**UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA**

**SEDE BOCA DEL MONTE**

**ING. SISTEMAS DE INFORMACIÓN Y CIENCIAS DE LA COMPUTACIÓN**

**CURSO: BASES DE DATOS**

Logotipo, Calendario

Descripción generada automáticamente**SECCIÓN: “B”**

**MANUAL TÉCNICO  
PROYECTO BASES DE DATOS DISTRIBUIDAS**

**CATEDRÁTICO:  
EDGAR LÓPEZ**

**INTEGRANTES**

Edras Fernando Tatuaca Alvarado 7690-22-19119

Cesar Alberto Tecún Leiva 7690-22-11542

Pablo Antonio Ispache Arriaga 7690-17940

Jose Daniel Tobar Reyes 7690-21-13125

Bryan Manuel Pineda Orozco 7690-16-8869

**MANUAL TÉCNICO**

Se desarrollo una web app, en conjunto con una base de datos en MySQL , aplicando los conocimientos adquiridos durante el curos de Bases de Datos II , para el desarrollo de la web app se utilizo lenguaje Python , y se hizo uso del framework flask ; por otra parte, se utilizó Visual Estudio Code como el entorno de desarrollo.

Se trabajó la gestión de la base de datos de un sistema bancario el cuál cuenta con diferentes sucursales que necesitan acceder a la información e ingresar nuevos valores a la base de datos, es decir, se desarrolló una base de datos distribuida, que nos permite manejar el acceso a la información por medio de roles.

Los roles tendrán acceso a una sucursal en específico, por otra parte también se cuenta con el usuario administrador de base de datos el cual tiene acceso a toda la información de la base de datos y a la gestión de roles , es decir , este puede dar permisos , crear roles y asignar roles a los usuarios.

**Requerimientos Técnicos**

Para que la web app funcione de forma correcta , se deben cumplir con los siguientes requisitos :

|  |  |
| --- | --- |
| Sistema Operativo | Windows o MacOS |
| Entorno de Desarrollo | Visual estudio Code ver. 17.14 |
| Desarrollo de la interfaz | Python |
| Memoria Ram mínima | 2 GB |
| Framework | Flask |
| Herramienta para manejo de contenedores | Docker ver. 28 |

**Diagrama entidad relación**

A continuación, se muestra el diagrama de nuestra base de datos

**Una captura de pantalla de una red social

El contenido generado por IA puede ser incorrecto.Descripción de las Entidades**

**Tabla Usuarios**

En ella se almacenan los datos principales de los usuarios del sistema como lo es su nombre , usuario, contraseña y sucursal asignada.

Se cuenta con un ID único para cada usuario esto para evitar conflictos al momento de autenticar un usuario o que la información de este este repetida.

**Tabla Clientes.**

Esta tabla contiene la información de los clientes , también contiene información de contacto , esto si en determinado momento el banco necesitase contactar con el cliente o necesite compartirle algún documento. Esta tabla esta directamente relacionada con la tabla cuentas a través del atributo cliente\_id , y de forma indirecta con compañas y tarjetas haciendo uso del mismo atributo

**Tabla Cuentas**

En esta tabla se almacenan las cuentas asociadas a los clientes, cabe recalcar que un cliente puede tener mas de una cuenta , esto dependiendo del fin de esta , pudiendo ser de ahorro , corriente o bien de plazo . Se relaciona directamente con la tabla de clientes , en una relación de uno a muchos , como se mencionaba anteriormente un cliente puede tener más de una cuenta.

**Tabla transacciones**

Acá se registran todos los movimientos que se realizan a las cuentas , como lo puede ser un depósito , un retiro o bien una transferencia . En esta tabla observamos una relación de uno a muchos con la tabla de cuentas , ya que en una misma cuenta se pueden realizar diversas transacciones.

**Tabla tarjetas**

En esta tabla es donde se almacenará la información de tarjetas , ya sean estas de crédito o debito , también acá se podrá definir el estado de la tarjeta , es decir si esta esta bloqueada, activa o en su defecto vencida. Se tiene una relación de uno a muchos con la tabla de clientes, dado que un cliente puede tener varias tarjetas , pero una tarjeta solo puede permanecer a un cliente.

**Tabla campañas**

Está tabla almacena información de campañas de marketing realizadas por el banco para atraer más clientes o bien fortalecer las relaciones existentes. Se pretende almacenar cuando ocurrió la campaña y por que medio fue que se ejecuto .

**Tablas auxiliares**

En ciertos casos es necesario la elaboración de una tabla auxiliar que nos permita el manejo de relaciones de muchos a muchos . Para efectos de la web app , fue necesario crear una tabla que relacione el cliente y la tarjeta , así como el cliente y la campaña , haciendo más eficiente el realizar consultas de dichas relaciones.

**Autenticación de roles**

El sistema garantiza que solo determinados usuarios tengan acceso a la información y funcionalidades , según su rol dentro del banco . Esto con el objetivo de tener un mejor rendimiento , y un incremento en la seguridad, dado que los usuarios solo tendrán acceso a la información que sea pertinente.

Se cuentan con varios usuarios como admin\_user , el cual es el administrador y tienen permisos y acceso a la base de datos , luego tenemos la replica que es de solo lectura y que nos servirá para la distribución de la base , por ultimo tenemos un usuario ejemplo por cada sucursal , dichos usuarios solo tienen acceso a la información en su respectiva sucursal.

Cada usuario se autentica usando su nombre de usuario y contraseña , definidos mediante le plugin mysql\_native\_password , compatible con la mayoría de conectores y sistemas de control de acceso.

**Análisis de la interface**

Al ejecutar el programa este nos desplegará una pantalla de autenticación la cual esta enlazada a la tabla de usuarios , al ingresar se nos desplegará las opciones disponibles , como lo es crear un registro , eliminarlo , etc . De acuerdo a el nivel de permisos de rol y la información a la que cuenta acceso el usuario en cuestión .

La interfaz de la web app es posible a partir de el uso de ngrok. Esta herramienta es un proxy inverso distribuido globalmente que asegura , protege y acelera aplicaciones y servicios de red . Para este caso en particular se esta compartiendo el puerto 5001 de un computador , y de esta forma se logra acceder a la interfaz de usuario.

**Fragmentación y distribución de los datos**

Los datos se dividieron en 5 bases de datos , los cuales son: sucursal1 , sucursal2,sucursal3 estas 3 se encargan del manejo de clientes y cuentas . Contamos con credit que maneja las tarjetas y la asociación con clientes y por ultimo mercadeo que maneja las campañas.

**Transparencia para el usuario**

Los usuarios desde el navegador , no deben preocuparse en qué base esta cada dato , el DDBMS en flask , junto con ProxySQL y SQLAlchemy , oculta la distribución y ofrece una interfaz unificada.

**Coordinación mediante middleware (DDBMS central )**

Existe un archivo llamado ddbms , que actúa como un coordinador inteligente , consultando a las distintas bases según el tipo de operación y sucursal.

**Control de acceso distribuido y consulta federada**

Cada una de las bases utiliza un usuario ProxySQL restringido (read\_sucursalX , read\_credit, etc) respetando seguridad por dominio.

Por otra parte en la consulta federada se recuperan datos desde múltiples nodos y los consolida como si fueran una sola tabla lógica.

**Actualizaciones Futuras**

Se plantearon las siguientes mejoras que pueden mejorar el desempeño y eficiencia del programa posteriormente, las cuales se describen a continuación

* **Auditoria :** Se busca registrar logs de acceso , cambios o acciones obre cuentas, usuarios o tarjetas, para facilitar el seguimiento ante errores o fraudes.
* **Cifrado de contraseñas :** Con el fin de garantizar una mejor seguridad se plantea el cifrado de contraseñas de 16 bits como mínimo para evitar accesos no deseados.
* **Reglas de negocio :** Implementación de restricciones como triggers o procedimientos que eviten retiros sin fondos, actualicen el saldo en forma automática al realizar una transacción y bloqueen las tarjetas vencidas en forma automática.
* **Procedimientos y vistas :** Crear procedimientos almacenados para operaciones comunes: abrir una cuenta , asignar una tarjeta , aplicar una transacción