**PROYECTO FINAL**

****

**BASES DE DATOS AVANZADAS**

**ROBINSON ALZA LARA – 20222578159**

**EDWARD GIOVANNI CASTILLO BELTRAN -     20222578040**

**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS**

**FACULTAD TECNOLÓGICA**

**PROYECTO CURRICULAR TECNOLOGÍA EN SISTEMATIZACIÓN DE DATOS**

**BOGOTÁ D.C. 10, NOVIEMBRE**

**2024**

Contenido

[Enunciado 3](#_Toc184672294)

[Definición del problema 4](#_Toc184672295)

[1. Descripción de funcionalidades: 4](#_Toc184672296)

[2. Descripción del entorno de ejecución 5](#_Toc184672297)

[3. Definición de las reglas de negocio. 7](#_Toc184672298)

[Diagrama del modelo Entidad Relación. 10](#_Toc184672299)

[Diagrama del modelo Relacional 10](#_Toc184672300)

[Diccionario de datos 11](#_Toc184672301)

[1. Descripción de tablas y columnas (Pendiente cambio de formato) 11](#_Toc184672302)

[Enlace repositorio 26](#_Toc184672303)

# Enunciado

La empresa PRONTOMUEBLE se dedica a la comercialización de muebles y desea llevar el control de las diferentes actividades que se generan alrededor de la venta de estos muebles.

Cuenta con algunos proveedores, que se especializan en fabricación de diferentes tipos de muebles como, por ejemplo: para cocina, para baño, y de alcoba. De estos fabricantes la empresa conserva el nombre, la dirección, números de teléfono y la persona de contacto. Es importante saber también que tipos de muebles son los que este proveedor le proporciona a PRONTOMUEBLE. Con respecto a los muebles, se debe determinar el tipo de mueble, material en el cual esta hecho, dimensiones, color y precio. La venta de los muebles no incluye la instalaci

Un vendedor podría vender varios de estos muebles a un cliente o a varios clientes. De los clientes de la empresa le interesa conocer su identificación, nombre, teléfonos de contacto, dirección de correspondencia y correo electrónico. Al final del mes, PRONTOMUEBLE acostumbra a generar los reportes mensuales en los cuales se puede determinar cual vendedor realizo mas ventas durante el mes, cuales son los clientes nuevos, cuales clientes realizaron las mayores compras, cuales son los muebles más vendidos, entre otros.

# Definición del problema

## Descripción de funcionalidades:

**Inicio de Sesión**

La aplicación dispondrá de un apartado de inicio de sesión donde tanto vendedores como el administrador podrán acceder con sus credenciales correo y contraseña. Cada tipo de usuario contará con permisos específicos de acuerdo con su rol dentro de la empresa.

**Home**

Al ingresar, se mostrará una página principal con un menú de navegación que da acceso a los diferentes servicios de la aplicación.

En el servicio (Home) se encontrara un collage de graficas, sobre el producto mas vendido en el mes, productos próximos a acabar, vendedor con mayores ventas y recaudos del mes, estas estadísticas pueden varias dependiendo de los tipos de acceso.

Accesos

1. **Administrador**
   1. Este es considerado un super usuario y se solicita que sea el único que pueda tener acceso a la modificación directa de la información exceptuando las facturas.
   2. Puede ver las estadísticas generales de cada vendedor.
   3. Puede ver, modificar y agregar, proveedores, vendedores, muebles con sus respectivas características.
2. **Vendedor**
   1. Este tipo de usuarios solamente podrá crear, cotizaciones, clientes, y facturas, siendo modificables únicamente los dos primeros ademas de su perfil.
   2. Puede ver sus estadísticas generales de ventas, recaudos y ventas pendientes (Cotizaciones)
3. **Cliente**
   1. El cliente solamente existe como una representación de la información a quien se esta realizando una venta, debido a ello no interactúa directamente con la aplicación sino como un vendedor, debido a que se solicita realizar un sitio web para el seguimiento de los mismos.
   2. Si desea conocer su factura deberá solicitársela al vendedor.
4. **Proveedores** 
   1. Como funciona con los clientes, estos únicamente existen como medio informativo para conocer quien está proveyendo el mueble, a que precio y cuanta cantidad.

**Cotización**

La cotización se manejará como una preventa, donde se podrá agregar el cliente a quien se va a realizar dicha acción, el producto a cotizar con su respectivo proveedor, con su precio y la cantidad limite a pedir.

Se podrá conocer la cantidad y el precio total de la cotización

Realizado el proceso de preventa el vendedor generara una factura con la información de la cotización

**Factura**

Se solicita que el servicio factura sea mostrado en un formato tabla donde se podrá visualizar IdFactura, nombre del cliente, fecha y hora de creación, cantidad vendida, precio final y si es un administrador se sumara el campo de vendedor que realizo la venta, sumado a esto las acciones de ver a detalle

**Vista a detalle**

La vista a detalle mostrara los campos previamente mencionados en el formato de la tabla, además a que proveedor se compró el mueble, e información a detalle del vendedor.

Peoductos

Se solicita que el servicio productos sea mostrado en un formato tabla donde se podrá visualizar IdProucto, nombre, cantidad, precio, tipo, listado de proveedores y adicional a esto la visualización y la edición de este.

En la modificación del producto se podrá agregar el proveedor con el precio y cantidad, las propiedades con su respectiva descripción y la imagen del producto.

En la visualización se mostrarán las propiedades anteriormente mencionadas.

**Estadísticas y Reportes**

Al final de cada mes, se generarán reportes que incluirán:

1. Indicación de los vendedores con mayor número de ventas realizadas en el período.
2. Listado de los nuevos clientes registrados en el último mes.
3. Identificación de los clientes que realizaron las compras más importantes.
4. Resumen de los muebles con mayor demanda.
5. Gráficas interactivas para facilitar la interpretación de los datos de ventas, clientes y muebles vendidos.

Manejo de roles

1. Administrador:

* Creación
* Consulta
* Eliminar
* Asociar

1. Cliente:
   * Comprar
2. Vendedor:
   * Consultar
   * Creación
   * Vender

## Descripción del entorno de ejecución

Este proyecto tiene como objetivo desarrollar un sistema de gestión de relaciones con el cliente (CRM) utilizando una combinación de PHP, HTML5, Bootstrap, JavaScript y MySQL para la gestión de bases de datos. Este CRM facilitará la gestión de datos de clientes, interacciones, ventas y otros procesos comerciales.

1. **Tecnologías usadas:**
   * **PHP:** Lenguaje de secuencias de comandos del lado del servidor utilizado para contenido dinámico y lógica del lado del servidor.
   * **HTML 5:** Lenguaje de marcado utilizado para estructurar y presentar contenido en la web.
   * **Bootstrap 5:** Marco de interfaz de usuario para desarrollar páginas web responsivas y orientadas a dispositivos móviles.
   * **Java Script:** Lenguaje de scripting para la creación de páginas web interactivas y dinámicas.
   * **MySQL:** Sistema de gestión de bases de datos relacionales para almacenar y gestionar datos.
   * **Patrón DAO:** Data Access Object estamos hablando de un patrón de diseño muy clásico en el cual una clase se encarga de las operaciones de persistencia contra una tabla de la base de datos
2. **Entorno de ejecución:**
   * Servidor web:
     + Software: Apache, este software se utilizará porque soporta el entorno de PHP incluyendo sus plataformas
     + Sistema operativo: compatible con Windows, Linux o MacOS debido a que será un servicio web
   * PHP: Dado al tipo de patrón que se utilizara para el desarrollo del proyecto, sumado a esto PHP es utilizado un 77,4% de los casi 2 mil millones de sitios web utilizan PHP permitiendo así un fácil mantenimiento.
     + Versión: PHP 8 o superiores
   * Motor base de datos: Se utilizara el motor desarrollado por PHP MySQL debido a la compatibilidad que tiene con el lenguaje, sumado a esto la interfaz para el manejo de la base de datos es intuitiva y fácil de utilizar.
     + Software: MySQL
     + Versión: MySQL 5.7 o superiores
   * Desarrollo Front-End:
     + HTML5
     + Bootstrap 5 o superiores, se utilizará para que el sitio web sea responsive.
     + Java Script: incluyendo JQuery y algunas librerías extras, este se utilizará para los eventos y cambios de estado que tenga el sitio web.
   * Herramientas y librerías:
     + Editor de código: VS Code.
     + Control de versiones: GitHub y Git
     + Mockups: Adobe XD

## Definición de las reglas de negocio.

|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripción** |
| R001 | Correo personal de la empresa | Todo usuario de la empresa posee una extensión de correo electrónico de dominio propio para ingresar al sitio web |
| R002 | Cuanta de administrador | Los administradores poseen una cuenta como perfil de administrador |
| R003 | Registro de compra | La generación de facturas que realice un cliente quedara registrada con el vendedor que la realizo y únicamente la registrara el vendedor |
| R004 | Cotización | El vendedor podrá generar una cotización (preventa) para el cliente, si este desea conocer la cantidad en stock y el precio unitario actual, además el valor total del pedido a realizar |
| R005 | Creación | El Administrador es el único usuario que puede realizar creaciones en todas las entidades exceptuando facturas. |
| R006 | Propiedades independientes | El mueble por vender posee diferentes propiedades que el administrador seleccionara y agregar una descripción a la hora de la creación del producto |
| R007 | Venta de muebles | Un mueble (Producto) puede ser suministrado por diferentes proveedores con una cantidad y precio que pueden ser diferentes |
| R008 | Reportes mensuales | El administrador es el único que tiene reportes, dichos reportes son: Ventas realizadas con su respectivo vendedor, clientes nuevos, recaudo mensual, el vendedor con más ventas, el cliente con más compras y el producto más vendido. |
| R009 | Consultas | El vendedor puede conocer únicamente las ventas, cotizaciones realizadas por él y los clientes realizados por él. |
| R010 | Finalización de venta | Al finalizar la venta, la cotización será vaciada automáticamente y pasará al detalle de la factura. |
| R011 | Persistencia de la cotización | Solo puede existir una cotización por cliente |
| R012 | Persistencia de las ventas | Aunque se modifique el precio del producto la factura tendrá el valor de producto en su día de venta. |
| R013 | Creación del cliente | El cliente es creado por el vendedor, a su vez se tomará la fecha de creación. |
| R014 | Inicio de sesión | El inicio de sesión será general para todos los integrantes de la empresa. |
| B015 | Registro de números | El proveedor como el cliente podrá registrar cuanto número quiera, para que la empresa pueda tener contacto directo (Telefonos) |
| B016 | Registrar Pedido de compra | ProntoMueble desea abarcar la mayor cantidad de solicitud de productos por parte de los clientes, para esto realizara un registro del pedido de los productos suministrados por el proveedor |
| B017 | Concretación Venta | La empresa va a utilizar dos tipos de venta, una preventa y venta, siendo la preventa la cotización y venta la concertación de la compra (Factura). |
| B018 | Detalle de venta | Para cada factura debe existir un detalle de venta, que especifique que mueble se vendió a que proveedore pertenece ese mueble, la cantidad y el valor unitario al momento de realizar la venta. |
| B019 | Modificación de Factura. | La factura ni el Detalle de factura de la venta realiza se podrán modificar. |
| B020 | Propiedades del mueble. | Con el fin de optimizar procesos, al momento de realizar el registro de los productos el administrador podrá modificar en cualquier momento las propiedades del mueble. |

# Diagrama del modelo Entidad Relación.

Diagrama

Descripción generada automáticamente

**(Figura 1 Modelo ER PRONTOMUEBLE)**

# Diagrama del modelo Relacional

**(Figura 2 Modelo R PRONTOMUEBLE)**

# Diccionario de datos

## 1. Descripción de tablas y columnas (Pendiente cambio de formato)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Proveedor** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
| PK | idProveedor | Identificador único del proveedor | INT | 10 | Not null |
|  | razonSocial | Razón social del proveedor | VARCHAR | 100 | Not null |
|  | direccion | Dirección física del proveedor | VARCHAR | 150 | Not null |
|  | nit | Número de identificación tributaria del proveedor | VARCHAR | 10 | Not null |
|  | personaContacto | Nombre de la persona de contacto del proveedor | VARCHAR | 45 | Not null |
|  | img | Nombre de la imagen del proveedor | VARCHAR | 45 | **Null** |
| FK | administrador\_idAdministrador | Identificador del administrador responsable | INT | 10 | Not null |
| **Descripción General** | | | | | |
| El atributo “nombrePersonaContacto” es la persona que representa al proveedor, la cual se cargada de realizar las negociaciones entre el proveedor y la empresa de muebles | | | | | |
| **Script SQL** | | | | | |
| CREATE TABLE Proveedor (  `idProveedor` INT NOT NULL AUTO\_INCREMENT,  `personaContacto` VARCHAR(45) NOT NULL,  `razonSocial` VARCHAR(100) NOT NULL,  `direccion` VARCHAR(45) NOT NULL,  `nit` VARCHAR(45) NOT NULL,  `img` VARCHAR(45) NULL,  `Administrador\_idAdministrador` INT NOT NULL,  PRIMARY KEY (`idProveedor`),  INDEX `fk\_Proveedor\_Administrador1\_idx` (`Administrador\_idAdministrador` ASC),  CONSTRAINT `fk\_Proveedor\_Administrador1`  FOREIGN KEY (`Administrador\_idAdministrador`)  REFERENCES `PRONTOMUEBLE`.`Administrador` (`idAdministrador`)  ON DELETE NO ACTION  ON UPDATE NO ACTION)  ENGINE = InnoDB; | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TelefonoP** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
| PK | idTelefonoP | Identificador único del teléfono del proveedor | **BIGINT** | **10** | **Not Null** |
| FK | idProveedor | Identificador del proveedor al que pertenece | INT | **10** | **Not Null** |
| **Descripción General** | | | | | |
| La tabla TelefonoProvedor almacenara todos los números que se registren del proveedor, se idéntica cada uno con el número y se relaciona con el Provedor | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE TelefonoP (**  **`idTelefonoP` BIGINT NOT NULL,**  **`Proveedor\_idProveedor` INT NOT NULL,**  **PRIMARY KEY (`idTelefonoP`),**  **INDEX `fk\_TelefonoP\_Proveedor1\_idx` (`Proveedor\_idProveedor` ASC),**  **CONSTRAINT `fk\_TelefonoP\_Proveedor1`**  **FOREIGN KEY (`Proveedor\_idProveedor`)**  **REFERENCES `PRONTOMUEBLE`.`Proveedor` (`idProveedor`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Administrador** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
| PK | idAdministrador | Identificador único del administrador | **INT** | **10** | **Not Null** |
|  | nombre | Nombre del administrador | VARCHAR | 45 | **Not Null** |
|  | apellido | Apellido del administrador | VARCHAR | 45 | **Not Null** |
|  | correo | Correo electrónico del administrador | VARCHAR | 100 | **Not Null** |
|  | identificacion | Identificación única del administrador | VARCHAR | 10 | **Not Null** |
|  | contraseña | Contraseña del administrador | VARCHAR | 100 | **Not Null** |
|  | img | Nombre de la imagen del administrador | VARCHAR | 45 | **Null** |
| **Descripción General** | | | | | |
| La tabla Administrador almacenara todos los datos del administrador del sitio web, además de sus credenciales de ingreso. | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE Administrador (**  **`idAdministrador` INT NOT NULL AUTO\_INCREMENT,**  **`nombre` VARCHAR(45) NOT NULL,**  **`apellido` VARCHAR(45) NOT NULL,**  **`correo` VARCHAR(100) NOT NULL,**  **`identificacion` VARCHAR(10) NOT NULL,**  **`contraseña` VARCHAR(100) NOT NULL,**  **`img` VARCHAR(45) NULL,**  **PRIMARY KEY (`idAdministrador`))**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Vendedor** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
| PK | idVendedor | Identificador único del vendedor | INT | **10** | **Not Null** |
|  | nombre | Nombre del vendedor | VARCHAR | 45 | **Not Null** |
|  | apellido | Apellido del vendedor | VARCHAR | 45 | **Not Null** |
|  | correo | Correo electrónico del vendedor | VARCHAR | 100 | **Not Null** |
|  | identificacion | Identificación única del vendedor | VARCHAR | 10 | **Not Null** |
|  | contraseña | Contraseña del vendedor | VARCHAR | 100 | **Not Null** |
|  | img | Nombre de la imagen del vendedor | VARCHAR | 45 | **Null** |
| **FK** | Administrador\_idAdministrador | Llave foránea para identificar la persona que relizo el registro | INT | 10 | **Not Null** |
| **Descripción General** | | | | | |
| La tabla vendedor almacenara todos los datos de los vendedores de la empresa registrados en el sitio web, además de sus credenciales de ingreso. | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE Vendedor (**  **`idVendedor` INT NOT NULL AUTO\_INCREMENT,**  **`nombre` VARCHAR(45) NOT NULL,**  **`apellido` VARCHAR(45) NOT NULL,**  **`correo` VARCHAR(100) NOT NULL,**  **`identificacion` VARCHAR(10) NOT NULL,**  **`contraseña` VARCHAR(100) NOT NULL,**  **`img` VARCHAR(45) NULL,**  **`Administrador\_idAdministrador` INT NOT NULL,**  **PRIMARY KEY (`idVendedor`),**  **INDEX `fk\_Vendedor\_Administrador1\_idx` (`Administrador\_idAdministrador` ASC),**  **CONSTRAINT `fk\_Vendedor\_Administrador1`**  **FOREIGN KEY (`Administrador\_idAdministrador`)**  **REFERENCES `PRONTOMUEBLE`.`Administrador` (`idAdministrador`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TelefonoV** | | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** | |
| PK | idTelefonoV | Identificador único del teléfono del cliente | BIGINT | **10** | **Not Null** | |
| FK | idVendedor | Identificador del cliente al que pertenece | INT | **10** | **Not Null** | |
| **Descripción General** | | | | | |
| La tabla TelefonoVendedor almacenara todos los números que se registren del vendedor, se idéntica cada uno con el número y se relaciona con el Vendedor | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE TelefonoV (**  **`idTelefonoV` BIGINT NOT NULL,**  **`Vendedor\_idVendedor` INT NOT NULL,**  **PRIMARY KEY (`idTelefonoV`),**  **INDEX `fk\_TelefonoV\_Vendedor1\_idx` (`Vendedor\_idVendedor` ASC),**  **CONSTRAINT `fk\_TelefonoV\_Vendedor1`**  **FOREIGN KEY (`Vendedor\_idVendedor`)**  **REFERENCES `PRONTOMUEBLE`.`Vendedor` (`idVendedor`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cliente** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
| PK | idCliente | Identificador único del cliente | BIGINT | **10** | **Not Null** |
|  | nombre | Nombre del cliente | VARCHAR | 45 | **Not Null** |
|  | apellido | Apellido del cliente | VARCHAR | 45 | **Not Null** |
|  | correo | Correo electrónico del cliente | VARCHAR | 100 | **Not Null** |
|  | identificacion | Identificación única del cliente | VARCHAR | 10 | **Not Null** |
|  | contraseña | Contraseña del cliente | VARCHAR | 100 | **Not Null** |
|  | fecha\_creacion | Fecha de registro del cliente | DATE |  | **Not Null** |
| FK | vendedor\_idVendedor | Referencia al vendedor asignado | INT | 10 | **Not Null** |
| **Descripción General** | | | | | |
| La tabla Cliente almacenara todos los datos del cliente, se relaciona con el vendedor que lo registra en el sitio web. El atributo fecha\_creacion la toma el sitio web de manera automática y corresponde a la fecha local | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE Cliente (**  **`idCliente` INT NOT NULL AUTO\_INCREMENT,**  **`nombre` VARCHAR(45) NOT NULL,**  **`apellido` VARCHAR(45) NOT NULL,**  **`correo` VARCHAR(100) NOT NULL,**  **`identificacion` VARCHAR(10) NOT NULL,**  **`fecha` DATE NOT NULL,**  **`Vendedor\_idVendedor` INT NOT NULL,**  **PRIMARY KEY (`idCliente`),**  **INDEX `fk\_Cliente\_Vendedor1\_idx` (`Vendedor\_idVendedor` ASC),**  **CONSTRAINT `fk\_Cliente\_Vendedor1`**  **FOREIGN KEY (`Vendedor\_idVendedor`)**  **REFERENCES `PRONTOMUEBLE`.`Vendedor` (`idVendedor`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TelefonoC** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
| PK | idTelefonoC | Identificador único del teléfono del cliente | BIGINT | **10** | **Not Null** |
| FK | idCliente | Identificador del cliente al que pertenece | INT | **10** | **Not Null** |
| **Descripción General** | | | | | |
| La tabla **TelefonoCliente** almacenara todos los números que se registren del Cliente, se idéntica cada uno con el número y se relaciona con el Cliente | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE TelefonoC (**  **`idTelefonoC` BIGINT NOT NULL,**  **`Cliente\_idCliente` INT NOT NULL,**  **PRIMARY KEY (`idTelefonoC`),**  **INDEX `fk\_TelefonoC\_Cliente1\_idx` (`Cliente\_idCliente` ASC),**  **CONSTRAINT `fk\_TelefonoC\_Cliente1`**  **FOREIGN KEY (`Cliente\_idCliente`)**  **REFERENCES `PRONTOMUEBLE`.`Cliente` (`idCliente`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mueble** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
| PK | idMueble | Identificador único del mueble | INT | **10** | **Not Null** |
|  | nombre | Nombre del mueble | VARCHAR | 45 | **Not Null** |
|  | descripcion | Descripción del mueble | VARCHAR | 150 | **Not Null** |
|  | img | Nombre de la imagen del mueble | VARCHAR | 45 | **Null** |
| FK | administrador\_idAdministrador | Referencia al administrador responsable | INT | **10** | **Not Null** |
| FK | idTipo | Referencia al tipo de mueble | INT | **10** | **Not Null** |
| **Descripción General** | | | | | |
| La tabla **Mueble** almacenara todos los datos relacionados al mueble(en general). | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE Mueble (**  **`idMueble` INT NOT NULL AUTO\_INCREMENT,**  **`nombre` VARCHAR(45) NOT NULL,**  **`descripcion` VARCHAR(45) NOT NULL,**  **`img` VARCHAR(45) NULL,**  **`Administrador\_idAdministrador` INT NOT NULL,**  **`Tipo\_idTipo` INT NOT NULL,**  **PRIMARY KEY (`idMueble`),**  **INDEX `fk\_Mueble\_Administrador1\_idx` (`Administrador\_idAdministrador` ASC),**  **INDEX `fk\_Mueble\_Tipo1\_idx` (`Tipo\_idTipo` ASC),**  **CONSTRAINT `fk\_Mueble\_Administrador1`**  **FOREIGN KEY (`Administrador\_idAdministrador`)**  **REFERENCES `PRONTOMUEBLE`.`Administrador` (`idAdministrador`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION,**  **CONSTRAINT `fk\_Mueble\_Tipo1`**  **FOREIGN KEY (`Tipo\_idTipo`)**  **REFERENCES `PRONTOMUEBLE`.`Tipo` (`idTipo`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PropiedadMueble** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
|  | descripcion | Descripción de la propiedad | VARCHAR | 100 | **Not Null** |
| FK | propiedad\_idPropiedades | Referencia a la propiedad específica | INT | **10** | **Not Null** |
| FK | mueble\_idMueble | Referencia al mueble al que pertenece | INT | **10** | **Not Null** |
| **Descripción General** | | | | | |
| La tabla **MueblePropiedad** es una tabla intermedia que almacenara todas las propiedades con las que cuenta un mueble, se relaciona con el muble para que pueda agregar cuantas propiedades desea a cada mueble. | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE PropiedadMueble (**  **`descripcion` VARCHAR(45) NOT NULL,**  **`Propiedad\_idPropiedad` INT NOT NULL,**  **`Mueble\_idMueble` INT NOT NULL,**  **PRIMARY KEY (`Propiedad\_idPropiedad`, `Mueble\_idMueble`),**  **INDEX `fk\_PropiedadMueble\_Mueble1\_idx` (`Mueble\_idMueble` ASC),**  **CONSTRAINT `fk\_PropiedadMueble\_Propiedad1`**  **FOREIGN KEY (`Propiedad\_idPropiedad`)**  **REFERENCES `PRONTOMUEBLE`.`Propiedad` (`idPropiedad`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION,**  **CONSTRAINT `fk\_PropiedadMueble\_Mueble1`**  **FOREIGN KEY (`Mueble\_idMueble`)**  **REFERENCES `PRONTOMUEBLE`.`Mueble` (`idMueble`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Propiedad** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
| PK | idPropiedades | Identificador único de la propiedad | INT | **10** | **Not Null** |
|  | nombre | Nombre de la propiedad (ej. color, material) | VARCHAR | 45 | **Not Null** |
| FK | tipo\_idTipo | Referencia al tipo de mueble al que aplica | INT | **10** | **Not Null** |
| **Descripción General** | | | | | |
| La tabla **Propiedad** almacenara todas las propiedades con las que cuenta un mueble, se relaciona con el Tipo para que se pueda categorizar las propiedades de cada mueble. | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE Propiedad (**  **`idPropiedad` INT NOT NULL AUTO\_INCREMENT,**  **`nombre` VARCHAR(45) NOT NULL,**  **`Tipo\_idTipo` INT NOT NULL,**  **PRIMARY KEY (`idPropiedad`),**  **INDEX `fk\_Propiedad\_Tipo1\_idx` (`Tipo\_idTipo` ASC),**  **CONSTRAINT `fk\_Propiedad\_Tipo1`**  **FOREIGN KEY (`Tipo\_idTipo`)**  **REFERENCES `PRONTOMUEBLE`.`Tipo` (`idTipo`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PedidoMueble** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
|  | cantidad | Cantidad de muebles disponibles | INT | **10** | **Not Null** |
|  | precio | Precio del mueble | DOUBLE | 10,2 | **Not Null** |
| FK | Proveedor\_idProveedor | Referencia al proveedor responsable | INT | **10** | **Not Null** |
| FK | Mueble\_idMueble | Referencia al mueble | INT | **10** | **Not Null** |
| **Descripción General** | | | | | |
| La tabla **ProveedorMueble** es una tabla intermedia que almacena los datos de un mueble(único), además la relación entre el proveedor y el mueble. | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE PedidoMueble (**  **`cantidad` INT NOT NULL,**  **`precio` DOUBLE NOT NULL,**  **`Proveedor\_idProveedor` INT NOT NULL,**  **`Mueble\_idMueble` INT NOT NULL,**  **PRIMARY KEY (`Proveedor\_idProveedor`, `Mueble\_idMueble`),**  **INDEX `fk\_ProveedorMueble\_Mueble1\_idx` (`Mueble\_idMueble` ASC),**  **CONSTRAINT `fk\_ProveedorMueble\_Proveedor`**  **FOREIGN KEY (`Proveedor\_idProveedor`)**  **REFERENCES `PRONTOMUEBLE`.`Proveedor` (`idProveedor`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION,**  **CONSTRAINT `fk\_ProveedorMueble\_Mueble1`**  **FOREIGN KEY (`Mueble\_idMueble`)**  **REFERENCES `PRONTOMUEBLE`.`Mueble` (`idMueble`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cotizacion** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
|  | cantidad | Cantidad de muebles cotizados | INT | **10** | **Not Null** |
| FK | mueble\_idMueble | Referencia al mueble cotizado | INT | **10** | **Not Null** |
| FK | cliente\_idCliente | Referencia al cliente que solicitó la cotización | INT | **10** | **Not Null** |
| FK | vendedor\_idVendedor | Referencia al vendedor que generó la cotización | INT | **10** | **Not Null** |
| **Descripción General** | | | | | |
| La tabla Cotizacio es una tabla que almacenara todas temporales que las compras no concretadas por un cliente. | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE Cotizacion (**  **`cantidad` INT NOT NULL,**  **`Cliente\_idCliente` INT NOT NULL,**  **`Vendedor\_idVendedor` INT NOT NULL,**  **`ProveedorMueble\_Proveedor\_idProveedor` INT NOT NULL,**  **`ProveedorMueble\_Mueble\_idMueble` INT NOT NULL,**  **PRIMARY KEY (`Cliente\_idCliente`, `Vendedor\_idVendedor`, `ProveedorMueble\_Proveedor\_idProveedor`, `ProveedorMueble\_Mueble\_idMueble`),**  **INDEX `fk\_Cotizacion\_Cliente1\_idx` (`Cliente\_idCliente` ASC),**  **INDEX `fk\_Cotizacion\_Vendedor1\_idx` (`Vendedor\_idVendedor` ASC),**  **INDEX `fk\_Cotizacion\_ProveedorMueble1\_idx` (`ProveedorMueble\_Proveedor\_idProveedor` ASC, `ProveedorMueble\_Mueble\_idMueble` ASC),**  **CONSTRAINT `fk\_Cotizacion\_Cliente1`**  **FOREIGN KEY (`Cliente\_idCliente`)**  **REFERENCES `PRONTOMUEBLE`.`Cliente` (`idCliente`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION,**  **CONSTRAINT `fk\_Cotizacion\_Vendedor1`**  **FOREIGN KEY (`Vendedor\_idVendedor`)**  **REFERENCES `PRONTOMUEBLE`.`Vendedor` (`idVendedor`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION,**  **CONSTRAINT `fk\_Cotizacion\_ProveedorMueble1`**  **FOREIGN KEY (`ProveedorMueble\_Proveedor\_idProveedor` , `ProveedorMueble\_Mueble\_idMueble`)**  **REFERENCES `PRONTOMUEBLE`.`PedidoProveedor` (`Proveedor\_idProveedor` , `Mueble\_idMueble`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Factura** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
| PK | idFactura | Identificador único de la factura | INT | **10** | **Not Null** |
|  | cantidad | Cantidad total de muebles en la factura | INT | **10** | **Not Null** |
|  | subTotal | Subtotal de la factura | DECIMAL | 10,2 | **Not Null** |
|  | fechaCreacion | Fecha de emisión de la factura | DATE |  | **Not Null** |
|  | horaCreacion | Hora de emisión de la factura | TIME |  | **Not Null** |
|  | iva | IVA aplicado en la factura | DECIMAL | 5,2 | **Not Null** |
|  | total | Total de la factura | DECIMAL | 10,2 | **Not Null** |
| FK | vendedor\_idVendedor | Referencia al vendedor que realizó la venta | INT | **10** | **Not Null** |
| FK | cliente\_idCliente | Referencia al cliente que realizó la compra | INT | **10** | **Not Null** |
| **Descripción General** | | | | | |
| La tabla Factura es una tabla que recopila los datos mas relevantes para generar la información de la factura | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE Factura (**  **`idFactura` INT NOT NULL AUTO\_INCREMENT,**  **`cantidadTotal` INT NOT NULL,**  **`subTotal` DOUBLE NOT NULL,**  **`fecha` DATE NOT NULL,**  **`hora` TIME NOT NULL,**  **`iva` INT NOT NULL,**  **`total` DOUBLE NOT NULL,**  **`Vendedor\_idVendedor` INT NOT NULL,**  **`Cliente\_idCliente` INT NOT NULL,**  **PRIMARY KEY (`idFactura`),**  **INDEX `fk\_Factura\_Vendedor1\_idx` (`Vendedor\_idVendedor` ASC),**  **INDEX `fk\_Factura\_Cliente1\_idx` (`Cliente\_idCliente` ASC),**  **CONSTRAINT `fk\_Factura\_Vendedor1`**  **FOREIGN KEY (`Vendedor\_idVendedor`)**  **REFERENCES `PRONTOMUEBLE`.`Vendedor` (`idVendedor`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION,**  **CONSTRAINT `fk\_Factura\_Cliente1`**  **FOREIGN KEY (`Cliente\_idCliente`)**  **REFERENCES `PRONTOMUEBLE`.`Cliente` (`idCliente`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DetalleFactura** | | | | | |
| **Tipo llave** | **Nombre de campo** | **Descripción** | **Tipo de dato** | **Longitud** | **Valores Nulos** |
|  | cantidad | Cantidad de muebles en el detalle | INT | **10** | **Not Null** |
|  | precio | Precio unitario del mueble en el detalle | DECIMAL | 10,2 | **Not Null** |
| FK | mueble\_idMueble | Identificador del mueble facturado | INT | **10** | **Not Null** |
| FK | factura\_idFactura | Identificador de la factura | INT | **10** | **Not Null** |
| **Descripción General** | | | | | |
| La tabla DetalleFactura es una tabla que almacena todos los datos de los productos comprados, además de la relación entre la factura y el mueble comprado. Cuenta con atributos como cantidad y precio que se toman al momento de realizar la compra. | | | | | |
| **Script SQL** | | | | | |
| **CREATE TABLE DetalleFactura (**  **`cantidad` INT NOT NULL,**  **`precio` DOUBLE NOT NULL,**  **`Factura\_idFactura` INT NOT NULL,**  **`ProveedorMueble\_Proveedor\_idProveedor` INT NOT NULL,**  **`ProveedorMueble\_Mueble\_idMueble` INT NOT NULL,**  **PRIMARY KEY (`Factura\_idFactura`, `ProveedorMueble\_Proveedor\_idProveedor`, `ProveedorMueble\_Mueble\_idMueble`),**  **INDEX `fk\_DetalleFactura\_Factura1\_idx` (`Factura\_idFactura` ASC),**  **INDEX `fk\_MuebleFactura\_ProveedorMueble1\_idx` (`ProveedorMueble\_Proveedor\_idProveedor` ASC, `ProveedorMueble\_Mueble\_idMueble` ASC),**  **CONSTRAINT `fk\_DetalleFactura\_Factura1`**  **FOREIGN KEY (`Factura\_idFactura`)**  **REFERENCES `PRONTOMUEBLE`.`Factura` (`idFactura`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION,**  **CONSTRAINT `fk\_MuebleFactura\_ProveedorMueble1`**  **FOREIGN KEY (`ProveedorMueble\_Proveedor\_idProveedor` , `ProveedorMueble\_Mueble\_idMueble`)**  **REFERENCES `PRONTOMUEBLE`.`PedidoProveedor` (`Proveedor\_idProveedor` , `Mueble\_idMueble`)**  **ON DELETE NO ACTION**  **ON UPDATE NO ACTION)**  **ENGINE = InnoDB;** | | | | | |

**Descripción de reglas (Pendiente)**

Los registros de teléfonos deben asociarse a un cliente, vendedor o proveedor.

Las identificaciones (identificacion) de administradores, vendedores y clientes deben ser únicas.

Cada clave foránea debe hacer referencia a una clave primaria válida en otra tabla.

**Descripción de procedimientos y triggers (Pendiente)**

**Descripción de vistas (Pendiente)**

# Enlace repositorio

https://github.com/EdwardGCB/PRONTOMUEBLE.git