

Elabora documentos técnicos y de usuario del software.

GA10-220501097-AA10-EV01

Juan David Martín Moreno

Aprendiz

Jairo Antonio Muñoz Arango

Instructor

Centro agropecuario La Granja

SENA Regional Tolima

Tecnólogo Análisis y Desarrollo de Software

Ficha 2675782

2024

Tabla de contenido

Introducción	4
Manual Técnico del Software	5
Sistema De Gestión De Ferretería Construmanta P.	5
Prerrequisitos de Instalación del Sistema.....	6
Hardware:.....	6
Software:.....	6
Frameworks y Estándares Utilizados	6
Diagrama de Casos de Uso del Sistema.....	7
Modelo Entidad-Relación de la Base de Datos.....	8
Diccionario de Datos.....	9
Diagrama de clases.....	12
Scripts de Instalación	13
Instalación de MySQL-Server:	13
Creación de la Base de Datos:	13
Diagrama de Componentes	18
Configuración del Backend.....	19
Instalar Node.js:	19
Clonar el repositorio y configurar el entorno:	19
Configuración del Frontend	19

Clonar el repositorio y configurar el entorno:	19
Conclusiones	20
Bibliografía.	21

Introducción

Este manual técnico detalla la instalación, configuración y mantenimiento del software de gestión de ferretería desarrollado para operar en un entorno Linux Ubuntu. Incluye la instalación de MySQL-Server y Node.js, así como la configuración del backend y frontend.

Manual Técnico del Software

Sistema De Gestión De Ferretería Construmanta P.

El producto "Sistema de Gestión de Ferretería" se enfoca en brindar mejoras significativas en la eficiencia operativa, la atención al cliente y la administración de inventario en una ferretería. Este sistema permite a los clientes solicitar cotizaciones, ser ayuda en las compras y alquilar herramientas de manera conveniente al poder ver que productos y herramientas están disponibles, además de facilitar la comunicación directa con el personal de la ferretería a través de un chat en línea. También ofrece ventajas en la seguridad de datos y mantenimiento eficiente, todo ello con el objetivo de mejorar la experiencia tanto para los clientes como para el personal de la ferretería.

Identificador del documento de especificación de requerimientos	RER-SGF-002
Identificador de análisis de viabilidad	AV-SGF-001
Fecha	2024-02-10
Proceso solicitante	Departamento de Ventas y Gestión de Ferretería
Solicitado por	Ferretería Construmanta P.
Proyecto	Sistema de Gestión de Ferretería
Módulo	N.A
Submódulo	N.A
Usuario Líder área o proceso	Juan David Martín Moreno
Coordinador Desarrollo FS	Juan David Martín Moreno
Profesional Funcional FS	Juan David Martín Moreno
Profesional de Producto FS	Juan David Martín Moreno

Prerrequisitos de Instalación del Sistema

Para poder instalar y posteriormente usar el sistema debemos cumplir algunos requisitos de hardware y software. En este caso se muestran los usados en las pruebas y despliegue local.

Hardware:

- CPU: Intel Core I3 5005U o similar como minimo.
- Memoria: 8GB RAM DDR3 a 1600MHz minimo.
- Disco Duro: 1TB HDD a 5200 RPM o con espacio suficiente para almacenar el sistema.

Software:

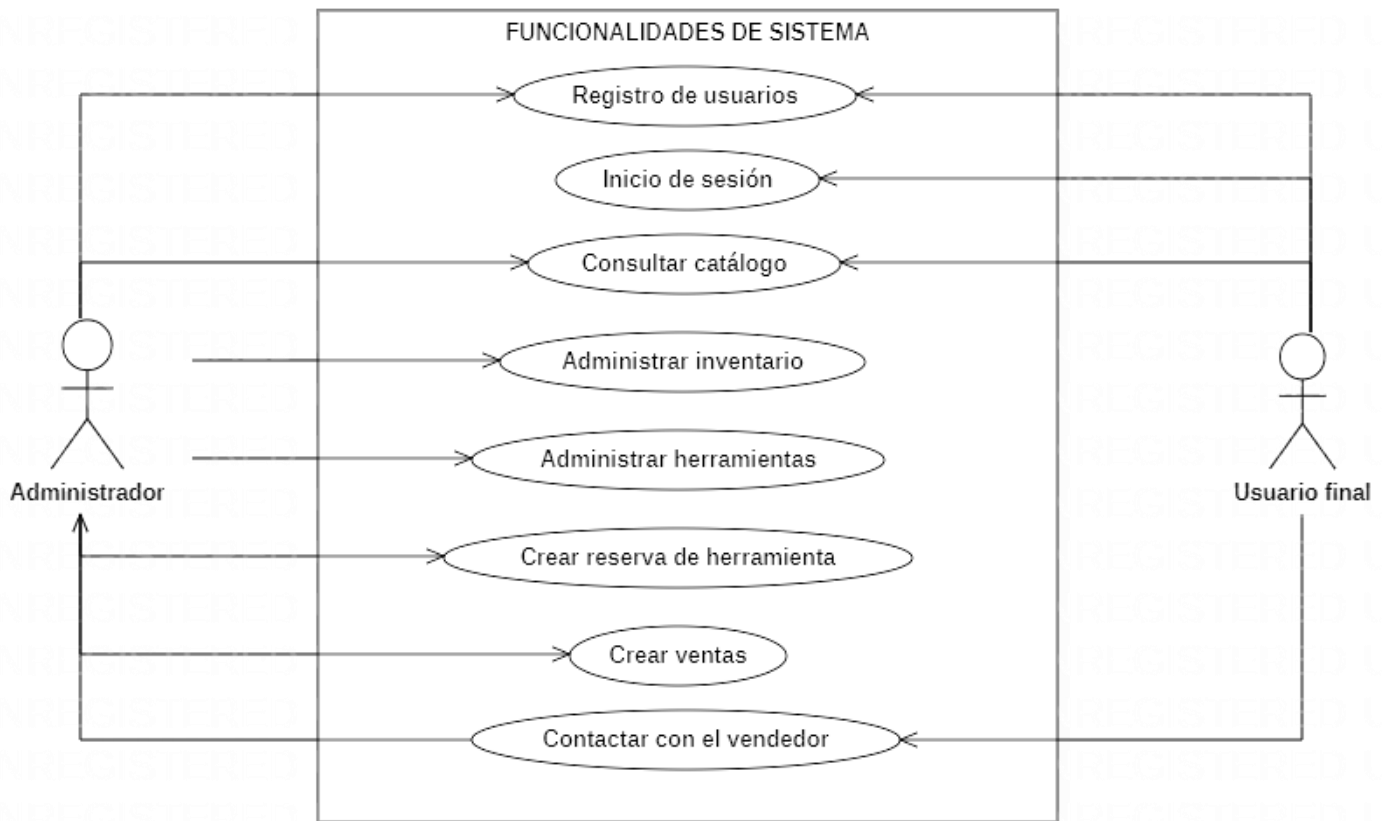
- Sistema Operativo: Linux Lubuntu

Frameworks y Estándares Utilizados

- Backend: Node.js con express.
- Base de Datos: MySQL
- Frontend: Node.js con React.

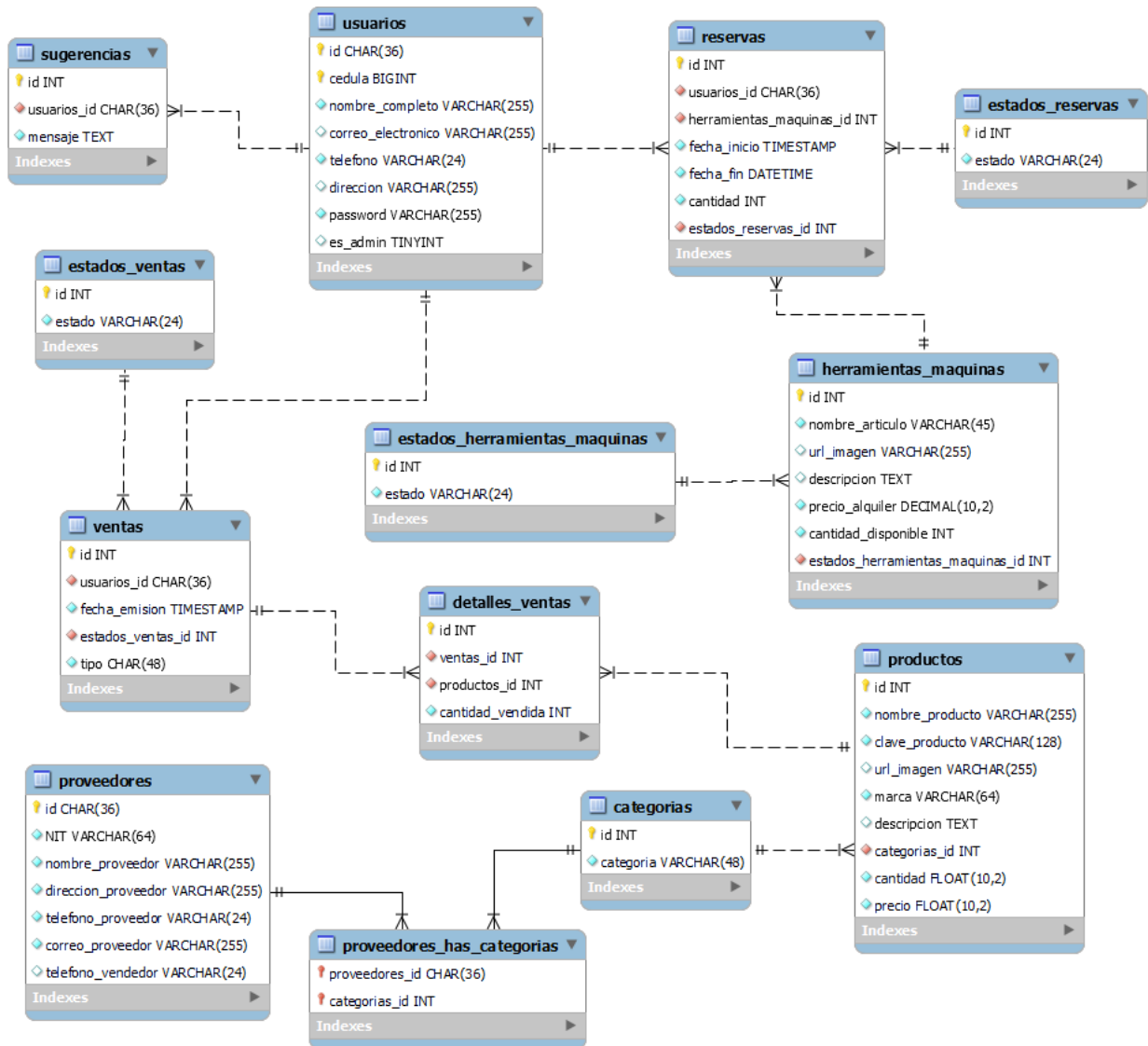
Diagrama de Casos de Uso del Sistema

Este es el diagrama de casos de uso del sistema donde se muestran los 2 tipos de usuarios con los que cuenta el sistema y como se relacionan con la funcionalidad del sistema.



Modelo Entidad-Relación de la Base de Datos

En este diagrama se muestra de forma grafica la base de datos, es decir, sus entidades y sus relaciones, tipos de datos, entre otros.



Diccionario de Datos

A continuación se muestra el diccionario de datos de la base de datos. En este se muestra la tabla, sus columnas, tipo de datos y la descripción de la columna.

usuarios

Nombre de la Columna	Tipo de Dato	Descripción
id	CHAR(36)	Identificador único de usuario
cedula	BIGINT	Número de cédula del usuario
nombre_completo	VARCHAR(255)	Nombre completo del usuario
correo_electronico	VARCHAR(255)	Correo electrónico del usuario
telefono	VARCHAR(24)	Número de teléfono del usuario
direccion	VARCHAR(255)	Dirección del usuario
password	VARCHAR(255)	Contraseña del usuario
es_admin	TINYINT	Indica si el usuario es administrador

sugerencias

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de sugerencia
usuarios_id	CHAR(36)	Identificador del usuario que sugiere
mensaje	TEXT	Mensaje de la sugerencia

estados_reservas

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de estado de reserva
estado	VARCHAR(24)	Estado de la reserva

estados_herramientas_maquinas

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de estado de herramienta o máquina
estado	VARCHAR(24)	Estado de la herramienta o máquina

herramientas_maquinas

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de herramienta o máquina
nombre_articulo	VARCHAR(45)	Nombre del artículo
url_imagen	VARCHAR(255)	URL de la imagen del artículo
descripcion	TEXT	Descripción del artículo
precio_alquiler	DECIMAL(10,2)	Precio de alquiler del artículo
cantidad_disponible	INT	Cantidad disponible del artículo
estados_herramientas_maquinas_id	INT	Identificador del estado del artículo

reservas

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de reserva
usuarios_id	CHAR(36)	Identificador del usuario que reserva
herramientas_maquinas_id	INT	Identificador de la herramienta o máquina reservada
fecha_inicio	TIMESTAMP	Fecha de inicio de la reserva
fecha_fin	DATETIME	Fecha de fin de la reserva
cantidad	INT	Cantidad reservada
estados_reservas_id	INT	Identificador del estado de la reserva

estados_ventas

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de estado de venta
estado	VARCHAR(24)	Estado de la venta

ventas

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de venta
usuarios_id	CHAR(36)	Identificador del usuario que realiza la venta
fecha_emision	TIMESTAMP	Fecha de emisión de la venta
estados_ventas_id	INT	Identificador del estado de la venta
tipo	CHAR(48)	Tipo de venta (por defecto 'FACTURA DE VENTA')

categorias

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de categoría
categoria	VARCHAR(48)	Nombre de la categoría

productos

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de producto
nombre_producto	VARCHAR(255)	Nombre del producto
clave_producto	VARCHAR(128)	Clave del producto
url_imagen	VARCHAR(255)	URL de la imagen del producto
marca	VARCHAR(64)	Marca del producto
descripcion	TEXT	Descripción del producto
categorias_id	INT	Identificador de la categoría del producto
cantidad	FLOAT(10,2)	Cantidad del producto
precio	FLOAT(10,2)	Precio del producto

detalles_ventas

Nombre de la Columna	Tipo de Dato	Descripción
id	INT	Identificador único de detalle de venta
ventas_id	INT	Identificador de la venta
productos_id	INT	Identificador del producto vendido
cantidad_vendida	INT	Cantidad vendida

proveedores

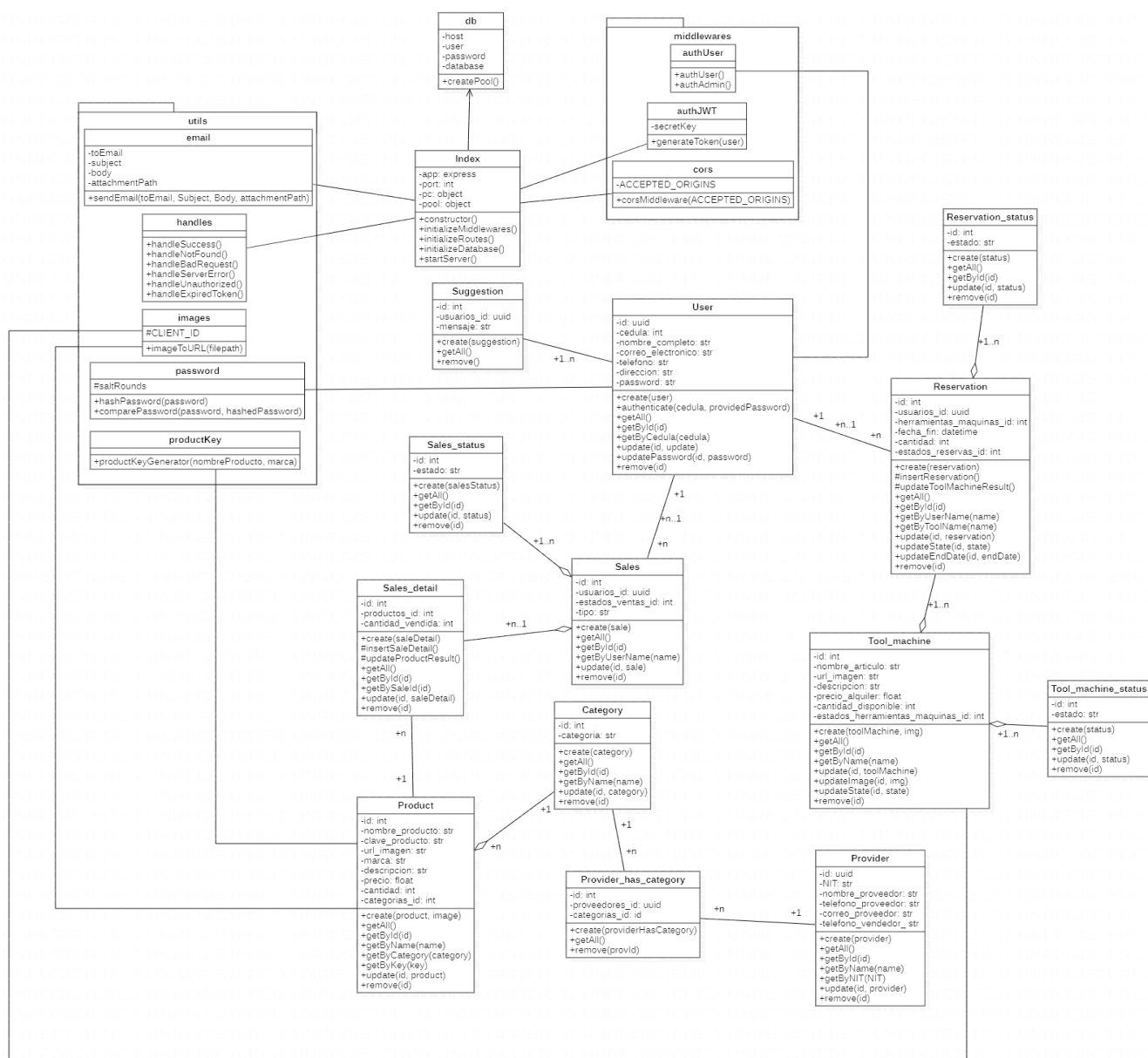
Nombre de la Columna	Tipo de Dato	Descripción
id	CHAR(36)	Identificador único de proveedor
NIT	VARCHAR(64)	NIT del proveedor
nombre_proveedor	VARCHAR(255)	Nombre del proveedor
direccion_proveedor	VARCHAR(255)	Dirección del proveedor
telefono_proveedor	VARCHAR(24)	Teléfono del proveedor
correo_proveedor	VARCHAR(255)	Correo del proveedor
telefono_vendedor	VARCHAR(24)	Teléfono del vendedor

proveedores_has_categorias

Nombre de la Columna	Tipo de Dato	Descripción
proveedores_id	CHAR(36)	Identificador del proveedor
categorias_id	INT	Identificador de la categoría

Diagrama de clases

Acá se muestra la forma en que está estructurado el backend del sistema, sus clases, atributos, métodos y relaciones.



Scripts de Instalación

Instalación de MySQL-Server:

```
sudo apt install mysql-server
sudo systemctl start mysql.service
sudo mysql_secure_installation
```

Creación de la Base de Datos:

```
-- MySQL Script generated by MySQL Workbench
-- Thu Jun  6 17:32:26 2024
-- Model: New Model      Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS = @@UNIQUE_CHECKS,
    UNIQUE_CHECKS = 0;
SET @OLD_FOREIGN_KEY_CHECKS = @@FOREIGN_KEY_CHECKS,
    FOREIGN_KEY_CHECKS = 0;
SET GLOBAL sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
SET @OLD_SQL_MODE = @@SQL_MODE,
    SQL_MODE =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

--
-- Schema ferreteria-construmanta-p
--
DROP SCHEMA IF EXISTS `ferreteria-construmanta-p` ;

--
-- Schema ferreteria-construmanta-p
--
CREATE SCHEMA IF NOT EXISTS `ferreteria-construmanta-p` DEFAULT CHARACTER SET utf8 ;
USE `ferreteria-construmanta-p` ;

--
-- Table `ferreteria-construmanta-p`.`usuarios`
--
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`usuarios` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`usuarios` (
  `id` CHAR(36) NOT NULL DEFAULT (UUID()),
  `cedula` BIGINT NOT NULL,
  `nombre_completo` VARCHAR(255) NOT NULL,
  `correo_electronico` VARCHAR(255) NULL DEFAULT NULL,
  `telefono` VARCHAR(24) NOT NULL,
  `direccion` VARCHAR(255) NULL DEFAULT NULL,
  `password` VARCHAR(255) NOT NULL,
  `es_admin` TINYINT NULL DEFAULT 0,
  PRIMARY KEY (`id`, `cedula`),
  UNIQUE INDEX `correo_electronico_UNIQUE` (`correo_electronico` ASC) VISIBLE,
  UNIQUE INDEX `telefono_UNIQUE` (`telefono` ASC) VISIBLE
) ENGINE = InnoDB;
```

```

-- Table `ferreteria-construmanta-p`.`sugerencias`
-----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`sugerencias` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`sugerencias` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `usuarios_id` CHAR(36) NOT NULL,
  `mensaje` TEXT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_sugerencias_usuarios1_idx` (`usuarios_id` ASC) VISIBLE,
  CONSTRAINT `fk_sugerencias_usuarios1`
    FOREIGN KEY (`usuarios_id`)
      REFERENCES `ferreteria-construmanta-p`.`usuarios` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE)
ENGINE = InnoDB;

-- Table `ferreteria-construmanta-p`.`estados_reservas`
-----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`estados_reservas` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`estados_reservas` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `estado` VARCHAR(24) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- Table `ferreteria-construmanta-p`.`estados_herramientas_maquinas`
-----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`estados_herramientas_maquinas` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`estados_herramientas_maquinas` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `estado` VARCHAR(24) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- Table `ferreteria-construmanta-p`.`herramientas_maquinas`
-----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`herramientas_maquinas` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`herramientas_maquinas` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `nombre_articulo` VARCHAR(45) NOT NULL,
  `url_imagen` VARCHAR(255) NULL DEFAULT NULL,
  `descripcion` TEXT NULL DEFAULT NULL,
  `precio_alquiler` DECIMAL(10,2) NOT NULL,
  `cantidad_disponible` INT NOT NULL,
  `estados_herramientas_maquinas_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_herramientas_maquinas_estados_herramientas_maquinas1_idx`
  (`estados_herramientas_maquinas_id` ASC) VISIBLE,
  CONSTRAINT `fk_herramientas_maquinas_estados_herramientas_maquinas1`
    FOREIGN KEY (`estados_herramientas_maquinas_id`)

```

```

REFERENCES `ferreteria-construmanta-p`.`estados_herramientas_maquinas` (`id`)
ON DELETE CASCADE
ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----
-- Table `ferreteria-construmanta-p`.`reservas`
-- -----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`reservas` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`reservas` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `usuarios_id` CHAR(36) NOT NULL,
  `herramientas_maquinas_id` INT NOT NULL,
  `fecha_inicio` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP(),
  `fecha_fin` DATETIME NOT NULL,
  `cantidad` INT NOT NULL,
  `estados_reservas_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_reservas_usuarios1_idx` (`usuarios_id` ASC) VISIBLE,
  INDEX `fk_reservas_herramientas_maquinas1_idx` (`herramientas_maquinas_id` ASC) VISIBLE,
  INDEX `fk_reservas_estados_reservas1_idx` (`estados_reservas_id` ASC) VISIBLE,
  CONSTRAINT `fk_reservas_usuarios1`
    FOREIGN KEY (`usuarios_id`)
      REFERENCES `ferreteria-construmanta-p`.`usuarios` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  CONSTRAINT `fk_reservas_herramientas_maquinas1`
    FOREIGN KEY (`herramientas_maquinas_id`)
      REFERENCES `ferreteria-construmanta-p`.`herramientas_maquinas` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  CONSTRAINT `fk_reservas_estados_reservas1`
    FOREIGN KEY (`estados_reservas_id`)
      REFERENCES `ferreteria-construmanta-p`.`estados_reservas` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----
-- Table `ferreteria-construmanta-p`.`estados_ventas`
-- -----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`estados_ventas` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`estados_ventas` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `estado` VARCHAR(24) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- -----
-- Table `ferreteria-construmanta-p`.`ventas`
-- -----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`ventas` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`ventas` (
  `id` INT NOT NULL AUTO_INCREMENT,

```

```

`usuarios_id` CHAR(36) NOT NULL,
`fecha_emision` TIMESTAMP NOT NULL DEFAULT (CURRENT_TIMESTAMP()),
`estados_ventas_id` INT NOT NULL,
`tipo` CHAR(48) NOT NULL DEFAULT 'FACTURA DE VENTA',
PRIMARY KEY (`id`),
INDEX `fk_ventas_usuarios1_idx` (`usuarios_id` ASC) VISIBLE,
INDEX `fk_ventas_estados_ventas1_idx` (`estados_ventas_id` ASC) VISIBLE,
CONSTRAINT `fk_ventas_usuarios1`
  FOREIGN KEY (`usuarios_id`)
    REFERENCES `ferreteria-construmanta-p`.`usuarios` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
CONSTRAINT `fk_ventas_estados_ventas1`
  FOREIGN KEY (`estados_ventas_id`)
    REFERENCES `ferreteria-construmanta-p`.`estados_ventas` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `ferreteria-construmanta-p`.`categorias`
-----

```

```
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`categorias`;
```

```

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`categorias` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `categoria` VARCHAR(48) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `categoria_UNIQUE` (`categoria` ASC) VISIBLE)
ENGINE = InnoDB;

```

```

-----
-- Table `ferreteria-construmanta-p`.`productos`
-----

```

```
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`productos`;
```

```

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`productos` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `nombre_producto` VARCHAR(255) NOT NULL,
  `clave_producto` VARCHAR(128) NOT NULL,
  `url_imagen` VARCHAR(255) NULL DEFAULT NULL,
  `marca` VARCHAR(64) NOT NULL,
  `descripcion` TEXT NULL DEFAULT NULL,
  `categorias_id` INT NOT NULL,
  `cantidad` FLOAT(10,2) NOT NULL,
  `precio` FLOAT(10,2) NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_productos_categorias1_idx` (`categorias_id` ASC) VISIBLE,
  CONSTRAINT `fk_productos_categorias1`
    FOREIGN KEY (`categorias_id`)
      REFERENCES `ferreteria-construmanta-p`.`categorias` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `ferreteria-construmanta-p`.`detalles_ventas`
-----

```



```

-----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`detalles_ventas` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`detalles_ventas` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `ventas_id` INT NOT NULL,
  `productos_id` INT NOT NULL,
  `cantidad_vendida` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_detalles_ventas_ventas1_idx` (`ventas_id` ASC) VISIBLE,
  INDEX `fk_detalles_ventas_productos1_idx` (`productos_id` ASC) VISIBLE,
  CONSTRAINT `fk_detalles_ventas_ventas1`
    FOREIGN KEY (`ventas_id`)
      REFERENCES `ferreteria-construmanta-p`.`ventas` (`id`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
  CONSTRAINT `fk_detalles_ventas_productos1`
    FOREIGN KEY (`productos_id`)
      REFERENCES `ferreteria-construmanta-p`.`productos` (`id`)
        ON DELETE CASCADE
        ON UPDATE CASCADE)
ENGINE = InnoDB;

-----
-- Table `ferreteria-construmanta-p`.`proveedores`
-----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`proveedores` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`proveedores` (
  `id` CHAR(36) NOT NULL DEFAULT (UUID()),
  `NIT` VARCHAR(64) NOT NULL,
  `nombre_proveedor` VARCHAR(255) NOT NULL,
  `direccion_proveedor` VARCHAR(255) NOT NULL,
  `telefono_proveedor` VARCHAR(24) NOT NULL,
  `correo_proveedor` VARCHAR(255) NOT NULL,
  `telefono_vendedor` VARCHAR(24) NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `NIT_UNIQUE` (`NIT` ASC) VISIBLE,
  UNIQUE INDEX `nombre_proveedor_UNIQUE` (`nombre_proveedor` ASC) VISIBLE,
  UNIQUE INDEX `correo_proveedor_UNIQUE` (`correo_proveedor` ASC) VISIBLE,
  UNIQUE INDEX `telefono_proveedor_UNIQUE` (`telefono_proveedor` ASC) VISIBLE)
ENGINE = InnoDB;

-----
-- Table `ferreteria-construmanta-p`.`proveedores_has_categorias`
-----
DROP TABLE IF EXISTS `ferreteria-construmanta-p`.`proveedores_has_categorias` ;

CREATE TABLE IF NOT EXISTS `ferreteria-construmanta-p`.`proveedores_has_categorias` (
  `proveedores_id` CHAR(36) NOT NULL,
  `categorias_id` INT NOT NULL,
  PRIMARY KEY (`proveedores_id`, `categorias_id`),
  INDEX `fk_proveedores_has_categorias_categorias1_idx` (`categorias_id` ASC) VISIBLE,
  INDEX `fk_proveedores_has_categorias_proveedores1_idx` (`proveedores_id` ASC) VISIBLE,
  CONSTRAINT `fk_proveedores_has_categorias_proveedores1`
    FOREIGN KEY (`proveedores_id`)
      REFERENCES `ferreteria-construmanta-p`.`proveedores` (`id`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,

```

```

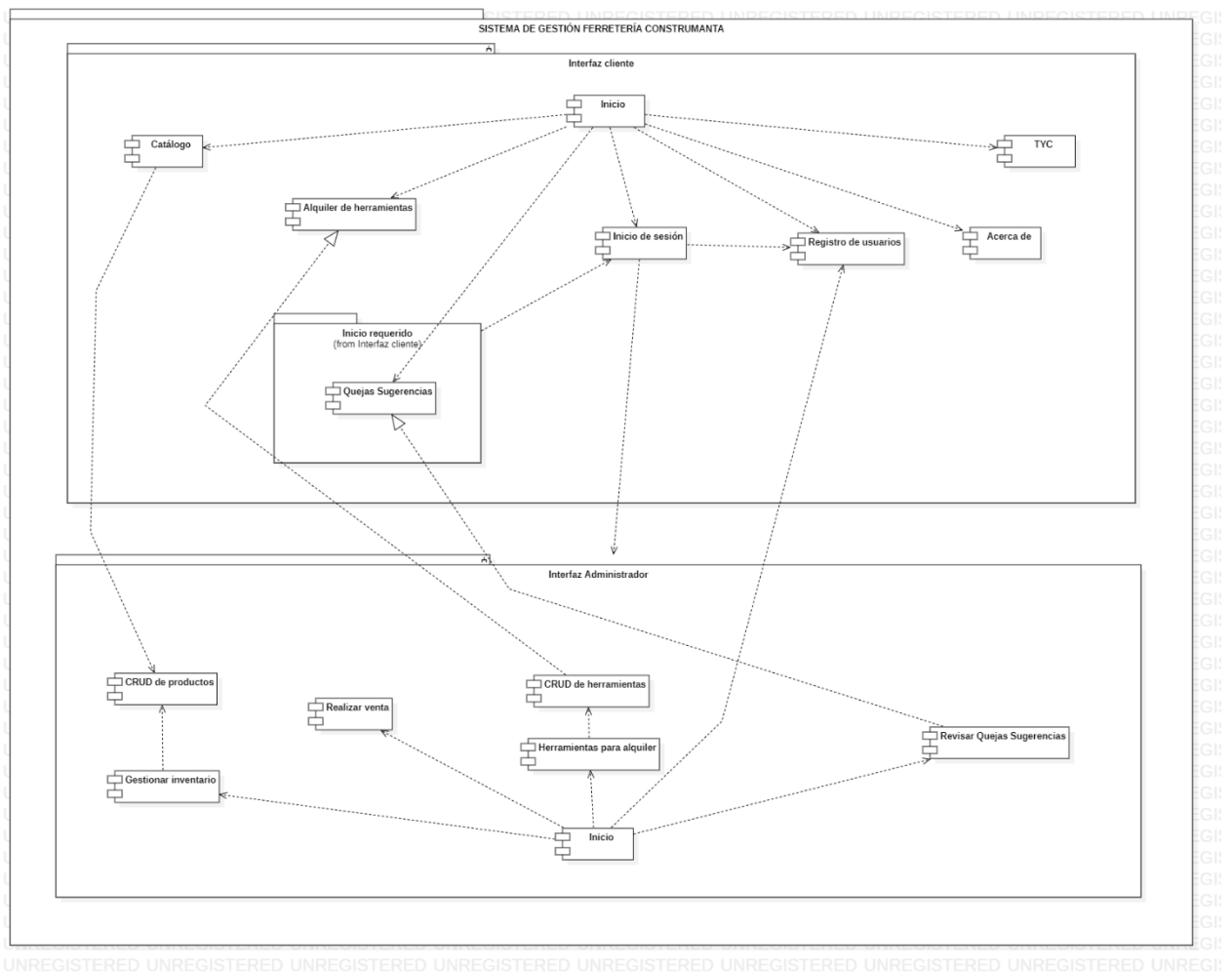
CONSTRAINT `fk_proveedores_has_categorias_categorias1`
  FOREIGN KEY (`categorias_id`)
  REFERENCES `ferreteria-construmanta-p`.`categorias` (`id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE)
ENGINE = InnoDB;

SET SQL_MODE = @OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS = @OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS = @OLD_UNIQUE_CHECKS;

```

Diagrama de Componentes

En este diagrama se muestra como está estructurada la lógica del proyecto, las 2 interfaces usadas y con que cuenta cada una. Como se relacionan los componentes.



Configuración del Backend

En este apartado se enseña el proceso de instalación del backend para la ejecución del servidor.

Instalar Node.js:

```
curl -fsSL https://fnm.vercel.app/install | bash
fnm install v20.10.0
node -v
```

Clonar el repositorio y configurar el entorno:

```
git clone https://github.com/judamar/FerreteriaConstrumantaP-backend
backend
cd backend
touch .env
nano .env
npm install
node ./src/index.js
```

Configuración del Frontend

En este apartado se enseña el proceso de instalación del frontend para la ejecución del servidor.

Clonar el repositorio y configurar el entorno:

```
git clone https://www.github.com/judamar/FerreteriaConstrumantaP-frontend frontend
cd frontend
touch .env
nano .env
npm install
npm run dev
```

Conclusiones

- El manual técnico es esencial para garantizar una instalación y configuración correctas del software, facilitando la gestión y el mantenimiento del sistema. Proporciona información detallada y estructurada sobre los requisitos del sistema, los procedimientos de instalación, los diagramas y la configuración necesaria, asegurando así una comprensión clara y precisa del funcionamiento del software para los administradores y técnicos responsables de su implementación y soporte.

Bibliografía.

Oracle Corporation. (2024). MySQL Workbench 8.0 Documentation. Recuperado de <https://dev.mysql.com/doc/workbench/en/>

Bitvise Limited. (2024). Bitvise SSH Client Documentation. Recuperado de <https://www.bitvise.com/ssh-client-documentation>

MKLabS. (2024). StarUML Documentation. Recuperado de <https://staruml.io/docs>

Actividades de documentación y entrega de software en procesos de implantación. (s. f.). Edu.co.

Recuperado 28 de junio de 2024, de

<https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF46/index.html#/>