

# **OPEN MARKET**



**Universidad  
del Cauca**

Vigilada Mineducación

**Ingeniería de Software II**

## **Presentado por:**

Brayan Majin Gaviria

Freider Escobar Cueltan

Jorge Andres Ayerbe Caicedo

Julian Ruano Majin

**Universidad del Cauca**

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Ingeniería de Sistemas**

*Popayán, 2 junio 2023*

# Índice

<b>Resumen.....</b>	<b>3</b>
Wireframe.....	3
<b>Escalabilidad.....</b>	<b>6</b>
Funcionalidad.....	6
<b>Mantenibilidad.....</b>	<b>7</b>
<b>Rendimiento.....</b>	<b>8</b>
<b>Modelo C4.....</b>	<b>10</b>
Contexto del Sistema.....	10
Diagrama de contenedores.....	11
Diagrama de componentes.....	12
Diagrama de códigos.....	13
<b>Vista de módulos.....</b>	<b>14</b>
<b>Vista de C&amp;C.....</b>	<b>15</b>
<b>Diagramas E-R.....</b>	<b>16</b>
<b>Vista de Instalación.....</b>	<b>17</b>
<b>Rationale (Justificación de las decisiones de diseño).....</b>	<b>18</b>
<b>Elección de diseños.....</b>	<b>19</b>
<b>Historias de usuario.....</b>	<b>20</b>

# Resumen

El programa a desarrollar tiene el objetivo de permitir a sus usuarios comprar, vender y distribuir productos, por medio de una aplicación rápida, segura y fácil de usar. Esto implica que se debe plantear una buena arquitectura de software que se adapte a los distintos cambios que se puedan presentar en el tiempo.

Para la aplicación se decide usar la arquitectura cliente-servidor y por capas, donde se puede tener organización y claridad de las responsabilidades, ya que al ser una aplicación de compra y venta de productos, se puede presentar un flujo grande de usuarios, permitiendo la escalabilidad, por lo que podemos tener más servidores para manejar la carga y distribuir el procesamiento de manera eficiente.

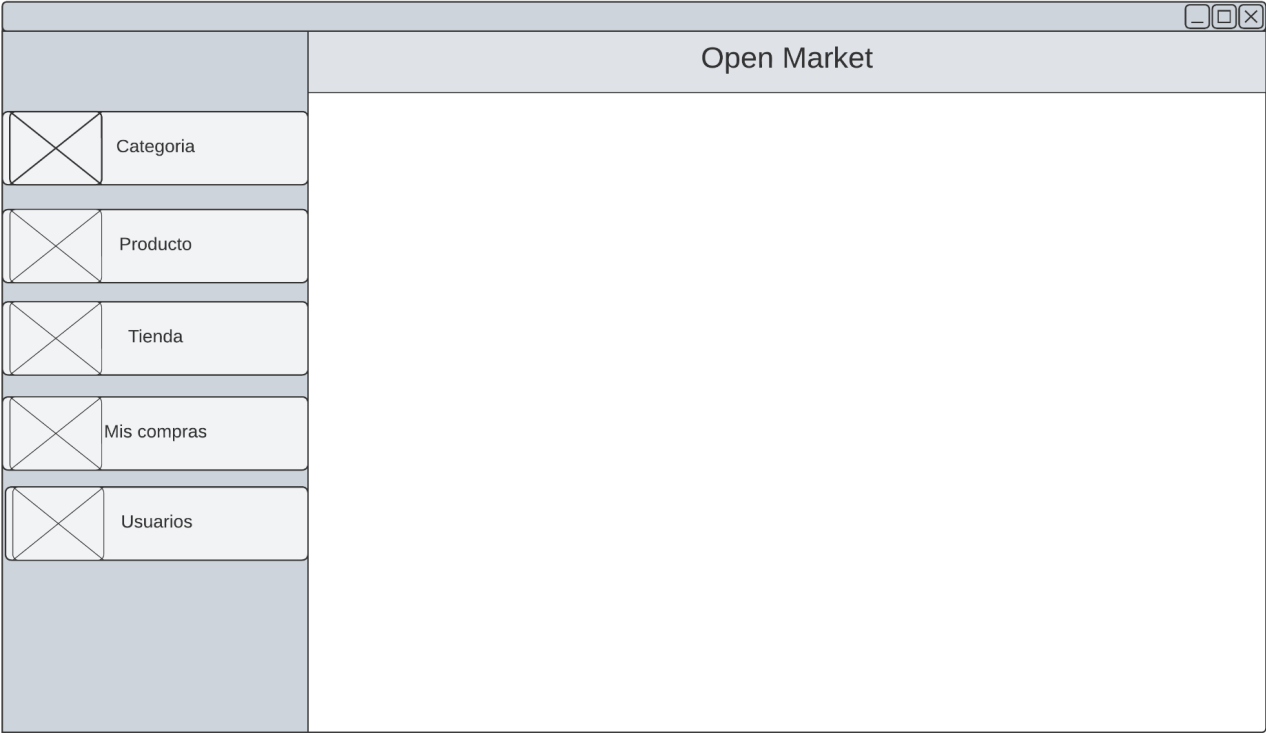
En tanto a patrones de diseño, se decide implementar algunos de ellos tales como el patrón observer, que permite que cuando un objeto cambie de estado, a los objetos dependientes se les notifique y así mantener actualizada la información, el patrón command nos permite convertir una solicitud en un objeto, donde podemos parametrizar clientes con diferentes solicitudes, hacer cola o registrar solicitudes y admitir operaciones como deshacer.

## Wireframe

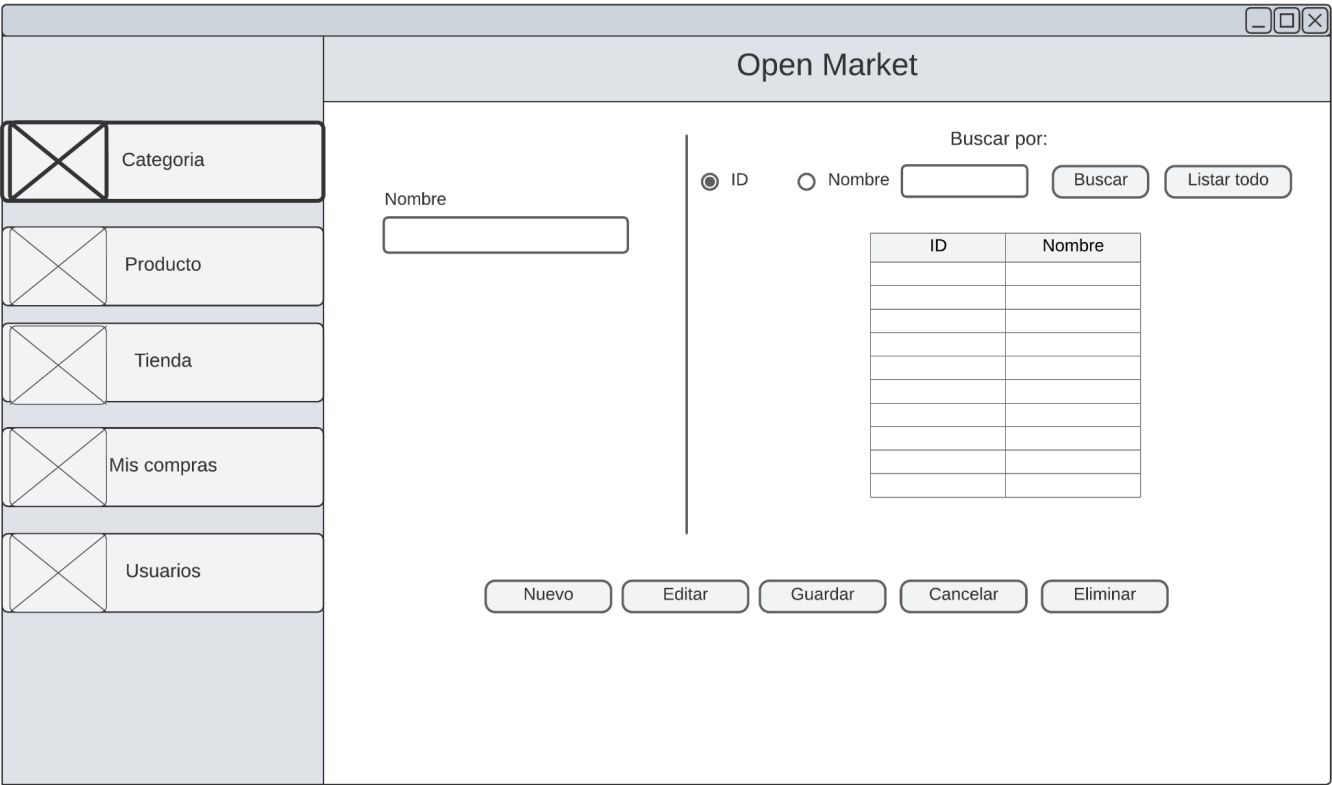
Se representan 4 diseños básicos:

- El primero es de la vista general cuando un usuario está logueado donde aparecen los botones categoría, producto, tienda, mis compras y usuarios.
- El segundo representa la sección de la categoría.
- El tercero representa la sección de productos
- El cuarto representa la sección de mis compras

Wireframe de sección de inicio (usuario logueado)



Wireframe de sección de categoria



Categoría

Producto

Tienda

Mis compras

Usuarios

Open Market

Nombre

Descripción

Precio

Stock

Dirección

Categoría

Seleccionar

Seleccionar imagen

Guardar

Cancelar

☒ Codigo producto

☐ Codigo categoria

☐ Nombre producto

Buscar

Listar todo

ID	Nombre	Descripción	Precio	Dirección	ID Categoría	Stock	Imagen

Categoría

Producto

Tienda

Mis compras

Usuarios

Open Market

Referencia	Producto	Precio	Fecha

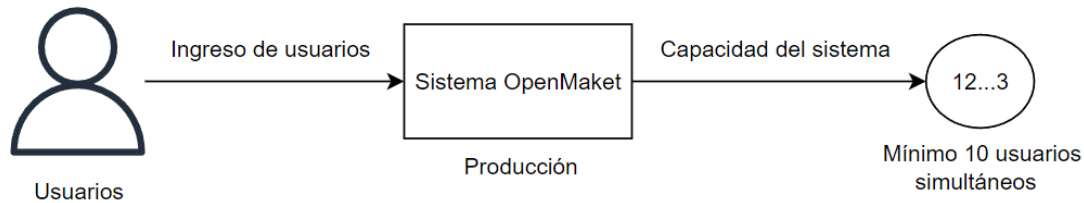
Ingrese la calificación

Referencia de compra

Confirmar

Especificación de **cualidades del software** mediante **escenarios de calidad**.

## Escalabilidad



**Nombre:** Escalabilidad :: Capacidad para manejar un aumento de usuarios

**Sinopsis:** A medida que se expande la tienda de productos, se espera un aumento significativo tanto en la cantidad de usuarios. Para satisfacer esta creciente demanda, el sistema debe poder expandirse adecuadamente sin comprometer el rendimiento ni la experiencia del usuario.

**Entorno:** La tienda está funcionando normalmente con un número moderado de clientes y una cantidad limitada de productos.

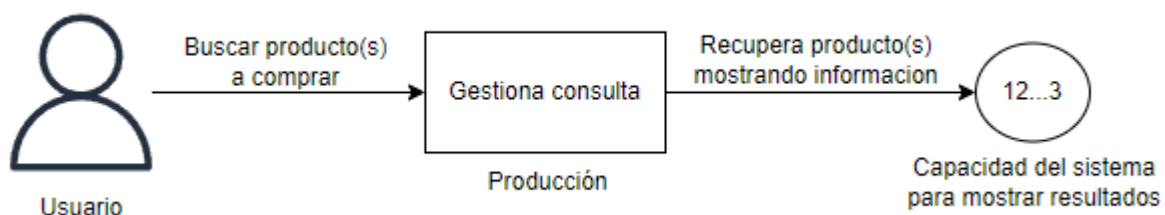
**Cambio en el entorno:** La popularidad de la tienda aumenta rápidamente, lo que aumenta significativamente el número de usuarios concurrentes y la cantidad de productos disponibles.

**Medida:** El sistema debe ser capaz de manejar al menos 10 usuarios simultáneos.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Local

## Funcionalidad



**Nombre:** Funcionalidad :: Capacidad para mostrar la información del producto

**Sinopsis:** Cuando un usuario registrado desea comprar un producto, se le permite aplicar filtros para encontrar más rápido lo deseado y el sistema debe tener la capacidad de responder rápidamente con la información del producto.

**Entorno:** La tienda está funcionando normalmente cuando el usuario es anónimo y solamente cuando se registra o inicia sesión se le permite comprar.

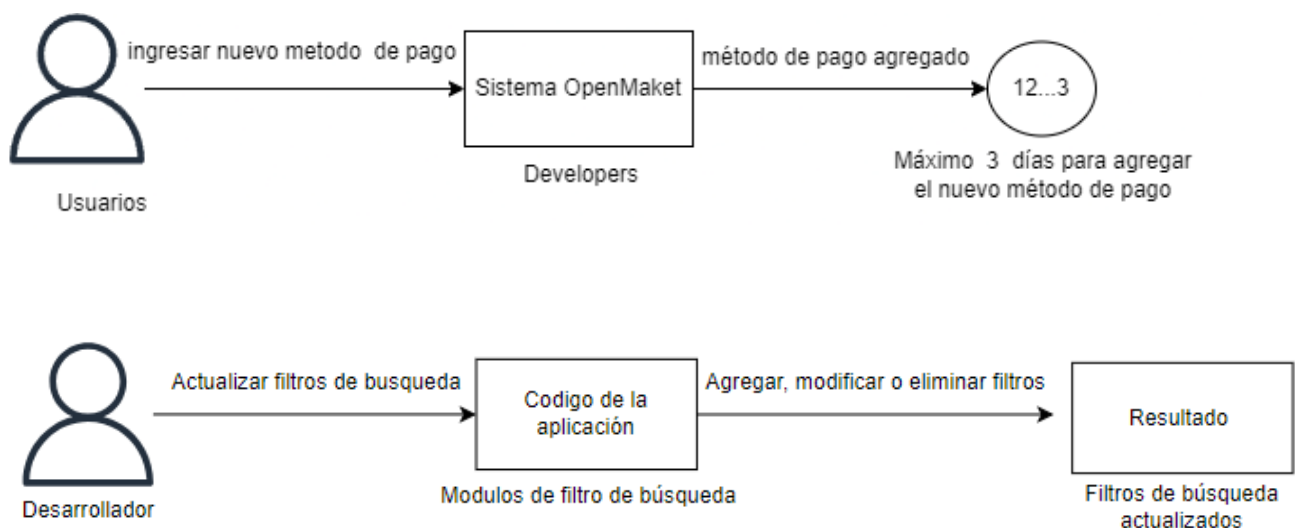
**Cambio en el entorno:** Se implementan productos con categoría, nombre y descripción similares.

**Medida:** El sistema debe ofrecer los filtros necesarios para que el usuario pueda encontrar el producto deseado sin ningún problema.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Local

## Mantenibilidad



**Nombre:** Mantenibilidad - Facilidad de actualización de los filtros de búsqueda de productos

**Sinopsis:** El sistema debe permitir a los desarrolladores agregar, modificar o eliminar fácilmente filtros de búsqueda de productos para adaptarse a cambios en los requisitos del negocio.

**Entorno:** La tienda en línea permite a los usuarios registrados aplicar filtros de búsqueda para encontrar productos más rápidamente. Actualmente, se pueden filtrar productos por categoría, nombre y descripción.

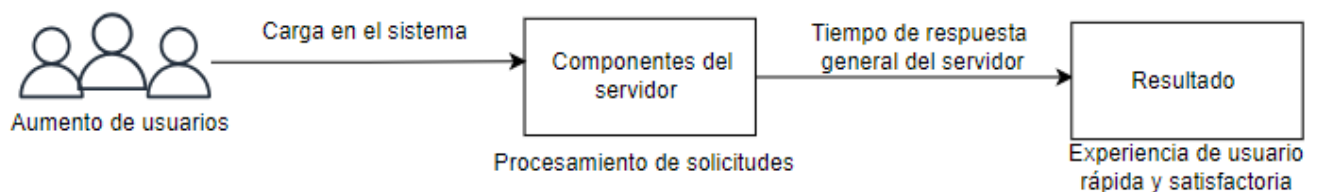
**Cambio en el entorno:** El negocio decide agregar nuevos atributos a los productos, como marca, precio y calificación. Estos nuevos atributos también deben ser utilizados como filtros de búsqueda para mejorar la experiencia del usuario.

**Medida:** El tiempo requerido para realizar cambios en los filtros de búsqueda no debe exceder de 2 días de trabajo, y el proceso de actualización debe involucrar a lo sumo 2 desarrolladores.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Local

## Rendimiento



**Nombre:** Rendimiento - Tiempo de respuesta general del servidor

**Sinopsis:** El sistema debe proporcionar un tiempo de respuesta rápido para todas las solicitudes de los usuarios para garantizar una experiencia de usuario satisfactoria.

**Entorno:** La tienda en línea permite a los usuarios registrados navegar, buscar productos y realizar compras.



**Cambio en el entorno:** Aumento en la cantidad de usuarios concurrentes que realizan diversas acciones en el sistema, como buscar y comprar productos.

**Respuesta:** El sistema debe mantener un tiempo de respuesta rápido para todas las solicitudes de los usuarios, incluso con un mayor volumen de usuarios concurrentes.

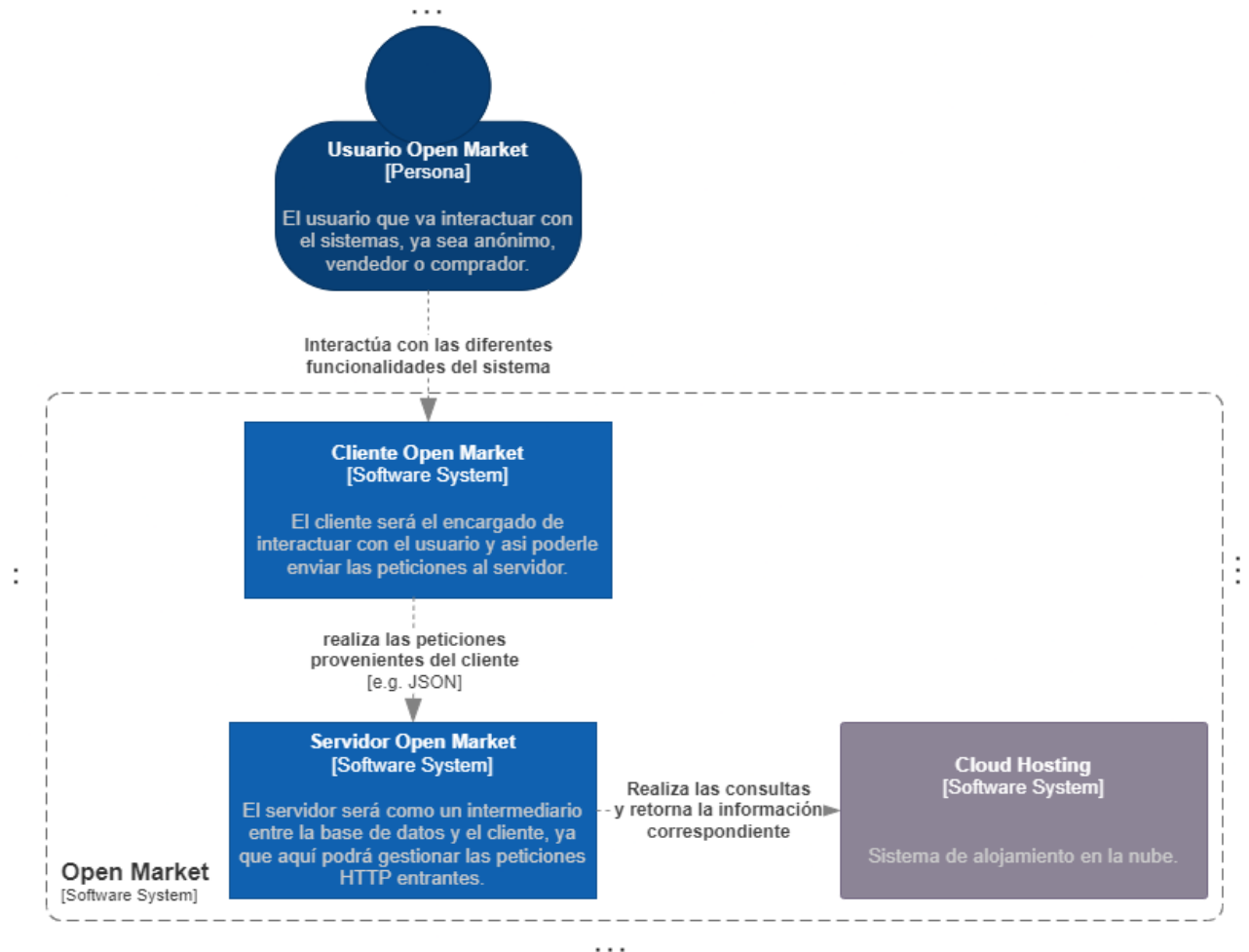
**Medida:** El 95% de todas las solicitudes de usuarios deben completarse en menos de 1 segundo.

**Prioridad Arquitectónica:** Alta

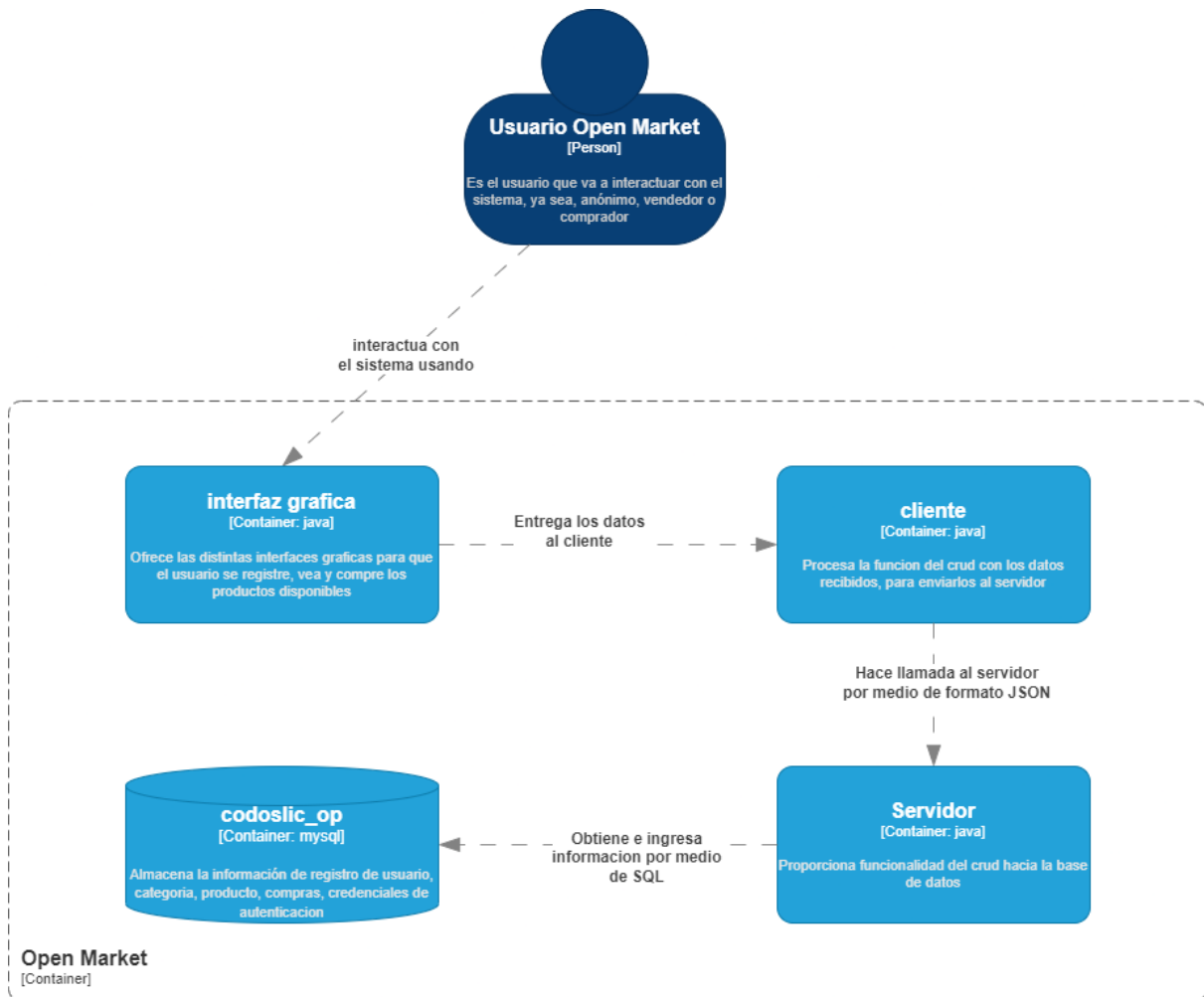
**Aplicación:** Local

# Modelo C4

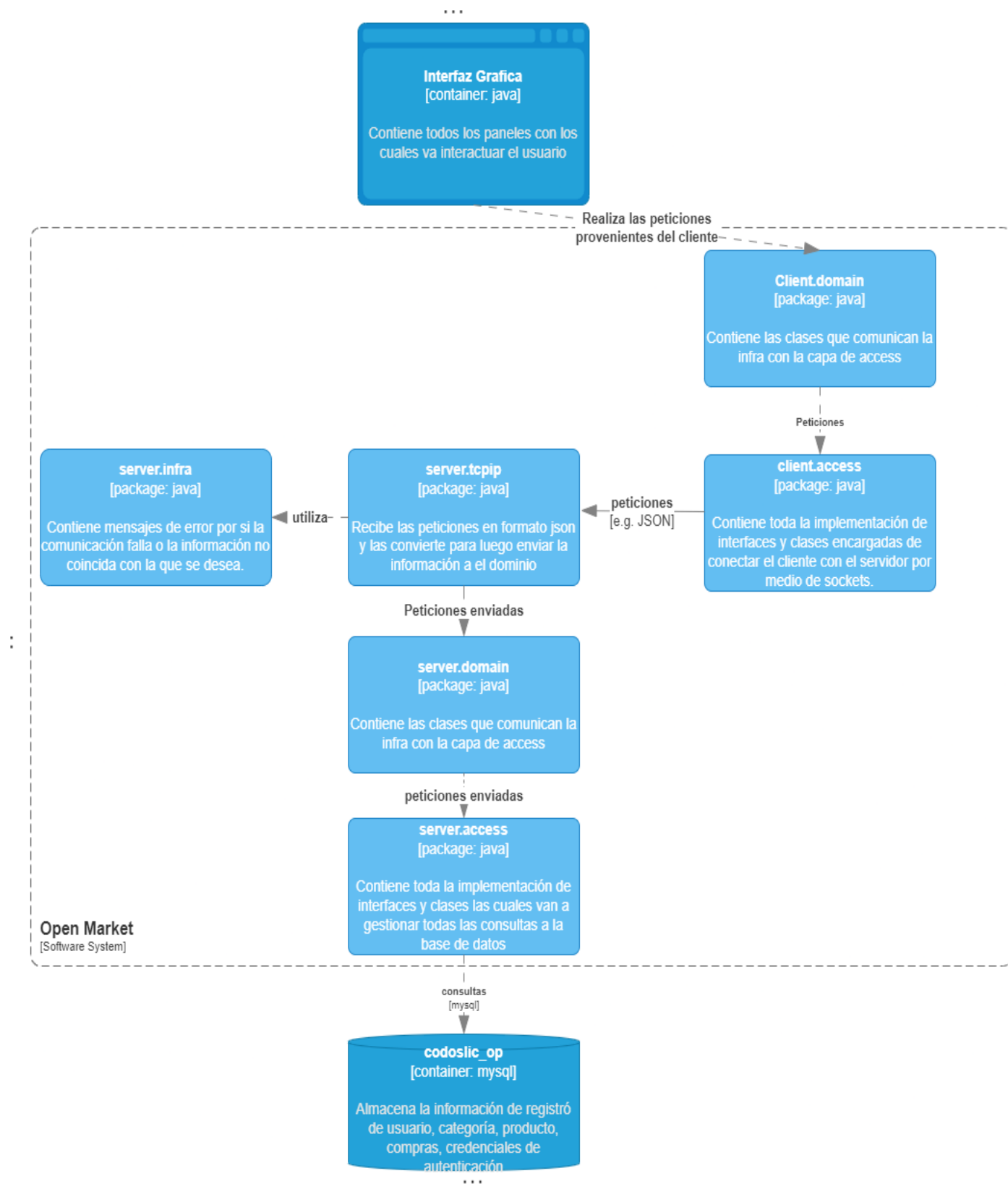
## Contexto del Sistema



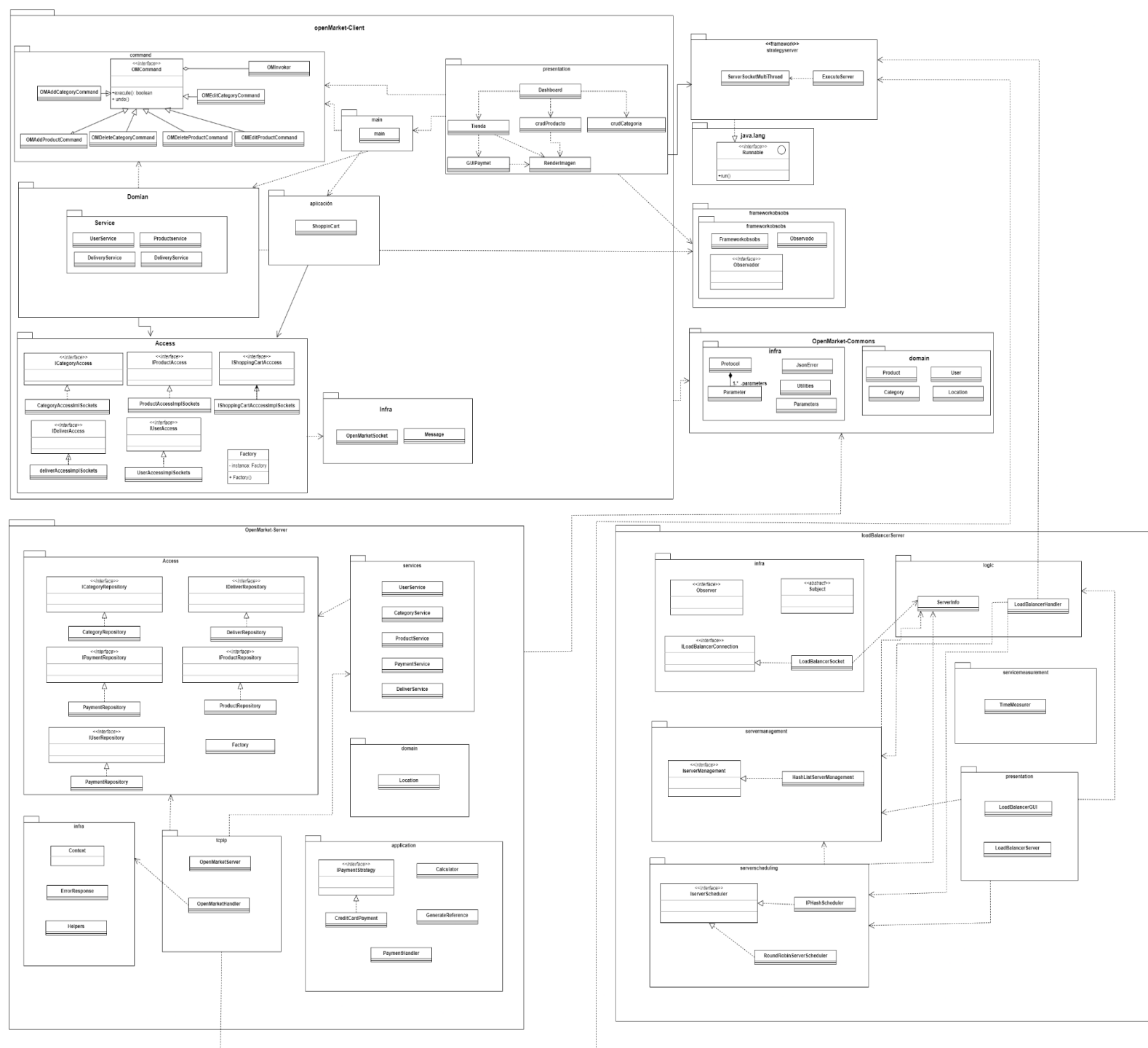
## Diagrama de contenedores



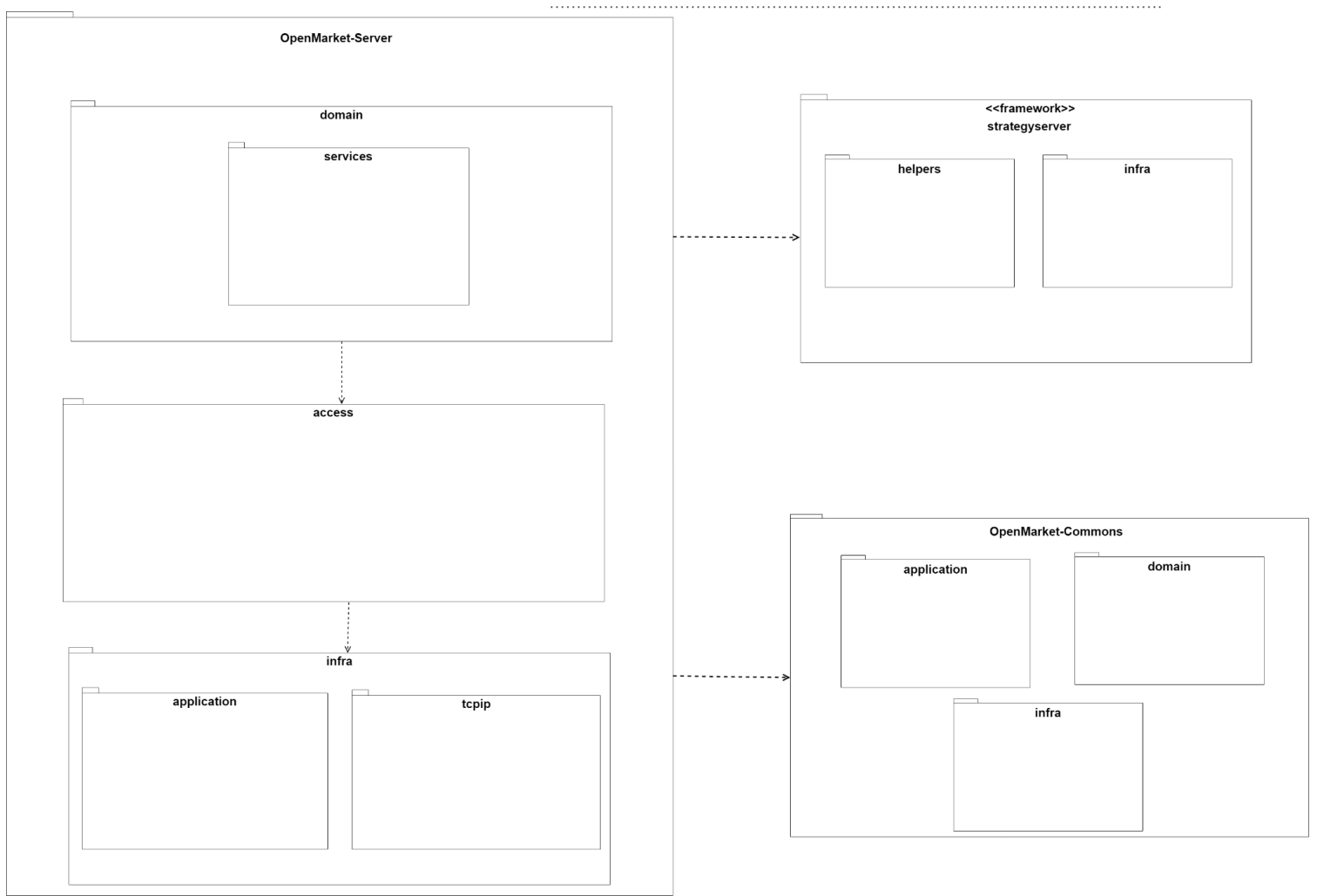
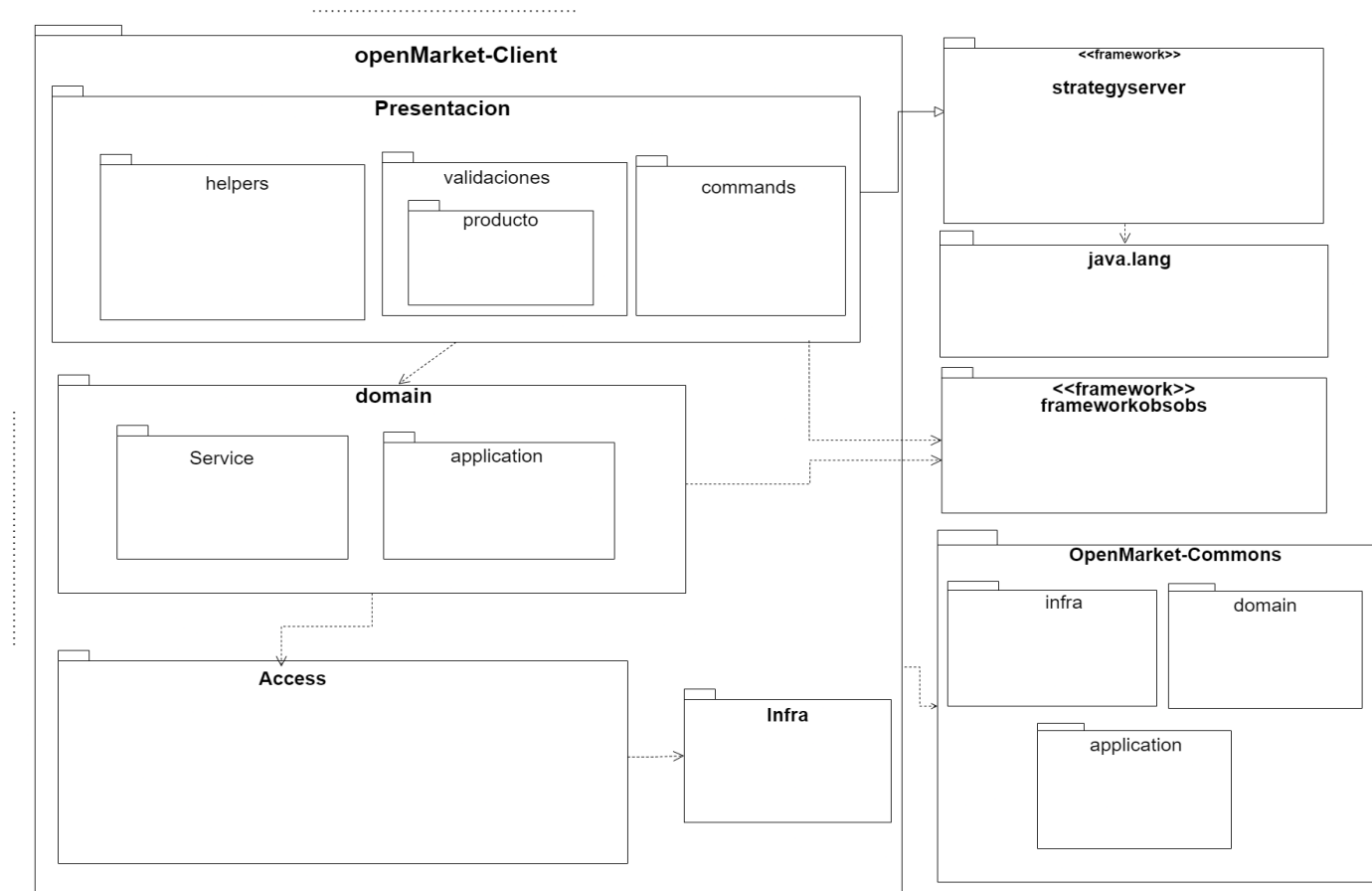
## Diagrama de componentes



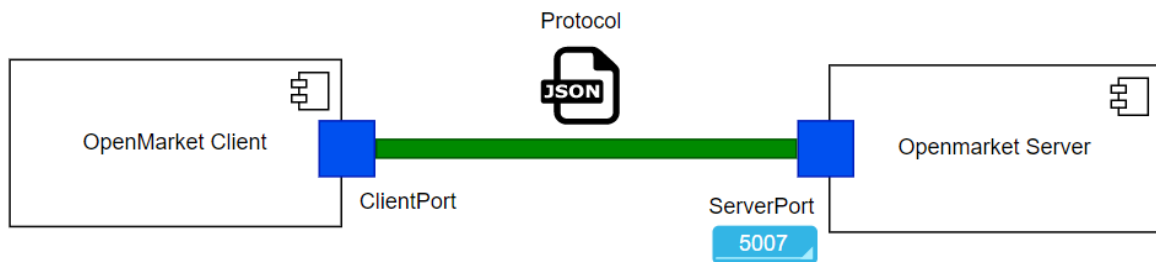
## Diagrama de códigos



# Vista de módulos



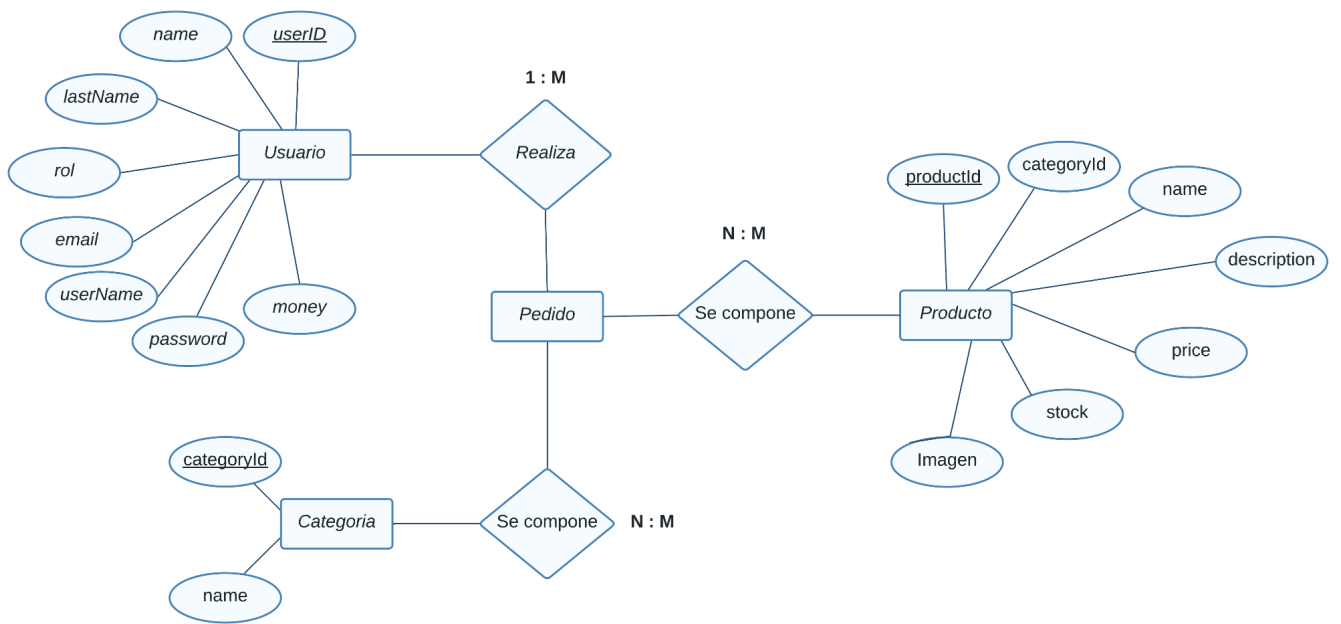
## Vista de C&C



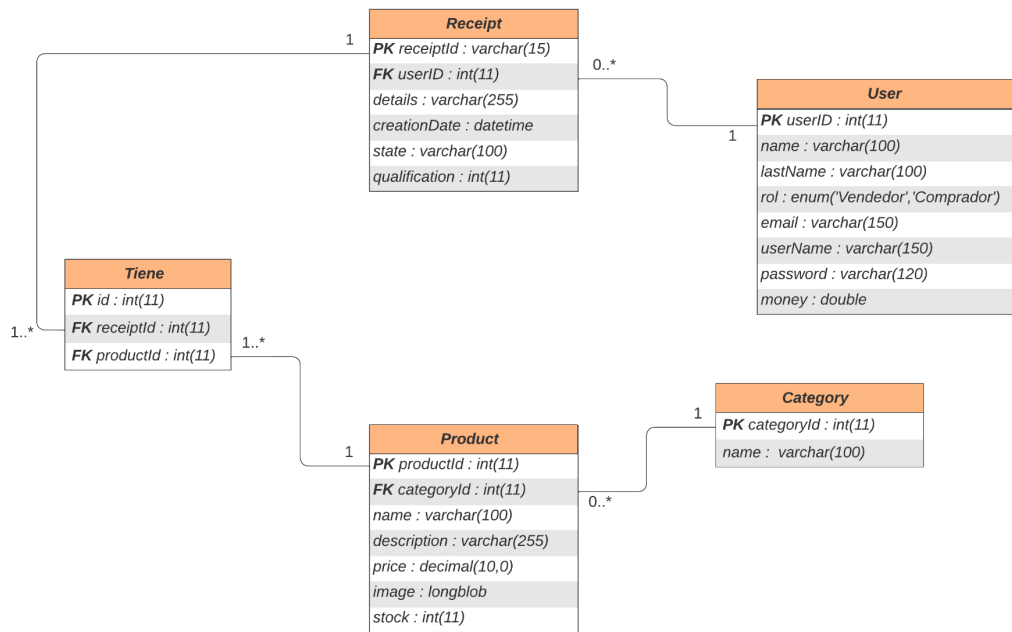
Rationale	El patrón arquitectónico Cliente-Servidor Se considera adecuado porque permite agregar nuevos clientes con gran facilidad sin tener un mayor impacto en la arquitectura. Debido a la naturaleza del problema, se pueden utilizar soluciones como un balanceador de carga si aumenta el número de usuarios.
Componente	OpenMarket-Client
Descripción:	Este componente es el encargado de los procesos relacionados con la gestión, venta y compra de productos.

Rationale	El patrón arquitectónico de Cliente-Servidor facilita la integración entre diferentes sistemas. Si en algún momento deseamos cambiar el sistema de pagos este se podría cambiar solo en el servidor sin afectar la lógica del cliente.
Componente	OpenMarket-Server
Descripción:	Este componente es el encargado de procesar las solicitudes del cliente y de gestionar las peticiones a la base de datos.

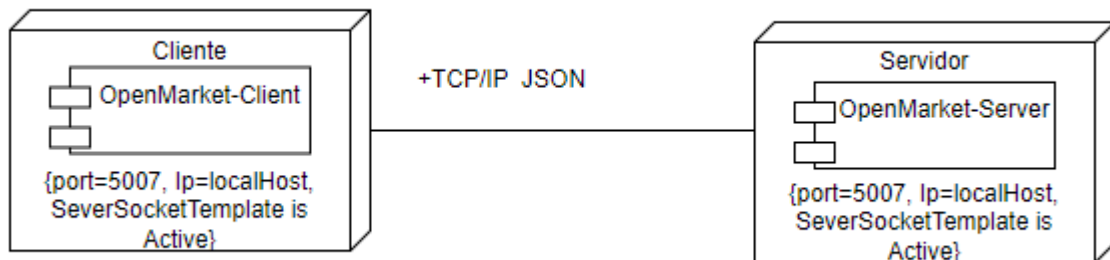
## Diagramas E-R







## Vista de Instalación



El cliente es el componente de la aplicación que interactúa directamente con el usuario. En este caso es una aplicación de escritorio, el cliente envía solicitudes al servidor para acceder a datos o realizar operaciones específicas.

El servidor es el componente responsable de procesar las solicitudes del cliente y ofrecer respuestas adecuadas. Además, tiene la capacidad de interactuar con una base de datos para almacenar o recuperar datos.

## Rationale (Justificación de las decisiones de diseño)

Atributo de Calidad	Concern	Táctica	Patrón
Escalabilidad	<ul style="list-style-type: none"><li>• Si aumenta los clientes se debe aumentar la capacidad para poder atender todas sus peticiones.</li><li>• Si la aplicación crece, debe poder expandirse brindando su servicio de igual manera.</li></ul>	<ul style="list-style-type: none"><li>• Concurrencia</li><li>• redundancia</li><li>• excepciones</li><li>• Transacciones</li><li>• Encriptación</li><li>• separación de responsabilidades</li><li>• Escalabilidad</li></ul>	Cliente-Servidor
Usabilidad	<ul style="list-style-type: none"><li>• Evitar sobrecargar a los usuarios con demasiada información o opciones</li></ul>	<ul style="list-style-type: none"><li>• Separar interfaz de usuario.</li><li>• Simplificación de flujo de trabajo</li><li>• separación de responsabilidades</li><li>• pruebas unitaria e integración</li><li>• escalabilidad</li><li>• abstracción</li></ul>	Capas
Modificabilidad	<ul style="list-style-type: none"><li>• Cuando se busca expandir el sistema o agregar nuevas funcionalidades, es importante asegurarse de que el proceso de implementación de estos cambios no sea tan complicado o engorroso.</li></ul>	<ul style="list-style-type: none"><li>• Desacoplamiento</li><li>• Separación de responsabilidades</li><li>• reutilización de código</li><li>• mantenibilidad</li><li>• Testing</li><li>• Recuperación de fallos</li></ul>	capas Patrón Commands

## Relational

En el contexto de la historia agregar productos, con el fin de resolver la interacción con varios clientes, nos decidimos por usar el patrón de diseño cliente-servidor y descartamos la opción de hacer una implementación directa, para lograr escalabilidad del sistema cuando se requieren conectar varios clientes, aceptando que esto afectara el desempeño de la aplicación al conectar muchos clientes porque serán muchas peticiones que el servidor debe atender.

en el contexto del historia de usuario listar productos, con el fin de resolver que el cliente interactúe y visualiza mejor la sección de productos, nos decidimos a usar el patrón de

diseño observador y cliente-servidor, y descartamos la opción de colocar más interacciones con el sistema para poder ver los productos sino que el usuario los pueda ir mirando automáticamente a medida que se los va agregando para lograr la usabilidad del sistema cuando se listen los productos, aceptando que esto afectara al rendimiento de la aplicación porque cada que se agregue un usuario se realizará la petición para que se actualicen la lista de productos.

En el contexto de la historia de usuario eliminar productos disponibles, con el fin de resolver la interacción con la eliminación de productos, nos decidimos por usar el patrón de diseño Command y descartamos la opción de agregar mas campos para eliminar un producto para lograr una mayor usabilidad a la hora de eliminara un producto, aceptado que se afectara la fiabilidad de la aplicación porque puede eliminar productos que no quieras.

en el contexto de las historias de usuario agregar,eliminar, listar productos, buscar,comprar, con el fin de resolver la interacción con estas funcionalidades, nos decidimos por usar el patrón de arquitectura Layers y descartamos la opción de acoplamiento de clases, para lograr la modularidad y mantenibilidad en caso de que se quiera añadir más funcionalidades futura, aceptando que esto afectará la complejidad del trabajo sino se diseña bien porque traerá dependencias excesivas entre capas

## Elección de diseños

### Tono azul para el Sidebar del dashboard

El azul es un color ampliamente utilizado en el diseño de interfaces de usuario, especialmente en aplicaciones financieras y negocios. Esto se debe a varias razones:

- **Profesionalismo:** El azul es un color que transmite seriedad y profesionalismo. En el contexto de un software de Open Market, es importante transmitir confianza y seguridad a los usuarios.
- **Relajación y enfoque:** El azul es un color que se asocia con la calma y la relajación. En un entorno de mercado abierto, donde los usuarios podrían estar sometidos a altos niveles de estrés y toma de decisiones rápidas, el uso de un color relajante puede ayudar a mantener visualmente la concentración y reducir la fatiga visual.

- **Accesibilidad:** El uso del azul en el fondo del dashboard también garantiza una buena accesibilidad en términos de contraste con otros colores( por ejemplo, texto blanco o iconos). De esta manera se mejora la legibilidad y el uso general del software.

### Tono gris para elementos y componentes

Se ha optado por utilizar tonos grises para los elementos y componentes del diseño por las siguientes razones:

- **Neutralidad:** El gris es un color neutral que no compite con otros colores en términos de atención. De esta manera, permite que los usuarios se concentren en la información y los datos personales en el dashboard.
- **Elegancia y sofisticación:** El uso de tonos grises en los elementos y componentes del diseño confiere al software una apariencia elegante y sofisticada, lo cual es apropiada para una aplicación destinada a personas que se dedican al sector financiero.
- **Contraste:** Los tonos grises combinan bien con el fondo azul del dashboard y proporcionan un contraste adecuado para mejorar la legibilidad y el uso general del software.

## Historias de usuario

<b>HU01</b>	Registro
Como	Usuario anónimo
Quiero	Poder registrarse al sistema
Para	Poder realizar compras
Criterios de aceptación	<ul style="list-style-type: none"> <li>● Como usuario anónimo, quiero poder acceder a la página de registro desde la página de inicio de sesión.</li> </ul>

	<ul style="list-style-type: none"> <li>• Como usuario anónimo, quiero poder ingresar mi nombre, dirección de correo electrónico, usuario y crear una contraseña durante el proceso de registro.</li> </ul>
--	--

<b>HU02</b>	Iniciar sesión
Como	Usuario del sistema
Quiero	Ingresar mi nombre de usuario y contraseña
Para	Poder acceder a mi cuenta
Criterios de aceptación	<ul style="list-style-type: none"> <li>• El usuario debe proporcionar un nombre de usuario y contraseña válidos.</li> </ul>

<b>HU03</b>	Agregar productos
Como	Vendedor
Quiero	Poder agregar productos al sistema
Para	Que los usuarios puedan verlos y comprarlos desde la página web
Criterios de aceptación	<ul style="list-style-type: none"> <li>• El vendedor debe proporcionar la información requerida del producto (nombre, descripción, precio, imagen, etc.).</li> <li>• El vendedor debe indicar la cantidad disponible en stock.</li> </ul>

<b>HU05</b>	Eliminar productos disponibles
Como	Vendedor
Quiero	Poder modificar la disponibilidad de un producto
Para	Mantener actualizada la disponibilidad de un producto
Criterios de aceptación	<ul style="list-style-type: none"> <li>• El vendedor debe tener permisos para eliminar productos.</li> <li>• El producto debe ser eliminado de las listas y búsquedas de productos.</li> </ul>

<b>HU06</b>	Suspender publicación
Como	Vendedor
Quiero	Pausar la publicación temporalmente
Para	Poderla activar en un futuro
Criterios de aceptación	<ul style="list-style-type: none"> <li>• El vendedor debe tener permisos para suspender la publicación de productos.</li> <li>• El producto no debe aparecer en las listas ni en las búsquedas de productos mientras esté pausado.</li> </ul>

<b>HU07</b>	Modificar productos
Como	Vendedor
Quiero	Modificar las características del producto
Para	Mantener la información actualizada
Criterios de aceptación	<ul style="list-style-type: none"> <li>• El vendedor debe tener permisos para editar la información del producto.</li> <li>• El vendedor debe proporcionar la nueva información del producto (nombre, descripción, precio, imagen, etc.).</li> </ul>

<b>HU08</b>	Listar productos
Como	Usuario
Quiero	Visualizar la lista de productos
Para	Conocer la oferta disponible en el sistema
Criterios de aceptación	<p>La lista de productos debe estar actualizada.</p> <p>La lista debe mostrar la información relevante de cada producto (nombre, precio, imagen, etc.).</p>

--	--

<b>HU09</b>	Buscar productos
Como	Usuario
Quiero	Buscar productos por nombre o descripción
Para	Encontrar el producto que necesito
Criterios de aceptación	<ul style="list-style-type: none"> <li>• El sistema debe ofrecer una función de búsqueda que permita filtrar los productos según los términos ingresados por el usuario.</li> <li>• Los resultados de la búsqueda deben mostrar la información relevante de cada producto (nombre, precio, imagen, etc.).</li> </ul>

<b>HU10</b>	Realizar Compra
Como	Usuario registrado
Quiero	Poder comprar el producto seleccionado
Para	Poder obtener el producto deseado y completar mi transacción de compra de manera satisfactoria.
Criterios de aceptación	<ul style="list-style-type: none"> <li>• El usuario debe estar autenticado en el sistema.</li> <li>• El producto seleccionado debe estar disponible en stock.</li> <li>• El usuario debe proporcionar la información de envío requerida.</li> </ul>

<b>HU11</b>	Realizar Pago
Como	Usuario registrado
Quiero	Poder pagar mis productos de manera segura
Para	Culminar la compra

Criterios de aceptación	<ul style="list-style-type: none"> <li>• El usuario debe estar autenticado en el sistema.</li> <li>• El sistema debe ofrecer opciones de pago seguras y confiables.</li> <li>• El usuario debe recibir una confirmación del pago exitoso.</li> </ul>
-------------------------	--

<b>HU12</b>	Confirmación de compra
Como	Usuario Registrado
Quiero	Confirmar la entrega de mi producto
Para	Verificar si fue satisfactoria la entrega
Criterios de aceptación	<ul style="list-style-type: none"> <li>• El usuario debe estar autenticado en el sistema.</li> <li>• El usuario debe haber realizado previamente una compra.</li> <li>• El sistema debe permitir al usuario confirmar la entrega y proporcionar un espacio para calificar la experiencia de compra y el producto recibido.</li> </ul>