A

ADO.NET: Es el sistema que nos permite conectar nuestra app de C# con bases de datos, como MySQL. Piensa en esto como el mensajero que lleva nuestras instrucciones (consultas SQL) hasta la base y nos trae de vuelta los resultados. Incluye clases como MySqlConnection, MySqlCommand, MySqlDataReader y muchas más.

AutoIncrement: Es una forma cómoda de decirle a MySQL: "cada vez que inserte un nuevo usuario, cuenta, etc., aumenta automáticamente este número". Se usa en IDs.

B

Base de Datos (BD): Es donde guardamos toda la información de los usuarios, cuentas, remesas, etc. Es como un Excel gigante, pero estructurado y preparado para trabajar con muchas consultas al mismo tiempo.

BeginTransaction(): En C#, nos permite agrupar varias operaciones SQL como si fueran una sola. Si algo falla, se puede deshacer todo (con Rollback()), y si todo sale bien, se confirma con Commit().

Button Events (C#): Son las acciones que ocurren cuando un usuario hace clic en un botón. Por ejemplo, hacer login, realizar un retiro, etc.

C

CHAR(n): Tipo de dato que guarda texto con longitud fija. Si dices CHAR(4) y guardas "AB", igual ocupa 4 espacios.

Check Constraint: Es una regla que se aplica a un campo para limitar los valores permitidos. Por ejemplo, asegurarse de que el monto de un servicio nunca supere $50.

Commit(): Si abriste una transacción, este método confirma todos los cambios que hiciste. Es como decir: "Guarda todo lo que acabo de hacer".

ConnectionString: Cadena de texto que indica dónde está la base de datos, el usuario, la contraseña y el nombre de la BD. Es la llave para conectar tu app a la BD.

CRUD: Es el acrónimo de las operaciones básicas en una BD: Crear, Leer, Actualizar y Eliminar (Create, Read, Update, Delete).

CURDATE(): Función SQL que devuelve la fecha actual (sin hora). Muy útil para comparar vencimientos.

C# (C Sharp): Es el lenguaje de programación que usamos para construir toda la interfaz del cajero, manejar eventos y conectarnos a la base.

D

DataGridView (DGV): Control visual en Windows Forms que muestra tablas de datos. Ideal para mostrar cosas como transacciones, remesas, pagos pendientes.

DataTable: Objeto de C# que guarda en memoria los datos que se traen desde MySQL. Se llena usando MySqlDataAdapter.

DateTime: Tipo de dato que representa una fecha y una hora. Se usa mucho para vencimientos, pagos, fechas de envío, etc.

DELETE: Comando SQL que elimina filas de una tabla.

DialogResult: Valor que representa la decisión del usuario tras un cuadro de diálogo (ej. Aceptar o Cancelar). Muy usado para confirmar pagos, retiros, etc.

E

ENUM: Tipo especial de campo que solo acepta ciertos valores predefinidos. Por ejemplo, tipo_servicio solo puede ser 'agua', 'luz' o 'telefono'.

ExecuteNonQuery(): Ejecuta una consulta SQL que no devuelve datos, como un INSERT, UPDATE o DELETE.

ExecuteReader(): Ejecuta una consulta SELECT y permite recorrer fila por fila con un reader.

ExecuteScalar(): Ejecuta una consulta que solo devuelve un valor. Se usa mucho para obtener el saldo de una cuenta.

F

FOREIGN KEY: Clave foránea. Relaciona dos tablas para mantener coherencia. Ejemplo: cada cuenta está relacionada con un usuario.

FormClosed: Evento de Windows Forms que se dispara cuando un formulario se cierra. Usado para volver a mostrar el formulario anterior.

G

GUI (Graphical User Interface): Es la parte visual de la app con la que interactúa el usuario: botones, cajas de texto, labels, etc.

I

IF EXISTS / IF NOT EXISTS: Condicionales SQL para crear o borrar tablas solo si existen o no existen. Evitan errores de ejecución.

INSERT INTO: Comando SQL para insertar una nueva fila en una tabla.

J

JOIN: Función SQL que permite unir dos tablas usando una relación común. Muy usado para unir usuarios y cuentas.

L

LIMIT: Limita la cantidad de resultados en una consulta SQL. Ejemplo: LIMIT 1 devuelve sólo una fila.

LIKE: Operador SQL que permite hacer búsquedas por patrones, usando % como comodín.

Login: Proceso para acceder al sistema ingresando usuario y pin (o hash).

M

MessageBox.Show(): Función de C# que muestra mensajes al usuario. Se usa para errores, confirmaciones o notificaciones.

MySqlCommand: Clase en ADO.NET que representa un comando SQL (SELECT, INSERT, etc.) que se va a ejecutar.

MySqlConnection: Representa una conexión abierta a una base de datos MySQL.

MySqlDataAdapter: Clase que llena un DataTable con los datos devueltos por una consulta.

MySqlDataReader: Objeto que permite leer los resultados fila por fila tras un SELECT.

N

NOW(): Función SQL que devuelve la fecha y hora actuales del servidor.

NULL: Valor que representa "vacío" o "sin valor". En este sistema se evita usarlo para campos importantes.

NumericUpDown (nud): Control de la interfaz para seleccionar números. Usado para montos personalizados en retiros y depósitos.

P

Parámetros (@nombre, @pin, etc.): Valores que se pasan a una consulta SQL de forma segura. Previenen inyección SQL.

Password Hash: Versión cifrada del pin/contraseña del usuario. En este proyecto usamos SHA256.

Primary Key: Clave que identifica de manera única cada fila. Ejemplo: id_usuario, id_cuenta.

R

Reader.Read(): Método que avanza una fila en el resultado de una consulta y permite acceder a sus columnas.

Remesa: Transferencia que un usuario puede enviar a otro, quien podrá retirarla sin tarjeta usando un código único.

Rollback(): Si una transacción falla, este método deshace todos los cambios hechos desde que se empezó.

S

SELECT: Comando SQL que permite consultar datos de una o más tablas.

String Interpolation: En C# se refiere a construir una cadena insertando variables dentro de ella usando $"Hola {nombre}".

Stored Procedure: Bloque de código SQL guardado en la base. No lo usamos, pero es una forma de reutilizar lógica compleja del lado del servidor.

T

Tabla: Estructura que contiene datos en filas y columnas. Por ejemplo: usuarios, cuentas, transacciones.

Transaction: Grupo de instrucciones SQL que se ejecutan juntas. Si una falla, ninguna se guarda.

Transacción (tabla): Registro que guarda información sobre operaciones como retiros, depósitos, pagos, etc.

TRY-CATCH: Bloque en C# que permite manejar errores sin que la aplicación se caiga. Muy usado para conexiones y consultas.

U

UPDATE: Instrucción SQL para modificar valores existentes en una tabla.

Usuario: Persona que usa el sistema. Puede ser un cliente o una entidad bancaria. Cada uno tiene su propia cuenta, datos y acciones permitidas.

V

Validación de campos: Revisión de que los datos que el usuario ingresó no estén vacíos, tengan el formato correcto o existan en la base.

VARCHAR(n): Tipo de dato que almacena texto de longitud variable, hasta un máximo definido por n.