempo válido | 01:20 | El usuario debe darle al botón de “Crear receta” para registrarla en la base de datos | Se registra la receta | Se registra la receta | Éxito |
| 1 | Registrar receta | Comprobar que te deja registrar una receta insertando un formato de tiempo inválido | 25 | El usuario debe darle al botón de “Crear receta” para registrarla en la base de datos | No se registra la receta | Se registra la receta con un mensaje de error en el apartado del tiempo: “Error en el formato” | Éxito |

## Pruebas de seguridad

El programa cuenta con un **login** el cual usaremos para las pruebas de seguridad. Para ello, haremos una prueba en la que se intentará iniciar sesión **sin introducir ningún dato** en los campos de usuario y contraseña.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Módulo a probar** | **Objetivo** | **Datos de entrada** | **Condición de ejecución** | **Resultado esperado** | **Resultado Obtenido** | **Estado** |
| 1 | Pantalla de login | Comprobar que no te deja iniciar sesión si no se introduce ningún valor | **Campo usuario**: ninguno  **Campo contraseña**: ninguno | El usuario debe existir en la base de datos | Mensaje de error | Mensaje de error | Éxito |

## Pruebas de carga

Para las pruebas de carga supondremos que nuestra aplicación **funciona** correctamente con **100 usuarios** conectados al mismo tiempo. Para ello usaremos la técnica de valores límites.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Módulo a probar** | **Objetivo** | **Datos de entrada** | **Condición de ejecución** | **Resultado esperado** |
| 1 | Carga del sistema | Comprobar el funcionamiento de nuestra aplicación dependiendo del número de personas conectadas | 99 usuarios conectados | Debe haber menos de 99 usuarios conectados | El sistema funciona correctamente |
| 2 | Carga del sistema | Comprobar el funcionamiento de nuestra aplicación dependiendo del número de personas conectadas | 100 usuarios conectados | Debe haber 100, o menos de 100, usuarios conectados | El sistema funciona correctamente |
| 3 | Carga del sistema | Comprobar el funcionamiento de nuestra aplicación dependiendo del número de personas conectadas | 101 usuarios conectados | Debe haber menos de 101 usuarios conectados | El sistema comienza a fallar |

## Pruebas de uso de recursos

Teniendo en cuenta que nuestra aplicación es **cliente-servidor**, el servidor siempre se mantiene en espera a que se conecte un cliente. Cuando un cliente se conecta a la aplicación y se sale, el servidor no termina ese hilo, sino que se mantiene en espera a que se conecte de nuevo. Cuando el número de clientes aumenta en gran medida, eso provoca que, conforme pase el tiempo, se **aumente el uso** de memoria **RAM** y se acabe **ralentizando el servidor**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Módulo a probar** | **Objetivo** | **Datos de entrada** | **Condición de ejecución** | **Resultado esperado** |
| 1 | Uso de recursos del sistema | Comprobar el funcionamiento de nuestro servidor dependiendo del tiempo que lleva activo y el número de clientes conectados | 15 horas con el servidor activo y más de 100 conexiones en total | El sistema debe haber estado iniciado varias horas | El servidor comienza a ralentizarse cada vez más |

## Pruebas de aceptación

Paras las pruebas de aceptación hemos realizado una **beta cerrada** con muchos usuarios con el fin de conocer sus opiniones y que nos digan donde han encontrado fallos. Cuando acabaron las pruebas, uno de los errores más reportado ha sido que, cuando se pincha en **el botón de descarga de las estadísticas, no** se **abre ninguna pestaña**. Tras investigarlo hemos descubierto que, en el lado cliente, no se puede abrir de forma correcta el informe de Jasper Report porque lo intentamos hacer en el lado cliente, cuando debería ser en el lado servidor. Esto se debe a que el cliente no puede conectarse a la base de datos del servidor de forma correcta. No obstante, el lado servidor podría pasar al cliente un archivo como puede ser un pdf y el cliente descargarlo, pero es algo que no nos sirve porque pretendemos que, antes de descargar nada, el usuario pueda ver el informe generado.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Módulo a probar** | **Objetivo** | **Datos de entrada** | **Condición de ejecución** | **Resultado esperado** | **Resultado obtenido** |
| 1 | Botón de descarga | Comprobar que el botón de descarga funciona correctamente | Pinchar en el botón | El usuario debe darle al botón de descarga en la ventana de estadísticas | Se abre una ventana con información | Error al abrir la ventana con información |

# Conclusiones

## Grado de consecución de los objetivos inicialmente planteados

No se ha completado el 100% de objetivos impuestos inicialmente pero sí su mayoría, habiendo trabajado casi todos los días en el proyecto.

## Dificultades encontradas

El proyecto es demasiado grande como para hacerlo una única persona, y más teniendo en cuenta que hemos tenido que hacer las prácticas al mismo tiempo. Habría estado bien que la envergadura del proyecto fuese menor o que se pudiese empezar con más antelación. Algunas de las cosas que he tenido que implementar nunca la habíamos visto antes y apenas existe documentación decente.

## Propuestas de mejora y posibles ampliaciones

Para el futuro se espera que, si la aplicación se hace eco entre las empresas de supermercados, se pueda llegar a algún acuerdo con ellas para traer más promociones a ésta, por lo que habrá que rediseñar la base de datos y parte de la aplicación.

A la hora de hacer la documentación, esta viene poco explicada, por lo que, en algunos puntos, no sabía muy qué poner. El ejemplo existente sirve para algunos puntos, otros ni si quiera los tiene así que no puedes guiarte.

# Bibliografía y recursos on-line

# Glosario de términos

## Informáticos

## Problema

(abreviaturas, términos propios usados del problema,...)