



# MI ECOMMERCE

GRUPO 7

#CENCOTECH





# ¿QUIÉNES SOMOS?

01



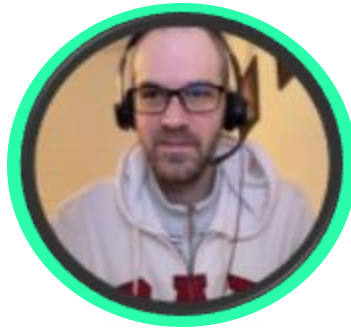
# NUESTRO EQUIPO



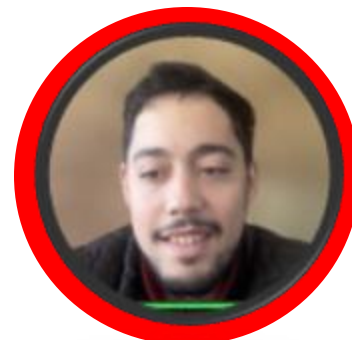
Diego  
Perez



Ballester  
Bermudez



Federico  
Sierra



Hernan  
Gutierrez



Juan  
Ventura





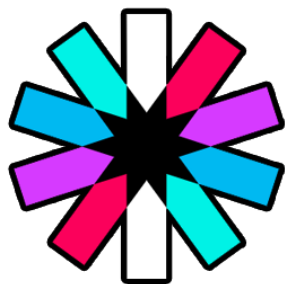
# HERRAMIENTAS UTILIZADAS

02





Sequelize



DB DESIGNER

- JAVASCRIPT
- NPM
- GIT
- JWT
- VSCODE
- NODEJS
- DBDESIGNER
- MYSQL
- SEQUELIZE
- EXPRESS
- SWAGGER
- POSTMAN
- PHPMYADMIN
- CLICKUP





# MEJORAS DE LA NUEVA VERSION

03



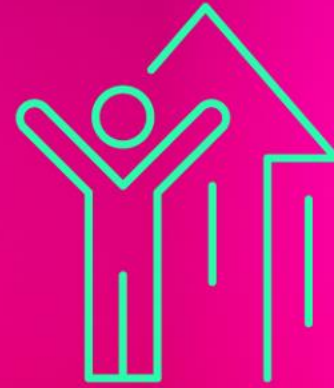


# PERSISTENCIA DE DATOS

03



# SQL Y MYSQL



## SQL es un lenguaje de bases de datos, mientras MySQL es un sistema de gestión de bases de datos basado en SQL

- SEGURO

Tiene un sistema de autorización flexible que nos permite otorgar privilegios a diferentes usuarios.

- RAPIDO

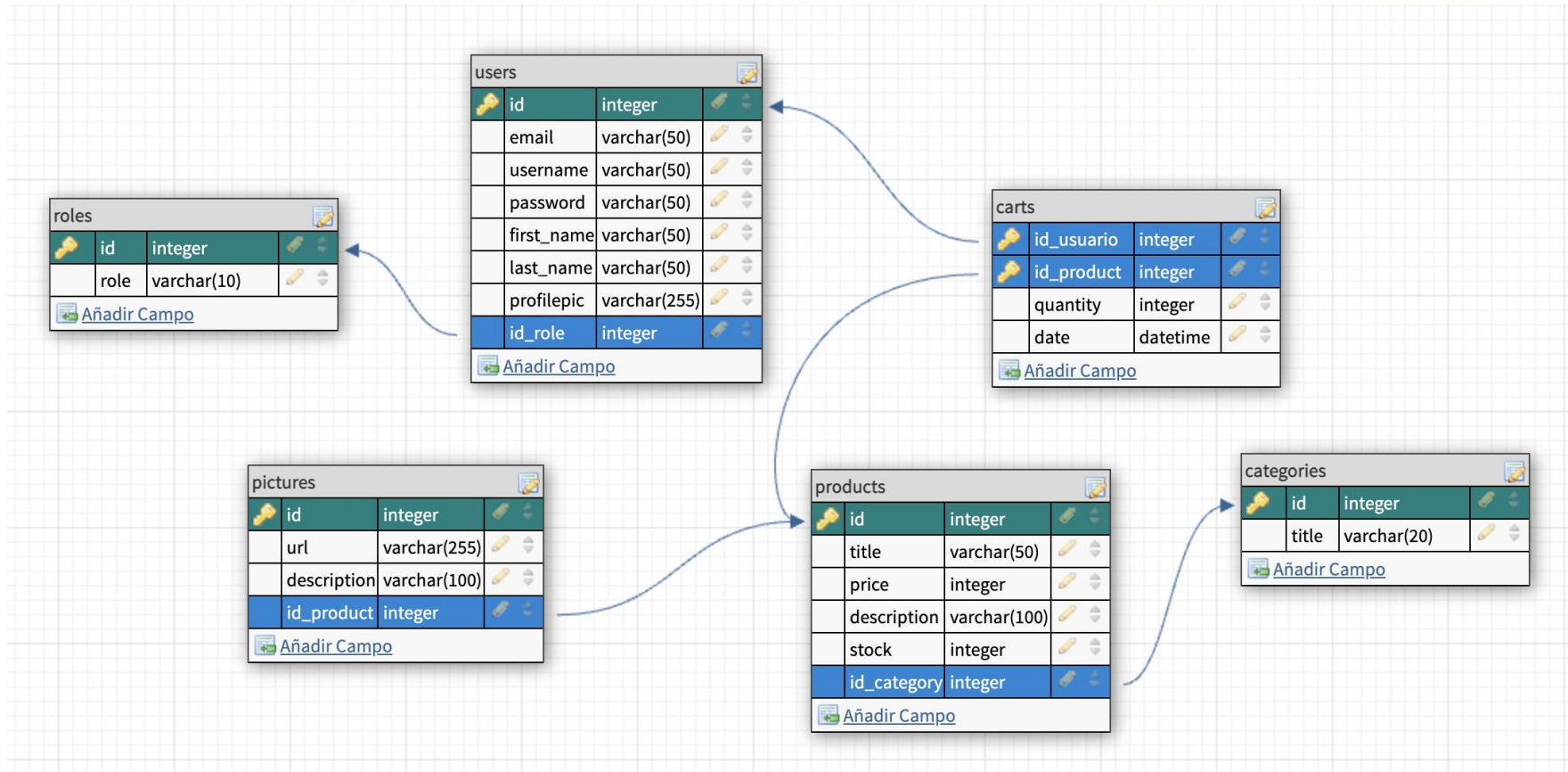
Fue un software diseñado desde un principio pensando en la rapidez

- MULTIPLATAFORMA

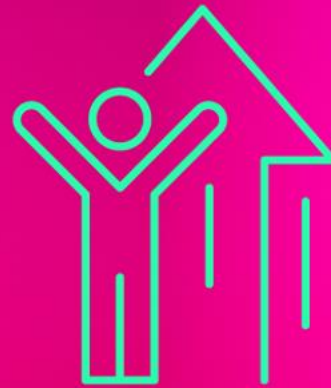
Es un software soportado por una amplia cantidad de sistemas operativos



# DISEÑO BASE DE DATOS



# SEQUELIZE

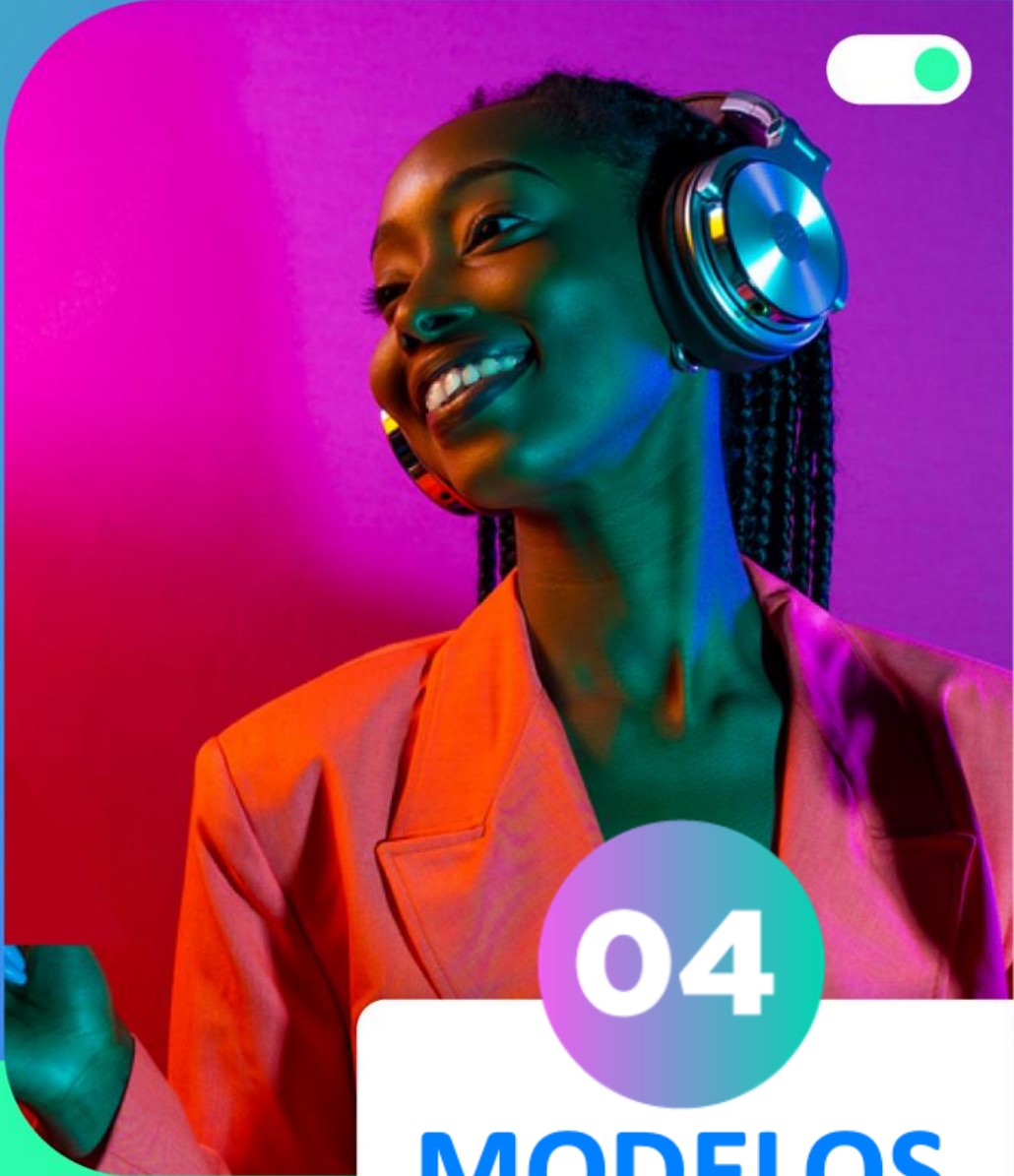


- QUE ES SEQUELIZE?

Es un ORM basado en promesas para Node.js que soporta bases de datos relacionales como MySQL.

- PARA QUE SIRVE UN ORM?

Sirve para "transformar" las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas para manipular los datos.



models

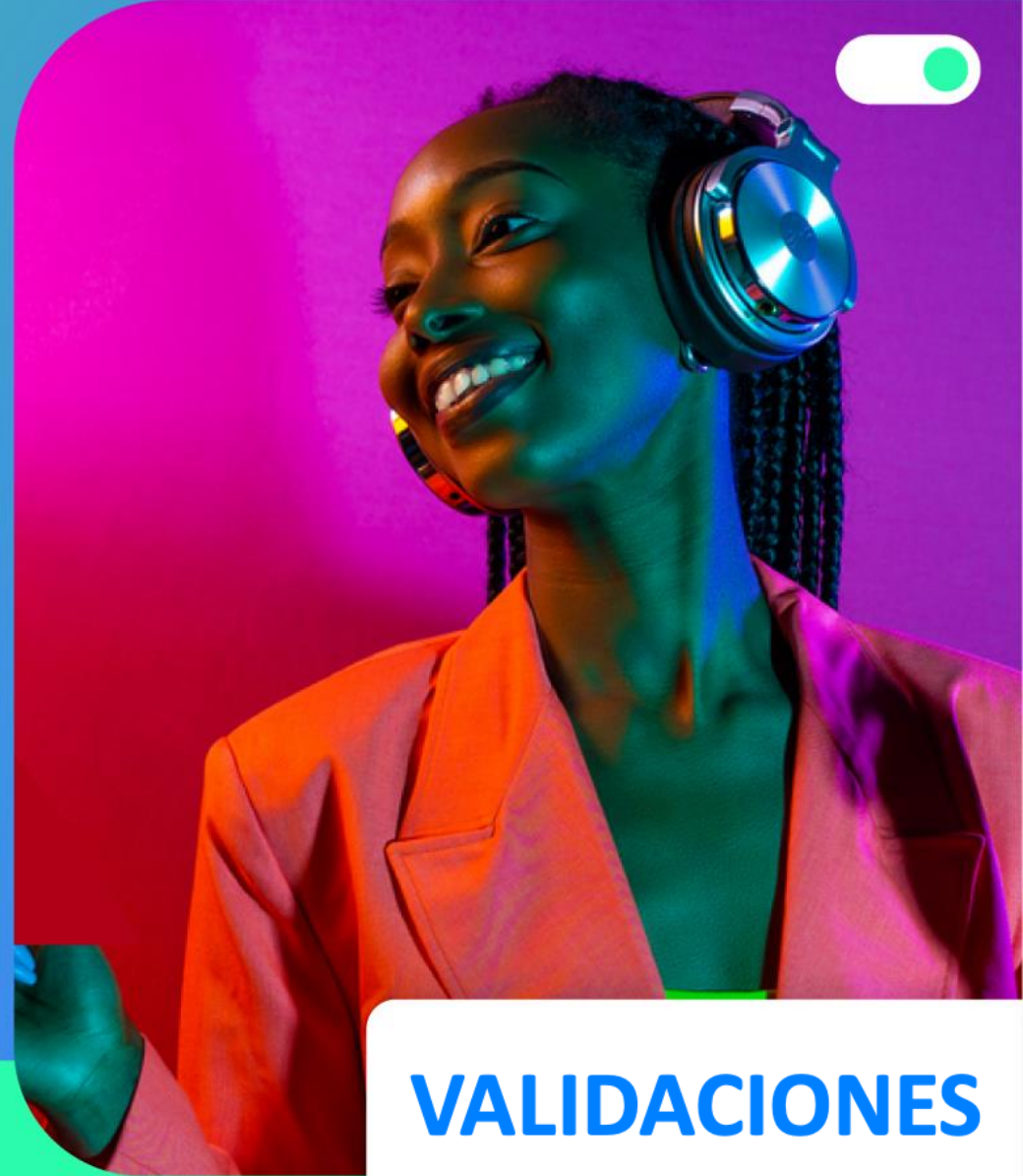
- JS Cart.js
- JS Category.js
- JS index.js
- JS Picture.js
- JS Product.js
- JS Role.js
- JS User.js

04

**MODELOS**



```
1  stock: {  
2    type: DataTypes.INTEGER,  
3    allowNull: false,  
4    defaultValue: 0,  
5    validate: {  
6      isInt :{  
7        args: true,  
8        msg: "Stock debe ser un entero"  
9      },  
10   min: {  
11     args: [0],  
12     msg: "Stock no puede ser negativo"  
13   }  
14 }  
15 },
```



## VALIDACIONES Y CONSTRAINTS



```
modifyCart: async (req, res) => {
  const t = await sequelize.transaction();
  try {
    const user = await persistence.getCartByUserID(req.params.id);
    const cart = user.cart;
    //PRIMER FOR:
    //Destruir carrito antiguo.
    for (let i = 0; i < cart.length; i++) { ...
    }
    //SEGUNDO FOR:
    //Crear nuevo carrito.
    const nuevoCart = req.body;
    for (let i = 0; i < nuevoCart.length; i++) { ...
    }
    const cartById = await persistence.getCartByUserID(req.params.id);
    await t.commit();
    res.status(200).json({ ok: true, newCart: cartById });
  } catch (error) {
    t.rollback();
    res.status(500).json({ ...
    })
  }
},
```



**TRANSACCIONES**





# METODOLOGIA DE TRABAJO

05





### **DESARROLLO**

Ayuda a la administracion de sprints de desarrollo



### **ORGANIZACION**

Orquestador de flujos de órdenes de compra.  
Centralizador de información y estados de las órdenes.



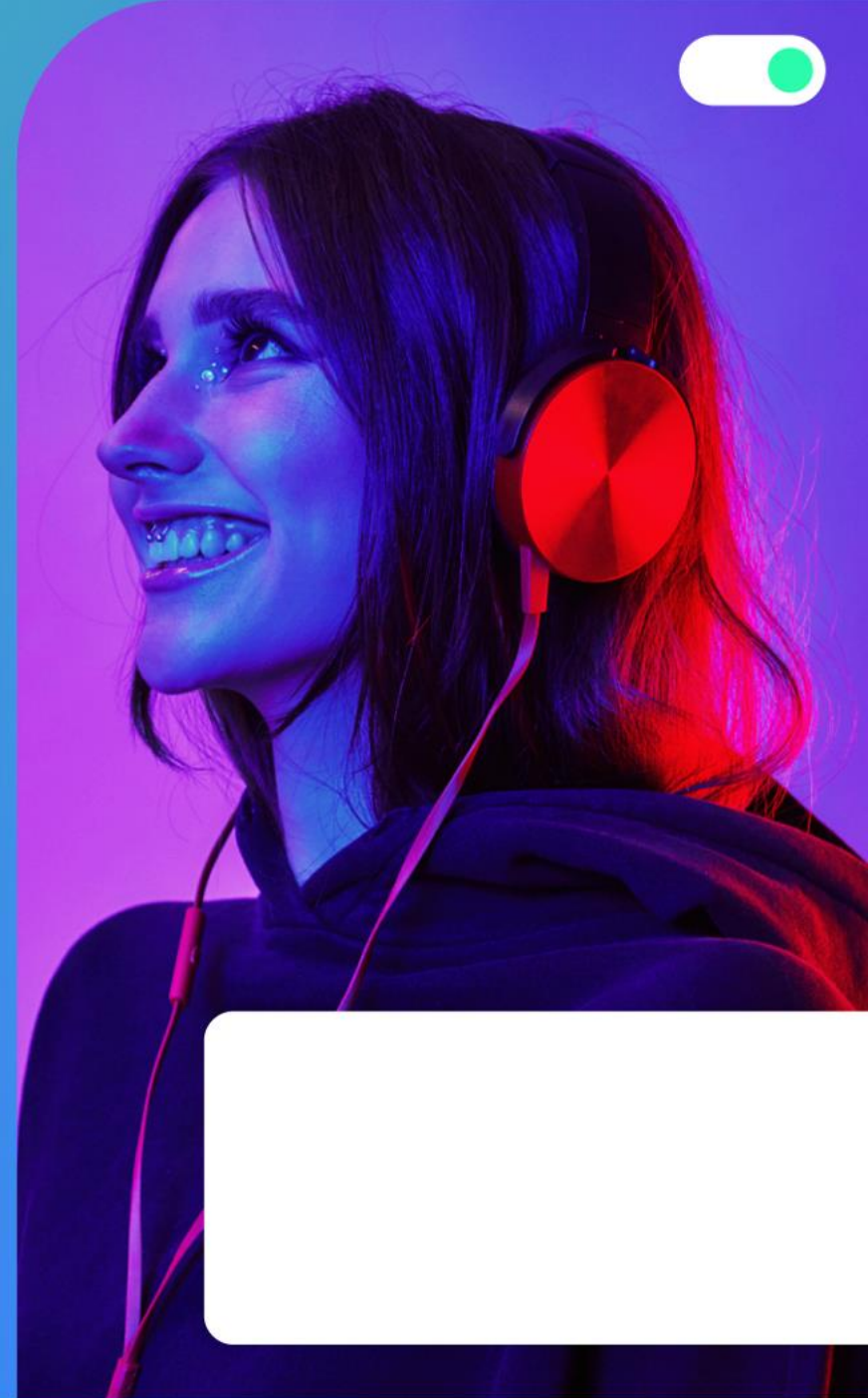
### **COMUNICACION**

Nos ofrece un canal de comunicación y nos permite ver lo que realiza el resto del equipo



#SOMOSCENCOTECH

ANY  
QUESTIONS?







# DEMO

06

