UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO



FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES

Sistema de Control de Mantenimiento de Productos para la empresa “IMPRISOL S.R.L”.

PROYECTO FINAL

*Grupo # 13*

DOCENTE : Ing. Evans Balcázar Veizaga

MATERIA : Tecnología Web

INTEGRANTES: Guzman Martinez Vinx Junior

: Trujillo Camacho Vladimir

: Zeballos Suarez Daniel

GRUPO : ´SC´

GESTION : 02-2019

Santa Cruz de la Sierra – Bolivia

2019

Tabla de contenido

[Introducción 9](#_Toc13459287)

[Capítulo I: Problema de investigación 10](#_Toc13459288)

[1. Definición del Problema 10](#_Toc13459289)

[1.1. Situación Problemática 10](#_Toc13459290)

[1.2. Situación deseada 10](#_Toc13459291)

[1.3 Objetivo de investigación 11](#_Toc13459292)

[2. Objetivos 12](#_Toc13459293)

[2.1. Objetivo General 12](#_Toc13459294)

[2.2. Objetivo Especifico 12](#_Toc13459295)

[3. Metodología 12](#_Toc13459296)

[3.1. Fase de Inicio 12](#_Toc13459297)

[3.2. Fase de Elaboración 13](#_Toc13459298)

[3.3. Fase de Construcción 13](#_Toc13459299)

[3.4. Fase de Transición 13](#_Toc13459300)

[4. Alcance 14](#_Toc13459301)

[4.1 Requisitos Funcionales 14](#_Toc13459302)

[4.2. Requisitos no funcionales 14](#_Toc13459303)

[Capítulo 2: IMPRISOL S.R.L 15](#_Toc13459304)

[5. Descripción Breve 15](#_Toc13459305)

[5.1 Localización 15](#_Toc13459306)

[5.2. Misión 16](#_Toc13459307)

[5.3. Visión 16](#_Toc13459308)

[5.4. Organización 16](#_Toc13459309)

[5.5. Modelo de negocio 17](#_Toc13459310)

[Capítulo 3: Marco Teórico 18](#_Toc13459311)

[6. Tecnologías 18](#_Toc13459312)

[6.1 Tecnología Web 18](#_Toc13459313)

[6.2. Laravel 18](#_Toc13459314)

[6.3. PHP 19](#_Toc13459315)

[Capítulo 4: Captura de requisitos 20](#_Toc13459316)

[7. Modelo de dominio 20](#_Toc13459317)

[8. Captura de Requisitos 21](#_Toc13459318)

[8.1. Identificar actores y casos de uso 21](#_Toc13459319)

[8.2. Diagrama General de Casos de Uso 32](#_Toc13459320)

[Capítulo 5: Análisis 33](#_Toc13459321)

[9. Análisis 33](#_Toc13459322)

[9.1. Identificación de Paquetes 33](#_Toc13459323)

[9.2. Análisis de la arquitectura 34](#_Toc13459324)

[9.3. Análisis de Casos de Uso: Diagramas de Comunicación 36](#_Toc13459325)

[Capítulo 6: Diseño 41](#_Toc13459326)

[10. Diseño 41](#_Toc13459327)

[10.1. Diseño de la arquitectura física 41](#_Toc13459328)

[10.2. Diagrama de Navegación 42](#_Toc13459329)

[10.3. Diseño de Casos de Uso: Diagramas de Secuencia 50](#_Toc13459330)

[10.4. Diseño Conceptual de la Base de Datos 59](#_Toc13459331)

[10.5. Diseño Lógico 60](#_Toc13459332)

[10.6. Diseño Físico 62](#_Toc13459333)

[10.7. Script 70](#_Toc13459334)

[Capítulo 7: Implementación 98](#_Toc13459335)

[11. Implementación de la arquitectura 98](#_Toc13459336)

[11.1. Identificación de los componentes 98](#_Toc13459337)

[12. Lenguaje de programación y Framework 98](#_Toc13459338)

[12.1. PHP 98](#_Toc13459339)

[12.2. Laravel 5.8 99](#_Toc13459340)

[13. Herramientas de desarrollo 100](#_Toc13459341)

[13.1 Sistema operativo 100](#_Toc13459342)

[13.2. Sistema Gestor de Base de Datos 100](#_Toc13459343)

[Capítulo 8: Pruebas 102](#_Toc13459344)

[14. Pruebas 102](#_Toc13459345)

[14.1 . Planificar pruebas 102](#_Toc13459346)

[14.2. Diseñar pruebas 102](#_Toc13459347)

[Conclusiones 104](#_Toc13459348)

[Recomendaciones 105](#_Toc13459349)

[Bibliografía 106](#_Toc13459350)

[Anexos 107](#_Toc13459351)

[**Ilustración 1** Objeto de investigación inventario 11](#_Toc13465693)

[**Ilustración 2** Localización de IMPRISOL S.R.L 15](#_Toc13465694)

[**Ilustración 3** Organigrama de la empresa IMPRISOL S.R.L. 16](#_Toc13465695)

[**Ilustración 4** Modelo de negocio. 17](#_Toc13465696)

[**Ilustración 5** Modelo de dominio inicial. 20](#_Toc13465697)

[**Ilustración 6** Diagrama de relación actor y caso de uso CU1. 23](#_Toc13465698)

[**Ilustración 7** Diagrama de relación actor y caso de uso CU2. 24](#_Toc13465699)

[**Ilustración 8** Diagrama de relación actor y caso de uso CU3. 25](#_Toc13465700)

[**Ilustración 9** Diagrama de relación actor y caso de uso CU4. 26](#_Toc13465701)

[**Ilustración 10** Diagrama de relación actor y caso de uso CU5. 27](#_Toc13465702)

[**Ilustración 11** Diagrama de relación actor y caso de uso CU6. 28](#_Toc13465703)

[**Ilustración 12** Diagrama de relación actor y caso de uso CU6. 29](#_Toc13465704)

[**Ilustración 13** Diagrama de relación actor y caso de uso CU8. 29](#_Toc13465705)

[**Ilustración 14** Diagrama de relación actor y caso de uso CU9. 30](#_Toc13465706)

[**Ilustración 15** Diagrama general de caso de usos. 32](#_Toc13465707)

[**Ilustración 16** Paquete de Administración. 33](#_Toc13465708)

[**Ilustración 17** Paquete de Administracion de Inventarios. 33](#_Toc13465709)

[**Ilustración 18** Paquete de Administracion de Ventas y Pedidos. 34](#_Toc13465710)

[**Ilustración 19** Vista del Paquete de Administración. 34](#_Toc13465711)

[**Ilustración 20** Vista del Paquete de Administracion de Inventario. 34](#_Toc13465712)

[**Ilustración 21** Vista del Paquete de Administracion de Ventas y Pedidos. 35](#_Toc13465713)

[**Ilustración 22** Analisis de Paquete. 35](#_Toc13465714)

[**Ilustración 23** Diagrama de comunicación Gestionar Usuario - Administrativo. 36](#_Toc13465715)

[**Ilustración 24** Diagrama de comunicación Gestionar Usuario - Cliente. 36](#_Toc13465716)

[**Ilustración 25** Diagrama de comunicación Gestionar Usuario - Proveedor. 37](#_Toc13465717)

[**Ilustración 26** Diagrama de comunicación Gestionar Producto. 37](#_Toc13465718)

[**Ilustración 27** Diagrama de comunicación Gestionar Venta. 38](#_Toc13465719)

[**Ilustración 28** Diagrama de comunicación Gestionar Inventario. 38](#_Toc13465720)

[**Ilustración 29** Diagrama de comunicación Gestionar Entrega. 39](#_Toc13465721)

[**Ilustración 30** Diagrama de comunicación Gestionar Pedido. 39](#_Toc13465722)

[**Ilustración 31** Diagrama de comunicación Visualizar Reportes. 40](#_Toc13465723)

[**Ilustración 32** Diagrama de comunicación Gestionar Compra Producto. 40](#_Toc13465724)

[**Ilustración 33** Diagrama de despliegue. 41](#_Toc13465725)

[**Ilustración 34** Diagrama de Navegación - Gestionar Usuario – Proveedor. 42](#_Toc13465726)

[**Ilustración 35** Diagrama de Navegación - Gestionar Usuario – Cliente. 43](#_Toc13465727)

[**Ilustración 36** Diagrama de Navegación - Gestionar Usuario – Administrativo. 44](#_Toc13465728)

[**Ilustración 37** Diagrama de Navegación - Gestionar Producto. 45](#_Toc13465729)

[**Ilustración 38** Diagrama de Navegación - Gestionar Venta. 46](#_Toc13465730)

[**Ilustración 39** Diagrama de Navegación - Gestionar Inventario. 47](#_Toc13465731)

[**Ilustración 40** Diagrama de Navegación - Gestionar Entrega. 48](#_Toc13465732)

[**Ilustración 41** Diagrama de Navegación - Gestionar Pedido. 49](#_Toc13465733)

[**Ilustración 42** Diagrama de Navegación – Visualizar Reportes. 50](#_Toc13465734)

[**Ilustración 43** Diagrama de Navegación – Visualizar Estadísticas. 50](#_Toc13465735)

[**Ilustración 44** Diagrama de Secuencia Gestionar Usuario - Proveedor. 50](#_Toc13465736)

[**Ilustración 45** Diagrama de Secuencia Gestionar Usuario - Cliente. 51](#_Toc13465737)

[**Ilustración 46** Diagrama de Secuencia Gestionar Usuario - Administrativo. 52](#_Toc13465738)

[**Ilustración 47** Diagrama de Secuencia Gestionar Producto. 53](#_Toc13465739)

[**Ilustración 48** Diagrama de Secuencia Gestionar Venta. 54](#_Toc13465740)

[**Ilustración 49** Diagrama de Secuencia Gestionar Venta. 55](#_Toc13465741)

[**Ilustración 50** Diagrama de Secuencia Gestionar Entrega. 56](#_Toc13465742)

[**Ilustración 51** Diagrama de Secuencia Gestionar Pedido. 57](#_Toc13465743)

[**Ilustración 52** Diagrama de Secuencia Gestionar Compra Producto. 58](#_Toc13465744)

[**Ilustración 53** Diagrama de conceptual de base de datos. 60](#_Toc13465745)

[**Ilustración 54** Diagrama de componentes. 98](#_Toc13465746)

# Introducción

El constante crecimiento del internet a nivel mundial, la globalización, la tecnología y las comunicaciones, hacen que el público demande nuevas maneras de realizar reservas o poder realizar cotizaciones, demandando no solo calidad, sino que también esperan que se le brinde mayor accesibilidad a los servicios ya disponibles y una mejor atención al cliente.

El correo electrónico es una de las aplicaciones TCP/IP más utilizadas en estos días. El correo electrónico es un recurso tecnológico que nos permite comunicarnos desde cualquier parte del mundo a través de Internet. El correo electrónico (también conocido como e-mail, un término inglés derivado de electronic mail) es un servicio que permite el intercambio de mensajes a través de sistemas de comunicación electrónicos. El concepto se utiliza principalmente para denominar al sistema que brinda este servicio vía Internet mediante el protocolo SMTP (Simple Mail Transfer Protocol), pero también permite nombrar a otros sistemas similares que utilicen distintas tecnologías. Los mensajes de correo electrónico posibilitan el envío, además de texto, de cualquier tipo de documento digital.

El funcionamiento del correo electrónico es similar al del correo postal. Ambos permiten enviar y recibir mensajes, que llegan a destino gracias a la existencia de una dirección. El correo electrónico también tiene sus propios buzones: son los servidores que guardan temporalmente los mensajes hasta que el destinatario los revisa.

# Capítulo I: Problema de investigación

## 1. Definición del Problema

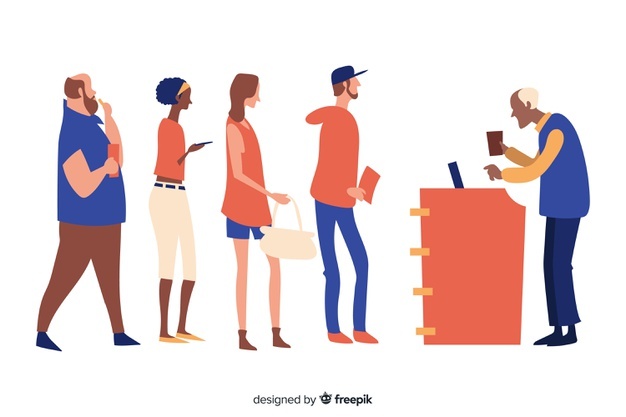
### 1.1. Situación Problemática

En la empresa IMPRISOL S.R.L. existe una deficiencia en el control de las garantías y mantenimientos de los productos de sus clientes, las cuales la atención es numerosa, al no ser estas atendidas rápidamente generan pérdidas.

### 1.2. Situación deseada

Permitir al administrativo el sistema poder llevar un control de los mantenimientos de los productos, proporcionando un sistema fiabilidad en la información. Como también se desea permitir al cliente realizar una reserva de manera directa mediante correo.

### 1.3 Objetivo de investigación



**Ilustración 1** Objeto de investigación cliente en espera



**Ilustración 2** Objeto de investigación garantía

## 2. Objetivos

### 2.1. Objetivo General

Desarrollar un Sistema de Control de Mantenimiento de Productos para la empresa “IMPRISOL S.R.L”.

### 2.2. Objetivo Especifico

* Recopilar información y flujo de negocio de la empresa.
* Definir los requerimientos funcionales y no funcionales del software.
* Realizar el diseño del software con el objetivo de describir como deberá ser implementado el sistema, satisfaciendo los requisitos de este.
* Aplicación de los protocolos de correo electrónico para capturar la información.
* Realizar la implementación del software de acuerdo con el diseño realizado.
* Realizar las pruebas necesarias para garantizar el correcto funcionamiento del software.

## 3. Metodología

La metodología que adoptamos para el proyecto es el Proceso Unificado de Desarrollo de Software (PUDS), Rumbaugh, Jacobson, y Booch, (2001).

Además, se utilizará al Lenguaje Unificado de Modelado (UML) como lenguaje de representación de los modelos resultantes en cada actividad del PUDS, Jacobson et al. (2000).

A continuación, veremos las Fases, “Flujo de Trabajo”, y “Actividades” correspondientes al PUDS que se llevarán a cabo durante el desarrollo del sistema.

### 3.1. Fase de Inicio

**Captura de Requisitos**

* Requisitos funcionales y no funcionales
* Lista de actores y casos de uso
* Detalle de casos de uso
* Prototipo de la interfaz de los casos de uso

**Resultado**

* Modelo de Casos de uso

### 3.2. Fase de Elaboración

**Actividad**

* Diseño de arquitectura
* Diseño de casos de uso
* Diseño de datos
* Resultado
* Descripción del diseño de software

### 3.3. Fase de Construcción

**Implementación**

* Traducción de los modelos de diseño, a lenguaje de programación.

Además, utilizaremos el UML (Lenguaje Unificado de Modelado) como lenguaje de representación visual.

### 3.4. Fase de Transición

* En esta fase final, el programa debe estar listo para ser probado, instalado y utilizado por el cliente sin ningún problema.

## 4. Alcance

### 4.1 Requisitos Funcionales

Gestión Usuarios (Técnico, Administrativo, Cliente)

Gestión Productos

Gestión Servicios de Mantenimiento

Gestión de Garantía

Gestión Reserva

Gestión Cotizaciones.

Gestión Promociones

Reportes y Estadisticas.

### 4.2. Requisitos no funcionales

#### Rendimiento

* **Base de datos:** se usará una base de datos normalizados, pero con un pequeño grado de redundancia.
* **Arquitectura:** totalmente escalable.
* **Interfaz:** se interactuará a través de comandos por correo.

#### Fiabilidad

* Se realizarán las pruebas necesarias para garantizar que el software funcione de forma correcta siempre.

# Capítulo 2: IMPRISOL S.R.L

## 5. Descripción Breve

IMPRISOL S.R.L. Es una empresa que provee servicios de ventas y logística de distribución a empresas y/o minoristas que requieran equipos de impresión, repuestos y todo tipo de insumos electrónicos de oficina. Esta cuenta con equipo profesional calificado para el asesoramiento e instalación de los insumos electrónicos a empresas que lo requieran como también chóferes para la distribución de los equipos e insumos de oficina a empresas minoristas dentro del radio urbano.

### 5.1 Localización

C. / Gustavo Parada 100, Santa Cruz de la Sierra.

**Ilustración 2** Localización de IMPRISOL S.R.L

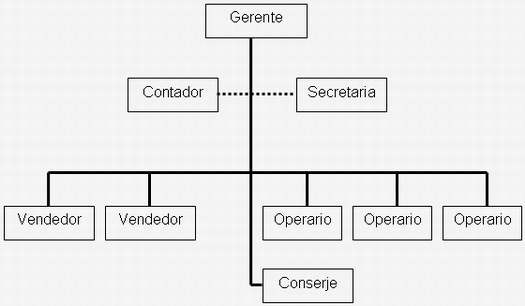
### 5.2. Misión

Satisfacer en sumo grado las necesidades de impresión de nuestros clientes a través de la oferta de un completo portafolio de productos de alta calidad, tanto compatibles como re manufacturados, para impresoras matriciales y láser a precios muy económicos.

### 5.3. Visión

Por su nivel de crecimiento, calidad y servicio, está llamada a constituirse en 5 años unas de las empresas líderes a nivel nacional en la comercialización de productos consumibles para impresoras

### 5.4. Organización



**Ilustración 3** Organigrama de la empresa IMPRISOL S.R.L.

### 5.5. Modelo de negocio

Proceso de Venta

******

**Ilustración 4** Modelo de negocio.

# Capítulo 3: Marco Teórico

## 6. Tecnologías

### 6.1 Tecnología Web

**Aspectos tecnológicos de la Web**

La World Wide Web, WWW o Web se ha convertido en el principal servicio de Internet. La WWW utiliza la estructura de comunicación existente en Internet y comparte protocolos de comunicaciones comunes, estándares y otras notaciones de protocolos de comunicación que permiten el acceso universal a los servicios de información presentes en la Web a través del llamado modelo de cliente-servidor, esto es, mediante la conexión remota red de ordenadores entre una red de ordenadores o máquinas llamadas servidores, y los ordenadores clientes. En la red, la Web utiliza el protocolo HTTP o HyperText Transfer Protocol, el Protocolo de Transferencia de Hipertexto que permite el intercambio de información hipertextual de las páginas web y que ha sido utilizado por los servidores World Wide Web desde su inicio en 1990. Estos protocolos de comunicaciones estandarizados y, en concreto, el protocolo HTTP, realizan las conexiones (el direccionamiento, la negociación entre recursos y el control de transferencias). Por su parte, las páginas web utilizan el lenguaje HTML o Hypertext MarkUp Language, el lenguaje de marcas de hipertexto que hace posible la WWW. (Lamarca, 2013)

### 6.2. Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su objetivo es desarrollar aplicaciones con código PHP de forma elegante y simple. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

Laravel es un framework joven con gran futuro. Cuenta con una comunidad llena de energía, documentación atractiva de contenido claro y completo; y, además, ofrece las funcionalidades necesarias para desarrollar aplicaciones modernas de manera fácil y segura. Está equipado con un montón de características interesantes, incluyendo enrutamiento RESTful, PHP nativo o atrativo motor ligero y muchos más. Construido con varios componentes de Symfony, Laravel ofrece a las aplicaciones web una increíble base de código confiable y bien probado. (Ecured, 2015)

### 6.3. PHP

PHP. Lenguaje de programación, interpretado, diseñado originalmente para la creación de Páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. PHP es un Acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre. (Ecured, 2018)

# Capítulo 4: Captura de requisitos

## 7. Modelo de dominio



**Ilustración 5** Modelo de dominio inicial.

## 8. Captura de Requisitos

### 8.1. Identificar actores y casos de uso

Lista de actores

|  |
| --- |
| ACTORES |
| 1. Administrativo 2. Cliente 3. Proveedor |

Describir los roles de los actores de los casos de uso

|  |  |
| --- | --- |
| ACTORES | DESCRIPCIÓN |
| Administrativo | Encargado de llevar control y gestión de inventarios. |
| Cliente | Es el interesado en adquirir un producto mediante pedidos. |
| Proveedor | Es el que proveerá los productos para su comercialización. |

#### 8.1.1. Lista de casos de uso

|  |
| --- |
| Casos de Uso |
| CU1 Gestionar usuario  CU2 Gestionar Producto  CU3 Gestionar Venta  CU4 Gestionar Inventario  CU5 Gestionar Entrega  CU6 Gestionar Pedido  CU7 Visualizar Reportes  CU8 Visualizar Estadísticas  CU9 Gestionar Compra Producto |

#### 8.1.2. Priorizar Casos de Uso

|  |  |
| --- | --- |
| Nombre de Caso de Uso | Prioridad |
| CU1: Gestionar Usuario | Normal |
| CU2: Gestionar Producto | Normal |
| CU3: Gestionar Venta | Crítico |
| CU4: Gestionar Inventario | Critico |
| CU5: Gestionar Entrega | Critico |
| CU6: Gestionar Pedido | Critico |
| CU7: Visualizar Reportes | Significativo |
| CU8: Visualizar Estadísticas | Significativo |
| CU9: Gestionar Compra Producto | Critico |

#### 8.1.3. Detalle de los casos de uso

**CU1: Gestionar Usuario**

****

**Ilustración 6** Diagrama de relación actor y caso de uso CU1.

|  |  |
| --- | --- |
| Nombre del Caso de Uso | CU1 Gestionar usuario |
| Propósito | Registrar, modificar y dar de baja a los usuarios del sistema |
| Actores | Administrativo |
| Actor Iniciador | Administrativo |
| Precondición | Ninguna |
| Postcondición | Se ha registrado, modificado o dado de baja a un usuario |
| Flujo Principal | 1. Registrar    1. El administrativo ingresará a la ventana de registrar nuevo usuario, dentro se le pedirá la siguiente información:    * Carnet de Identidad    * Nombre    * Correo electrónico    * Teléfono    * Nombre de usuario    * Contraseña de usuario    1. Luego deberá confirmar el registro del nuevo usuario. 2. Modificar   2.1. El administrativo seleccionará el usuario del que modificará sus datos.  2.2. Realiza la modificación de los datos del usuario.  2.3. Confirma los cambios del usuario.   1. Dar de baja    1. El administrativo selecciona al usuario que dará de baja.    2. Confirma la baja del usuario. |
| Excepciones | * 1. No se rellenó algún campo importante   2. Los tipos de datos no son válidos   3. Los tipos de datos no son válidos |

**CU2: Gestionar Producto**

****

**Ilustración 7** Diagrama de relación actor y caso de uso CU2.

|  |  |
| --- | --- |
| Nombre del Caso de Uso | CU2 Gestionar Producto |
| Propósito | Llevar el registro, actualización y baja de los productos. |
| Actores | Administrativo |
| Actor Iniciador | Administrativo |
| Precondición | Ninguna |
| Postcondición | Se ha llevado la gestión correspondiente en los productos. |
| Flujo Principal | 1. Registrar   * 1. El administrativo ingresará a la ventana de registrar nuevo producto, dentro se le pedirá la siguiente información:   + Codigo   + Nombre   + Marca   + Tipo   + Precio   + Cantidad   1. Luego deberá confirmar el registro del nuevo usuario.  1. Modificar   2.1. El administrativo seleccionará el producto del que modificará sus datos.  2.2. Realiza la modificación de los datos del producto.  2.3. Confirma los cambios del producto.   1. Dar de baja    1. El administrativo selecciona al producto que dará de baja. |
| Excepciones | * 1. El tipo de dato es inválido |

**CU3: Gestionar Venta**

****

**Ilustración 8** Diagrama de relación actor y caso de uso CU3.

|  |  |
| --- | --- |
| Nombre del Caso de Uso | CU3 Gestionar Venta |
| Propósito | Registrar, modificar y anular las ventas en el sistema. |
| Actores | Administrativo |
| Actor Iniciador | Administrativo |
| Precondición | Un pedido debe haber sido registrada |
| Postcondición | Se ha registrado una nueva venta en el sistema. |
| Flujo Principal | 1. Registrar   1.1. El administrativo ingresará la siguiente información:   * Código * Monto total * Pedido   1. Confirmar el registro de venta.  1. Modificar   2.1 El administrativo podrá modificar la siguiente información.   * Código * Monto total  1. Anular   3.1 El administrativo podrá anular una venta por el motivo que sea. |
| Excepciones | * 1. El tipo de dato es inválido |

**CU4: Gestionar Inventario**

****

**Ilustración 9** Diagrama de relación actor y caso de uso CU4.

|  |  |
| --- | --- |
| Nombre del Caso de Uso | CU4 Gestionar Inventario |
| Propósito | Cerrar el inventario del sistema en el sistema. |
| Actores | Administrativo |
| Actor Iniciador | Administrativo |
| Precondición | Gestionar Producto |
| Postcondición | Nuevo cierre de inventario |
| Flujo Principal | 1. Corte de inventario   1.1 Confirmar corte de inventario |
| Excepciones | Ninguna |

**CU5: Gestionar Entrega**

****

**Ilustración 10** Diagrama de relación actor y caso de uso CU5.

|  |  |
| --- | --- |
| Nombre del Caso de Uso | CU5 Gestionar Entrega |
| Propósito | Registrar, modificar y anular las entregas en el sistema. |
| Actores | Administrativo, cliente |
| Actor Iniciador | Administrativo |
| Precondición | Gestionar Venta |
| Postcondición | Ninguna |
| Flujo Principal | 1. Registrar   1.1. El administrador ingresará la siguiente información:   * Código * Fecha entrega * Destino   1. Confirmar el registro de entrega.  1. Modificar   2.1 El administrador podrá modificar la siguiente información.   * Código * Fecha * Destino  1. Anular   3.1 El administrativo podrá anular una entrega a por el motivo que sea. |
| Excepciones | Ninguna |

**CU6: Gestionar Pedido**

******

**Ilustración 11** Diagrama de relación actor y caso de uso CU6.

|  |  |
| --- | --- |
| Nombre del Caso de Uso | CU6 Gestionar Pedido |
| Propósito | Un cliente podrá realizar los pedidos de los productos para su preparación, como también anularlo. |
| Actores | Cliente, Administrativo |
| Actor Iniciador | Cliente |
| Precondición | Gestionar producto |
| Postcondición | Ninguna |
| Flujo Principal | 1. Registrar   1.1. El cliente ingresará la siguiente información:   * Código * Descripción * Producto * Cantidad   1. Confirmar el registro del producto.  1. Modificar   2.1 El administrador podrá modificar la siguiente información.   * Código * Descripción * Producto * Cantidad  1. Anular   3.1 El Cliente podrá anular su pedido a por el motivo que sea. |
| Excepciones | Ninguna |

**CU7: Visualizar Reportes**

****

**Ilustración 12** Diagrama de relación actor y caso de uso CU6.

|  |  |
| --- | --- |
| Nombre del Caso de Uso | CU7 Visualizar Reportes |
| Propósito | Visualizar los reportes sobre los que desee un informe. |
| Actores | Administrativo |
| Actor Iniciador | Administrativo |
| Precondición | Se debe haber realizado todos los casos de uso. |
| Postcondición | Ninguna |
| Flujo Principal | 1. Visualizar    1. El administrativo podrá realizar la visualización de los reportes que seleccione al criterio. |
| Excepciones | Ninguna |

**CU8: Visualizar Estadística**

******

**Ilustración 13** Diagrama de relación actor y caso de uso CU8.

|  |  |
| --- | --- |
| Nombre del Caso de Uso | CU8 Visualizar Estadística |
| Propósito | Visualizar las estadísticas sobre los que desee información. |
| Actores | Administrativo |
| Actor Iniciador | Administrativo |
| Precondición | Los casos de uso deben estar implementados. |
| Postcondición | Ninguna |
| Flujo Principal | 1. Delegar    1. El administrativo podrá visualizar las estadísticas generadas por el sistema. |
| Excepciones | Ninguna |

**CU9: Gestionar Compra Producto**

****

**Ilustración 14** Diagrama de relación actor y caso de uso CU9.

|  |  |
| --- | --- |
| Nombre del Caso de Uso | CU9 Gestionar Compra Producto |
| Propósito | Registrar, modificar y anular las notas de compra de productos entrantes en el sistema. |
| Actores | Administrativo, Proveedor |
| Actor Iniciador | Administrativo |
| Precondición | Gestionar Producto |
| Postcondición | Se actualizo el stock del producto. |
| Flujo Principal | 1. Registrar   1.1. El administrativo ingresará la siguiente información:   * Código * Monto * Descripción * Cantidad * Producto   1. Confirmar el registro de la compra.  1. Modificar   2.1 El administrativo podrá modificar la siguiente información.   * Descripción * Monto * Cantidad  1. Anular   3.1 El Administrativo podrá anular la nota de compra por el motivo que sea. |
| Excepciones | Ninguna |

### 8.2. Diagrama General de Casos de Uso



**Ilustración 15** Diagrama general de caso de usos.

# Capítulo 5: Análisis

## 9. Análisis

### 9.1. Identificación de Paquetes

**Paquete de Administración.**



**Ilustración 16** Paquete de Administración.

**Paquete de Administración de Inventario**



**Ilustración 17** Paquete de Administracion de Inventarios.

**Paquete de Administración de Ventas y Pedidos**



**Ilustración 18** Paquete de Administracion de Ventas y Pedidos.

### 9.2. Análisis de la arquitectura

#### 9.2.1. Vista de paquetes

**Paquete de Administración.**

****

**Ilustración 19** Vista del Paquete de Administración.

**Paquete de Administración de Inventario**

**Ilustración 20** Vista del Paquete de Administracion de Inventario.

**Paquete de Administración de Ventas y Pedidos.**



**Ilustración 21** Vista del Paquete de Administracion de Ventas y Pedidos.

#### 9.2.2. Análisis de paquete



**Ilustración 22** Analisis de Paquete.

### 9.3. Análisis de Casos de Uso: Diagramas de Comunicación

Gestionar Usuario – Administrativo



**Ilustración 23** Diagrama de comunicación Gestionar Usuario - Administrativo.

**Gestionar Usuario – Cliente**

****

**Ilustración 24** Diagrama de comunicación Gestionar Usuario - Cliente.

**Gestionar Usuario – Proveedor**

****

**Ilustración 25** Diagrama de comunicación Gestionar Usuario - Proveedor.

**Gestionar Producto**

****

**Ilustración 26** Diagrama de comunicación Gestionar Producto.

**Gestionar Venta**

****

**Ilustración 27** Diagrama de comunicación Gestionar Venta.

**Gestionar Inventario**

****

**Ilustración 28** Diagrama de comunicación Gestionar Inventario.

**Gestionar Entrega**

****

**Ilustración 29** Diagrama de comunicación Gestionar Entrega.

**Gestionar Pedido**

****

**Ilustración 30** Diagrama de comunicación Gestionar Pedido.

**Visualizar Reportes**

****

**Ilustración 31** Diagrama de comunicación Visualizar Reportes.

**Gestionar Compra Producto**



**Ilustración 32** Diagrama de comunicación Gestionar Compra Producto.

# Capítulo 6: Diseño

## 10. Diseño

### 10.1. Diseño de la arquitectura física



**Ilustración 33** Diagrama de despliegue.

### 10.2. Diagrama de Navegación

**Sistema Web**

**Gestionar Usuario – Proveedor.**

**Ilustración 34** Diagrama de Navegación - Gestionar Usuario – Proveedor.

**Gestionar Usuario – Cliente**

****

**Ilustración 35** Diagrama de Navegación - Gestionar Usuario – Cliente.

**Gestionar Usuario – Administrativo**

****

**Ilustración 36** Diagrama de Navegación - Gestionar Usuario – Administrativo.

**Gestionar Producto**

****

**Ilustración 37** Diagrama de Navegación - Gestionar Producto.

**Gestionar Venta**

****

**Ilustración 38** Diagrama de Navegación - Gestionar Venta.

**Gestionar Inventario**

****

**Ilustración 39** Diagrama de Navegación - Gestionar Inventario.

**Gestionar Entrega**

****

**Ilustración 40** Diagrama de Navegación - Gestionar Entrega.

**Gestionar Pedido**

****

**Ilustración 41** Diagrama de Navegación - Gestionar Pedido.

**Visualizar Reportes**

**Ilustración 42** Diagrama de Navegación – Visualizar Reportes.

**Visualizar Estadísticas**

**Ilustración 43** Diagrama de Navegación – Visualizar Estadísticas.

### 10.3. Diseño de Casos de Uso: Diagramas de Secuencia

**Gestionar Usuario – Proveedor**

****

**Ilustración 44** Diagrama de Secuencia Gestionar Usuario - Proveedor.

### Gestionar Usuario - Cliente

****

**Ilustración 45** Diagrama de Secuencia Gestionar Usuario - Cliente.

### Gestionar Usuario - Administrativo

****

**Ilustración 46** Diagrama de Secuencia Gestionar Usuario - Administrativo.

### Gestionar Producto

****

**Ilustración 47** Diagrama de Secuencia Gestionar Producto.

### Gestionar Venta

****

**Ilustración 48** Diagrama de Secuencia Gestionar Venta.

### Gestionar Venta

****

**Ilustración 49** Diagrama de Secuencia Gestionar Venta.

### Gestionar Entrega

****

**Ilustración 50** Diagrama de Secuencia Gestionar Entrega.

### Gestionar Pedido

****

**Ilustración 51** Diagrama de Secuencia Gestionar Pedido.

### Gestionar Compra Producto

****

**Ilustración 52** Diagrama de Secuencia Gestionar Compra Producto.

### 10.4. Diseño Conceptual de la Base de Datos

#### 10.4.1. Diagrama de Clases



**Ilustración 53** Diagrama de conceptual de base de datos.

### 10.5. Diseño Lógico

**USUARIO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id(PK)** | User | Password | estado | Id\_Administrativo(FK) |

**TipoEntrega**

|  |  |
| --- | --- |
| **Id(PK)** | descripcion |

**Proveedor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id(PK)** | Nombre | Telefono | Direccion | Codigo | Estado |

**Tipo Producto**

|  |  |
| --- | --- |
| **Id(PK)** | Descripcion |

**Cliente**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id(PK)** | Código | Nit | Fecha | Descripcion | Estado |

**Administrativo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id(PK)** | Nombre | Código | Cargo | Teléfono | Fecha\_ingreso | estado |

**Entrega**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id(PK)** | Codigo | Fecha | Descripcion | Estado | Monto\_total | idAdministrativo(PK) | Id\_TipoEntrega(FK) |

**NotaVenta**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | Código | fecha\_emision | Monto\_total | estado | IdPedido(FK) | idEntrega(PK) |

**Pedido**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | Código | Fecha | Descripción | Estado | Monto\_total | idCliente(FK) |

**Almacen**

|  |  |
| --- | --- |
| **Id** | Codigo |

**Lote**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | Codigo | Fecha\_ingreso | Cantidad | Estado | idAlmacen(FK) | idProveedor(FK) |

**Producto**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | Código | Marca | Modelo | Precio | Costo | estado | idLote(FK) | idTipoProducto(FK) |

**Stock**

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | Cantidad | idStock(FK) | idAlmacen(FK) |

**DetallePedido**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cantidad | Descripcion | Total | idProducto(FK) | idPedido(FK) |

### 10.6. Diseño Físico

**Usuario**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | integer |  | No | Pk |  |
| 2 | user | Varchar |  |  |  |  |
| 3 | Password | varchar |  |  |  |  |
| 4 | Estado | varchar |  |  |  |  |
| 5 | idAdministrativo | Integer |  | No | Fk | Llave Foranea de Administrativo |

**TipoEntrega**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | integer |  | No | Pk |  |
| 2 | Descripcion | Varchar |  |  |  |  |

**Proveedor**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Serial |  | No | Pk |  |
| 2 | Nombre | Varchar |  |  |  |  |
| 3 | Telefono | varchar |  |  |  |  |
| 4 | Direccion | varchar |  |  |  |  |
| 5 | Codigo | varchar |  |  |  |  |
| 6 | Estado | varchar |  |  |  |  |

**TipoProducto**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Integer |  | No | Pk |  |
| 2 | Descripcion | Varchar |  |  |  |  |

**Cliente**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Integer |  | No | Pk |  |
| 2 | Codigo | Varchar |  |  |  |  |
| 3 | nit | Integer |  |  |  |  |
| 4 | nombre | Varchar |  |  |  |  |

**Administrativo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Serial |  | No | Pk |  |
| 2 | Nombre | Varchar |  |  |  |  |
| 3 | Codigo | varchar |  |  |  |  |
| 4 | Cargo | varchar |  |  |  |  |
| 5 | Telefono | varchar |  |  |  |  |
| 6 | Fecha\_Ingreso | integer |  |  |  |  |
| 7 | Estado | Varchar |  |  |  |  |

**Entrega**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Integer |  | No | Pk |  |
| 2 | Codigo | Varchar |  |  |  |  |
| 3 | Fecha\_registro | Timestamp |  |  |  |  |
| 4 | Fecha\_Entrega | Timestamp |  |  |  |  |
| 5 | Destino | Varchar |  |  |  |  |
| 6 | Estado | Varchar |  |  |  |  |
| 7 | idTipoEntrega | Integer |  | No | Fk | Llave foránea de tipo de Entrega |
| 8 | idAdministrativo | Integer |  | No | Fk | Llave foránea de administrativo |

**NotaVenta**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Integer |  | No | Pk |  |
| 2 | codigo |  |  |  |  |  |
| 3 | Fecha\_emision |  |  |  |  |  |
| 4 | Monto\_total |  |  |  |  |  |
| 5 | Estado |  |  |  |  |  |
| 6 | idPedido | Integer |  | No | Fk | Llave Foranea de Pedido |
| 7 | idEntrega | Integer |  | No | Fk | Lllave Foranea de Entrega |

**Pedido**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Integer |  | No | Pk |  |
| 2 | Codigo | varchar |  |  |  |  |
| 3 | Fecha | Fecha |  |  |  |  |
| 4 | Descripcion | Varchar |  |  |  |  |
| 5 | Estado | Varchar |  |  |  |  |
| 6 | Monto\_total | Decimal |  |  |  |  |
| 7 | idCliente | Integer |  | No | Fk | Llave Foranea de Cliente |

**Almacen**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Integer |  | No | Pk |  |
| 2 | Codigo | Varchar |  |  |  |  |

**Lote**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Integer |  | No | Pk |  |
| 2 | Codigo | varchar |  |  |  |  |
| 3 | Fecha\_ingreso | date |  |  |  |  |
| 4 | Cantidad | Integer |  |  |  |  |
| 5 | Estado | Varchar |  |  |  |  |
| 6 | idProveedor | Integer |  | No | Fk | Llave Foranea de Proveedor |
| 7 | idAlmacen | Integer |  | No | Fk | Lllave Foranea de Almacen |

**Producto**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Integer |  | No | Pk |  |
| 2 | Codigo | Varchar |  |  |  |  |
| 3 | Marca | Varchar |  |  |  |  |
| 4 | Modelo | Varchar |  |  |  |  |
| 5 | Precio | Decimal |  |  |  |  |
| 6 | Costo | Varchar |  |  |  |  |
| 7 | Estado | Varchar |  |  |  |  |
| 8 | idLote | Integer |  | No | Fk | Llave Foranea de Lote |
|  | idTipoProducto | Integer |  | No | Fk | Llave Foranea de TipoProducto |

**Stock**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Id | Integer |  | No | Pk |  |
| 2 | Cantidad | varchar |  | No |  |  |
| 3 | idProducto | integer |  | No | Fk | Llave Foranea de Producto |
| 4 | idAlmacen | Integer |  | No | Fk | Llave Foranea de Almacen |

**DetallePedido**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nro | Nombre de Campo | Tipo Dato | Longitud | Nulo? | Llave | Observación |
| 1 | Cantidad | Integer |  | No | Pk |  |
| 2 | Descripcion | Varchar |  |  |  |  |
| 3 | total | Decimal |  |  |  |  |
| 4 | idProducto | Integer |  | No | Fk | Llave Foarena de Producto |
| 5 | idPedido | Integer |  | No | Fk | Llave Foranea de Pedido |

### 10.7. Script

--

-- PostgreSQL database dump

--

-- Dumped from database version 9.5.9

-- Dumped by pg\_dump version 9.5.9

-- Started on 2019-07-08 06:22:19

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SET check\_function\_bodies = false;

SET client\_min\_messages = warning;

SET row\_security = off;

--

-- TOC entry 1 (class 3079 OID 12355)

-- Name: plpgsql; Type: EXTENSION; Schema: -; Owner:

--

CREATE EXTENSION IF NOT EXISTS plpgsql WITH SCHEMA pg\_catalog;

--

-- TOC entry 2293 (class 0 OID 0)

-- Dependencies: 1

-- Name: EXTENSION plpgsql; Type: COMMENT; Schema: -; Owner:

--

COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';

SET search\_path = public, pg\_catalog;

SET default\_tablespace = '';

SET default\_with\_oids = false;

--

-- TOC entry 188 (class 1259 OID 71054)

-- Name: administratives; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE administratives (

id integer NOT NULL,

name character varying(255) NOT NULL,

phone character varying(255) NOT NULL,

date\_admission date NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE administratives OWNER TO musci;

--

-- TOC entry 190 (class 1259 OID 71064)

-- Name: categories; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE categories (

id bigint NOT NULL,

description character varying(255) NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone

);

ALTER TABLE categories OWNER TO musci;

--

-- TOC entry 189 (class 1259 OID 71062)

-- Name: categories\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE categories\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE categories\_id\_seq OWNER TO musci;

--

-- TOC entry 2294 (class 0 OID 0)

-- Dependencies: 189

-- Name: categories\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE categories\_id\_seq OWNED BY categories.id;

--

-- TOC entry 186 (class 1259 OID 71038)

-- Name: clients; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE clients (

id integer NOT NULL,

name character varying(255) NOT NULL,

phone character varying(255) NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE clients OWNER TO musci;

--

-- TOC entry 200 (class 1259 OID 71135)

-- Name: deliveries; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE deliveries (

id bigint NOT NULL,

code character varying(255) NOT NULL,

register\_date date NOT NULL,

delivery\_date date,

destine character varying(255),

estado character varying(255) NOT NULL,

sale\_id bigint NOT NULL,

administrative\_id bigint,

deleted\_at timestamp(0) without time zone,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone

);

ALTER TABLE deliveries OWNER TO musci;

--

-- TOC entry 199 (class 1259 OID 71133)

-- Name: deliveries\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE deliveries\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE deliveries\_id\_seq OWNER TO musci;

--

-- TOC entry 2295 (class 0 OID 0)

-- Dependencies: 199

-- Name: deliveries\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE deliveries\_id\_seq OWNED BY deliveries.id;

--

-- TOC entry 204 (class 1259 OID 71167)

-- Name: inventory\_cuts; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE inventory\_cuts (

id bigint NOT NULL,

date date NOT NULL,

description character varying(255) NOT NULL,

deleted\_at timestamp(0) without time zone,

administrative\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone

);

ALTER TABLE inventory\_cuts OWNER TO musci;

--

-- TOC entry 203 (class 1259 OID 71165)

-- Name: inventory\_cuts\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE inventory\_cuts\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE inventory\_cuts\_id\_seq OWNER TO musci;

--

-- TOC entry 2296 (class 0 OID 0)

-- Dependencies: 203

-- Name: inventory\_cuts\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE inventory\_cuts\_id\_seq OWNED BY inventory\_cuts.id;

--

-- TOC entry 206 (class 1259 OID 71180)

-- Name: inventory\_details; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE inventory\_details (

id bigint NOT NULL,

previous\_stock integer NOT NULL,

new\_stock integer NOT NULL,

product\_id bigint NOT NULL,

inventory\_cuts\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE inventory\_details OWNER TO musci;

--

-- TOC entry 205 (class 1259 OID 71178)

-- Name: inventory\_details\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE inventory\_details\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE inventory\_details\_id\_seq OWNER TO musci;

--

-- TOC entry 2297 (class 0 OID 0)

-- Dependencies: 205

-- Name: inventory\_details\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE inventory\_details\_id\_seq OWNED BY inventory\_details.id;

--

-- TOC entry 182 (class 1259 OID 71009)

-- Name: migrations; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE migrations (

id integer NOT NULL,

migration character varying(255) NOT NULL,

batch integer NOT NULL

);

ALTER TABLE migrations OWNER TO musci;

--

-- TOC entry 181 (class 1259 OID 71007)

-- Name: migrations\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE migrations\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE migrations\_id\_seq OWNER TO musci;

--

-- TOC entry 2298 (class 0 OID 0)

-- Dependencies: 181

-- Name: migrations\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE migrations\_id\_seq OWNED BY migrations.id;

--

-- TOC entry 196 (class 1259 OID 71104)

-- Name: order\_\_details; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE order\_\_details (

id bigint NOT NULL,

description character varying(255) NOT NULL,

quantity integer NOT NULL,

subtotal numeric(8,2),

order\_id bigint NOT NULL,

product\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE order\_\_details OWNER TO musci;

--

-- TOC entry 195 (class 1259 OID 71102)

-- Name: order\_\_details\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE order\_\_details\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE order\_\_details\_id\_seq OWNER TO musci;

--

-- TOC entry 2299 (class 0 OID 0)

-- Dependencies: 195

-- Name: order\_\_details\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE order\_\_details\_id\_seq OWNED BY order\_\_details.id;

--

-- TOC entry 194 (class 1259 OID 71088)

-- Name: orders; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE orders (

id bigint NOT NULL,

code character varying(255) NOT NULL,

date date NOT NULL,

description character varying(255) NOT NULL,

total\_amount numeric(8,2),

client\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE orders OWNER TO musci;

--

-- TOC entry 193 (class 1259 OID 71086)

-- Name: orders\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE orders\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE orders\_id\_seq OWNER TO musci;

--

-- TOC entry 2300 (class 0 OID 0)

-- Dependencies: 193

-- Name: orders\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE orders\_id\_seq OWNED BY orders.id;

--

-- TOC entry 185 (class 1259 OID 71031)

-- Name: password\_resets; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE password\_resets (

email character varying(255) NOT NULL,

token character varying(255) NOT NULL,

created\_at timestamp(0) without time zone

);

ALTER TABLE password\_resets OWNER TO musci;

--

-- TOC entry 192 (class 1259 OID 71072)

-- Name: products; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE products (

id bigint NOT NULL,

code integer NOT NULL,

name character varying(255) NOT NULL,

brand character varying(255) NOT NULL,

model character varying(255) NOT NULL,

stock integer,

purchase\_price double precision NOT NULL,

sale\_cost double precision NOT NULL,

category\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE products OWNER TO musci;

--

-- TOC entry 191 (class 1259 OID 71070)

-- Name: products\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE products\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE products\_id\_seq OWNER TO musci;

--

-- TOC entry 2301 (class 0 OID 0)

-- Dependencies: 191

-- Name: products\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE products\_id\_seq OWNED BY products.id;

--

-- TOC entry 187 (class 1259 OID 71046)

-- Name: providers; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE providers (

id integer NOT NULL,

code character varying(255) NOT NULL,

name character varying(255) NOT NULL,

phone character varying(255) NOT NULL,

address character varying(255) NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE providers OWNER TO musci;

--

-- TOC entry 210 (class 1259 OID 71211)

-- Name: purchase\_details; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE purchase\_details (

id bigint NOT NULL,

description character varying(255),

quantity integer NOT NULL,

cost numeric(8,2) NOT NULL,

product\_id bigint NOT NULL,

purchase\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE purchase\_details OWNER TO musci;

--

-- TOC entry 209 (class 1259 OID 71209)

-- Name: purchase\_details\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE purchase\_details\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE purchase\_details\_id\_seq OWNER TO musci;

--

-- TOC entry 2302 (class 0 OID 0)

-- Dependencies: 209

-- Name: purchase\_details\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE purchase\_details\_id\_seq OWNED BY purchase\_details.id;

--

-- TOC entry 208 (class 1259 OID 71198)

-- Name: purchases; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE purchases (

id bigint NOT NULL,

code character varying(255) NOT NULL,

emission\_date date NOT NULL,

total\_cost numeric(8,2) NOT NULL,

provider\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE purchases OWNER TO musci;

--

-- TOC entry 207 (class 1259 OID 71196)

-- Name: purchases\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE purchases\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE purchases\_id\_seq OWNER TO musci;

--

-- TOC entry 2303 (class 0 OID 0)

-- Dependencies: 207

-- Name: purchases\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE purchases\_id\_seq OWNED BY purchases.id;

--

-- TOC entry 198 (class 1259 OID 71122)

-- Name: sales; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE sales (

id bigint NOT NULL,

code character varying(255) NOT NULL,

emission\_date date NOT NULL,

total\_amount numeric(8,2) NOT NULL,

order\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE sales OWNER TO musci;

--

-- TOC entry 197 (class 1259 OID 71120)

-- Name: sales\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE sales\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE sales\_id\_seq OWNER TO musci;

--

-- TOC entry 2304 (class 0 OID 0)

-- Dependencies: 197

-- Name: sales\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE sales\_id\_seq OWNED BY sales.id;

--

-- TOC entry 184 (class 1259 OID 71017)

-- Name: users; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE users (

id bigint NOT NULL,

email character varying(255) NOT NULL,

email\_verified\_at timestamp(0) without time zone,

password character varying(255) NOT NULL,

color character varying(255) DEFAULT 'primary'::character varying NOT NULL,

font\_size character varying(255) DEFAULT '15px'::character varying NOT NULL,

root character varying(255) DEFAULT '0'::character varying NOT NULL,

remember\_token character varying(100),

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

deleted\_at timestamp(0) without time zone

);

ALTER TABLE users OWNER TO musci;

--

-- TOC entry 183 (class 1259 OID 71015)

-- Name: users\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE users\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE users\_id\_seq OWNER TO musci;

--

-- TOC entry 2305 (class 0 OID 0)

-- Dependencies: 183

-- Name: users\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE users\_id\_seq OWNED BY users.id;

--

-- TOC entry 202 (class 1259 OID 71156)

-- Name: warehouses; Type: TABLE; Schema: public; Owner: musci

--

CREATE TABLE warehouses (

id bigint NOT NULL,

codigo character varying(255) NOT NULL,

nombre character varying(255) NOT NULL,

deleted\_at timestamp(0) without time zone,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone

);

ALTER TABLE warehouses OWNER TO musci;

--

-- TOC entry 201 (class 1259 OID 71154)

-- Name: warehouses\_id\_seq; Type: SEQUENCE; Schema: public; Owner: musci

--

CREATE SEQUENCE warehouses\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE warehouses\_id\_seq OWNER TO musci;

--

-- TOC entry 2306 (class 0 OID 0)

-- Dependencies: 201

-- Name: warehouses\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: musci

--

ALTER SEQUENCE warehouses\_id\_seq OWNED BY warehouses.id;

--

-- TOC entry 2083 (class 2604 OID 71067)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY categories ALTER COLUMN id SET DEFAULT nextval('categories\_id\_seq'::regclass);

--

-- TOC entry 2088 (class 2604 OID 71138)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY deliveries ALTER COLUMN id SET DEFAULT nextval('deliveries\_id\_seq'::regclass);

--

-- TOC entry 2090 (class 2604 OID 71170)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY inventory\_cuts ALTER COLUMN id SET DEFAULT nextval('inventory\_cuts\_id\_seq'::regclass);

--

-- TOC entry 2091 (class 2604 OID 71183)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY inventory\_details ALTER COLUMN id SET DEFAULT nextval('inventory\_details\_id\_seq'::regclass);

--

-- TOC entry 2078 (class 2604 OID 71012)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY migrations ALTER COLUMN id SET DEFAULT nextval('migrations\_id\_seq'::regclass);

--

-- TOC entry 2086 (class 2604 OID 71107)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY order\_\_details ALTER COLUMN id SET DEFAULT nextval('order\_\_details\_id\_seq'::regclass);

--

-- TOC entry 2085 (class 2604 OID 71091)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY orders ALTER COLUMN id SET DEFAULT nextval('orders\_id\_seq'::regclass);

--

-- TOC entry 2084 (class 2604 OID 71075)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY products ALTER COLUMN id SET DEFAULT nextval('products\_id\_seq'::regclass);

--

-- TOC entry 2093 (class 2604 OID 71214)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY purchase\_details ALTER COLUMN id SET DEFAULT nextval('purchase\_details\_id\_seq'::regclass);

--

-- TOC entry 2092 (class 2604 OID 71201)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY purchases ALTER COLUMN id SET DEFAULT nextval('purchases\_id\_seq'::regclass);

--

-- TOC entry 2087 (class 2604 OID 71125)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY sales ALTER COLUMN id SET DEFAULT nextval('sales\_id\_seq'::regclass);

--

-- TOC entry 2079 (class 2604 OID 71020)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY users ALTER COLUMN id SET DEFAULT nextval('users\_id\_seq'::regclass);

--

-- TOC entry 2089 (class 2604 OID 71159)

-- Name: id; Type: DEFAULT; Schema: public; Owner: musci

--

ALTER TABLE ONLY warehouses ALTER COLUMN id SET DEFAULT nextval('warehouses\_id\_seq'::regclass);

--

-- TOC entry 2263 (class 0 OID 71054)

-- Dependencies: 188

-- Data for Name: administratives; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY administratives (id, name, phone, date\_admission, created\_at, updated\_at, deleted\_at) FROM stdin;

1 Daniel Zeballos 78066791 2019-01-13 \N \N \N

\.

--

-- TOC entry 2265 (class 0 OID 71064)

-- Dependencies: 190

-- Data for Name: categories; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY categories (id, description, created\_at, updated\_at) FROM stdin;

1 impresora \N \N

2 cartuchos \N \N

3 accesorios \N \N

4 portatiles \N \N

\.

--

-- TOC entry 2307 (class 0 OID 0)

-- Dependencies: 189

-- Name: categories\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('categories\_id\_seq', 4, true);

--

-- TOC entry 2261 (class 0 OID 71038)

-- Dependencies: 186

-- Data for Name: clients; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY clients (id, name, phone, created\_at, updated\_at, deleted\_at) FROM stdin;

2 Alex Dominguez 72150495 \N \N \N

\.

--

-- TOC entry 2275 (class 0 OID 71135)

-- Dependencies: 200

-- Data for Name: deliveries; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY deliveries (id, code, register\_date, delivery\_date, destine, estado, sale\_id, administrative\_id, deleted\_at, created\_at, updated\_at) FROM stdin;

3 212 2019-07-05 2019-07-19 los lotes R 1 1 \N 2019-07-08 09:39:17 2019-07-08 09:42:09

\.

--

-- TOC entry 2308 (class 0 OID 0)

-- Dependencies: 199

-- Name: deliveries\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('deliveries\_id\_seq', 3, true);

--

-- TOC entry 2279 (class 0 OID 71167)

-- Dependencies: 204

-- Data for Name: inventory\_cuts; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY inventory\_cuts (id, date, description, deleted\_at, administrative\_id, created\_at, updated\_at) FROM stdin;

1 2019-07-09 inventory \N 1 2019-07-08 09:32:26 2019-07-08 09:32:26

\.

--

-- TOC entry 2309 (class 0 OID 0)

-- Dependencies: 203

-- Name: inventory\_cuts\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('inventory\_cuts\_id\_seq', 1, true);

--

-- TOC entry 2281 (class 0 OID 71180)

-- Dependencies: 206

-- Data for Name: inventory\_details; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY inventory\_details (id, previous\_stock, new\_stock, product\_id, inventory\_cuts\_id, created\_at, updated\_at, deleted\_at) FROM stdin;

\.

--

-- TOC entry 2310 (class 0 OID 0)

-- Dependencies: 205

-- Name: inventory\_details\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('inventory\_details\_id\_seq', 1, false);

--

-- TOC entry 2257 (class 0 OID 71009)

-- Dependencies: 182

-- Data for Name: migrations; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY migrations (id, migration, batch) FROM stdin;

1 2014\_10\_12\_000000\_create\_users\_table 1

2 2014\_10\_12\_100000\_create\_password\_resets\_table 1

3 2019\_07\_01\_034158\_create\_clients\_table 1

4 2019\_07\_01\_034249\_create\_providers\_table 1

5 2019\_07\_01\_034315\_create\_administratives\_table 1

6 2019\_07\_07\_040208\_create\_categories\_table 1

7 2019\_07\_07\_040209\_create\_products\_table 1

8 2019\_07\_07\_173125\_create\_orders\_table 1

9 2019\_07\_07\_174106\_create\_order\_\_details\_table 1

10 2019\_07\_07\_183235\_create\_sales\_table 1

11 2019\_07\_07\_195911\_create\_deliveries\_table 1

12 2019\_07\_08\_030855\_create\_warehouses\_table 1

13 2019\_07\_08\_031659\_create\_inventory\_cuts\_table 1

14 2019\_07\_08\_031810\_create\_inventory\_details\_table 1

15 2019\_07\_08\_065225\_create\_purchases\_table 1

16 2019\_07\_08\_065618\_create\_purchase\_details\_table 1

\.

--

-- TOC entry 2311 (class 0 OID 0)

-- Dependencies: 181

-- Name: migrations\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('migrations\_id\_seq', 16, true);

--

-- TOC entry 2271 (class 0 OID 71104)

-- Dependencies: 196

-- Data for Name: order\_\_details; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY order\_\_details (id, description, quantity, subtotal, order\_id, product\_id, created\_at, updated\_at, deleted\_at) FROM stdin;

1 nose que pedo 5 500.45 1 1 \N \N \N

2 nose que pedo 5 1525.45 2 2 \N \N \N

\.

--

-- TOC entry 2312 (class 0 OID 0)

-- Dependencies: 195

-- Name: order\_\_details\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('order\_\_details\_id\_seq', 2, true);

--

-- TOC entry 2269 (class 0 OID 71088)

-- Dependencies: 194

-- Data for Name: orders; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY orders (id, code, date, description, total\_amount, client\_id, created\_at, updated\_at, deleted\_at) FROM stdin;

1 565adsd 2019-07-20 mi primera orden 500.45 2 \N \N \N

2 456465assss 2019-07-10 mi segunda orden 1525.45 2 \N \N \N

\.

--

-- TOC entry 2313 (class 0 OID 0)

-- Dependencies: 193

-- Name: orders\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('orders\_id\_seq', 2, true);

--

-- TOC entry 2260 (class 0 OID 71031)

-- Dependencies: 185

-- Data for Name: password\_resets; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY password\_resets (email, token, created\_at) FROM stdin;

\.

--

-- TOC entry 2267 (class 0 OID 71072)

-- Dependencies: 192

-- Data for Name: products; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY products (id, code, name, brand, model, stock, purchase\_price, sale\_cost, category\_id, created\_at, updated\_at, deleted\_at) FROM stdin;

1 1121 impresora Epson 456px 50 45.200000000000003 50 1 \N 2019-07-08 09:43:47 \N

2 1231 tinta Epson Cartucho de tinta 41 20.300000000000001 25.5 2 \N 2019-07-08 09:43:47 \N

\.

--

-- TOC entry 2314 (class 0 OID 0)

-- Dependencies: 191

-- Name: products\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('products\_id\_seq', 2, true);

--

-- TOC entry 2262 (class 0 OID 71046)

-- Dependencies: 187

-- Data for Name: providers; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY providers (id, code, name, phone, address, created\_at, updated\_at, deleted\_at) FROM stdin;

3 20194 IMPRIMAYOR S.A. 72150495 Av. Cumavi N°16 \N \N \N

\.

--

-- TOC entry 2285 (class 0 OID 71211)

-- Dependencies: 210

-- Data for Name: purchase\_details; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY purchase\_details (id, description, quantity, cost, product\_id, purchase\_id, created\_at, updated\_at, deleted\_at) FROM stdin;

1 1221 50 880.00 1 1 2019-07-08 09:43:47 2019-07-08 09:43:47 \N

2 venta 41 50.00 2 1 2019-07-08 09:43:47 2019-07-08 09:43:47 \N

\.

--

-- TOC entry 2315 (class 0 OID 0)

-- Dependencies: 209

-- Name: purchase\_details\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('purchase\_details\_id\_seq', 2, true);

--

-- TOC entry 2283 (class 0 OID 71198)

-- Dependencies: 208

-- Data for Name: purchases; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY purchases (id, code, emission\_date, total\_cost, provider\_id, created\_at, updated\_at, deleted\_at) FROM stdin;

1 ci13 2019-07-11 50.00 3 2019-07-08 09:43:47 2019-07-08 09:43:47 \N

\.

--

-- TOC entry 2316 (class 0 OID 0)

-- Dependencies: 207

-- Name: purchases\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('purchases\_id\_seq', 1, true);

--

-- TOC entry 2273 (class 0 OID 71122)

-- Dependencies: 198

-- Data for Name: sales; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY sales (id, code, emission\_date, total\_amount, order\_id, created\_at, updated\_at, deleted\_at) FROM stdin;

1 4566 2019-01-20 500.45 1 \N \N \N

2 1212341 2019-07-20 1525.45 2 \N \N \N

\.

--

-- TOC entry 2317 (class 0 OID 0)

-- Dependencies: 197

-- Name: sales\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('sales\_id\_seq', 2, true);

--

-- TOC entry 2259 (class 0 OID 71017)

-- Dependencies: 184

-- Data for Name: users; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY users (id, email, email\_verified\_at, password, color, font\_size, root, remember\_token, created\_at, updated\_at, deleted\_at) FROM stdin;

1 daniel@gmail.com \N $2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi primary 15px 1 \N \N \N \N

2 alex@gmail.com \N $2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi primary 15px 0 \N \N \N \N

3 vinx@gmail.com \N $2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi primary 15px 0 \N \N \N \N

\.

--

-- TOC entry 2318 (class 0 OID 0)

-- Dependencies: 183

-- Name: users\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('users\_id\_seq', 3, true);

--

-- TOC entry 2277 (class 0 OID 71156)

-- Dependencies: 202

-- Data for Name: warehouses; Type: TABLE DATA; Schema: public; Owner: musci

--

COPY warehouses (id, codigo, nombre, deleted\_at, created\_at, updated\_at) FROM stdin;

\.

--

-- TOC entry 2319 (class 0 OID 0)

-- Dependencies: 201

-- Name: warehouses\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: musci

--

SELECT pg\_catalog.setval('warehouses\_id\_seq', 1, false);

--

-- TOC entry 2106 (class 2606 OID 71061)

-- Name: administratives\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY administratives

ADD CONSTRAINT administratives\_pkey PRIMARY KEY (id);

--

-- TOC entry 2108 (class 2606 OID 71069)

-- Name: categories\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY categories

ADD CONSTRAINT categories\_pkey PRIMARY KEY (id);

--

-- TOC entry 2102 (class 2606 OID 71045)

-- Name: clients\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY clients

ADD CONSTRAINT clients\_pkey PRIMARY KEY (id);

--

-- TOC entry 2118 (class 2606 OID 71143)

-- Name: deliveries\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY deliveries

ADD CONSTRAINT deliveries\_pkey PRIMARY KEY (id);

--

-- TOC entry 2122 (class 2606 OID 71172)

-- Name: inventory\_cuts\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY inventory\_cuts

ADD CONSTRAINT inventory\_cuts\_pkey PRIMARY KEY (id);

--

-- TOC entry 2124 (class 2606 OID 71185)

-- Name: inventory\_details\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY inventory\_details

ADD CONSTRAINT inventory\_details\_pkey PRIMARY KEY (id);

--

-- TOC entry 2095 (class 2606 OID 71014)

-- Name: migrations\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY migrations

ADD CONSTRAINT migrations\_pkey PRIMARY KEY (id);

--

-- TOC entry 2114 (class 2606 OID 71109)

-- Name: order\_\_details\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY order\_\_details

ADD CONSTRAINT order\_\_details\_pkey PRIMARY KEY (id);

--

-- TOC entry 2112 (class 2606 OID 71096)

-- Name: orders\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY orders

ADD CONSTRAINT orders\_pkey PRIMARY KEY (id);

--

-- TOC entry 2110 (class 2606 OID 71080)

-- Name: products\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY products

ADD CONSTRAINT products\_pkey PRIMARY KEY (id);

--

-- TOC entry 2104 (class 2606 OID 71053)

-- Name: providers\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY providers

ADD CONSTRAINT providers\_pkey PRIMARY KEY (id);

--

-- TOC entry 2128 (class 2606 OID 71216)

-- Name: purchase\_details\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY purchase\_details

ADD CONSTRAINT purchase\_details\_pkey PRIMARY KEY (id);

--

-- TOC entry 2126 (class 2606 OID 71203)

-- Name: purchases\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY purchases

ADD CONSTRAINT purchases\_pkey PRIMARY KEY (id);

--

-- TOC entry 2116 (class 2606 OID 71127)

-- Name: sales\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY sales

ADD CONSTRAINT sales\_pkey PRIMARY KEY (id);

--

-- TOC entry 2097 (class 2606 OID 71030)

-- Name: users\_email\_unique; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY users

ADD CONSTRAINT users\_email\_unique UNIQUE (email);

--

-- TOC entry 2099 (class 2606 OID 71028)

-- Name: users\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY users

ADD CONSTRAINT users\_pkey PRIMARY KEY (id);

--

-- TOC entry 2120 (class 2606 OID 71164)

-- Name: warehouses\_pkey; Type: CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY warehouses

ADD CONSTRAINT warehouses\_pkey PRIMARY KEY (id);

--

-- TOC entry 2100 (class 1259 OID 71037)

-- Name: password\_resets\_email\_index; Type: INDEX; Schema: public; Owner: musci

--

CREATE INDEX password\_resets\_email\_index ON password\_resets USING btree (email);

--

-- TOC entry 2135 (class 2606 OID 71149)

-- Name: deliveries\_administrative\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY deliveries

ADD CONSTRAINT deliveries\_administrative\_id\_foreign FOREIGN KEY (administrative\_id) REFERENCES administratives(id);

--

-- TOC entry 2134 (class 2606 OID 71144)

-- Name: deliveries\_sale\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY deliveries

ADD CONSTRAINT deliveries\_sale\_id\_foreign FOREIGN KEY (sale\_id) REFERENCES sales(id);

--

-- TOC entry 2136 (class 2606 OID 71173)

-- Name: inventory\_cuts\_administrative\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY inventory\_cuts

ADD CONSTRAINT inventory\_cuts\_administrative\_id\_foreign FOREIGN KEY (administrative\_id) REFERENCES administratives(id);

--

-- TOC entry 2138 (class 2606 OID 71191)

-- Name: inventory\_details\_inventory\_cuts\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY inventory\_details

ADD CONSTRAINT inventory\_details\_inventory\_cuts\_id\_foreign FOREIGN KEY (inventory\_cuts\_id) REFERENCES inventory\_cuts(id);

--

-- TOC entry 2137 (class 2606 OID 71186)

-- Name: inventory\_details\_product\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY inventory\_details

ADD CONSTRAINT inventory\_details\_product\_id\_foreign FOREIGN KEY (product\_id) REFERENCES products(id);

--

-- TOC entry 2131 (class 2606 OID 71110)

-- Name: order\_\_details\_order\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY order\_\_details

ADD CONSTRAINT order\_\_details\_order\_id\_foreign FOREIGN KEY (order\_id) REFERENCES orders(id);

--

-- TOC entry 2132 (class 2606 OID 71115)

-- Name: order\_\_details\_product\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY order\_\_details

ADD CONSTRAINT order\_\_details\_product\_id\_foreign FOREIGN KEY (product\_id) REFERENCES products(id);

--

-- TOC entry 2130 (class 2606 OID 71097)

-- Name: orders\_client\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY orders

ADD CONSTRAINT orders\_client\_id\_foreign FOREIGN KEY (client\_id) REFERENCES clients(id);

--

-- TOC entry 2129 (class 2606 OID 71081)

-- Name: products\_category\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY products

ADD CONSTRAINT products\_category\_id\_foreign FOREIGN KEY (category\_id) REFERENCES categories(id);

--

-- TOC entry 2140 (class 2606 OID 71217)

-- Name: purchase\_details\_product\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY purchase\_details

ADD CONSTRAINT purchase\_details\_product\_id\_foreign FOREIGN KEY (product\_id) REFERENCES products(id);

--

-- TOC entry 2141 (class 2606 OID 71222)

-- Name: purchase\_details\_purchase\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY purchase\_details

ADD CONSTRAINT purchase\_details\_purchase\_id\_foreign FOREIGN KEY (purchase\_id) REFERENCES purchases(id);

--

-- TOC entry 2139 (class 2606 OID 71204)

-- Name: purchases\_provider\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY purchases

ADD CONSTRAINT purchases\_provider\_id\_foreign FOREIGN KEY (provider\_id) REFERENCES providers(id);

--

-- TOC entry 2133 (class 2606 OID 71128)

-- Name: sales\_order\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: musci

--

ALTER TABLE ONLY sales

ADD CONSTRAINT sales\_order\_id\_foreign FOREIGN KEY (order\_id) REFERENCES orders(id);

--

-- TOC entry 2292 (class 0 OID 0)

-- Dependencies: 6

-- Name: public; Type: ACL; Schema: -; Owner: postgres

--

REVOKE ALL ON SCHEMA public FROM PUBLIC;

REVOKE ALL ON SCHEMA public FROM postgres;

GRANT ALL ON SCHEMA public TO postgres;

GRANT ALL ON SCHEMA public TO PUBLIC;

-- Completed on 2019-07-08 06:22:19

--

-- PostgreSQL database dump complete

--

# Capítulo 7: Implementación

## 11. Implementación de la arquitectura

### 11.1. Identificación de los componentes



**Ilustración 54** Diagrama de componentes.

## 12. Lenguaje de programación y Framework

### 12.1. PHP

Originalmente diseñado por el programador danés-canadiense Rasmus Lerdorf, en el año 1994 en base a la escritura de un grupo de CGI binarios escritos en el lenguaje C. En un comienzo, PHP sólo estaba compuesto por algunas macros que permitían trabajar más fácilmente en la creación de páginas web. En el año de 1995 Rasmus Lerdorf le añadió el analizador sintáctico y se llamó PHP/F1 Versión 2, sólo reconocía texto HTML y algunas directivas de mSQL. Después de esta fecha la contribución al código fue pública. PHP se caracteriza por ser un lenguaje gratuito y multiplataforma. Además de su posibilidad de acceso a muchos tipos de bases de datos, también es importante destacar su capacidad de crear páginas dinámicas, así como la posibilidad de separar el diseño del contenido de una web. PHP es la solución para la construcción de Webs con independencia de la Base de Datos y del servidor Web, válido para cualquier plataforma. (Red Grafica Latino Americana, 2010).

### 12.2. Laravel 5.8

Laravel es un Framework (metodología) de trabajo para PHP, también se puede decir que es una herramienta que cuenta con la metodología para realizar cualquier desarrollo de cualquier sistema que se requiera, ya que incorpora módulos que nos ayudaran a desarrollar de una mejor forma, y sobre todo de una manera estándar en donde cualquier otro desarrollador que conozca Laravel pueda colaborar en nuestros proyectos. (Andres, 2016).

Laravel 5.8 incluirá nuevas páginas de error con un diseño muy minimalista que pretende ser más adecuado para una variedad de sitios web y aplicaciones web sin necesidad de ser rediseñado para que se ajuste a un tema. Se pueden obtener los datos del usuario relacionado con una orden, pero no se puede obtener el perfil del usuario desde la orden porque no existe una relación directa entre las ordenes y los perfiles. Esta nueva relación nos permitirá crear una conexión entre los modelos sin tener una relación directa. (Fernandes, 2019).

## 13. Herramientas de desarrollo

### 13.1 Sistema operativo

La plataforma que soporta el software desarrollado es el S.O “Fedora 24”. Se eligió esta plataforma porque ofrece todas las herramientas necesarias para su implementación.

**Ventajas:**

* Es abierto.
* Es el que tiene más herramientas para desarrollar rapido.
* Se puede aprender intuitivamente.
* Tiene una gran cantidad de tutoriales.
* No viene con CD de instalación o de recuperación. El Starter posee una imagen en el disco rígido de la nueva computadora.

**Desventajas:**

* La compatibilidad de los comandos entre diferentes versiones.
* Conocimiento de comandos de la shell de nivel alto.
* Limitaciones de actualización de Hardware de acuerdo a sus controladores.

### 13.2. Sistema Gestor de Base de Datos

El Sistema Gestor de Base de Datos utilizado es PostgreSQL.

**Ventajas**:

* Su aparente facilidad.
* Buen rendimiento general.
* Compatibilidad, versatilidad.
* Es rápido y eficiente.
* Es gratuito.
* Importa base de datos, establece permisos de acceso.
* Es seguro.
* La construcción de las consultas es sencilla e intuitiva.

**Desventajas:**

* El mal uso.
* Tiempo de respuesta relativamente largo.

# Capítulo 8: Pruebas

## 14. Pruebas

### . Planificar pruebas

Con el propósito de asegurar el correcto funcionamiento, se ha desarrollado un plan de pruebas para la aplicación, estableciendo el alcance, diseño del software, objetivos, roles y recurrencia.

Para la realización de las pruebas a la aplicación se utilizará el método de prueba “caja negra”.

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisará por completo todos los requerimientos funcionales para un programa.

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

* Funciones incorrectas o faltantes.
* Errores de interfaz.
* Errores de la estructura de datos o en el acceso de la base de datos.
* Errores de comportamiento o rendimiento.
* Errores de inicialización y terminación.

Los casos de uso sobre los cuales se realizarán las respectivas pruebas son:

* Registrar Producto
* Registrar Pedido

### 14.2. Diseñar pruebas

#### 14.2.1 Registrar Producto

Se registrará un nuevo producto

**Entrada**

* Se colocan los datos del producto, asigna a una categoría.

**Resultado**

* Si se registró correctamente, el sistema habrá registrado dicho producto.

**Condiciones:**

* Ninguna

**Procedimiento de prueba para registrar nuevo producto:**

* Presionar el botón “Registrar Producto” del menú principal.
* Ingresar la información requerida.
* Presionar el botón “Confirmar”.

#### Registrar Pedido

Se realizara el registro de un pedido por parte del cliente.

**Entrada**

* Se ingresara los datos y detalles del pedido que se requiere de acuerdo al su cantidad.

**Resultado**

* Si se registró correctamente, el sistema habrá registrado dicho pedido.

**Condiciones**

* Stock en los productos.

**Procedimiento de prueba para aprobar registro de pedido:**

* Presionar la opción de “Registrar pedido”.
* Ingrese los datos requeridos.
* Presione guardar.

# Conclusiones

Al finalizar con este proyecto, se observó que se realizaron todas las actividades propuestas al inicio y las conclusiones a las que se llegaron son los siguientes:

* Se recopilo la información necesaria para definir los requisitos funcionales y no funcionales.
* El análisis de requisitos ayudó a entender la situación problemática.
* Se diseñaron interfaces para el uso fácil para el cliente.
* Se implementó el software de acuerdo con los distintos diseños.
* Se realizaron pruebas con los usuarios de la aplicación para garantizar un correcto funcionamiento.
* El software fue implementado satisfactoriamente de acuerdo con los requerimientos técnicos establecidos.

# Recomendaciones

El proyecto está terminado en su versión de prototipo, se recomienda realizar constantes pruebas para mejorar la integridad de este basándose en el documento presente el cual se describe los detalles del proceso de desarrollo del proyecto.

Se recomienda utilizar las siguientes recomendaciones:

• Instalar el servidor en un ambiente ventilado y seguro.

• Implementar políticas de privacidad dentro de la empresa sobre el manejo de recursos tecnológicos.

• Mantener los equipos limpios todo el tiempo, realizar un diagnóstico y una limpieza interna una vez al año.

• Instalar cámaras de seguridad en el ambiente del Servidor.

• Hacer ampliaciones de la funcionalidad del sistema a procesos y áreas, que no están incorporadas inicialmente.

• Realizar capacitación al personal sobre las funcionalidades.

# Bibliografía

Lamarca, M.J. (2013). *Aspectos de la Web.* Recuperado en http://www.hipertexto.info/documentos/web\_tecnolog.htm [2019, 1 de junio]

Ecured (2015). *Laravel.* Recuperado en [<https://www.ecured.cu/Laravel> [2019](http://eprints.uanl.mx/3619/1/SEGURIDAD_FISICA_PREVENCION_Y_DETECCION.pdf%20%5b2019), 1 de julio]

Ecured (2018). *PHP.* Recuperado en <https://www.ecured.cu/PHP> [2019, 1 de julio]

Red Grafica Latino Americana (2010). *Lenguaje de programación PHP.* Recuperado en [http://redgrafica.com/El-lenguaje-de-programacion-PHP [2019](http://redgrafica.com/El-lenguaje-de-programacion-PHP%20%5b2019), 4 de julio]

Andres. (2016). Introducción a Laravel desde Cero. Recuperado en <https://guiadev.com/introduccion-laravel/> [2019, 4 de julio]

Fernandes, C (2019). *Laravel 5.8.* Recuperado en <https://styde.net/novedades-en-laravel-5-8/> [2019, 5 de julio]

# Anexos



