

```
40 CREATE TABLE TIEN
41     codigo_tienda INT
42     codigo_producto INT
43     existencia INT NO
44     PRIMARY KEY(codig
45     FOREIGN KEY(codig
46 );
```

```
48 CREATE TABLE USER
49 _Code INT NOT NUL
```

```
<link rel="stylesheet" href="/css/Home.css"/>  
<title>Login</title>  
</head>  
<body>  
    <div class="login">  
        <div class="container">  
            <div class="logo">  
                  
            </div>  
            <div class="container-login">  
                <h1 class="title"> Login to you account</h1>  
                <div class="container-form">  
                    <form method="POST" action="ServletLogin" class="form">  
                        <label for="text" class="label">User Name</label>  
                        <input type="text" class="input" name="User_name" placeholder="Gerardo7897" required>  
                        <label for="password" class="label">Password:</label>  
                        <input type="password" name="password" placeholder="*****" class="input input-password" id="password" required>  
                    </form>  
                </div>  
            </div>  
        </div>  
    </body>
```

Luis Gerardo Marcelino Tax Mantanico 201930539

201930539

DIAGRAMA DE CLASE

Un diagrama de clases es una herramienta de modelado que se utiliza para representar las clases y sus relaciones, este tipo de diagrama es muy útil para visualizar la estructura de una clase, en este proyecto se utilizaron diferentes tipos de clase cada clase contaba con sus atributos que eran de tipo `private` y así mismo contaba con sus propios `get` y `set`, otras clases que más se utilizaron fueron las de `servlet` ya que se utiliza para procesar solicitudes y respuestas HTTP también otro de los beneficios es poder acceder a la base de datos y realizar las diferentes consultas.

DIAGRAMA E/R

El diagrama entidad-relación es una herramienta esencial para el diseño de bases de datos relacionales y el desarrollo de sistemas de información. Ayuda a identificar entidades, representar relaciones, definir atributos, facilitar la comunicación y mejorar la calidad de los datos almacenados en la base de datos.

DIAGRAMA DE TABLA

Un diagrama de tabla es una forma de visualizar la estructura de una base de datos en forma de tablas, cada una de ellas consiste en filas (también llamadas "tuplas") y columnas, como una hoja de cálculo. Por último, se tiene que definir qué atributo o atributos funcionarán como clave primaria para cada tabla. Una clave primaria (PK) es un identificador único para una entidad determinada.

MAPEO FÍSICO DE LA DB

En resumen, el mapeo físico es el proceso de asignar los objetos de la base de datos a los dispositivos físicos de almacenamiento. El objetivo es optimizar el rendimiento y la eficiencia del almacenamiento de datos, asegurándose de que los datos se almacenen y accedan de manera efectiva.

DIAGRAMA DE CLASE

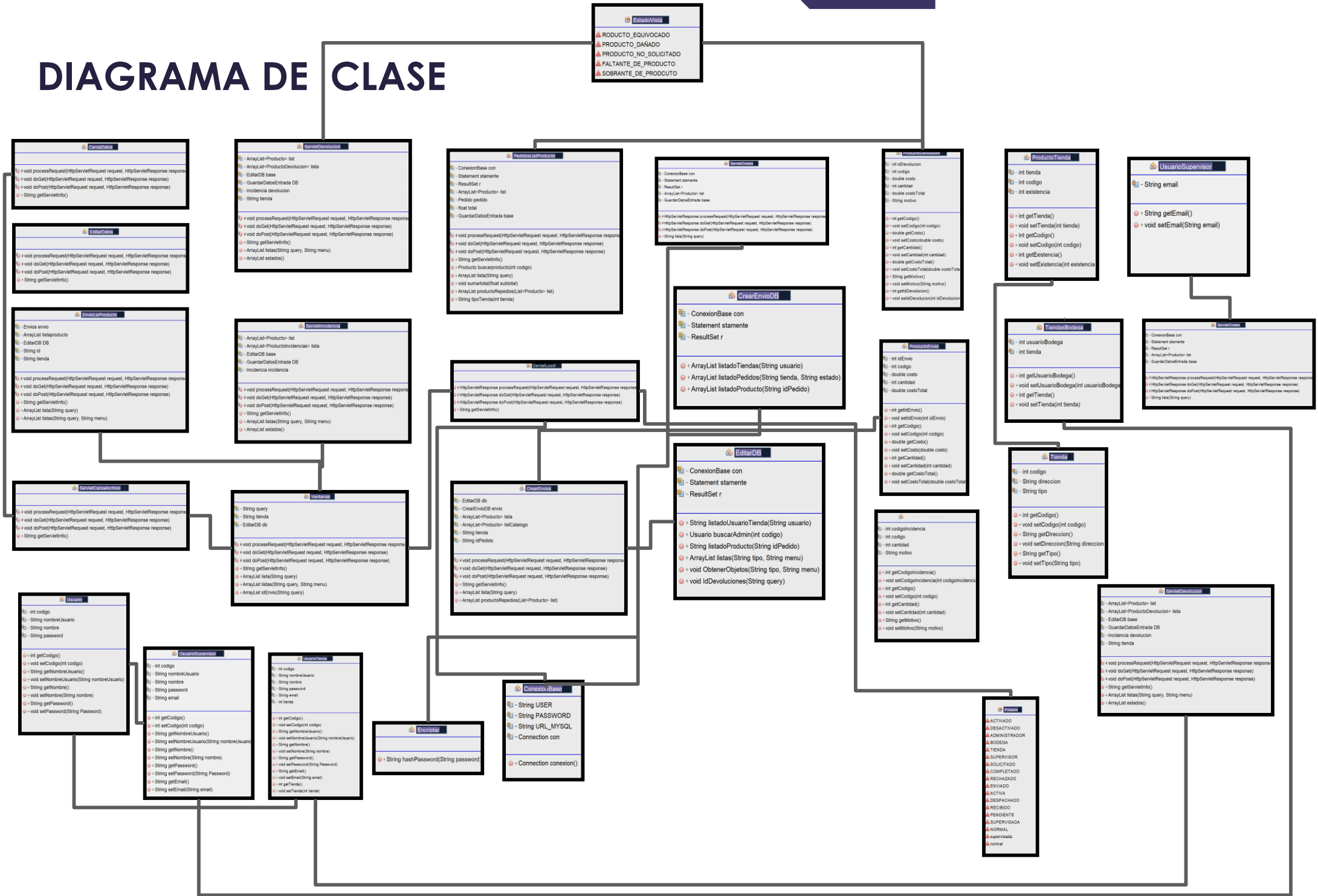
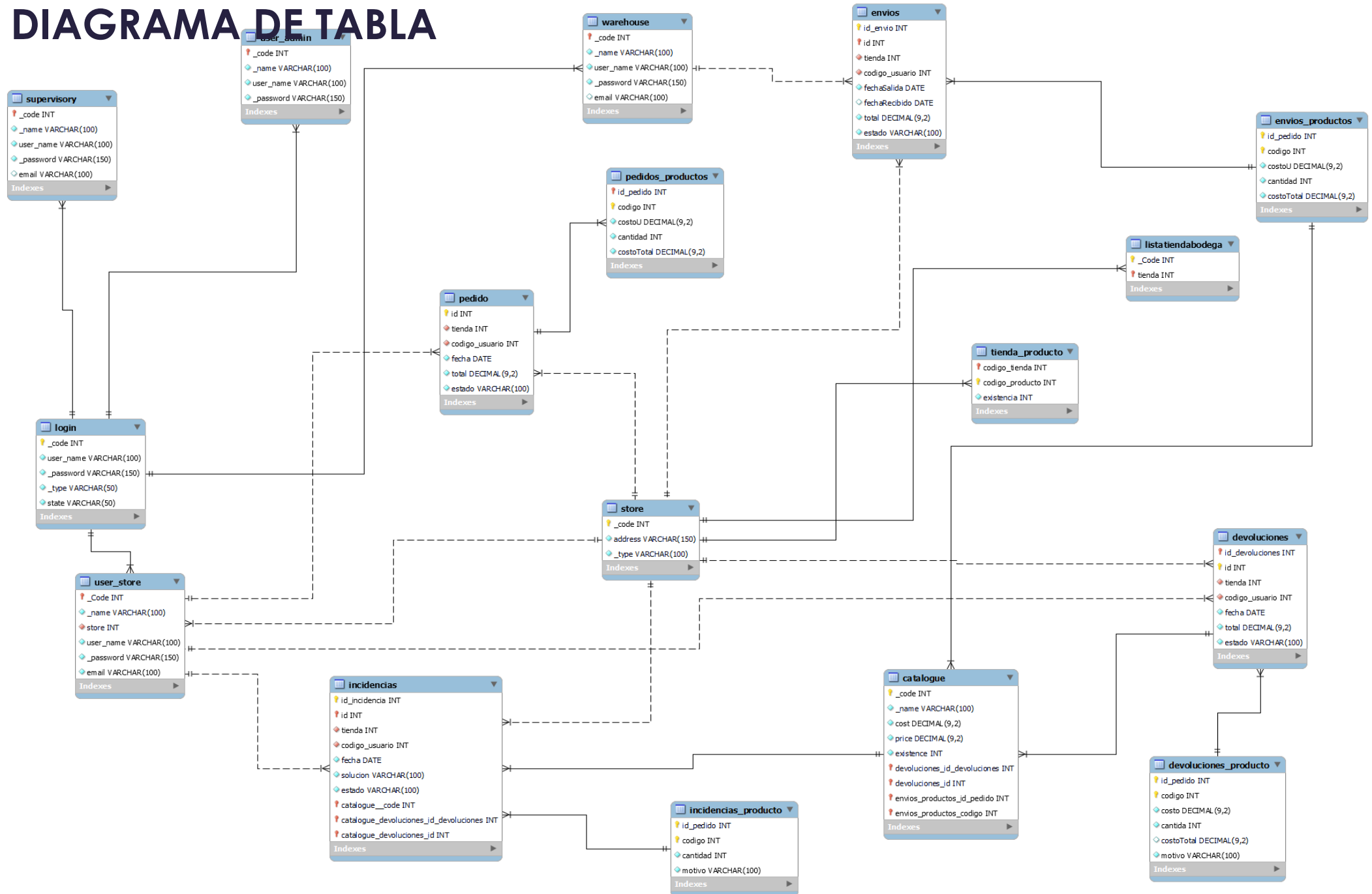


DIAGRAMA ER

DIAGRAMA DE TABLA



MAPEO FISICO

```
CREATE DATABASE data_tienda;
```

```
USE data_tienda;
```

```
CREATE TABLE LOGIN (  
  _code INT NOT NULL AUTO_INCREMENT,  
  user_name VARCHAR(100) NOT NULL UNIQUE,  
  _password VARCHAR(150) NOT NULL,  
  _type VARCHAR(50) NOT NULL,  
  state varchar(50) NOT NULL,  
  PRIMARY KEY (_code)  
);
```

```
CREATE TABLE USER_ADMIN(  
  _code INT NOT NULL,  
  _name VARCHAR(100) NOT NULL,  
  user_name VARCHAR(100) NOT NULL UNIQUE,  
  _password VARCHAR(150) NOT NULL,  
  PRIMARY KEY(_code),
```

```
FOREIGN KEY (_code) REFERENCES LOGIN(_code) ON UPDATE CASCADE
```

```
);
```

```
Insert into login value(1,'admin','8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918','ADMINISTRADOR','ACTIVADO');
```

```
insert into user_admin value(1,'admin','admin','8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918');
```

```
CREATE TABLE CATALOGUE(
```

```
_code INT NOT NULL AUTO_INCREMENT,
```

```
_name VARCHAR(100) NOT NULL,
```

```
cost DECIMAL(9,2) NOT NULL,
```

```
price DECIMAL(9,2) NOT NULL,
```

```
existence INT NOT NULL,
```

```
PRIMARY KEY (_code)
```

```
);
```

```
CREATE TABLE STORE(
```

```
_code INT NOT NULL AUTO_INCREMENT,
```

```
address VARCHAR(150) NOT NULL,
```

```
_type VARCHAR(100) NOT NULL,
```

```
PRIMARY KEY (_code)
```

```
);
```

```
CREATE TABLE TIENDA_PRODUCTO(  
codigo_tienda INT NOT NULL,  
codigo_producto INT NOT NULL,  
existencia INT NOT NULL,  
PRIMARY KEY(codigo_tienda, codigo_producto),  
FOREIGN KEY(codigo_tienda) REFERENCES STORE (_code) ON UPDATE CASCADE  
);
```

```
CREATE TABLE USER_STORE(  
_Code INT NOT NULL,  
_name VARCHAR(100) NOT NULL,  
store INT NOT NULL,  
user_name VARCHAR(100) UNIQUE NOT NULL,  
_password VARCHAR(150) NOT NULL,  
email VARCHAR(100) NOT NULL,  
PRIMARY KEY(_code),  
FOREIGN KEY(_code) REFERENCES LOGIN(_code) ON UPDATE CASCADE,  
FOREIGN KEY(store) REFERENCES STORE(_code)ON UPDATE CASCADE  
);
```

```
CREATE TABLE SUPERVISORY(  

```



```
_code INT NOT NULL,  
_name VARCHAR(100) NOT NULL,  
user_name VARCHAR(100) UNIQUE NOT NULL,  
_password VARCHAR(150) NOT NULL,  
email VARCHAR (100),  
PRIMARY KEY(_code),  
FOREIGN KEY(_code) REFERENCES LOGIN(_code) ON UPDATE CASCADE  
);
```

```
CREATE TABLE WAREHOUSE (  
_code INT NOT NULL,  
_name VARCHAR(100) NOT NULL,  
user_name VARCHAR(100) UNIQUE NOT NULL,  
_password VARCHAR(150) NOT NULL,  
email VARCHAR (100),  
PRIMARY KEY(_code),  
FOREIGN KEY(_code) REFERENCES LOGIN(_code) ON UPDATE CASCADE  
);
```

```
CREATE TABLE listaTiendaBodega(  
_Code INT NOT NULL,  
tienda INT NOT NULL,
```

```
primary key(_code, tienda),  
FOREIGN KEY(tienda) REFERENCES STORE(_code) ON UPDATE CASCADE  
);
```

```
CREATE TABLE PEDIDO(  
  id INT NOT NULL AUTO_INCREMENT,  
  tienda INT NOT NULL,  
  codigo_usuario INT NOT NULL,  
  fecha DATE NOT NULL,  
  total Decimal(9,2) NOT NULL,  
  estado VARCHAR(100) NOT NULL,  
  PRIMARY KEY( id),  
  FOREIGN KEY(codigo_usuario) REFERENCES USER_STORE (_code) ON UPDATE CASCADE,  
  FOREIGN KEY(tienda) REFERENCES STORE(_code) ON UPDATE CASCADE  
  
);
```

```
CREATE TABLE PEDIDOS_PRODUCTOS(  
  id_pedido INT NOT NULL,  
  codigo INT NOT NULL,  
  costoU DECIMAL(9,2) NOT NULL,
```

```
cantidad INT NOT NULL,  
costoTotal DECIMAL(9,2) NOT NULL,  
PRIMARY KEY(id_pedido, codigo),  
FOREIGN KEY(id_pedido) REFERENCES PEDIDO (id) ON UPDATE CASCADE  
);
```

```
CREATE TABLE ENVIOS(  
  id_envio INT NOT NULL AUTO_INCREMENT,  
  id INT NOT NULL,  
  tienda INT NOT NULL,  
  codigo_usuario INT NOT NULL,  
  fechaSalida DATE NOT NULL,  
  fechaRecibido DATE,  
  total DECIMAL(9,2) NOT NULL,  
  estado VARCHAR(100) NOT NULL,  
  PRIMARY KEY(id_envio,id),  
  FOREIGN KEY(codigo_usuario) REFERENCES WAREHOUSE (_code) ON UPDATE CASCADE,  
  FOREIGN KEY(tienda) REFERENCES STORE(_code) ON UPDATE CASCADE  
);
```

```
CREATE TABLE ENVIOS_PRODUCTOS(  
  id_envio INT NOT NULL,  
  id_producto INT NOT NULL,  
  cantidad INT NOT NULL,  
  PRIMARY KEY(id_envio, id_producto),  
  FOREIGN KEY(id_envio) REFERENCES ENVIOS (id_envio) ON UPDATE CASCADE,  
  FOREIGN KEY(id_producto) REFERENCES PRODUCTOS (id_producto) ON UPDATE CASCADE  
);
```

```
id_pedido INT NOT NULL,  
codigo INT NOT NULL,  
costoU DECIMAL(9,2) NOT NULL,  
cantidad INT NOT NULL,  
costoTotal DECIMAL(9,2) NOT NULL,  
PRIMARY KEY(id_pedido, codigo)  
);
```

```
CREATE TABLE INCIDENCIAS(  
id_incidencia INT NOT NULL AUTO_INCREMENT,  
id INT NOT NULL,  
tienda INT NOT NULL,  
codigo_usuario INT NOT NULL,  
fecha DATE NOT NULL,  
solucion VARCHAR(100) NOT NULL,  
estado VARCHAR(100) NOT NULL,  
PRIMARY KEY(id_incidencia,id),  
FOREIGN KEY(codigo_usuario) REFERENCES USER_STORE (_code) ON UPDATE CASCADE,  
FOREIGN KEY(tienda) REFERENCES STORE(_code) ON UPDATE CASCADE  
);
```

```
CREATE TABLE INCIDENCIAS_PRODUCTO(  
id_pedido INT NOT NULL,  
codigo INT NOT NULL,  
cantidad INT NOT NULL,  
motivo VARCHAR(100) NOT NULL,  
PRIMARY KEY(id_pedido, codigo)  
);
```

```
CREATE TABLE DEVOLUCIONES(  
id_devoluciones INT NOT NULL AUTO_INCREMENT,  
id INT NOT NULL,  
tienda INT NOT NULL,  
codigo_usuario INT NOT NULL,  
fecha DATE NOT NULL,  
total DECIMAL(9,2) NOT NULL,  
estado VARCHAR(100) NOT NULL,  
PRIMARY KEY(id_devoluciones,id),  
FOREIGN KEY(codigo_usuario) REFERENCES USER_STORE (_code) ON UPDATE CASCADE,  
FOREIGN KEY(tienda) REFERENCES STORE(_code) ON UPDATE CASCADE  
);
```

```
CREATE TABLE DEVOLUCIONES_PRODUCTO(  
  id_pedido INT NOT NULL,  
  codigo INT NOT NULL,  
  costo DECIMAL(9,2) NOT NULL,  
  cantida INT NOT NULL,  
  costoTotal DECIMAL(9,2),  
  motivo VARCHAR(100) NOT NULL,  
  PRIMARY KEY(id_pedido, codigo)  
);
```