**PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS**

## **TALLER DE BASES DE DATOS SQL JOIN**

|  |  |
| --- | --- |
| **Presentado a:** | Instructor Franco Reyna |
|  |  |
| **Por Aprendiz:** | Yuri Tatiana Ruco Morales Anderson Vladimir Mosquera Vega Cristian Camilo Lara Montero  Jorje Luis Crucerira Molina  Laura Sofia Pijimue  Melva Ceron Buitrago |
| **Ficha:** | 3064975 |
| **Competencia:** | BASES DE DATOS |

**Resultado de Aprendizaje:**

Fortalecer las competencias en programación y bases de datos

Tecnólogo en Análisis y Desarrollo de Software

Servicio Nacional de Aprendizaje SENA

Centro de Teleinformática y Producción Industrial Regional Cauca

Popayán, día 29 de mes 09 del año 2025

**PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL**

**FORMATO ENTREGA DE EVIDENCIAS**

## **Tabla de Contenido**

Contenido

[**TALLER DE BASES DE DATOS SQL JOIN** 1](#_Toc210143041)

[**Tabla de Contenido** 2](#_Toc210143042)

[**introducción** 3](#_Toc210143043)

[**Objetivos** 3](#_Toc210143044)

[General 3](#_Toc210143045)

[Específicos 3](#_Toc210143046)

[**Marco teórico** 3](#_Toc210143047)

[Go 3](#_Toc210143048)

[Lenguajes compilados vs interpretados 3](#_Toc210143049)

[Framework Fyne 4](#_Toc210143050)

[Driver MySQL en Go 4](#_Toc210143051)

[**Solución propuesta** 4](#_Toc210143052)

[Tecnologías utilizadas 4](#_Toc210143053)

[Justificación 4](#_Toc210143054)

[**Desarrollo** 4](#_Toc210143055)

[Instalación y configuración 4](#_Toc210143056)

[Diseño de la base de datos 5](#_Toc210143057)

[Funcionalidades 5](#_Toc210143058)

[Interfaz gráfica 5](#_Toc210143059)

[**Resultados** 6](#_Toc210143060)

[**Conclusiones** 7](#_Toc210143061)

[**Bibliografía** 7](#_Toc210143062)

## **introducción**

En el marco del ejercicio planteado, se buscó desarrollar una aplicación de escritorio que permitiera realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre una base de datos MySQL. Para ello se eligió el lenguaje **Go**, debido a sus características de rendimiento, seguridad y simplicidad.

Asimismo, se utilizó el framework **Fyne** para implementar una interfaz gráfica amigable y multiplataforma, logrando una experiencia de usuario más dinámica que en una aplicación de consola.

## **Objetivos**

### General

Desarrollar una aplicación de escritorio en Go que permita gestionar productos mediante operaciones CRUD conectadas a una base de datos MySQL, utilizando el framework Fyne para la interfaz gráfica.

### Específicos

* Instalar y configurar el entorno de desarrollo con Go, MySQL y las librerías necesarias.
* Implementar la conexión a la base de datos desde Go utilizando el driver correspondiente.
* Diseñar una interfaz gráfica de usuario intuitiva con Fyne.
* Desarrollar las funcionalidades CRUD (crear, listar, actualizar y eliminar registros).
* Validar el correcto funcionamiento de la aplicación a través de pruebas y ejemplos.

## **Marco teórico**

### Go

Go es un lenguaje de programación creado por Google en 2007. A diferencia de lenguajes interpretados, Go es **compilado**, lo que le otorga velocidad, eficiencia y robustez. Es ampliamente utilizado en aplicaciones de backend, sistemas distribuidos y, con ayuda de librerías, en aplicaciones de escritorio.

### Lenguajes compilados vs interpretados

* **Compilados**: generan un ejecutable que se ejecuta directamente en el sistema (ej. Go, C, C++).
* **Interpretados**: requieren un intérprete en tiempo de ejecución (ej. Python, JavaScript).

### Framework Fyne

Fyne es un framework multiplataforma para construir interfaces gráficas en Go. Permite crear aplicaciones de escritorio modernas y ligeras, con componentes como botones, entradas de texto, tablas y cuadros de diálogo.

### Driver MySQL en Go

Go incluye un driver genérico para bases de datos, pero para conectarse a MySQL es necesario instalar el paquete:

go get github.com/go-sql-driver/mysql

Este driver facilita ejecutar consultas SQL y manejar resultados directamente desde Go.

## **Solución propuesta**

Se decidió implementar una aplicación de escritorio con interfaz gráfica en lugar de consola, con el fin de ofrecer una experiencia más intuitiva.

### Tecnologías utilizadas

* **Lenguaje**: Go (v1.20 o superior)
* **Base de datos**: MySQL
* **Framework gráfico**: Fyne (fyne.io/fyne/v2)
* **Driver MySQL**: github.com/go-sql-driver/mysql
* **Compilador**: GCC instalado con **MSYS2**

### Justificación

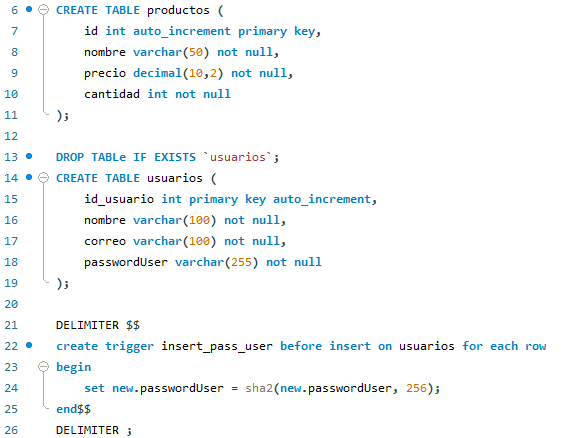
La combinación Go + Fyne + MySQL permite un desarrollo ágil, con buena integración entre backend y frontend en un único lenguaje, y sin necesidad de dependencias externas pesadas.

## **Desarrollo**

### Instalación y configuración

1. Instalación de Go desde <https://go.dev/doc/install>.
2. Instalación de MySQL.
3. Instalación de Fyne:
4. go get fyne.io/fyne/v2
5. Instalación del driver de MySQL:
6. go get github.com/go-sql-driver/mysql
7. Instalación de GCC mediante MSYS2 desde <https://www.msys2.org>.

### Diseño de la base de datos



### Funcionalidades

* **Agregar producto**: permite insertar registros en la base de datos.
* **Listar productos**: muestra todos los productos en una tabla con scroll.
* **Actualizar productos**: permite editar directamente las celdas de la tabla y confirmar los cambios antes de guardarlos en la BD.
* **Eliminar producto**: cada fila tiene un icono 🗑 para eliminar con confirmación.

### Interfaz gráfica

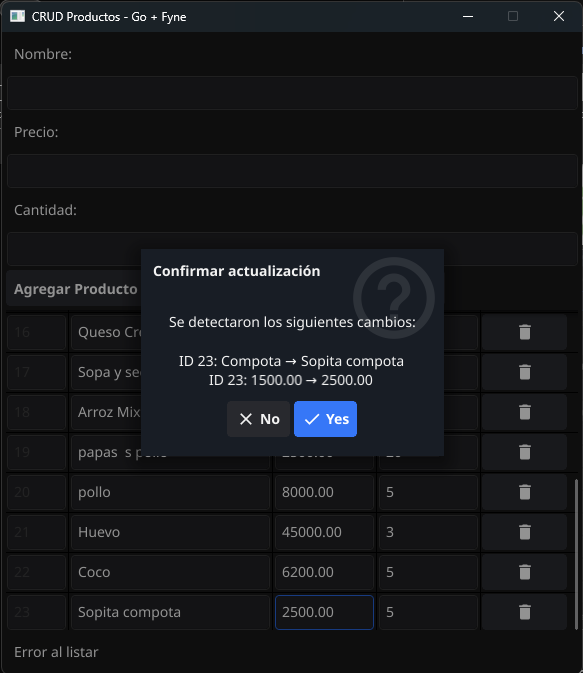
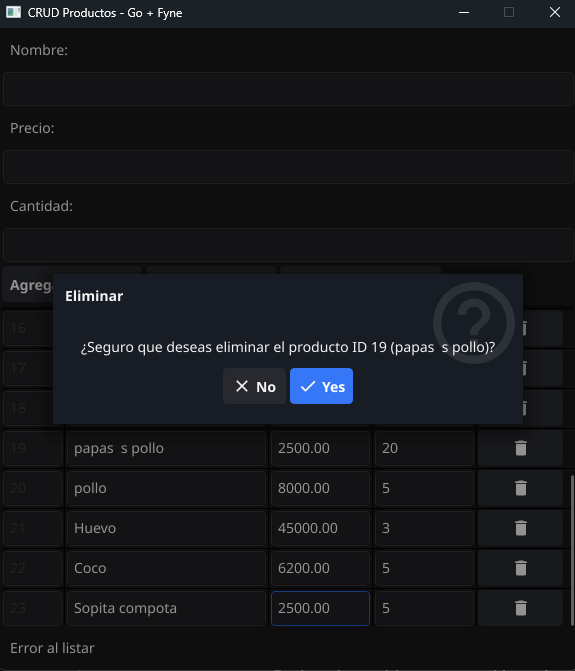
La ventana principal contiene:

* Entradas de texto para nombre, precio y cantidad.
* Botones: *Agregar*, *Listar*, *Actualizar*.
* Tabla con columnas editables y una columna de eliminar.

*A screenshot of a computer

AI-generated content may be incorrect.* A screenshot of a computer

AI-generated content may be incorrect.

## **Resultados**

* Se logró establecer conexión exitosa con MySQL desde Go.
* Se implementaron todas las operaciones CRUD.
* La tabla en Fyne permite edición directa de celdas, simulando una hoja de cálculo.
* El uso de íconos de eliminar en cada fila mejoró la experiencia de usuario.

## **Conclusiones**

* Go es un lenguaje versátil que, junto con Fyne, permite crear aplicaciones de escritorio robustas.
* El uso de drivers específicos simplifica la conexión con bases de datos sin requerir conectores adicionales.
* Fyne, aunque ligero, ofrece los componentes suficientes para construir CRUDs funcionales.
* La aplicación desarrollada demuestra que Go puede ser una alternativa interesante para proyectos de escritorio integrados con bases de datos.

## **Bibliografía**

* Documentación oficial de Go:  
  <https://go.dev/doc>
* Driver MySQL para Go:  
  <https://github.com/go-sql-driver/mysql>
* Framework Fyne:  
  <https://fyne.io/>
* MSYS2:  
  <https://www.msys2.org/>
* OpenIA-ChatGPT:   
  [chatgpt.com/](https://chatgpt.com/)