

# Programación Cliente/Servidor Concurrente – SC 303

FVGames

## **Integrantes:**

Nathalie Mariela Chaves Muñoz

## **Prof.**

Jose Antonio Ortega González

III CUATRIMESTRE

Fecha entrega: 25/08/2024

## **1 Contenido**

1	DESCRIPCIÓN DEL PROBLEMA .....	2
1.1	Problema general.....	2
1.2	Subproblemas.....	2
2	Requerimientos Funcionales y No Funcionales .....	6
3	Diagramas .....	8
3.1	Diagrama general .....	8
3.2	Diagrama de clases .....	9
3.3	Diagrama de paquetes .....	10
3.4	Descripción general de la solución.....	10
3.5	Diagrama de base de datos .....	12
3.6	Diagrama de Gantt.....	13
4	Enlace de GitHub: .....	13
5	Referencias Bibliográficas.....	14

# 1 DESCRIPCIÓN DEL PROBLEMA

## 1.1 *Problema general*

La empresa FVGames enfrenta la necesidad de un software integral para la gestión de inventarios, paquetes, clientes, un carrito de compras y las visualizaciones del listado de productos y las compras realizadas. Para abordar estas necesidades, han decidido contratar a una empresa de software la cual proveerá una solución adecuada a estos problemas.

## 1.2 *Subproblemas*

### **a) Carrito de compras y usuarios**

Se enfoca en la creación de un sistema que permita a los usuarios gestionar sus compras dentro de la plataforma. Se propone utilizar una estructura de datos llamada lista doble circular para organizar los datos de manera eficiente.

Detalles:

- a. Lista doble circular: Es una estructura de datos donde cada nodo tiene un enlace al siguiente nodo y al anterior, formando un ciclo. En este caso, se usa para representar a los clientes (usuarios) y sus compras.
- b. Listas anidadas: Cada cliente (nodo de la lista) contendrá otra lista que almacenará sus compras. Esto permite organizar los datos de forma que cada cliente tenga su propio historial de compras dentro de la estructura principal.

Objetivo:

Facilitar la gestión de clientes y sus respectivas compras, manteniendo un orden y estructura que permita acceder a la información de manera rápida y eficiente.

## **b) Gestión del inventario**

Se abordará cómo se organizará y gestionará el inventario de la empresa.

Detalles:

- a) Lista categorizada: Se utilizará una lista donde los artículos estarán organizados por categorías (por ejemplo, videojuegos, consolas, accesorios). Cada categoría tendrá su propia lista de artículos y sus respectivas cantidades.

Objetivo:

Asegurar una gestión eficiente del inventario, donde los artículos están organizados por categorías, y se facilita la búsqueda y actualización de la información de cada artículo.

## **a) Gestión de paquetes**

Se centra en la creación de paquetes de productos, donde se agrupan varios artículos y se venden juntos a un precio especial.

Detalles:

- a) Lista independiente: Los paquetes se manejarán en una lista separada que agrupa artículos individuales disponibles en la tienda.
- b) Reglas de negocio:
- Cada paquete puede contener hasta 5 artículos.

- Se aplicará un descuento del 15% sobre el precio total de los artículos en el paquete.
- El administrador del software será quien gestione estos paquetes y aplique los descuentos correspondientes.

Objetivo:

Permitir la creación de paquetes de productos con descuentos, lo que puede aumentar las ventas al ofrecer incentivos a los clientes.

### **c) Gestión de clientes**

Se centra en la gestión de la información de los clientes registrados.

Detalles:

- a) Clase Cliente: Cada cliente registrado tendrá un perfil que almacena su información personal y de compras.
- b) Funcionalidades:
  - Los clientes podrán visualizar productos disponibles en la tienda.
  - Podrán realizar compras y guardar productos en un carrito de compras para comprar más tarde.
- c) Seguridad y persistencia: Se debe asegurar que toda la información del cliente esté protegida y que se guarde de manera segura y persistente (por ejemplo, usando bases de datos).

**Objetivo:**

Proporcionar una experiencia de usuario completa y segura, donde los clientes puedan gestionar su información, ver productos, realizar compras y usar un carrito de compras.

## 2 Requerimientos Funcionales y No Funcionales

### 2.1 *Requerimientos Funcionales*

- a. Gestión de Inventario:
  - a. Editar artículos según la categoría.
  - b. Agregar la cantidad de artículos por producto.
  - c. Actualizar la cantidad de productos tras una compra.
  - d. Solicitar el total de ingresos basado en las compras del día.
- b. Gestión de Paquetes:
  - a. Crear paquetes de venta con 2 a 5 productos, que pueden ser repetidos.
  - b. Calcular el precio de los paquetes con un 15% de descuento por defecto.
  - c. Permitir al administrador asignar un descuento diferente.
- c. Gestión de Clientes:
  - a. Registrar clientes con nombre, apellidos, cédula, dirección, email, dinero en la cuenta y método de pago preferido.
  - b. Permitir a los clientes ingresar como invitados o como registrados.
  - c. Realizar compras solo como cliente registrado.
  - d. Validar fondos suficientes en la cuenta para completar la compra.
  - e. Permitir recargar dinero en la cuenta.
  - f. Editar datos de cliente y registrar transacciones de recarga.
  - g. Iniciar sesión con cédula y contraseña.
  - h. Permitir el acceso simultáneo de múltiples usuarios.
- d. Carrito de Compras:
  - a. Mostrar un error si un artículo no está disponible.
  - b. Permitir seleccionar la cantidad de artículos a comprar.
  - c. Agregar artículos al carrito de compras.
  - d. Calcular el precio final con impuestos del 13%.
- e. Visualización:
  - a. Mostrar un listado de productos con opción de filtrado por categoría.
  - b. Permitir la adición de productos al carrito desde el listado.
  - c. Permitir al administrador visualizar todas las compras realizadas.
  - d. Consultar compras por intervalo de fecha o por usuario.

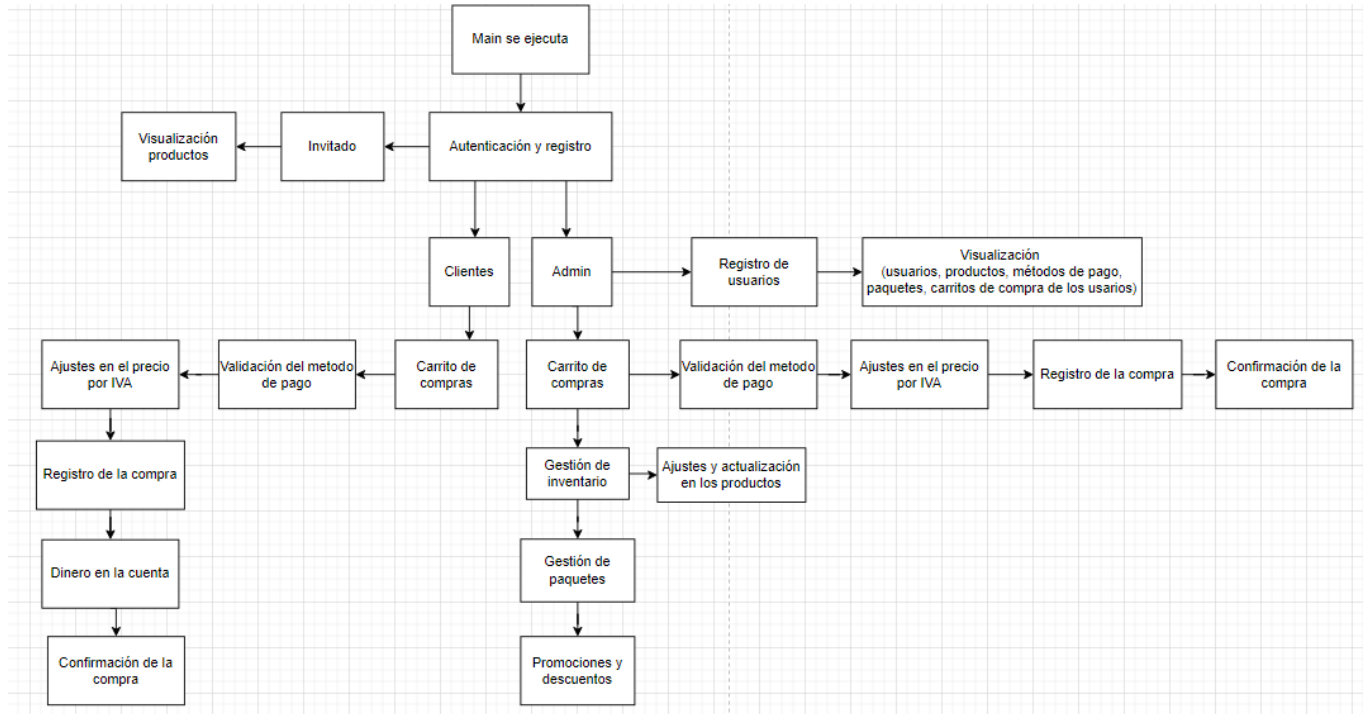
## **2.2 *Requerimientos No Funcionales***

- a. Patrón de Diseño:
  - a. Seguir el patrón de diseño MVC (Model-View-Controller).
- b. Persistencia de Datos:
  - a. Utilizar una base de datos MySQL para la persistencia de la información.
  - b. Mantener la información previa si el programa se cierra y se reinicia.
  - c. Permitir agregar, modificar y eliminar información en la base de datos.
- c. Interfaz Grafica
  - a. Utilizar la biblioteca Swing para la interfaz gráfica.
  - b. Gestionar accesos desde el cliente de manera gráfica.
  - c. Ofrecer capacidades adicionales al administrador a través de la interfaz gráfica.
- d. Concurrencia
  - a. Utilizar sockets para la conexión simultánea de clientes.
  - b. Procesar toda la lógica de negocio en el servidor.
  - c. Gestionar accesos, solicitudes y visualizaciones en el cliente.
  - d. Solicitar y procesar todas las operaciones desde el cliente al servidor.
- e. Visualización
  - a. Incluir una imagen descriptiva para cada artículo.
- f. Carrito de compras
  - a. Mostrar un error si la cantidad solicitada supera la disponible.
  - b. Mostrar información explícita de la factura final.

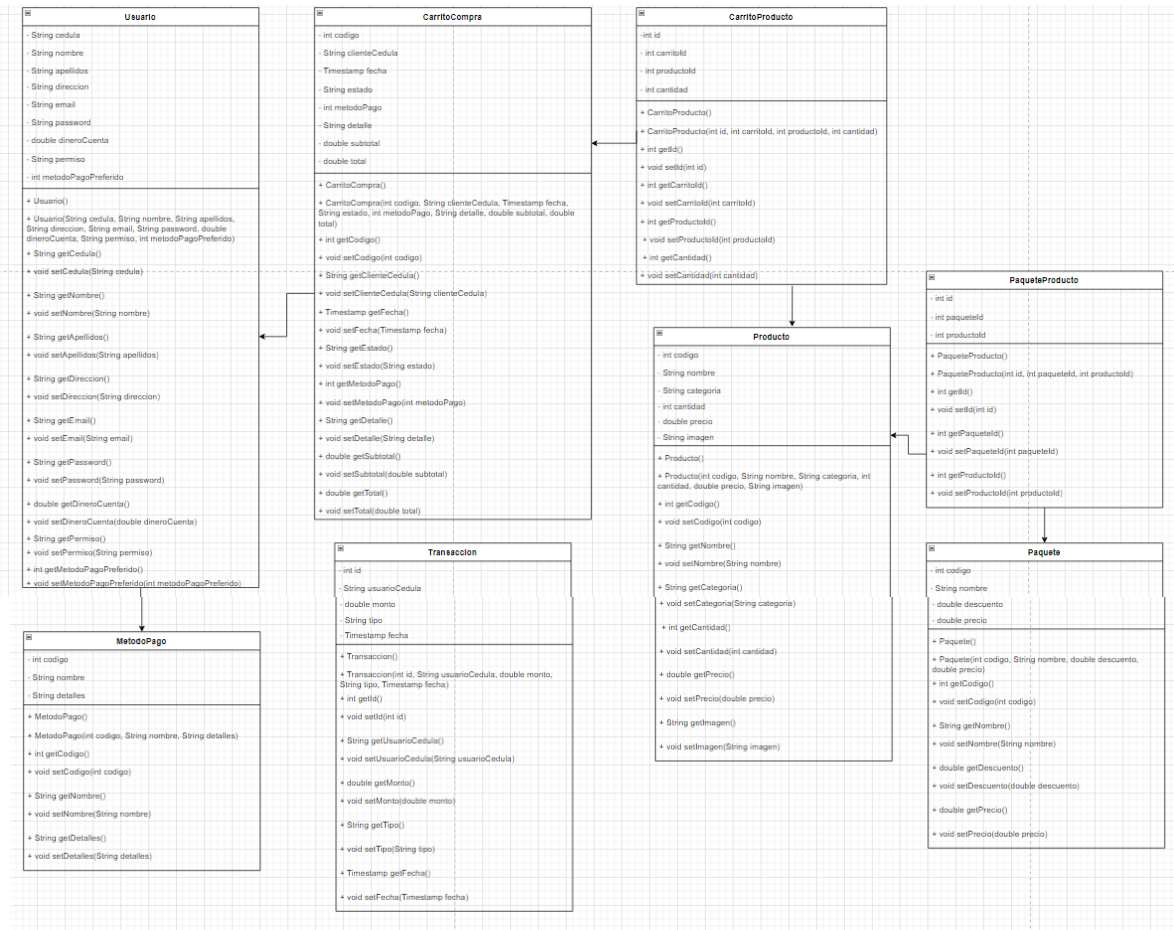


## 3 Diagramas

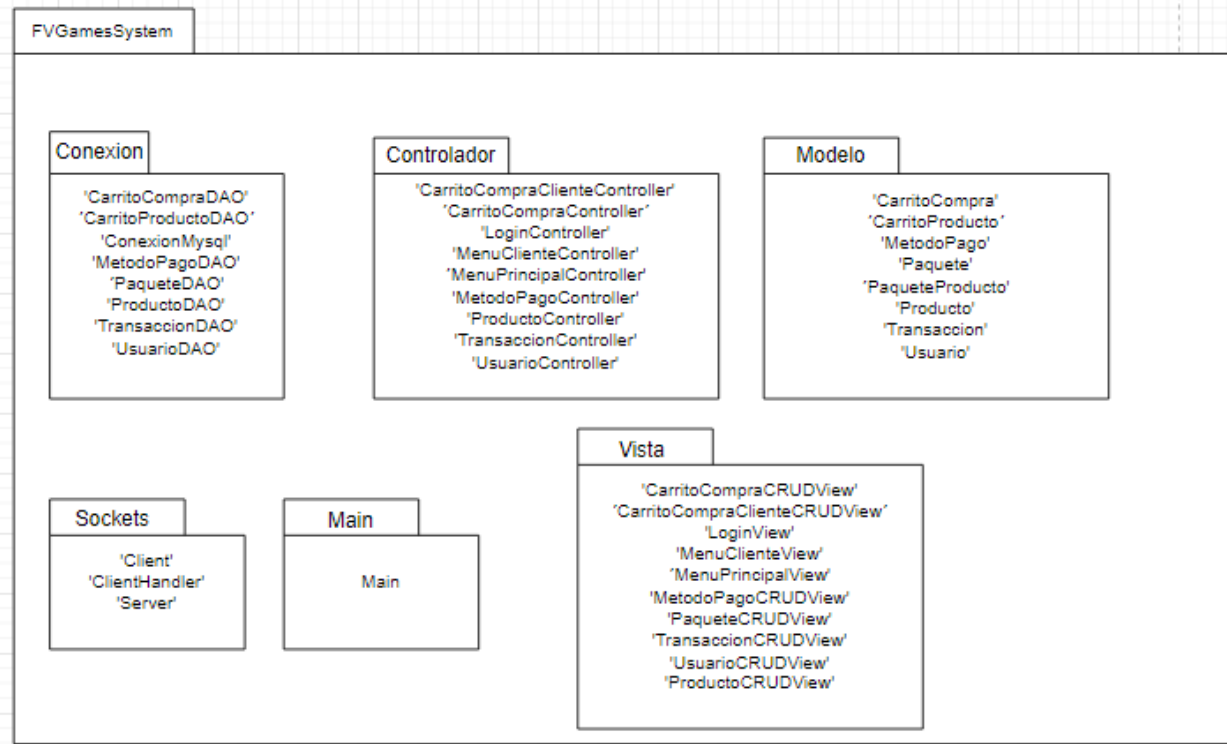
### 3.1 Diagrama general



## 3.2 Diagrama de clases



### 3.3 Diagrama de paquetes



### 3.4 Descripción general de la solución

La ejecución del proyecto se realizará utilizando software en Java, empleando NetBeans como entorno de desarrollo y Swing como interfaz gráfica. El programa permitirá tres posibles modos de acceso para los usuarios: administrador, usuario registrado e invitado.

Para los **administradores**, se ofrecerá una gama completa de funcionalidades. Los usuarios podrán visualizar y modificar los precios de los productos, restablecer o ajustar el stock, crear paquetes con hasta cinco artículos y acceder a un historial detallado de las compras realizadas y a las ganancias obtenidas. Este conjunto de herramientas permitirá a los administradores gestionar

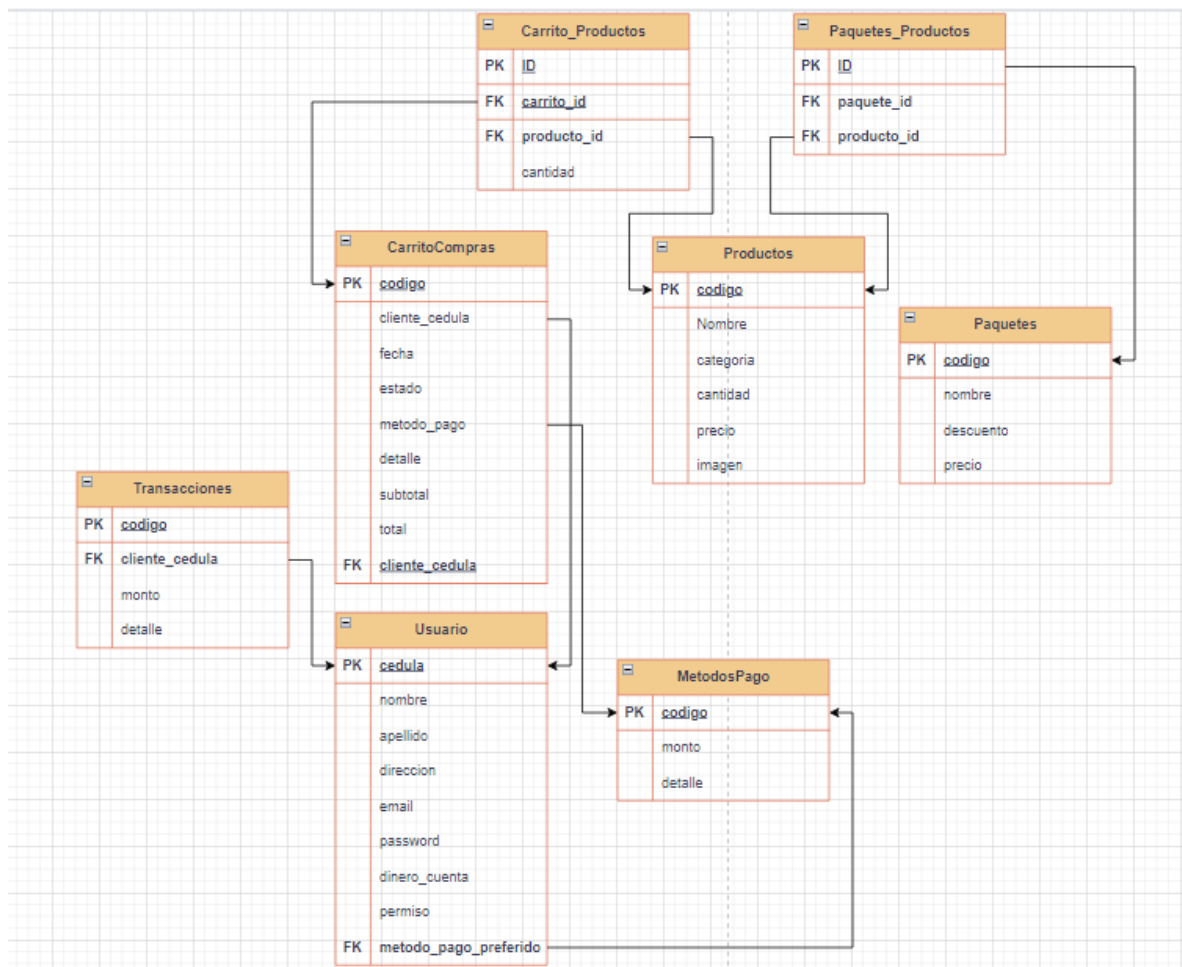
eficazmente el inventario y las ventas, asegurando que la empresa pueda mantener un control preciso sobre sus recursos y operaciones.

Los **usuarios registrados** tendrán acceso a funcionalidades diseñadas para mejorar la experiencia de compra. Podrán visualizar los productos disponibles y agregar artículos a su carrito de compras. Esta funcionalidad facilita el proceso de adquisición, permitiendo a los usuarios gestionar sus compras de manera sencilla y eficiente.

Por último, los **invitados** tendrán un acceso más limitado al sistema. Estos usuarios solo podrán visualizar los productos disponibles, sin la capacidad de realizar modificaciones ni agregar artículos a un carrito de compras. Este nivel de acceso está diseñado para que los clientes puedan explorar la oferta de productos sin registrarse, incentivando futuras compras.

Cada tipo de usuario cuenta con permisos específicos que garantizan una gestión eficiente y segura del inventario y las ventas. Este enfoque asegura que las funciones críticas del sistema las utilicen solo personas autorizadas, mientras que se da una experiencia de usuario adecuada a cada nivel de acceso.

### 3.5 Diagrama de base de datos



<https://github.com/nathi27/FVGames-Proyecto.git>

## 5 Referencias Bibliográficas

*Edrawsoft.* (s.f.). Obtenido de <https://www.edrawsoft.com/es/how-to-create-database-diagram.html>

*lucidchart.* (s.f.). Obtenido de <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-paquetes-uml>

*lucidchart.* (s.f.). Obtenido de <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>