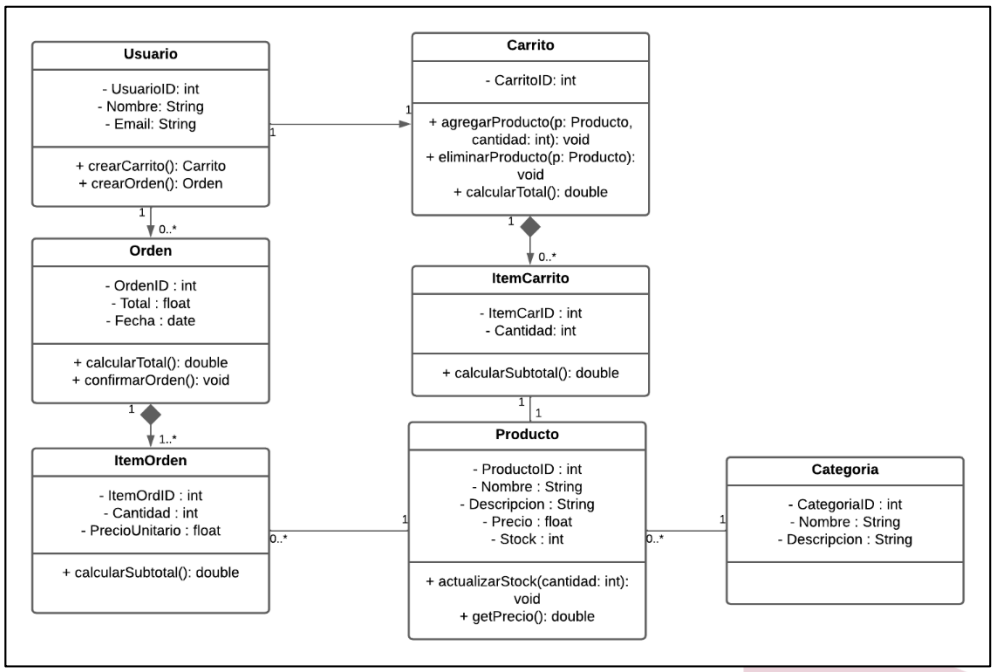
Nombre del estudiante: Arley Ivan Salgado Mañay
Asignatura: Programación orientada a objetos
Fecha: 15/02/2026
Título: Aprendizaje Autónomo 2

1 Clases

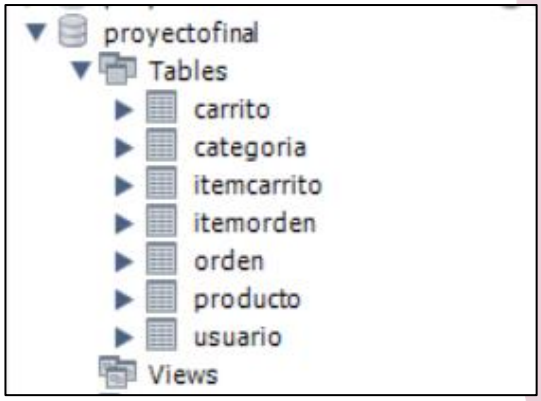
Para el desarrollo del sistema de E-commerce se creó el siguiente diagrama de clases. Este diagrama fue ingresado dentro de una base de datos en MySQL y desarrollado dentro del lenguaje de programación go.



https://lucid.app/lucidchart/5d019943-6f10-4c1e-8159-522356dba685/edit?view_items=Ep2oI2pBndjw&page=0_0&invitationId=inv_7e610642-4533-4268-a4d0-cf98333363a9

1.1 Base de datos

Dentro de la aplicación MySQL Workbench se crea la base de datos ***proyectofinal*** en el que se crean las tablas que pertenecen a las clases creadas en el diagrama anterior.

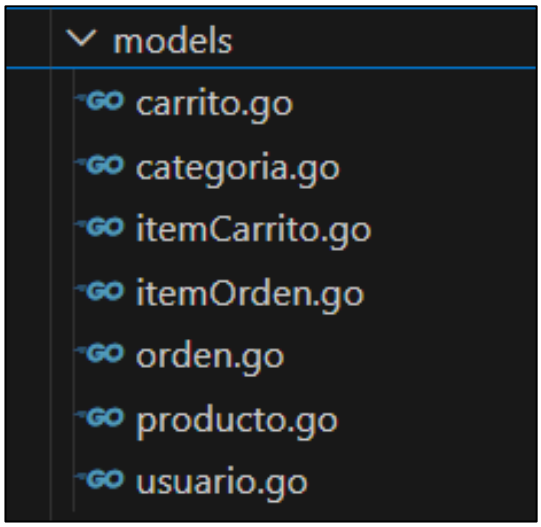


Cada tabla mantiene los atributos de cada clase

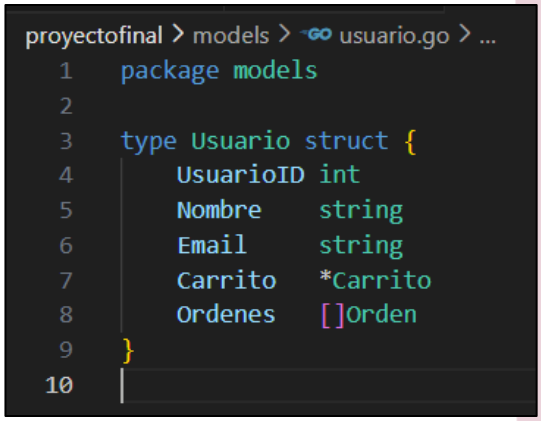
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
UsuarioID	INT	☑	☑	☐	☐	☐	☐	☐	☐	
Nombre	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
Email	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

1.2 GO

Para pode desarrollar el sistema dentro de Go se crea un archivo *.go* para cada clase, los cuales van a estar dentro de una carpeta *models*.



Dentro de cada archivo se colocan los atributos que tienen cada clase



2 Implementación de prerrequisitos

Para el desarrollo del sistema primero se va a conectar la base de datos MySQL por medio de GO.

2.1 Creación archivo .env

Para que el código pueda conectarse con la base se debe tener los datos de esta. Se crea un archivo con el nombre de *.env* en donde se va a tener el nombre de la base, el usuario, contraseña, host y puerto.

```
.env
proyectofinal > .env
1   DB_NAME = proyectofinal
2   DB_USER = root
3   DB_PASSWORD = 12345
4   DB_HOST = localhost
5   DB_PORT =3306
```

2.2 Instalar paquetes

Dentro del directorio en el que se encuentra el sistema se debe crear un archivo en el que se van a encontrar la información del proyecto, así como sus dependencias. Esto se logra colocando *go mod init nombre* en la línea de comandos y va a crear un archivo con el nombre de *go.mod*.

```
\proyectofinal> go mod init proyectofinal
```

Para poder lograr la interconexión entre la base de datos y GO se debe hacer uso de 2 paquetes dentro del directorio en el que se encuentra el sistema. El primer paquete a instalar es ***Go-MySQL-Driver*** el cual proporciona un controlador para que se pueda usar SQL en GO (*mysql package - github.com/go-sql-driver/mysql - Go Packages*, s. f.). El segundo paquete es ***GoDotEnv*** es el que permitela carga del archivo creado anteriormente con el nombre de *.env* (*godotenv package - github.com/joho/godotenv - Go Packages*, s. f.)

```
\proyectofinal> go get -u github.com/go-sql-driver

\proyectofinal> go get github.com/joho/godotenv
```

Una vez que se ejecutan ambos comandos dentro del archivo *go.mod* se va a colocar automáticamente cuales son los paquetes que se están utilizando, así como su versión.

```
go.mod
proyectofinal > go.mod
Reset go.mod diagnostics | Run govulncheck | Run go mod tidy | Create vendor directory
1   module proyectofinal
2
3   go 1.25.5
4
Check for upgrades | Upgrade transitive dependencies | Upgrade direct dependencies
5   require (
6       filippo.io/edwards25519 v1.1.0 // indirect
7       github.com/go-sql-driver/mysql v1.9.3 // indirect
8       github.com/joho/godotenv v1.5.1 // indirect
9   )
10
```

Al mismo tiempo se crea un archivo con el nombre *go.sum* en el que se van a encontrar las descargas que se realizaron.

go.sum

proyectofinal > go.sum

```
filippo.io/edwards25519 v1.1.0 h1:FNf4tywRC1HmFuKW5xopWpigGjJKiJSV0Cqo0cJWDaA=
filippo.io/edwards25519 v1.1.0/go.mod h1:BxyFTGdWcka3PhytdK4V28tE5sGfRvvvRV7EaN4VDT4=
github.com/go-sql-driver/mysql v1.9.3 h1:U/N249h2WzJ3Ukj8SowVFjdtZKfu9vlLZxjPXV1aweo=
github.com/go-sql-driver/mysql v1.9.3/go.mod h1:qn46aNg1333BRMNU69Lq93t8du/dwxI64Gl8i5p1WMU=
github.com/joho/godotenv v1.5.1 h1:7eLL/+HRGLY0ldzfGMeQkb7vMd0as4CfYvUVzLqw0N0=
github.com/joho/godotenv v1.5.1/go.mod h1:f4LDr5Voq0i2e/R5DDNOoa2zzDfwtkZa6DnEwAbqwq4=
```

3 Implementación de Software

Una vez que ya se tienen los requisitos para que pueda funcionar el código se crea un archivo con el nombre *connect.go* dentro de una carpeta *db*. Dentro de este archivo se va a tener la conexión del código con la base de datos.

connect.go

proyectofinal > db > connect.go > Connect

```go
package db

import (
    "database/sql"
    "fmt"
    "os"

    _ "github.com/go-sql-driver/mysql"
    "github.com/joho/godotenv"
)

// Connect crea y valida la conexión a MySQL
func Connect() (*sql.DB, error) {

    // Cargar archivo .env
    if err := godotenv.Load(); err != nil {
        return nil, fmt.Errorf("error cargando archivo .env: %w", err)
    }

    // Obtener variables de entorno
    user := os.Getenv("DB_USER")
    password := os.Getenv("DB_PASSWORD")
    host := os.Getenv("DB_HOST")
    port := os.Getenv("DB_PORT")
    name := os.Getenv("DB_NAME")

```

connect.go

proyectofinal > db > connect.go > Connect

```
49  }
26
27      // Validar que no estén vacías
28      if user == "" || password == "" || host == "" || port == "" || name == "" {
29          return nil, fmt.Errorf("Faltan variables de entorno para la base de datos")
30      }
31
32      // Crear DSN
33      dsn := fmt.Sprintf("%v:%v@tcp(%v:%v)/%v",
34          user, password, host, port, name,
35      )
36
37      // Abrir conexión
38      db, err := sql.Open("mysql", dsn)
39      if err != nil {
40          return nil, err
41      }
42
43      // Verificar conexión real
44      if err := db.Ping(); err != nil {
45          return nil, err
46      }
47
48      return db, nil
49  }
```

Para que se lleve a cabo la conexión se crea un archivo *main.go* que es en donde se va atener el funcionamiento del código

main.go

proyectofinal > main.go > main

```
1   package main
2
3   import (
4       "log"
5       "net/http"
6
7       "proyectofinal/db"
8   )
9
10  func main() {
11
12      // Conectar a la base de datos
13      database, err := db.Connect()
14      if err != nil {
15          log.Fatal("Error al conectar a la base de datos: ", err)
16      }
17      defer database.Close()
18
19      log.Println("Conexión exitosa a la base de datos")
20
21
22      // Iniciar servidor
23      if err := http.ListenAndServe(":8000", nil); err != nil {
24          log.Fatal("Error al iniciar el servidor: ", err)
25      }
26  }
```

4 GitHub

En el siguiente enlace se tiene acceso hacia GitHub en donde se encuentran los avances del sistema en desarrollo. El código en desarrollo se encuentra dentro de la carpeta *proyectofinal*

https://github.com/arsalgadoma/ProgramacionOrientadaAObtejos.git

5 Video explicativo y de funcionalidad

https://mailinternacionaledu-my.sharepoint.com/:v:/g/personal/arsalgadoma_uide_edu_ec/IQCYDRG3-jsCR6_ByCPWBfATAY1Z7Le3wA8C8Zwf3yvWbMY?e=L9wcyC

6 Referencias

Godotenv package—Github.com/joho/godotenv—Go Packages. (s. f.). Recuperado 14 de febrero de 2026, de https://pkg.go.dev/github.com/joho/godotenv#section-readme

Mysql package—Github.com/go-sql-driver/mysql—Go Packages. (s. f.). Recuperado 14 de febrero de 2026, de https://pkg.go.dev/github.com/go-sql-driver/mysql