**Índice.**

1. **Análisis.**
   1. Empresa
      1. Logo
      2. Nombre empresa
   2. Descripción del problema
   3. Descripción de la solución
   4. Stakeholders
   5. Lista de requerimientos (Código, nombre y descripción)
   6. Especificación de requerimientos (código, nombre, descripción, entrada, salida, excepción)
      1. Requerimientos funcionales
      2. Requerimientos no funcionales
2. **Diseño.**
   1. Mockups
   2. Arquitectura de software
      1. Diseño
      2. Documentación
      3. Evaluación
   3. Diagrama de clases
3. **Desarrollo**
   1. Pantallazos de la interfaz gráfica (awt – Swing)
   2. Principios SOLID
   3. Patrones Gof
4. **Análisis.**
   1. **Empresa.**
      1. Logo

Imagen que contiene dibujo

Descripción generada automáticamente

* + 1. Nombre empresa

Portal Games

* 1. **Descripción del problema.**

En las plataformas de videojuegos no hay un sistema que permita el acceso a una información detallada de las estadísticas del jugador respecto a varios juegos probados, de igual forma los usuarios no pueden compartir detalles de su progreso.

Así mismo, no hay variedad de contenido en diferentes temáticas de tal forma que la experiencia del usuario no cumple las expectativas para jugar variedad de títulos.

* 1. **Descripción de la solución.**

Mediante esta plataforma de juegos el usuario podrá visualizar, comprar y jugar siempre y cuando el videjojuego este disponible, de igual manera será más fácil ver las estadísticas del usuario.

* 1. **Stakeholders.**

Imagen que contiene texto

Descripción generada automáticamente

* 1. **Lista de requerimientos.**

|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripcion** |
| **RF01** | Iniciar sesión | Iniciar sesión mediante un usuario/correo y contraseña. |
| **RF02** | Registrar usuario | Llenar formulario de registro para crear usuario en la plataforma. |
| **RF03** | Recuperar contraseña | Mediante un código enviado por correo electrónico, se ingresa el código en la aplicación para modificar la contraseña. |
| **RF04** | Modificar perfil | Cada usuario puede modificar ciertos datos y opciones según convenga |
| **RF05** | Consultar juego en la tienda | Se cargará un listado filtrado por categoría de juego para comprar |
| **RF06** | Comprar juego | Dependiendo de las monedas que tenga el usuario, podrá comprar los juegos que desee |
| **RF07** | Recargar monedas | Mediante un código ingresado por el usuario en la aplicación, o por parte del administrador, se le otorgan monedas al usuario |
| **RF08** | Cargar juegos en biblioteca | Cuando el usuario compre un juego se añadirá a la biblioteca disponible para jugar |
| **RF09** | Jugar juego en biblioteca | Mediante la selección del juego se añade. |
| **RF10** | Tablero de juego | Una vez se haya marcado un hito en un juego, se registrarán las estadísticas tales como: tiempo jugado, puntaje, nivel, experiencia. |
| **RF11** | Ver estadísticas de usuario | Se registrará un listado de estadísticas tales como: juego, tiempo jugado, puntaje, nivel, experiencia de todos los juegos visitados por el usuario. |
| **RF12** | Recompensas | Por cada hito importante alcanzado en un juego, se recargarán monedas a la cuenta del usuario o el administrador publicará llaves únicas periódicamente el cual valen para desbloquear un juego gratis. |
| **RF13** | Añadir amigos | Se consultará un listado de usuarios el cual se podrán añadir siempre y cuando el usuario añadido acepte la solicitud. |
| **RF14** | Ver estadísticas de amigos | Una vez 2 usuarios sean amigos se podrá ver las estadísticas individuales y se hará una comparativa de las estadísticas propias. |
| **RF15** | Chat local | Una vez 2 usuarios sean amigos se podrá activar una ventana de chat, el cual podrán conversar desde 2 computadores simultáneamente. |
| **RF16** | Enviar correo | Por cada “transacción” se enviará un correo electrónico a la cuenta registrada en la plataforma. |
| **RNF01** | Interfaz grafica | Iniciar sesión mediante un usuario/correo y contraseña. |
| **RNF02** | Manuales | El programa debe contar con manuales de usuario para aprender el uso de la plataforma. |
| **RNF03** | Patrones | Se debe desarrollar el programa mediante patrones (Pizarrón y capas o cliente-servidor). |
| **RNF04** | Lenguaje de programación | Se debe desarrollar en el lenguaje de programación JAVA, siguiendo los estándares de patrones de arquitectura y estructuras de datos. |
| **RNF05** | Herramientas de desarrollo | Se debe usar el IDE Netbeans o Eclipse para el desarrollo de la aplicación. |
| **RNF06** | Persistencia de datos | El programa debe contar con persistencia de datos en un motor de base de datos tales como MySQL, Oracle o MariaDB |
| **RNF07** | Usabilidad | Todas las interfaces deben ser de fácil manejo e intuitivas para el usuario final. |
| **RNF08** | Contenido multimedia | El programa debe tener contenido multimedia para referenciar los juegos a los que se hacen alusión. |

* 1. **Especificación de requerimientos.**
     1. Requerimientos funcionales.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Código** | **Nombre** | **Descripción** | **Entrada** | **Salida** | **Excepción** |
| **RF01** | Iniciar sesión | Iniciar sesión mediante un usuario/correo y contraseña. | Usuario, contraseña | Sesión iniciada | Credenciales incorrectas |
| **RF02** | Registrar usuario | Llenar formulario de registro para crear usuario en la plataforma. | Nombre, Apellido, correo, nickname | Usuario creado | Datos incorrectos |
|
| **RF03** | Recuperar contraseña | Mediante un código enviado por correo electrónico, se ingresa el código en la aplicación para modificar la contraseña. | Correo, y código de recuperación | Ingreso de nueva contraseña | Orientado a conexión |
| **RF04** | Modificar perfil (Usuario) | Cada usuario puede modificar ciertos datos y opciones según convenga | Nombre, Apellido, correo, nickname, contraseña | Perfil modificado | Acceso a base de datos |
|
| **RF5** | Modificar perfil (Admin) | Cada usuario puede modificar ciertos datos y opciones según convenga | Nombre, Apellido, correo, nickname, contraseña, rol | Perfil modificado | Acceso a base de datos |
|
| **RF06** | Consultar juego en la tienda | Se cargará un listado filtrado por categoría de juego para comprar | Nombre del juego y categoría | Juego encontrado o no encontrado | Acceso a base de datos |
| **RF07** | Comprar juego | Dependiendo de las monedas que tenga el usuario, podrá comprar los juegos que desee | Selección de juego, método de pago | Juego comprado | Acceso a base de datos |
| **RF08** | Recargar monedas | Mediante un código ingresado por el usuario en la aplicación, o por parte del administrador, se le otorgan monedas al usuario | Cantidad de monedas que se ingresaran a una cuenta | Monedas añadidas | Acceso a base de datos |
| **RF09** | Cargar juegos en biblioteca | Cuando el usuario compre un juego se añadirá a la biblioteca disponible para jugar | Selección de juego | Juego añadido | Acceso a base de datos |
| **RF010** | Jugar juego en biblioteca | Mediante la selección del juego se añade. | Inicio de juego | Empezar el juego | Acceso a base de datos |
| **RF11** | Tablero de juego | Una vez se haya marcado un hito en un juego, se registrarán las estadísticas tales como: tiempo jugado, puntaje, nivel, experiencia. | Datos obtenidos del usuario | Estadísticas de los jugadores | Acceso a base de datos |
| **RF11** | Ver estadísticas de usuario | Se registrará un listado de estadísticas tales como: juego, tiempo jugado, puntaje, nivel, experiencia de todos los juegos visitados por el usuario. | Datos obtenidos del usuario del juego | Estadísticas del jugador | Acceso a base de datos |
| **RF12** | Recompensas | Por cada hito importante alcanzado en un juego, se recargarán monedas a la cuenta del usuario o el administrador publicará llaves únicas periódicamente el cual valen para desbloquear un juego gratis. | Estadísticas del jugador | Recargar monedas | Acceso a base de datos |
| **RF13** | Enviar correo | Por cada “transacción” se enviará un correo electrónico a la cuenta registrada en la plataforma. | correo del usuario | Correo enviado | Orientado a conexión |
| **RF14** | Ver estadísticas del juego | Una vez 2 usuarios sean amigos se podrá ver las estadísticas individuales y se hará una comparativa de las estadísticas propias. | Datos de usuario | Estadisticas totales del juego | Acceso a base de datos |

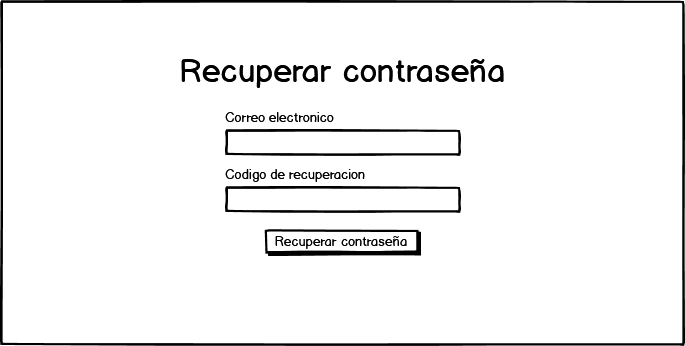
* + 1. Requerimientos no funcionales.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Código** | **Nombre** | **Descripción** | **Entrada** | **Salida** | **Excepción** |
| **RNF01** | Interfaz grafica | Aplicación contara con componentes gráficos para si respectivo uso. | N/A | N/A | Errores de diseño |
| **RNF04** | Lenguaje de programación | Se debe desarrollar en el lenguaje de programación JAVA, siguiendo los estándares de patrones de arquitectura y estructuras de datos. | N/A | N/A | Errores de sintaxis |
| **RNF05** | Ambiente de desarrollo | Se debe usar el IDE Netbeans o Eclipse para el desarrollo de la aplicación, arquitectua. |  |  | Errores de sintaxis |
| **RNF06** | Persistencia de datos | El programa debe contar con persistencia de datos en un motor de base de datos MySQL. | Query | Crud | Acceso a base de datos,  Errores de sintaxis |
| **RNF07** | Usabilidad | Todas las interfaces deben ser de fácil manejo e intuitivas para el usuario final. | N/A | N/A | Mala distribución de las interfaces |
| **RNF08** | Contenido multimedia | El programa debe tener contenido multimedia para referenciar los juegos a los que se hacen alusión. | N/A | N/A | Falta de imágenes |

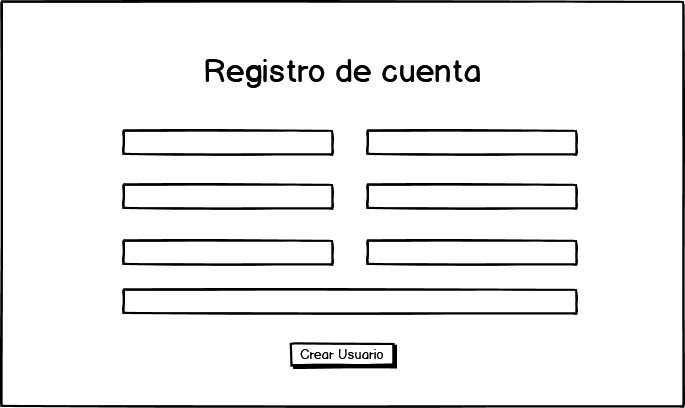
1. **Diseño.**
   1. **Mockups.**
      1. Login

****

* + 1. Recuperar contraseña

****

* + 1. Registro cuenta

****

* + 1. Inicio

Captura de pantalla de un celular

Descripción generada automáticamente

* + 1. Tienda

Captura de pantalla de un celular

Descripción generada automáticamente

* + 1. Biblioteca de juegos

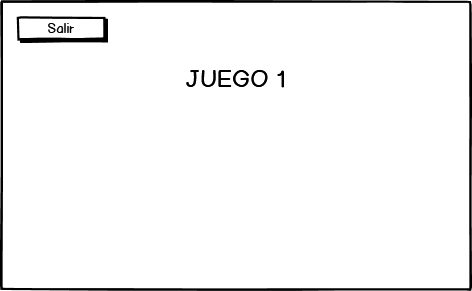
Captura de pantalla de un celular

Descripción generada automáticamente

* + 1. Detalles de juego

****

* + 1. Juego

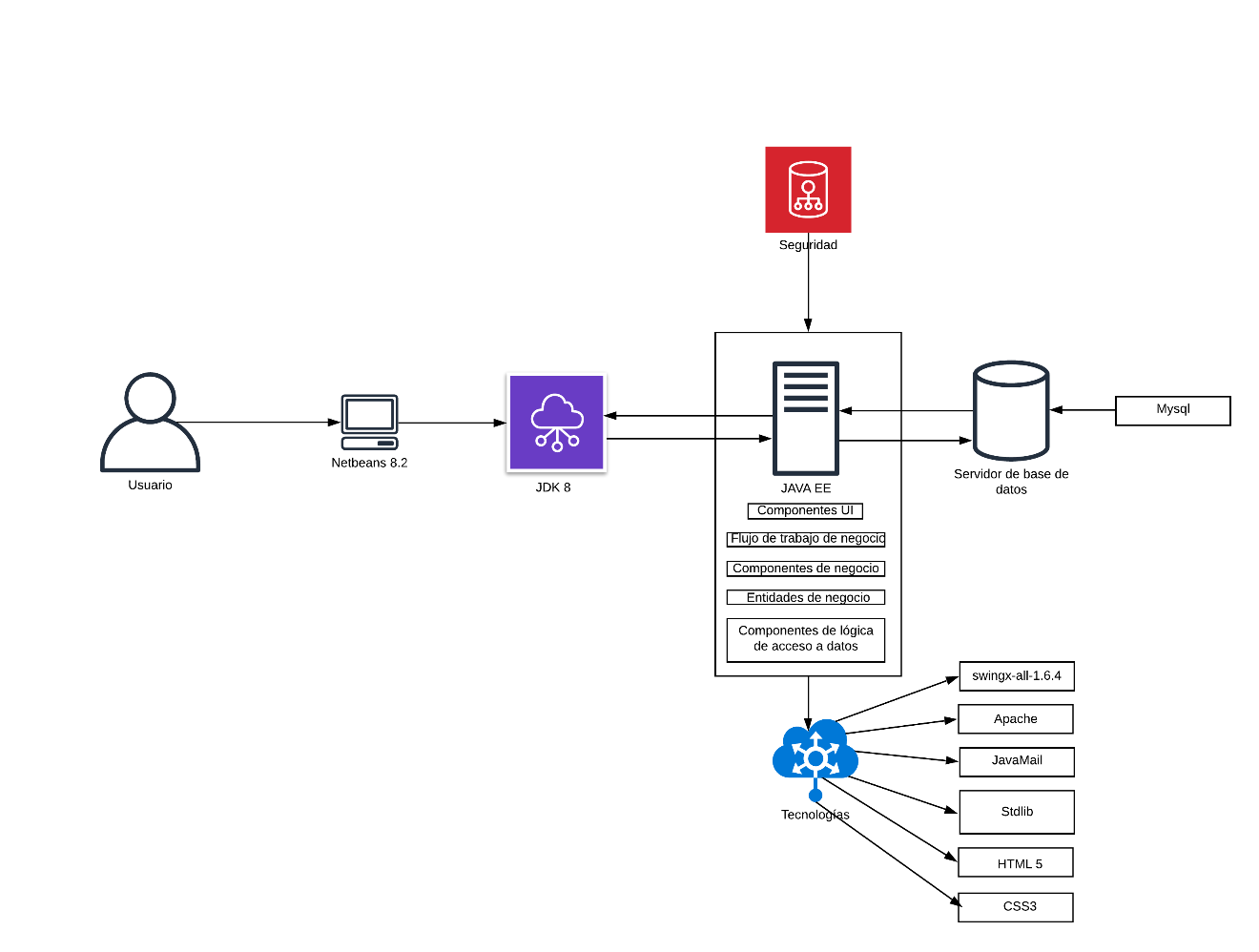
****

* + 1. Estadísticas usuario

Captura de pantalla de un celular

Descripción generada automáticamente

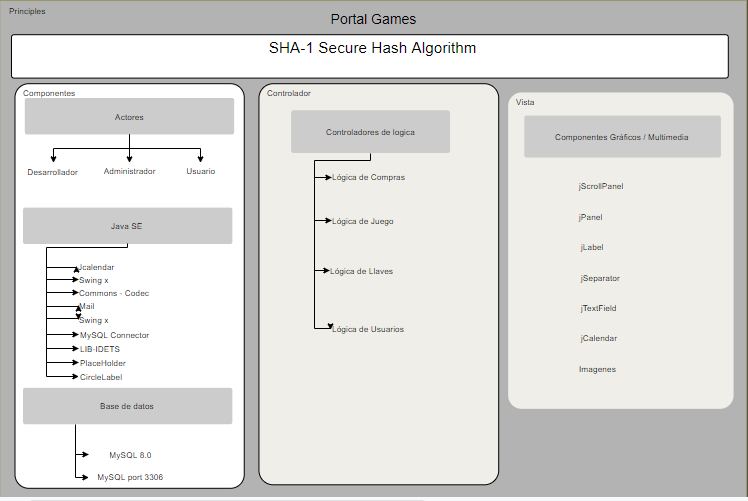
* 1. **Arquitectura de software.**
     1. Diseño.



v.1

Captura de pantalla de un celular con letras

Descripción generada automáticamente

v.2

v.3

* + 1. Documentación.

La arquitectura definida **cliente-servidor** comienza por un usuario, el cual accede a un cliente Windows y mediante el IDE Netbeans 8.2 obtiene los datos del servidor para su misma interacción. El servidor tiene componentes UI (gráficos), flujo de trabajo de datos, componentes de negocio, entidades de negocio y componentes de lógica de acceso a datos.

Los componentes de lógica de acceso a datos maneja tecnologías y librerías sirviendo como herramientas para el desarrollo de la aplicación, dichas herramientas son: swingx-all-1.6.4 (el cual permite el autocompletado de datos mediante un JCombobox), servidor local Apache, librería Java Mail (la cual permite la conexión de la aplicación por medio de un protocolo “smpt” a Gmail o hotmail), adicionalmente cuenta con la librería stdlib (la cual permite el desarrollo de juegos), finalmente cuenta con el lenguaje de etiquetas HTML y CSS3 para establecer una plantilla de diseño cuando llegan correos electrónicos generados desde la aplicación.

El servidor cuenta con encriptación de contraseñas, parametrización de funcionalidades asegurando principios de seguridad tales como la confiabilidad, confidencialidad e integridad.

El servidor se conecta a un servidor de base de datos (Mysql) el cual es administrado por un gestor de base de datos MySQL Workbench 8.0 CE.

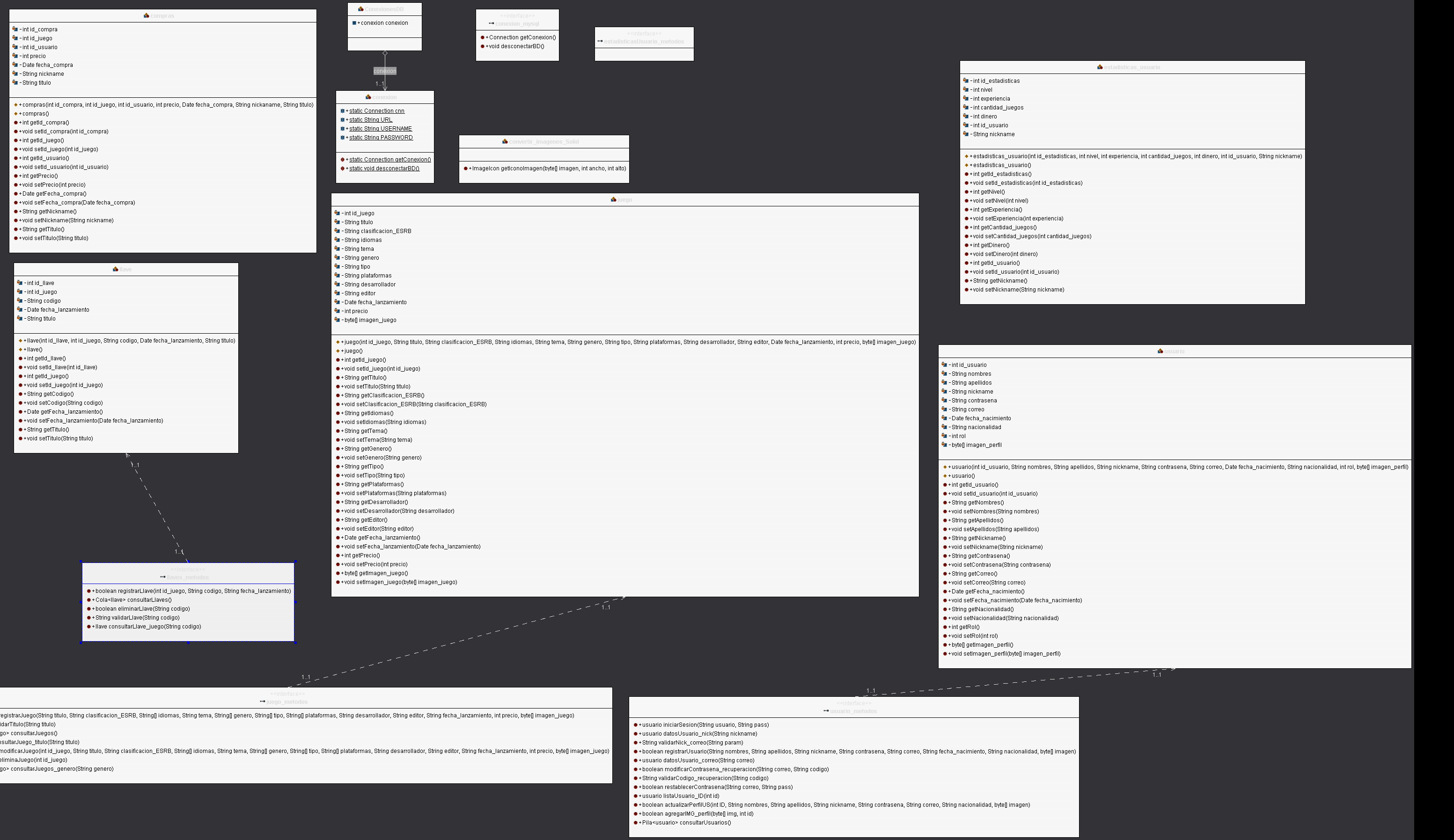
La arquitectura definida **Pizarra** representa las funcionalidades de la aplicación las cuales son Iniciar sesión, registrar usuario, recuperar contraseña, modificar perfil, consultar juegos, comprar juegos, recargar monedas, cargar juegos a biblioteca, jugar juego en biblioteca, tablero de juego, ver estadísticas de juego, donde mediante unos parámetros enviados desde la vista al controlador se conecta a la base de datos retornando una conexión y la lógica del controlador retorna un resultado a la vista, recuperar contraseña, de la vista al controlador del controlador al modelo, del modelo a la conexión, y de la conexión a la base de datos, el correo por el contrario.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Paquete** | **Clases** | **Descripción clase** | **Métodos** | **Descripción Métodos** |
| **Modelo** | conexion\_mysql (interface) | Conexión a la base de datos MySQL | Super clase, clase hija, constructor |  |
| estadisticasUsuario\_metodos (interface) | Ventana donde se calculan las horas de juego de cada usuario |  |  |
| juego\_metodos (interface) | Interfaz en la cual se genera la estructura para inicializar los métodos | juego\_metodos | En esta intefaz se raliza gran parte de la lógica para inicializar los juegos |
| llaves\_metodos (interface) | Interfaz en el cual se generan la estructura de las llaves | llaves\_metodos | Se inicializan los métodos de el CRUD de las llaves |
| usuario\_metodos (interface) | Método en el cual se genera la estructura de los usuarios | usuario\_metodos | En esta interfaz se ejecutan todos los métodos que contiene la inetrfaz |
| compras | Método en el que se adquiere un juego y se agrega a la biblioteca del usuario | compras | Clase en la que se puede hacer toda la edición de las compras |
| convertir\_imagenes | Convierte los bytes de una imagen por separado a una imagen completa en la interfaz | convertir\_  imagenes\_  Solid | En esta clase se uso el principio ingle Responsability para convertir bytes a una imagen completa |
| estadisticas\_usuario | Clase en la que se generan las estadísticas del usuario |  |  |
| juego | Se establecen los respectivos datos del juego | Juego | Se crean todos los métodos en los cuales se puede editar o modificar |
| llave | Clase en la que se generan las llaves | llave | En esta clase se crearon los Getter y setter de las llaves |
| usuario | Clase en la que se definen los datos del usuario | usuario | Clase en la que se hacen todas las operaciones básicas a la hora de crear un usuario |
| **Vista Modulo ADMIN** | CrearJuego | Ventana donde hay cuadros de texto y otros scrolls con opciones | CrearJuego | Se desarrolló para la creación de juegos de tal forma en la que pida datos y los almacene al nuevo juego |
| CrearLlave | Clase en la que el admisntrador puede asignar llaves a un juego | LogicaLlave  CrearLlave | En esta clase se crea una llave y se le asigna a una copia de un juego |
| DetallesJuego | Ventana donde se identifican las características del juego | Juego  DetallesJuego | En esta clase se crean los datos del juego que se mostraran al usuario antes de comprarlo |
| DetallesUsuario | Ventana donde se identifican las características del juego | Usuario  DetallesUsuario | En esta clase el administrador podrá cambiar el rol del usuario demás de poder agregar sado |
| GestionJuegos | Ventana en la que es posible editar eliminar los juegos | Juegos  Gestionjuegos | El administrador podrá añadir, eliminar, buscar, editar los juegos |
| GestionLlaves | Método en el que es posible agregar editar o elminar llaves ya ingresadas | Llaves  GestionLlaves | El administrador podrá crear las llaves y asignarla a los juegos |
| GestionUsuarios | Método en el que es posible agregar editar o eliminar usuarios | Usuario  GestionUsuarios | El administrador seleccionara un Usuario para poder hacer su correspondiente edición |
| PrincipalADM | Ventana en la que el admin tiene acceso a todas las ventanas | PrincipalADM | Pestaña en la cual se nave por toda la interfaz del menú del administrador |
| **Vista Modulo LOGIN** | Login | Ventana | Login | Se validaran los datos ingresados con la base de datos |
| RegistroCuenta | Pestaña donde el usuario ingresara su nombre, | RegistroCuenta | Se almacenaran los datos ingresados en la base de datos para ser calidados y llamados en un futuro |
| OlvideContrasena | Pestaña donde el usuario escribirá su correo ingresado a en el registro de su cuenta | Usuario  OlvideContrasena | El usuario podra recuperar su contraseña por medio de un código de recuperación que llegara de forma automática a su correo |
| CodigoRecuperacion\_cuenta | Código que se generara a partir | Usuario.  OlvideContrasena. CodigoRecuperacion  \_cuenta. | Se genera el código de recuperación el cual es enviado al correo |
| RestablecerContrasena | Dos cuadros de texto, uno contraseña nueva y la segunda confirmación de esta | CodigoRecuperacion  \_cuenta. | Se envía a una pestaña en la cual el usaurio podra cambiar su contraseña después de haber ingresado el código |
| **Vista Modulo USUARIO** | ActualizarContrasenaUS | Se actualizará la contraseña del usuario | Usuario  ActualizarContrasenaUS | El usuario podra cambiar sui contraseña siempre y cuando este dentro de la cuenta |
| ActualizarPerfilUS | Se editarán los datos del perfil | Usuario  ActualizarPerfilUS | El usuario tendrá la oportunidad de editar sus datos |
| BibliotecaUS | Se podrá visualizar los juegos en peretencia | BibliotecaUS | El usuario tendrá una |
| DetallesJuego\_tienda | Ventana o pestaña donde se ve el nombre del juego, una descripción, su costo y si está disponible | DetallesJuego | Se llaman los datos del juego para ser mostrados |
| Juego1 | Clase en la que se establecen los parámetros de un juego | LogicaJuego  Juego1 | Se inicializan los parámetros del juego |
| Modal\_imgPerfil | Imagen que se le asigna a un perfil | Modal\_imgPerfil | Se llamara la imagen de la base de datos y será mostrada |
| PrincipalUS | Ventana en la cual es lusuario podrá navegar por el programa | PrincipalUS | El usuario podrá navegar a la pestaña que prefiera |
| TableroJuego1 | Venta en la que se ejecuta el juego 1 | LogicaJuego  TableroJuego1 | Se llama e inicia el juego con sus parámetros |
| TiendaUS | Intefaz en la que el usuario podrá comprar los juegos de su preferencia | TiendaUS | Se añadirá a las pertenencias del usuario loego de haber efectuado el pago |
| **Controlador** | logicaCompras | Se gestionara la compra de juegos | TiensaUS  logicaCompras | Calse en la que se hace la resta o validación de los juegos |
| logicaEstadisticas\_usuario | Se gestionaran las estadísticas de los usuarios |  |  |
| logicaJuego | Se gestionara el funcionamiento del juego | logicaJuego | Se retornan todos los datos del juego para poder ejecutarse |
| logicaLlave | Se desarrollo la lógica del funcionamiento de las llaver para obtener juegos | logicaLlave | Clase en la cual se envían los datos ingresados de la creación para ser almacenados |
| logicaUsuario | Se desarrollaron las acciones que el usuario tendrá | Usuario  logicaJuego | Clase en la cual se envían los datos ingresados de la creación para ser almacenados |
| **Principipal** | principal | Clase en la cual es inicia la interfaz para dar un correcto funcionamiento al programa | principal | Se llaman todos los parámetros para dar al ejecución |

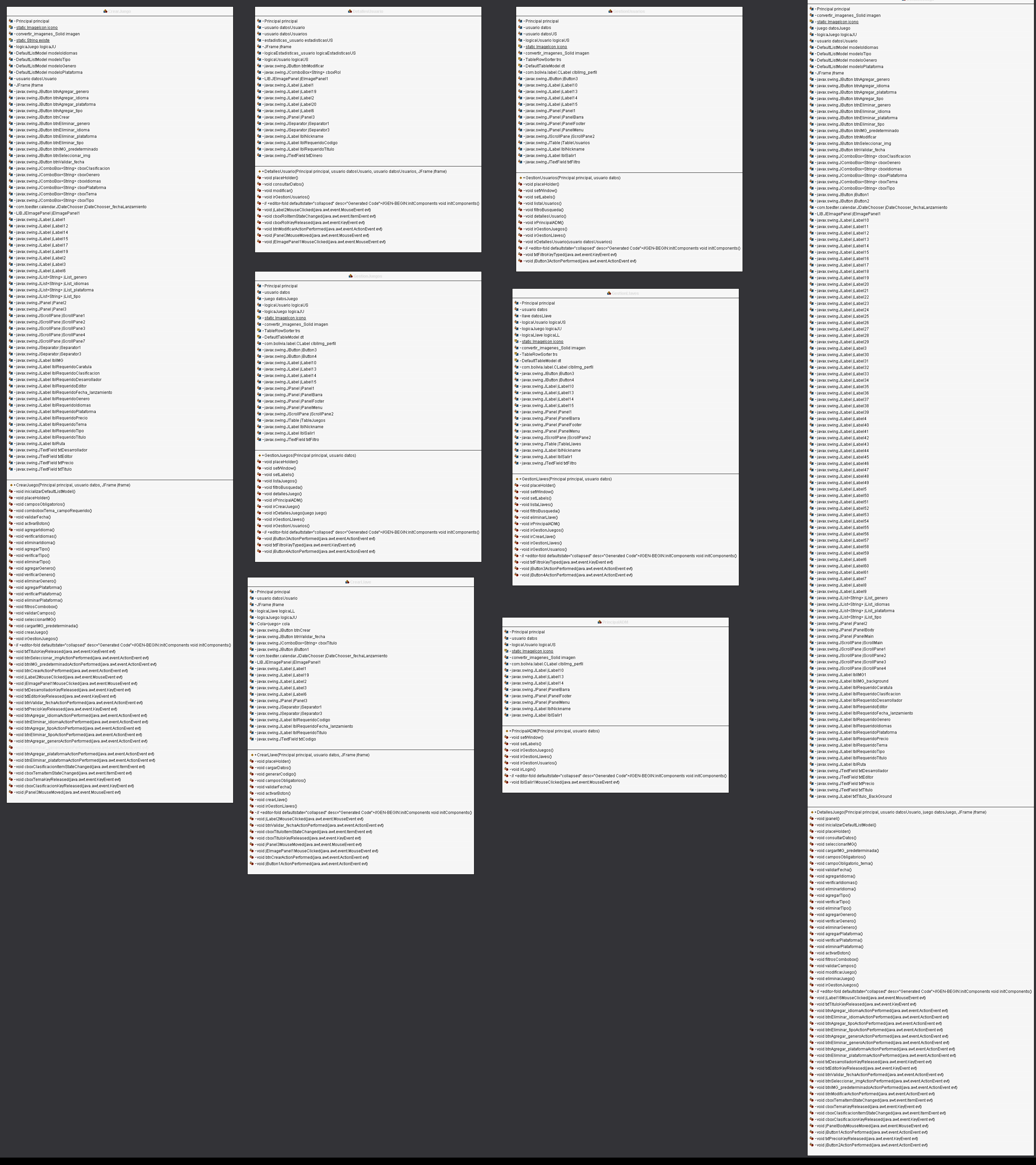
* + 1. Evaluación.

|  |  |
| --- | --- |
| **Criterio** | **Puntaje** |
| **Pertinencia/ Relevancia** | 8 |
| **Eficacia** | 7 |
| **Eficiencia** | 8 |
| **Impacto** | 6 |
| **Sostenibilidad** | 8 |
| **Comunicación** | 9 |
| **Creatividad** | 6 |
| **Diseño** | 8 |
| **Estilo arquitectónico** | 9 |
| **Funcionalidad** | 69 |

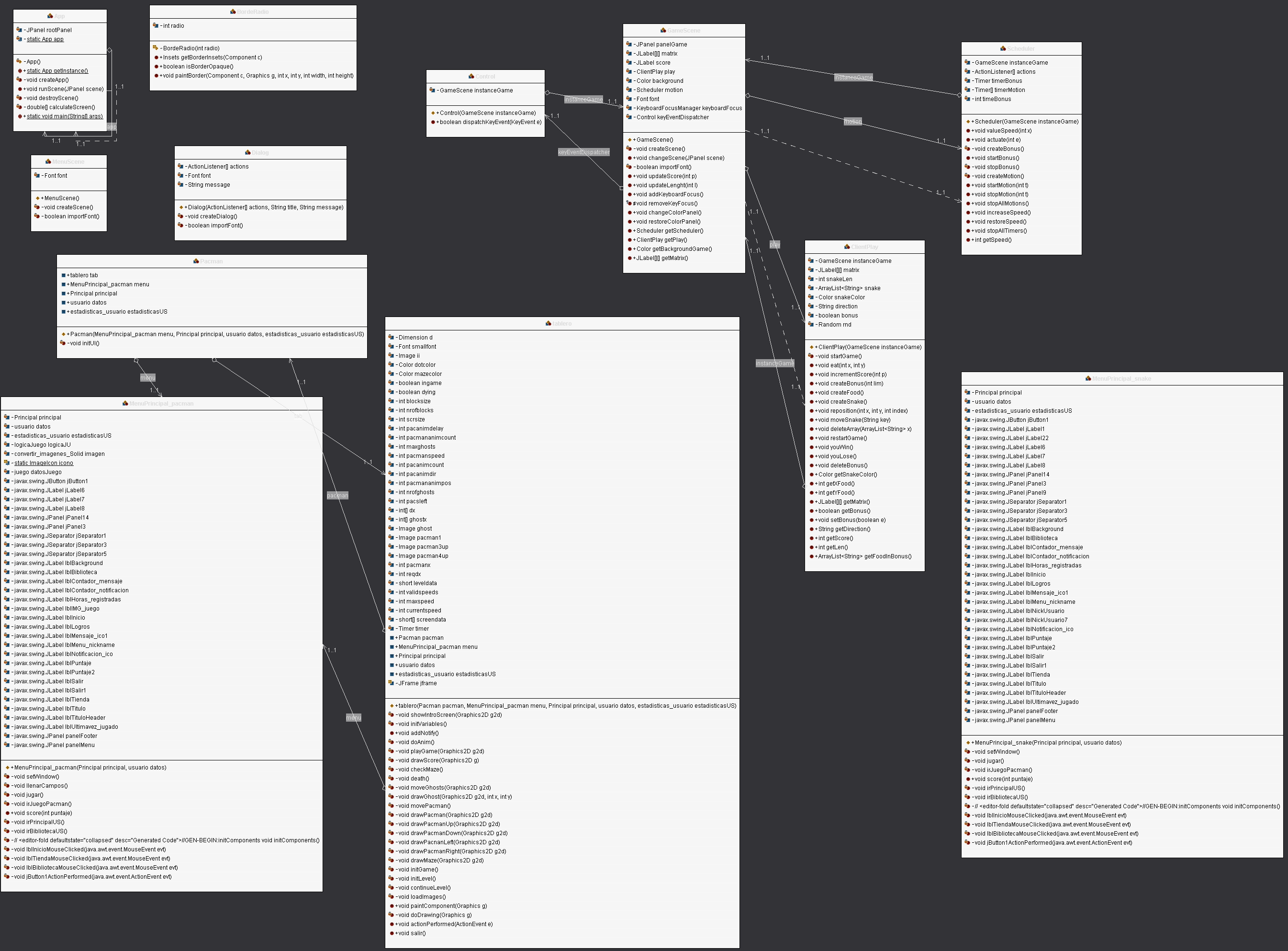
* 1. **Diagrama de clase.**

Modelo

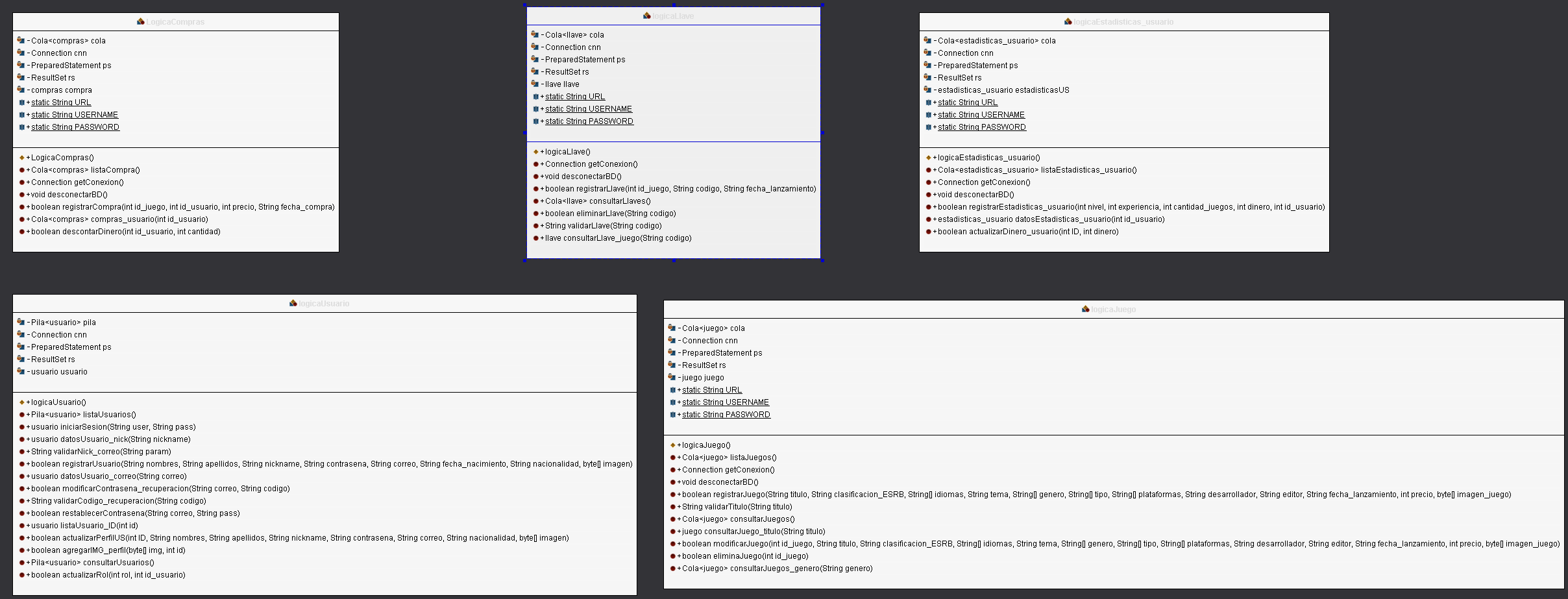
Vista – Administrador



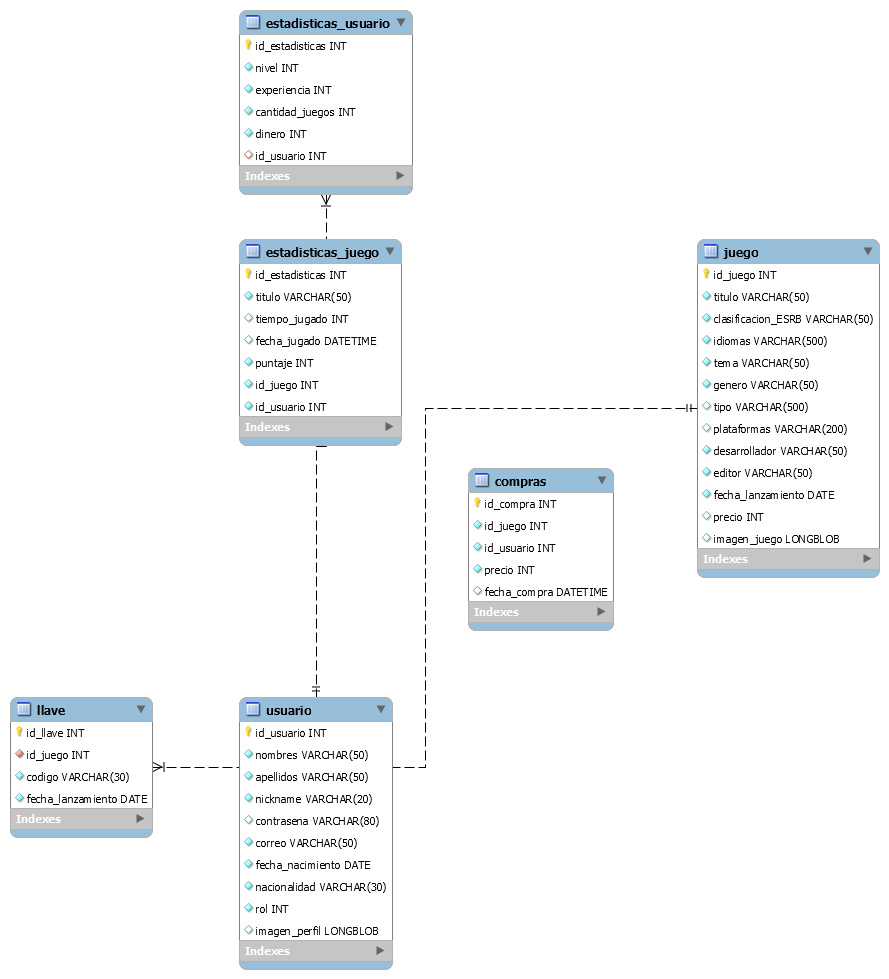
Vist.ususario



Controlador



Modelo Relacional



1. **Desarrollo.**
   1. **Interfaz gráfica.**
   2. **Principios SOLID**

|  |  |  |
| --- | --- | --- |
| **Principios** | **Uso** | **Código Fuente** |
| **Single responsability Principle** | Creación de imágenes | Conversión de bits a imagen |
| **Open Close principle** | Clase conexión | Se usa para hacer la conexión a diferentes bases de datos |
| **Interface Segregation Principle** | Lógica de juegos | Solo se llama la interfaz requerida en cada acción o proceso |
| **Singleton** | Creación de imágenes |  |
| **Builder** | Creación de interfaces | permite la creación de una variedad de objetos complejos desde un objeto fuente |

**3.3 Patrones Gof**

|  |  |  |
| --- | --- | --- |
| **Patrones** | **Uso** | **Código Fuente** |
| **Facade** | Creación de variables y estructuras de datos | Conversión de bits a imagen |
| **Adapter** | Conversión de divisas | Se usa para hacer la conexión a diferentes bases de datos |

En los patrones gof se encuentran los patones creacionales, estructurales y de comportamiento