

Manual Técnico – Gestor de Tarjetas

Realizado por: Javier Alejandro Mérida Gómez - 202230368

Descripción del Software:

¡Bienvenido al "Gestor de Tarjetas"!

Este sistema es una herramienta integral utilizada por una entidad bancaria para gestionar registros de tarjetas de crédito, sus respectivos movimientos y otras funciones adicionales. El sistema está conectado a una base de datos MySQL, la cual se encarga de almacenar y manejar la información de manera eficiente.

Requisitos del Sistema

- Plataforma: El juego fue desarrollado para ejecutarse en sistemas operativos Windows y Linux.
- Entorno de Desarrollo: Se utilizó Apache NetBeans IDE 20 para el desarrollo del código y MYSQL para el desarrollo de la base de datos.
- Lenguaje de Programación: El juego se implementó en Java (Programación Orientada a Objetos), versión 17.
- Sistema de Desarrollo: El desarrollo se llevó a cabo en un entorno Ubuntu Linux.

Componentes del Juego

Backen:

Se encarga de todos los procedimientos, funciones y conexiones para el funcionamiento del sistema de gestión de datos, cuenta con las siguientes clases:

- ✓ Movimientos: esta clase cuenta con un único método llamado realizarMovimiento, este acepta como parámetros el numero de tarjeta, la fecha, el tipo, la descripción, el establecimiento y el monto.

Obtiene primero el saldo y estado de la tarjeta, realiza la comprobación del estado, si el valor es verdadero continua con el movimiento. Verifica el tipo de movimiento que sea realizar, si es un cargo hace una verifcaron mas, si el usuario cuenta con mas saldo del monto del retiro para poder continuar. Realiza la operación correspondiente y actualiza el saldo y añade un nuevo movimiento en la base de datos.

- ✓ Tarjetas: esta clase maneja las tarjetas y cuenta con tres métodos.

CancelarTarjeta: acepta como parámetro un numero de tarjeta, obtenemos de primero el salo y limite de la tarjeta que se desea cancelar, si el usuario posee alguna deuda no

se podrá cancelar la tarjeta hasta que pague el monto de la deuda, se pide una comprobación para cancelar la tarjeta, si el usuario confirma cambia el estado de la tarjeta y no se puede utilizar para ningún movimiento.

GenerarNumeroTarjeta: este método acepta como parámetros un prefijo de la tarjeta y la cantidad de tarjetas existentes del tipo de tarjeta, realiza una operación para poder crear el número de tarjeta con su prefijo y que se encuentre en secuencia y de forma automática.

RegistrarTarjeta: este método acepta como parámetro el código de solicitud, el límite de crédito y el tipo de tarjeta, inicia verificando el tipo de tarjeta que es y selecciona el prefijo a utilizar, se crea una nueva tarjeta con los datos proporcionados.

- ✓ Solicitudes: esta clase maneja todo lo relacionado a las solicitudes, este cuenta con tres métodos:

numeroDeSolicitud: este es una forma automática de obtener el número de solicitud que se realiza, cuenta las rows creadas en la base de datos y le suma uno y así obtenemos el número de solicitud.

NuevaSolicitud: recibe como parámetros todos los datos solicitados para una nueva solicitud y los carga a la base de datos.

AutorizacionTarjeta: recibe como parámetro el código de solicitud, obtenemos primero el salario y tipo de tarjeta, verificamos que el 60% del salario del usuario sea mayor al límite por tipo de tarjeta y si es así continuamos con el proceso, si todo está correcto cargamos la información a la base de datos.

- ✓ Reportes: Se maneja en la búsqueda y creación de los reportes necesarios para la aplicación, tenemos cuatro tipos diferentes de búsqueda con diferentes parámetros, estos son los 4 métodos que tenemos en donde cada uno busca información diferente:

consultaTarjeta, reporteListadoSimple, reporteListadoSoliSimple, reporteEstadoCuentasSimple.

- ✓ Gestor: Esta clase se encarga de la conexión con la base de datos y la conexión entre clases.

Fronted:

Se encarga exclusivamente de la parte visual y de interacción con el usuario, el fin de ella es ser amigable con el usuario y facilitar el uso de la aplicación y hacerlo lo más sencillo posible para el uso ordinario de la aplicación.

Cuenta con un frame principal, el cual se encarga en conectar todos los internalFrame.

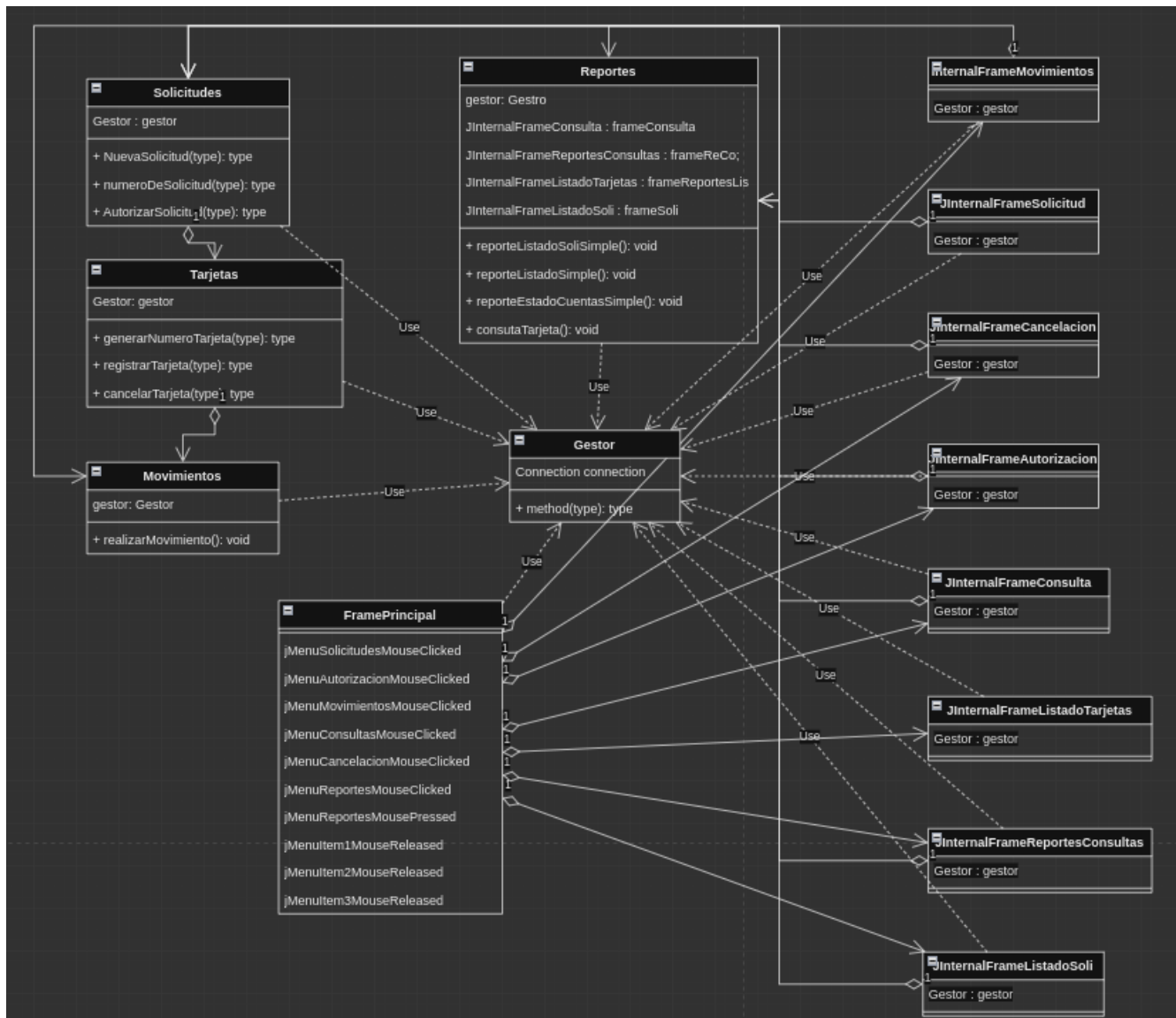
Cuenta con 8 internalFrame, cada uno de ellos son necesario para la interfaz grafica.

El nombre de sus clases son:

- FramePrincipal
- InternalFrameMovimientos
- JinternalFrameAutorizacion
- JinternalFrameCancelacion
- JinternalFrameConsulta
- JinternalFrameListadoSoli
- JinternalFrameListadoTarjetas
- JinternalFrameReportesConsultas
- JinternalFrameSolicitud

El uso de estas clases es exclusivo para la interfaz grafica.

Diagrama de clases



Mapeo físico de la DB.

DDL

Crear esquema:

```
CREATE SCHEMA gestor_tarjetas;
```

```
USE gestor_tarjetas ;
```

Crear tabla SOLICITUD

```
CREATE TABLE solicitud (  
    codigo_solicitud INTEGER NOT NULL,  
    nombre_solicitante VARCHAR(99) NOT NULL,  
    salario_solicitante DECIMAL(15, 2) NOT NULL,  
    Fecha_solicitud DATE,  
    tipo_tarjeta VARCHAR(13) NOT NULL,  
    direccion_solicitante VARCHAR(149) NOT NULL,  
    estado_solicitud BOOLEAN NOT NULL,  
    CONSTRAINT PK_SOLICITUD PRIMARY KEY(codigo_solicitud)  
);
```

Crear tabla TARJETA

```
CREATE TABLE tarjeta (  
    numero_tarjeta VARCHAR(19) NOT NULL,  
    cod_solicitud INTEGER NOT NULL,  
    limite_tarjeta DECIMAL(10, 2) NOT NULL,  
    estado_tarjeta BOOLEAN,  
    saldo_tarjeta DECIMAL(10, 2) NOT NULL,  
    CONSTRAINT PK_TARJETA PRIMARY KEY(numero_tarjeta),  
    CONSTRAINT FK_SOLICITUD_IN_COD_SOLICITUD  
        FOREIGN KEY (cod_solicitud) REFERENCES solicitud(codigo_solicitud)  
);
```

Crear tabla MOVIMIENTOS

```
CREATE TABLE movimientos (  
    numero_movimiento INTEGER NOT NULL AUTO_INCREMENT,  
    num_tarjeta VARCHAR(19) NOT NULL,  
    fecha_movimiento DATE,  
    tipo_movimiento VARCHAR(5) NOT NULL,  
    descripcion VARCHAR(200) NOT NULL,  
    codigo_establecimiento VARCHAR(10) NOT NULL,  
    monto DECIMAL(15, 2) NOT NULL,  
    CONSTRAINT PK_MOVIMIENTOS PRIMARY KEY(numero_movimiento),  
    CONSTRAINT FK_TARJETA_IN_NUM_TARJETA  
        FOREIGN KEY (num_tarjeta) REFERENCES tarjeta(numero_tarjeta)  
);
```

Diagrama E/R

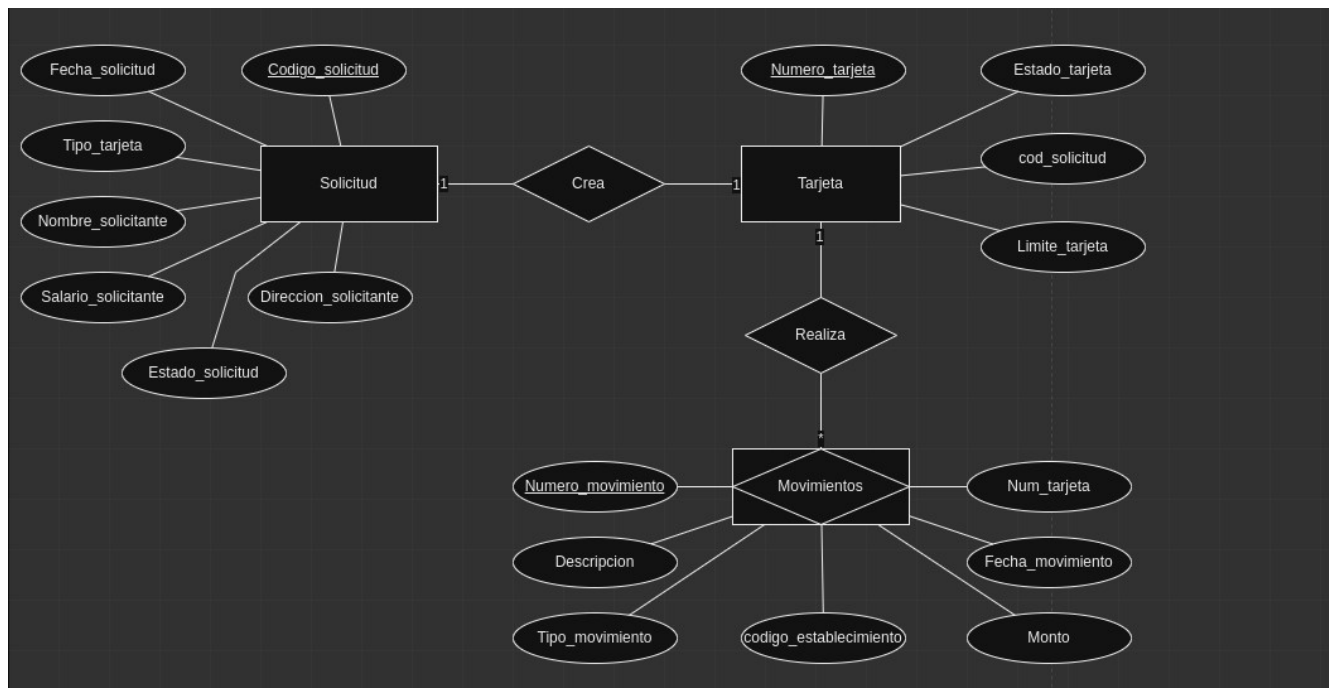


Diagrama de Tablas

