

INSTITUTO POLITECNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y
TECNOLOGIAS AVANZADAS

Ingeniería Web

Alumno: Vargas Sosa Carlos Adair

Grupo: 2TM4

Profesor: Polanco Montelongo Francisco Antonio

Segunda entrega departamental

Fecha de entrega 20 de octubre de 2024

❖ Arquitectura del sistema

1. Componentes principales:

El componente de inicio de sesión es una pieza clave en el proyecto ya que determina las funciones que pueden hacer ciertos tipos de usuarios, en este caso se cuenta con 3 tipos: Usuario registrado, no registrado y administrador

El componente de conexión con la base de datos es fundamental ya que en todas nuestras vistas se requiere extraer información guardada en la base de datos.

2. Patrones de comunicación:

Los pasos de mensaje entre cliente servidor serán principalmente llevados a cabo por medio del protocolo HTTP quien es el encargado de gestionar la comunicación.

En el presente proyecto debido a las limitantes de conocimiento de los programadores estudiantes se decidió hacer un monolito lo que significa que es un modelo de desarrollo de software tradicional que utiliza un código base para realizar varias funciones empresariales. Todos los componentes de software de un sistema monolítico son interdependientes debido a los mecanismos de intercambio de datos dentro del sistema.

En la creación del sistema de ventas de videojuegos se tiene contemplado apegarnos en el uso de la arquitectura MVC ampliamente utilizada en la web

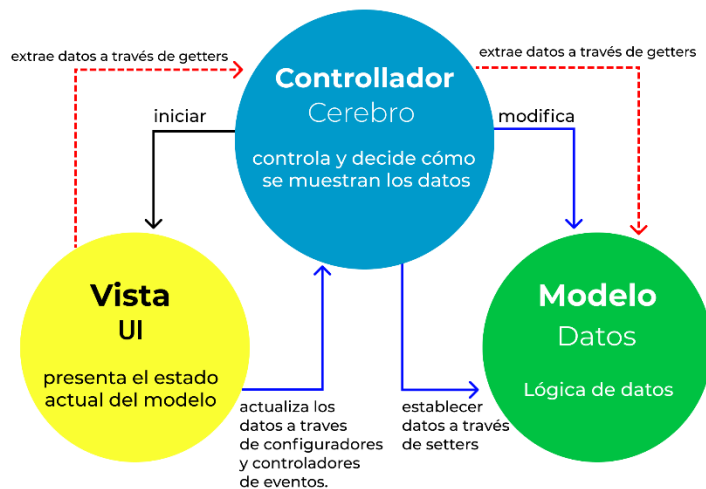
La arquitectura MVC (Modelo-Vista-Controlador) es un patrón de diseño que separa la aplicación en tres componentes distintos: Modelo, Vista y Controlador. Esta separación de responsabilidades facilita el mantenimiento y escalabilidad de la aplicación, lo cual es especialmente útil en plataformas de venta de videojuegos donde hay una gran cantidad de datos que deben gestionarse de manera eficiente.

El **Modelo** es el componente que se encarga de la lógica de negocio y de la interacción con la base de datos. En una plataforma de venta de videojuegos, el Modelo gestionará la información de los videojuegos, usuarios, compras, y transacciones.

La **Vista** es la interfaz de usuario de la aplicación. Es la parte que se encarga de mostrar la información al usuario y de recibir sus interacciones. En la plataforma de venta de videojuegos, la Vista incluirá las páginas que los usuarios ven, como la página de inicio, el catálogo de juegos, la página de detalles del juego, el carrito de compras, etc.

El **Controlador** es el intermediario entre el **Modelo** y la **Vista**. Se encarga de manejar las solicitudes del usuario, procesarlas (con la ayuda del Modelo) y luego actualizar la Vista con la información adecuada.

Patrones de Arquitectura MVC



3. Flujo de datos:

Los datos de los usuarios y datos de videojuegos se almacenan en una base de datos MySQL, en periodo de desarrollo se aloja en una maquina local que el alumno proporciona, una vez en producción esta base de datos se alojara en la nube. La información obtenida de la plataforma de videojuegos se obtiene mediante formularios de HTML estos se comunican con el servidor mediante el protocolo HTTP, la seguridad del envío de datos esta protegida por funciones nativas de html que impiden insertar código que pueda causar vulnerabilidades a nivel vasco.

Los componentes principales que hacen uso de los datos ingresados en la pagina web son principalmente utilizados por la base de datos para verificar que el usuario este con una inicio de sesión y poder verificar su identidad, en este proyecto se utiliza en correo electrónico como identificador de usuario por lo que cada usuario deberá registrase con un correo diferente.

4. Elementos técnicos:

Usar PHP y MySQL para desarrollar una plataforma de venta de videojuegos es una opción popular y viable debido a su costo accesible, facilidad de uso, alto rendimiento, y amplia comunidad de soporte. Estos elementos hacen que sea una solución robusta para crear un sitio web funcional y escalable para vender videojuegos en línea.

Hasta el momento se tiene contemplado hacer uso de Bootstrap es un framework multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web debido a que facilita estilizar la plataforma de videojuegos.

5. Aspectos no funcionales:

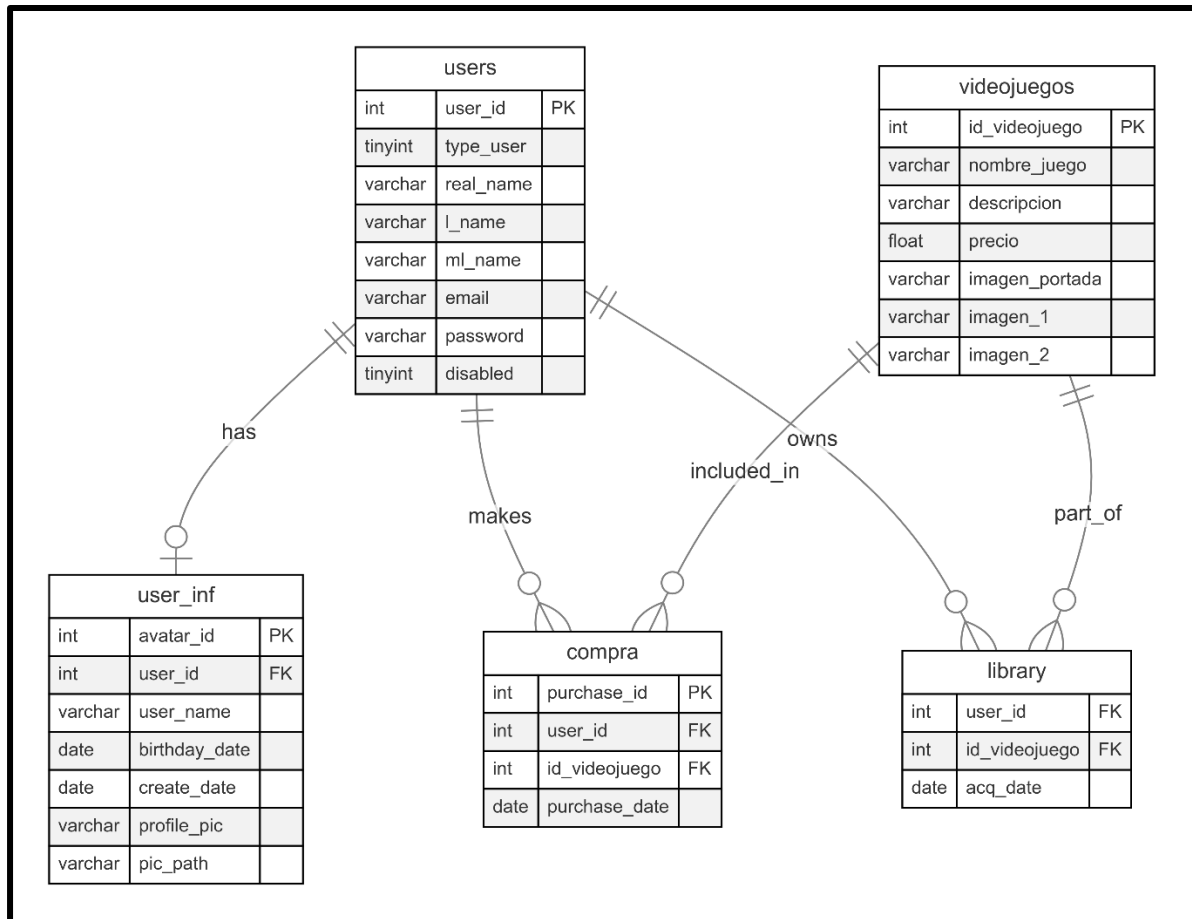
La escalabilidad de este proyecto es limitada debido a la estructura que contiene, ya que se trata de código monolítico y a la no implementación de micro servicios y programación concurrente que afecta severamente el rendimiento de la plataforma, una forma de mejorar este aspecto es implementado la arquitectura de sistema distribuido que su principal función es la de optimizar recursos de equipo que estañen red.

La implementación de técnicas que hacen mas robusto el proyecto en términos de seguridad es la implementación de formatos de contraseña ya que en la función del registro se debe colocar una contraseña que se considere robusta, además de utilizar funciones nativas de html para evitar posibles vulnerabilidades como e SQL inyección siguiendo la recomendación de W3C.

En este proyecto básico de una plataforma de venta de videojuegos no se tomaron las medidas para optimizar la plataforma web, sin embargo el uso de tecnologías como JQuery ayudaron a que el sistema tuviera un mejor rendimiento en cuanto a búsquedas, por otro lado en las bases de datos se implementaron técnicas como la creación de índices, que optimiza la búsqueda de elementos en la base de datos.

La limitación de la disponibilidad recae sobre el servidor ya que en este proyecto se considero que estuviera alojado en una maquina local para el desarrollo, para la puesta en producción se tiene contemplado hacer uso de la nube como AWS, Azure, Google Cloud etc., sin embargo, debido a que estos servicios ofrecen planes gratuitos por un corto periodo de tiempo, el funcionamiento de la plataforma web se limita a ese periodo de prueba.

❖ Modelo de datos



■ Entidades Principales:

- **users**: Almacena la información básica de los usuarios
- **user_inf**: Contiene información adicional del perfil de usuario
- **videojuegos**: Almacena los datos de los videojuegos
- **compra**: Registra las compras realizadas
- **library**: Mantiene un registro de los juegos que posee cada usuario

■ Relaciones:

- Un usuario (**users**) puede tener un perfil (**user_inf**) - Relación 1:1
- Un usuario (**users**) puede hacer múltiples compras (**compra**) - Relación 1:N
- Un usuario (**users**) puede tener múltiples juegos en su biblioteca (**library**) - Relación 1:N
- Un videojuego puede estar en múltiples compras y bibliotecas - Relación 1:N

■ Claves:

- Primarias (PK): user_id, avatar_id, id_videojuego, purchase_id
- Foráneas (FK): Las relaciones entre tablas se establecen mediante user_id e id_videojuego
 - Características importantes:
- La tabla users es la tabla central que conecta con user_inf, compra y library
- La tabla library actúa como una tabla de relación entre users y videojuegos
- Cada compra está asociada a un usuario y un videojuego específico
- La información del perfil de usuario se mantiene separada de los datos básicos de autenticación