**Tabla de Contenido**

[**I.** **Introducción:** 2](#_Toc512984695)

[**II.** **Objetivo** 2](#_Toc512984696)

[**III.** **Alcance** 2](#_Toc512984697)

[**IV.** **Decisiones de Diseño** 3](#_Toc512984698)

[**V.** **Estilo de Arquitectura** 3](#_Toc512984699)

[**VI.** **Vista de contexto** 3](#_Toc512984700)

[**VII.** **Vista Funcional** 4](#_Toc512984701)

[**VIII.** **Vista de Desarrollo** 5](#_Toc512984702)

[**IX.** **Vista de Información** 5](#_Toc512984703)

[**X.** **Vista de Despliegue** 6](#_Toc512984704)

[**XI.** **Diagrama de Componentes** 7](#_Toc512984705)

[**XII.** **Casos de uso** 8](#_Toc512984706)

[**XIII.** **Diagrama de clases** 8](#_Toc512984707)

[**XIV.** **Diseño de interfaces** 8](#_Toc512984708)

[**XV.** **Modelo de datos** 8](#_Toc512984709)

# **Introducción:**

En el documento de arquitectura y diseño se elabora la especificación del diseño de software, en donde se presentan el diseño de las partes del producto y sus interfaces, a través de la definición del alcance, las decisiones de diseño, del patron de arquitectura a utilizar, las vistas de diseño, el diagrama de clases, el modelo entidad relación y el diseño de interfaces.

Las vistas como medio de representación a características del sistema, tales como la representación de datos, los algoritmos e incluso el comportamiento nos permite mejorar del enfoque del desarrollo, disminuir la complejidad, y satisfacer inquietudes desde una descripción simplificada del sistema. Con lo cual nos apoyamos en unas vistas concretas para la representación de nuestro sistema, algunas en base a los requisitos funcionales, para delimitar el alcance del sistema y precisar lo que este debe brindar en términos de servicios a sus usuarios.

Estas vistas nos fueron de gran utilidad puesto que a partir del dominio del problema se logró formalizar una organización de módulos en el entorno de desarrollo representados en objetos y clases, así mismo comprender las abstracciones fundamentales del sistema y precisar cada uno de los componentes que forma la topología de hardware sobre la que se ejecutara el aplicativo, dichos componentes no representaron una dificultad mayor en cuanto al manejo, gracias a los distintos laboratorios que fueron propuestos y guiados por el docente.

# **Objetivo**

Con el documento de Arquitectura y Diseño de Software, se pretende establecer cómo se va a llevar el desarrollo del aplicativo empresarial en cuestión de la arquitectura, a través de las diferentes vistas en las cuales se ven parámetros fundamentales para el desarrollo, tales como que servidor web se va a utilizar, que servidor de base de datos, en que plataforma se va a utilizar la aplicación, el flujo de la información, entre otros.

## **Alcance**

Como se ha propuesto en el acta de iniciación, se desarrollará el aplicativo empresarial que requiere el problema de Rapicoop, en donde la arquitectura a utilizar, y el patrón de arquitectura a seguir, permitirá que se logre la distribución de los alimentos ofrecidos en la plataforma de manera eficiente, a un costo justo, y con estándares de higiene y calidad. Para esto se ha dividido el desarrollo en módulos que permitirán funcionalidades como que el usuario final pueda registrarse según su rol, el cliente pueda realizar pedidos a domicilio, los vendedores puedan recibir pedidos, realizar descuentos a sus productos, agregar nuevos productos, y llevar un control sobre sus ventas.

# **Decisiones de Diseño**

La principal decisión de diseño se presenta en el documento anexo Documento de Seguimiento Decisiones de Diseño de Software: Desempeño, en donde el problema a tratar es que no se garantice la correcta operación de los recursos de la plataforma.

# **Estilo de Arquitectura**

El patrón de arquitectura de software que se utilizara es Modelo Vista Controlador (MVC), el cual, utilizando estos 3 componentes, separa la lógica de la aplicación de la lógica de la vista, lo cual permite un desarrollo modular tal y como se ha planteado. De esta forma, el usuario solicitara el controlador, el controlador se comunicara con al modelo y enviara los datos obtenidos a la vista, la cual presentara visualmente los resultados esperados por el usuario.

En el componente modelo se encontrarán las interfaces a utilizar como Venta, Usuario, Producto y Detalle venta. En el controlador se desarrollarán los Servlet, como Servlet Controlador y Servlet Logueo y en el componente Vista se encontrarán los Java Server Page JSP como index, iniciar sesión, registrar producto, registrar venta, añadir y actualizar producto, entre otros.

# **Vista de contexto**

La vista de contexto presenta a los tres usuarios del sistema, y el contexto en el que utilizan el aplicativo para la gestión de pedidos, en el caso del cliente, se relaciona en cuanto a que realiza el pedido del producto, el domiciliario consulta de pedidos y productos que se encuentran en el sistema y en la plataforma se encarga del pago por pedido al cliente.

Pago Pedido

Pedido Producto

Cliente

Gestionar Pedidos

Cliente

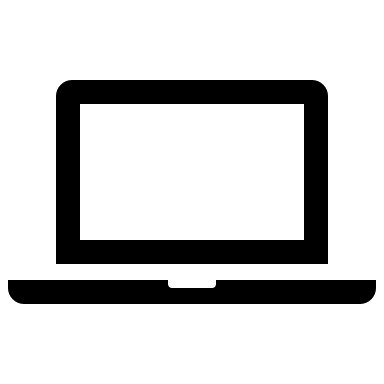
Consulta Productos y pedidos

Domiciliario

# **Vista Funcional**

El equipo ha decidido dividir su aplicación en tres componentes:

* Flota de domicilios: Donde el domiciliario tendrá el control de la flota que entrega los pedidos, como su ubicación y el tiempo de duración de la entrega del pedido.
* Préstamo de plantas de almacenamiento: El vendedor podrá llevar el control de las plantas de almacenamiento, un inventario de los insumos que conserva.
* Plataforma de manejo de pedidos: Donde el cliente podrá realizar sus pedidos.



Flota de domicilios

Plataforma de manejo de pedidos.

Préstamo de plantas de almacenamiento

Componente Web

Java Servlet

Session Beans

Java Server Pages

JSP

HTTP

Data

Base

# **Vista de Desarrollo**

Se ha dividido la vista de desarrollo en 5 componentes, se presenta el orden en el cual se utilizarán en el aplicativo.

Sistema Operativo

Gestor de bases de datos

Servidor Web

Aplicación web

Bases de datos

# **Vista de Información**

En la siguiente vista se presenta el flujo de información, la cual inicia en la base de datos, luego va el servidor web el cual permite la conexión con la aplicación web, que es la interfaz gráfica por la cual los clientes, los domiciliarios y los vendedores accedan a la información.

Aplicación Web

Cliente

Domiciliario

Vendedor

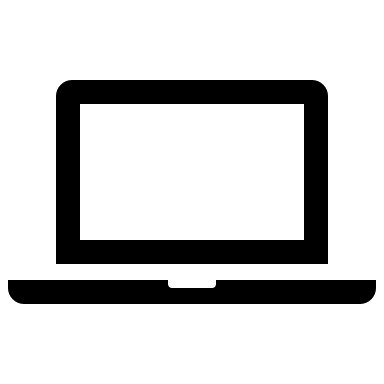
Aplicación Web

Servidor Web

Bases de datos

# **Vista de Despliegue**

En la vista de despliegue se muestra como es la arquitectura en cuanto a los servidores a utilizar, y se especifica que el Cliente accederá a la aplicación a través de un Navegador web por medio del protocolo HTTP.

Cliente

Web Browser

HTTP

Application server: Glassfish

Database Server: MySQL

Web Server: Apache Tomcat

# **Diagrama de Componentes**

Se tomaron como componentes externos, los usuarios del sistema, es decir cliente, domiciliario y vendedor, cada uno de estos ingresa por medio del componente de interfaz propio, la interfaz de cliente es dependiente del componente interfaz catálogo de producto y carrito de compras, la interfaz de domicilio accede a la interfaz de carrito de compras, que a su vez depende de la interfaz catálogo de productos. Por otro lado, la interfaz de vendedor accede al catálogo de productos a través de la interfaz de administración de catálogo de productos, y también accede a la interfaz de administración de puntos de venta. El componente Interfaz Catálogo de productos se conecta a la base de datos productos, y la Interfaz de administración de puntos de venta se conecta a la base de datos puntos de venta.

<<external>>

Vendedor

<<external>>

Domiciliario

<<external>>

Cliente

Interfaz domicilio.

Interfaz de administración de catálogo de productos.

Interfaz de administración de puntos de venta.

Interfaz Catálogo de productos.

Interfaz de vendedor

Interfaz de Cliente

Interfaz Carrito de compras

Base de datos productos.

Base de datos punto de venta.

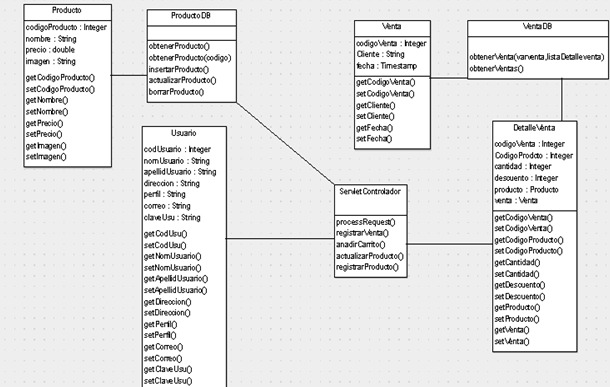
Representa

# **Casos de uso**

Durante la etapa de diseño es de suma importancia realizar la especificación funcional a través de los casos de uso. Los casos de uso planteados son los siguientes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | UC | Caso de Uso | Responsable |  |
|  | Ciclo 1 | | |  |
|  | 1 | Confirmar pedido | LE |  |
|  | 2 | Ver información pedido | LS |  |
|  | 3 | Agregar Producto | LD |  |
|  | 4 | Modificar Producto | LA |  |
|  | 5 | Borrar Producto | LP |  |
|  | 6 | Seleccionar Producto | LC |  |
|  | Ciclo 2 | | |  |
|  | 7 | Modificar Punto de venta | LC |  |
|  | 8 | Eliminar Punto de Venta | LA |  |
|  | 9 | Diligenciar Formulario | LP |  |
|  | 10 | Consultar formulario | LE |  |
|  | 11 | Confirmar Pedido | LS |  |
|  | 12 | Agregar Punto de Venta | D |  |
|  |  |  |  |  |

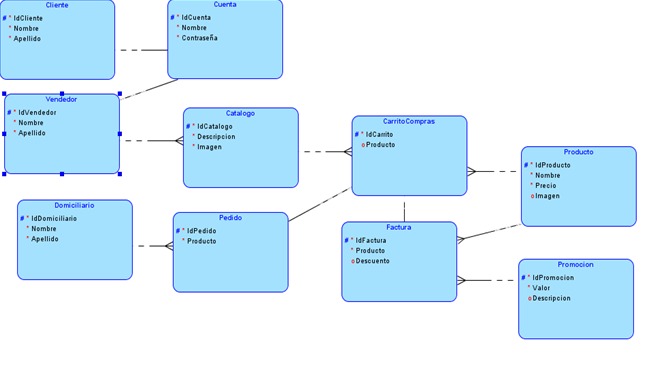
# **Diagrama de clases**



# **Diseño de interfaces**

Las interfaces gráficas de acuerdo con los casos de uso se encuentran en el documento anexado: Diseño de Interfaces.

# **Modelo de datos**



|  |  |  |
| --- | --- | --- |
| **CONTROL DE CAMBIOS** | | |
| **Fecha** | **Descripción** | **Autor(es)** |
| 5/03/2018 | Creación Vistas | LP - LE |
| 6/03/2018 | Creación Diagrama de componentes | LC - LA |
| 5/04/2018 | Introducción | LS |
| 5/04/2018 | Documentación vistas y diagramas | LA - LC |
| 1/05/2018 | Ajustes Ciclo II | LP - LE |