**UNIVERSIDAD CATÓLICA BOLIVIANA “SAN PABLO”**

**UNIDAD ACADÉMICA SANTA CRUZ**



**PROYECTO**

**“CALZADOS ELIZABETH”**

**INTEGRANTES:**

**Henry Alberto Coronado Villca**

**Carlos Samuel Torrico Lopez**

**Introducción**

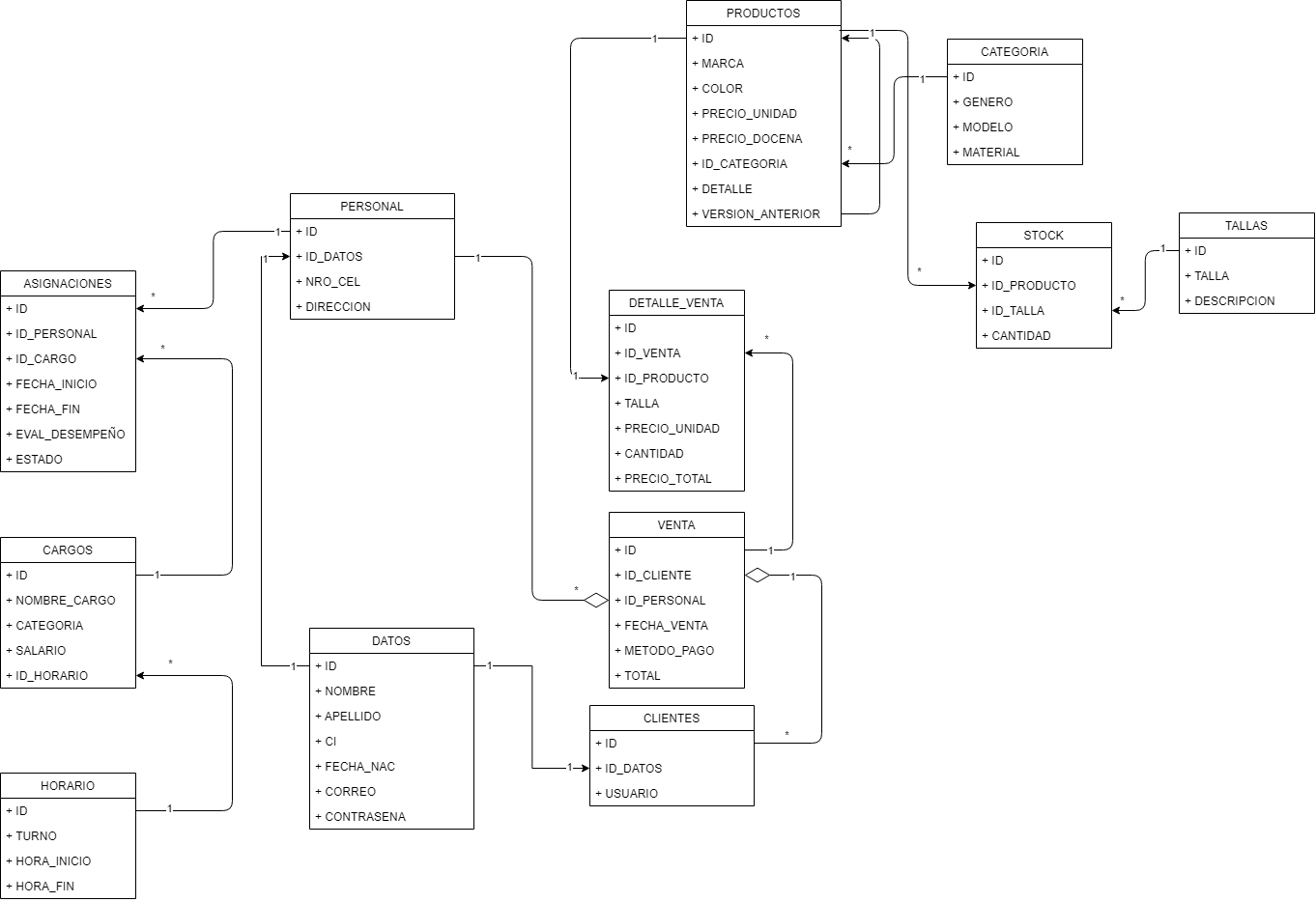
**El proyecto trata sobre una página de una tienda de zapatos llamada Calzados Elizabeth. En la página el usuario tiene un entorno donde puede ver el inventario desde una opción llamada catálogo, además tiene la opción de crearse una cuenta , y entrar en ella, el privilegio de crearse la cuenta es que estará habilitado para compras en nuestras tiendas, además de que podrá ver su registro de sus futuras compras, también hay otra opción de registro para personas que desean empleo en la empresa, y dejan sus datos que posteriormente serán revisadas por el administrador, además contamos con la opción de login para los clientes que crearon su cuenta y para el personal, también hay otra opción para el administrador donde se puede aceptar o no a los que enviaron su solicitud.**

**Objetivos**

**Nuestros objetivos son crear un entorno sencillo donde el usuario pueda interactuar y ver los catálogos sin dificultad.**

**También que, si hay personas interesadas en trabajar en la tienda, puedan enviar su solicitud y esperar a que el gerente vea si son aptos.**

**Estructura de proyecto**

****

**Base de datos**

Para nuestra base de datos creamos en total 12 tablas:

**HORARIO**

Propósito: Almacena información sobre turnos de trabajo y establece la hora de inicio y finalización de cada turno de trabajo.

Uso: Puede organizar y gestionar los horarios de trabajo de los empleados.

**DATOS**

Propósito: Contiene información personal como nombre, apellido, CI, fecha de nacimiento, correo y contraseña.

Uso: Se utiliza para almacenar información personal de clientes y empleados.

**PERSONAL**

Propósito: Almacena cierta información personal, como número de celular y dirección, y enlaces a la tabla DATOS.

Uso: Gestionar información detallada de las personas que trabajan en la empresa.

**CARGO**

Propósito: Almacenar información sobre diversos puestos de la empresa incluidos salarios y horas de trabajo relacionadas.

Uso: Puede organizar y gestionar las funciones y responsabilidades de los empleados.

**ASIGNACIONES**

Propósito: Almacenar pedidos de empleados para puestos específicos durante un periodo de tiempo específico con detalles como calificación y estado.

Uso: Puedes controlar las tareas de los empleados en diferentes roles.

**CATEGORÍA**

Propósito: Contiene información sobre categorías de productos como género, estilo y material.

Uso: Clasificar los productos de la tienda en diferentes categorías.

**TALLAS**

Propósito: Almacenar información sobre las tallas de productos disponibles.

Uso: Facilita la gestión de inventarios y ventas determinando los tamaños de los productos.

**STOCK**

Propósito: Registrar el número de productos en stock y combinarlos con determinadas tallas.

Uso: Gestionar inventario de productos a la venta.

**PRODUCTO**

Propósito: Contiene información detallada de cada producto, como marca, modelo, color y precios.

Uso: Gestionar información detallada sobre los productos de la tienda.

**CLIENTE**

Propósito: Almacena información sobre los clientes, incluidas las identificaciones de usuarios, y está vinculada a la tabla DATOS.

Uso: Gestionar información sobre los clientes que compran en la tienda.

**VENTA**

Propósito: Almacenar información sobre una venta, incluido el cliente, los empleados, la fecha, el método de pago y el monto total.

Uso: Facilita el seguimiento y gestión de transacciones de ventas.

**DETALLE\_VENTA**

Propósito: Registrar información detallada sobre los productos vendidos en cada transacción, incluyendo cantidad, precio unitario y total.

Uso: Permite realizar un seguimiento detallado de los productos vendidos en cada venta.

**Consultas**

Ver catálogo:

**\_ Seleccionamos algunas columnas de las siguientes tablas:**

Tabla PRODUCTO: Selecciona las columnas ID, MARCA, COLOR y PRECIO UNIDAD.

Tabla CATEGORÍA: Selecciona las columnas GÉNERO, ESTILO y MATERIAL.

Tabla TALLAS: Selecciona la columna TALLA.

**\_ Declaramos las tablas y relaciones:**

Tabla PRODUCTO: Se utiliza para obtener información básica del producto.

Tabla CATEGORÍA: Se realiza un JOIN basado en la relación entre el ID de la categoría del producto y el ID de la categoría.

Tabla STOCK: Se realiza un JOIN para obtener información relacionada con el stock del producto.

Tabla TALLAS: Se realiza un JOIN para obtener información de la talla del producto.

**\_ Resultado:**

Como resultado final podemos ver los detalles completos de cada producto, incluyendo ID, marca, color, precio por unidad, género, estilo, material y talla. Este procedimiento es útil para generar un catálogo detallado de productos para presentación o análisis.

Ver personal y su cargo

**\_ Selección de Columnas con Alias:**

Tabla PERSONAL: Selecciona el ID del personal y lo renombra como "ID Personal".

Tabla DATOS: Selecciona nombre, apellido, CI y correo del personal.

Tabla ASIGNACIONES: Selecciona fechas de inicio, fin y estado de las asignaciones.

Tabla CARGO: Selecciona nombre del cargo y salario.

Tabla HORARIO: Selecciona el turno del personal.

**\_ Declaración de Tablas y Relaciones:**

Tabla PERSONAL: Se utiliza como la tabla principal para obtener información básica del personal.

Tabla DATOS: Se realiza un JOIN basado en la relación entre el ID de datos del personal y el ID de datos.

Tabla ASIGNACIONES: Se realiza un JOIN para obtener información relacionada con las asignaciones del personal.

Tabla CARGO: Se realiza un JOIN para obtener información relacionada con el cargo del personal.

Tabla HORARIO: Se realiza un JOIN para obtener información del horario del cargo.

**\_ Resultado:**

La consulta proporciona un resultado detallado con información clave sobre el personal, sus asignaciones, cargo y horario.

El cliente que compro mas productos

Selección de Columnas y Funciones de Agregación:

Tabla VENTA: Selecciona el ID del cliente de la venta.

Tabla DETALLE\_VENTA: Selecciona el ID del producto de la venta.

Tabla CLIENTE: Selecciona el nombre y apellido del cliente.

COUNT(DISTINCT DV.ID\_PRODUCTO): Cuenta la cantidad total de productos distintos comprados por el cliente.

**\_ Declaración de Tablas y Relaciones:**

Tabla VENTA: Utilizada como la tabla principal para obtener información relacionada con las ventas.

Tabla DETALLE\_VENTA: Se realiza un JOIN para vincular las ventas con los detalles de la venta.

Tabla CLIENTE: Se realiza un JOIN para obtener información sobre el cliente.

**\_ Agrupación y Ordenamiento:**

GROUP BY: Agrupa los resultados por ID del cliente, nombre y apellido del cliente.

ORDER BY: Ordena los resultados en orden descendente según la cantidad total de productos comprados.

**\_ Limitación de Resultados:**

FETCH FIRST ROW ONLY: Limita el resultado a la primera fila, es decir, al cliente que ha comprado la mayor cantidad de productos.

**\_ Resultado:**

La consulta proporciona información detallada sobre el cliente que ha comprado la mayor cantidad de productos, incluyendo el ID del cliente, nombre, apellido y la cantidad total de productos comprados.

El producto más vendido

**\_ Selección de Columnas y Funciones de Agregación:**

Tabla DETALLE\_VENTA: Selecciona el ID del producto y la cantidad vendida.

Tabla PRODUCTO: Selecciona la información adicional del producto como marca, color y precio unitario.

SUM(DV.CANTIDAD): Suma la cantidad total vendida del producto.

**\_ Declaración de Tablas y Relaciones:**

Tabla DETALLE\_VENTA: Utilizada como la tabla principal para obtener información relacionada con las ventas.

Tabla PRODUCTO: Se realiza un JOIN para obtener detalles adicionales del producto.

\_ Agrupación y Ordenamiento:

GROUP BY: Agrupa los resultados por ID del producto, marca, color y precio unitario.

ORDER BY: Ordena los resultados en orden descendente según la cantidad total vendida.

**\_ Limitación de Resultados:**

FETCH FIRST ROW ONLY: Limita el resultado a la primera fila, es decir, al producto más vendido.

**\_ Resultado:**

La consulta proporciona información detallada sobre el producto más vendido, incluyendo el ID del producto, marca, color, precio unitario y la cantidad total vendida.

El cliente que ha comprado la mayor cantidad de productos

**\_ Selección de Columnas y Funciones de Agregación:**

Tabla VENTA: Selecciona el ID del cliente.

Tabla CLIENTE: Selecciona el nombre y apellido del cliente.

COUNT(DISTINCT DV.ID\_PRODUCTO): Cuenta la cantidad total de productos distintos comprados por el cliente.

**\_ Declaración de Tablas y Relaciones:**

Tabla VENTA: Utilizada como la tabla principal para obtener información sobre las ventas.

Tabla DETALLE\_VENTA: Se realiza un JOIN para relacionar las ventas con los detalles de la venta.

Tabla CLIENTE: Se realiza un JOIN para obtener información adicional del cliente.

**\_ Agrupación y Ordenamiento:**

GROUP BY: Agrupa los resultados por el ID del cliente, nombre y apellido del cliente.

ORDER BY: Ordena los resultados en orden descendente según el total de productos comprados.

**\_ Limitación de Resultados:**

FETCH FIRST ROW ONLY: Limita el resultado a la primera fila, es decir, al cliente con más productos comprados.

**\_ Resultado:**

La consulta proporciona información detallada sobre el cliente que ha comprado la mayor cantidad de productos, incluyendo el ID del cliente, nombre, apellido y el total de productos comprados.

**IMODULOS:  
  
CONECCION:** Claro, aquí tienes un resumen de las funciones del módulo que compartiste:

Funciones:

1. `\_\_init\_\_`:\*\* Inicia la conexión con la base de datos Oracle y crea un cursor para ejecutar consultas.

2. `consulta`:\*\* Realiza una consulta simple a la base de datos y muestra los resultados en la consola.

3. `crear\_cliente`:\*\* Inserta un nuevo cliente en la base de datos, utilizando la información proporcionada. Esta función realiza tres consultas SQL: para insertar en la tabla `DATOS`, obtener el ID recién insertado y finalmente insertar en la tabla `CLIENTE`.

4. `login\_user`:\*\* Permite iniciar sesión ya sea como cliente o personal, verificando las credenciales proporcionadas en la base de datos y devolviendo la información asociada al usuario.

5. `ver\_catalogo`:\*\* Obtiene información detallada sobre los productos disponibles en el catálogo, incluyendo detalles sobre marca, color, precio, género, estilo, material, talla y cantidad en stock.

Cada función cumple un propósito específico: desde la manipulación de datos hasta la autenticación de usuarios y la visualización del catálogo de productos.

**PRINCIPAL**

FUNCIONES

1. \*\*Inicialización de la Interfaz:\*\* La clase `Raiz` tiene un método `\_\_init\_\_` que configura la ventana principal de la interfaz con elementos como título, tamaño, icono, y configuraciones básicas de la ventana.

2. \*\*Manejo de Eventos y Ventanas Secundarias:\*\* Hay métodos que controlan la apertura y cierre de ventanas secundarias (`abrir\_ventana` y `cerrar\_ventana\_secundaria`).

3. \*\*Acciones para la Creación de Clientes y Usuarios:\*\* Se incluyen funciones como `crear\_cliente` y `logear\_usuario` que parecen estar relacionadas con la creación de usuarios y clientes en una base de datos, además del login de usuarios.

4. \*\*Diseño de la Interfaz:\*\* Se utilizan imágenes y botones (`CTkButton`) para diseñar la interfaz, además de cajas de entrada (`CTkEntry`) para la introducción de datos como correo, contraseña, etc.

5. \*\*Conexión a una Base de Datos:\*\* Hay referencias a una clase `BaseDato` que parece encargarse de operaciones con una base de datos, como la creación y el acceso a usuarios y clientes.

6. \*\*Validación de Datos:\*\* Se realizan verificaciones de datos ingresados por el usuario, como campos vacíos o contraseñas diferentes.

Parece ser un código bastante extenso y detallado que abarca múltiples aspectos de la interacción con el usuario y la lógica del programa. Comentar código es una práctica excelente para facilitar su comprensión y mantenimiento en el futuro, así como para explicar el propósito de las diferentes secciones. Por ejemplo, podrías agregar comentarios explicativos sobre cada método, su propósito y las operaciones que realiza dentro de la interfaz. ¿Hay alguna parte específica del código sobre la que te gustaría más información o algún aspecto en particular que quieras comentar?

**APP**

Este código principal importa un módulo llamado `GR.principal` y utiliza la clase `Raiz` de este módulo para inicializar y mostrar la interfaz principal. Luego, verifica el estado de la variable `gr.Estado` para identificar el tipo de usuario que ha iniciado sesión: cliente, personal o administrador. Dependiendo de este estado, se crea una instancia de las clases `Cliente`, `Personal` o `Administrator` del módulo `GR.principal`. Cada instancia se configura para representar la interfaz correspondiente al tipo de usuario y se muestra en la aplicación

**Interfaz**

CustomerTkinter, Tkinter, PIL

**Estilo personalizado:**

Con Tkinter, cada elemento de la interfaz, desde los botones hasta el cuadro de texto, tiene un estilo único que refleja la identidad de Calzados Elizabeth. El color, la fuente y el diseño se unen para crear una experiencia visual coherente y atractiva.

**Interacción sin preocupaciones:**

Se uso Tkinter para una navegación fluida y natural por el catálogo. La interacción es intuitiva y simula la experiencia de elegir zapatos en una tienda física.

**Eficiencia y rendimiento:**

Las interfaces diseñadas con Tkinter no sólo son elegantes, sino también eficientes. La carga y conversión de datos es rápida y sin complicaciones, lo que proporciona una experiencia de usuario sin complicaciones.