

**INSTITUTO SUPERIOR TECNOLÓGICO**

**“SAN GABRIEL”**



**ESPECIALIDAD:** INFORMÁTICA MENCIÓN ANÁLISIS DE SISTEMAS

**TEMA:**

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INVENTARIO CON CATÁLOGO 3D PARA LA VENTA DE PASTILLAS DE FRENOS AUTOMOTRIZ EN LA EMPRESA AUTOFRENO RIOBAMBA ELABORADO CON EL LENGUAJE DE PROGRAMACIÓN PYTHON, FRAMEWORK DJANGO Y EL GESTOR DE BASE DE DATOS POSTGRESQL.

**TRABAJO DE INVESTIGACIÓN**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE:**

**TECNÓLOGO EN INFORMÁTICA MENCIÓN ANÁLISIS DE SISTEMAS**

**AUTOR:**

JOSÉ EDUARDO SEGOVIA LARREA

**TUTOR:**

ING. MARÍA FERNANDA PATIÑO

RIOBAMBA - ECUADOR

OCTUBRE, 2022

## **CERTIFICACIÓN**

Certifico que el Sr. **JOSÉ EDUARDO SEGOVIA LARREA**, con N° de Cédula 1600874471 ha elaborado bajo mi Asesoría el proyecto de Investigación titulado:

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INVENTARIO CON CATÁLOGO 3D PARA LA VENTA DE PASTILLAS DE FRENOS AUTOMOTRIZ EN LA EMPRESA AUTOFRENO RIOBAMBA ELABORADO CON EL LENGUAJE DE PROGRAMACIÓN PYTHON, FRAMEWORK DJANGO Y EL GESTOR DE BASE DE DATOS POSTGRESQL.**

Por lo tanto, autorizo la presentación para la calificación respectiva.

---

**Ing. Fernanda Patiño**  
**DOCENTE DEL ISTSGA**

“El presente Proyecto de Investigación constituye un requisito previo para la obtención del Título de **Tecnólogo en Informática Mención Análisis de Sistemas**.



“Yo, **José Eduardo Segovia Larrea** con N° de cédula 1600874471, declaro que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que se han llegado es de mi absoluta responsabilidad.”

---

José Eduardo Segovia Larrea

**INSTITUTO SUPERIOR TECNOLÓGICO**

**“SAN GABRIEL”**

**ESPECIALIDAD: INFORMÁTICA**

**TRABAJO DE INVESTIGACIÓN**

**PREVIA A LA OBTENCIÓN DEL TÍTULO DE:**

**TECNÓLOGO EN INFORMÁTICA MENCIÓN ANÁLISIS DE SISTEMAS**

**TÍTULO:**

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INVENTARIO CON CATÁLOGO 3D PARA LA VENTA DE PASTILLAS DE FRENOS AUTOMOTRIZ EN LA EMPRESA AUTOFRENO RIOBAMBA ELABORADO CON EL LENGUAJE DE PROGRAMACIÓN PYTHON, FRAMEWORK DJANGO Y EL GESTOR DE BASE DE DATOS POSTGRESQL.

**APROBADO:**

**ASESOR DE TESIS DE GRADO:** .....

**PRESIDENTE DEL TRIBUNAL:** .....

**MIEMBRO DEL TRIBUNAL:** .....

**MIEMBRO DEL TRIBUNAL:** .....

**FIRMAS DE RESPONSABILIDAD**

**NOMBRES**

**FECHA**

**FIRMA**

**DIRECTOR DE TESIS**

.....

.....

**PRESIDENTE DEL TRIBUNAL**

.....

.....

**MIEMBRO DEL TRIBUNAL**

.....

.....

**MIEMBRO DEL TRIBUNAL**

.....

.....

**NOTA FINAL DE LA TESIS**

.....

## **DEDICATORIA**

El presente trabajo de investigación, se lo dedico a mis padres quienes siempre me apoyaron incondicionalmente para esforzarme y concluir con esta meta.

Me han enseñado a ser la persona que soy hoy, mis principios, mis valores, mi perseverancia y empeño.

**JOSÉ**

## **AGRADECIMIENTO**

A mis padres y hermanos quienes supieron apoyarme para que pueda terminar con mi formación académica.

Agradezco a todos los docentes que, con su sabiduría, conocimiento y apoyo, motivaron a desarrollarme como persona y profesionalidad.

**JOSÉ**

## **ABREVIATURAS**

**GPL:** Licencia Pública General

**MIT:** Instituto Tecnológico de Massachusetts

**CERN:** Organización Europea para la investigación nuclear

**HTML:** Lenguaje de Marcado de Hipertexto

**XML:** Lenguaje de Marcado Extensible

**MVC:** Modelo de control de vistas

**SO:** Sistema Operativo

**BD:** Base de Datos

**SQL:** Lenguaje de Consultas Estructuradas

## GLOSARIO DE TÉRMINOS

**PSEUDOCÓDIGO:** Es una forma de escribir los pasos que va a realizar un programa de la forma más cercana al lenguaje de programación que vamos a utilizar posteriormente.

**FRAMEWORK:** Es una estructura de base utilizada como punto de partida para elaborar con objetivos específicos.

**DJANGO:** Es un framework de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles.

**DOCSTRING:** Las cadenas de documentación son textos que escriben entre triple comillas dentro de los programas para documentarlos.

**.NET:** Es una plataforma de aplicaciones que permite la creación y ejecución de servicios web y aplicaciones de internet.

**BYTECODES:** Es un código intermedio más abstracto que el código máquina, habitualmente es tratado como un archivo binario que contiene un programa ejecutable similar a un módulo objeto, que es un archivo binario producido por el compilador cuyo contenido es el código objeto o código máquina.

**HASH:** Es un algoritmo matemático que transforma cualquier bloque arbitrario de datos a una nueva serie de caracteres de longitud fija.

**OPEN SOURCE:** Es un código diseñado de manera que sea accesible al público: todos pueden ver, modificar y distribuir el código de la forma que consideren conveniente.

**HOSTING:** Es un servicio donde se puede alojar todos los contenidos necesarios para que una página web y correo sea accesible por cualquier usuario de internet en cualquier momento.

## ÍNDICE GENERAL

CERTIFICACIÓN .....	II
FIRMAS DE RESPONSABILIDAD.....	V
DEDICATORIA.....	VI
AGRADECIMIENTO.....	VII
ABREVIATURAS .....	VIII
GLOSARIO DE TÉRMINOS .....	IX
INTRODUCCIÓN .....	16
RESUMEN.....	18
SUMMARY .....	19
CAPÍTULO I.....	20
MARCO REFERENCIAL .....	20
1.1. Antecedentes del problema .....	20
1.2. Definición del problema.....	20
1.3. Justificación.....	21
1.4. Objetivos .....	21
1.4.1. Objetivo general .....	21
1.4.2. Objetivos específicos.....	21
CAPÍTULO II .....	22
MARCO TEÓRICO .....	22
2.1. Sistema de inventario .....	22
2.1.1. Tipos de inventarios .....	22
2.1.2. Beneficios de un sistema de inventario .....	22
2.2. Catálogo .....	23
2.2.1. Catálogo 3D.....	23

2.2.2. 3D .....	23
2.2.2.1. Usos del modelado 3D .....	23
2.2.2.2. Características del modelado 3D.....	24
2.2.2.3. Tipos de modelados.....	24
2.3. Lenguaje de programación .....	24
2.3.1. Programación .....	25
2.3.2. Lenguaje interpretado.....	25
2.3.3. Lenguaje compilado .....	25
2.3.4. Lenguajes intermediarios .....	25
2.3.5. Técnica .....	25
2.4. Python.....	26
2.4.1. Cultura de Python .....	26
2.4.2. Sintaxis .....	27
2.4.3. Librería estándar.....	27
2.4.4. Rendimiento .....	28
2.4.5. Documentación.....	28
2.4.6. Licencia .....	28
2.4.7. Python para la educación.....	29
2.4.8. Python para la ciencia.....	29
2.4.9. Empresas que usan Python .....	29
2.5. Framework .....	29
2.6. Django .....	29
2.7. PostgreSQL .....	30
2.7.1. Características .....	30
2.7.2. Ventajas.....	30
CAPÍTULO III .....	32

ANÁLISIS Y DISEÑO .....	32
3.1. Recopilación de información.....	32
3.2. Análisis.....	32
3.2.1. Análisis de factibilidad.....	32
3.2.2. Técnica .....	32
3.2.3. Operativa .....	33
3.2.4. Económica.....	33
3.2.5. Análisis de requerimientos .....	33
3.2.5.1. Requerimientos funcionales .....	33
3.2.5.2. Requerimientos no funcionales .....	35
3.2.6. Casos de usos .....	35
3.3. Diseño.....	37
3.3.1. Diseño conceptual .....	37
3.4. Diccionarios de datos .....	38
3.5. Diseño de interfaces .....	40
CAPÍTULO IV .....	43
IMPLEMENTACIÓN DE SISTEMA .....	43
4.1. Configuración de las herramientas de desarrollo .....	43
4.2. Arquitectura del sistema.....	44
4.3. Implementación del sistema .....	45
4.3.1. Definición de módulos del sistema .....	45
4.3.2. Desarrollo del sistema .....	45
4.3.3. Implementación del sistema .....	46
4.4. Pruebas .....	46
4.5. Capacitación al personal.....	46
4.6. Mantenimiento .....	47

CAPÍTULO V .....	48
CONCLUSIONES Y RECOMENDACIONES .....	48
CONCLUSIONES .....	48
RECOMENDACIONES .....	49
BIBLIOGRAFÍA.....	50
ANEXOS.....	53
MANUAL TÉCNICO .....	54
MANUAL DE USUARIO .....	69

## ÍNDICE DE TABLAS

Tabla 1 Componentes hardware .....	32
Tabla 2 Componentes Software.....	33
Tabla 3 Requerimientos funcionales .....	34
Tabla 4 Requerimientos no funcionales .....	35
Tabla 5 Tabla de usuario.....	38
Tabla 6 Tabla de proveedor .....	38
Tabla 7 Tabla de producto .....	39
Tabla 8 Tabla pastilla heredada de producto .....	39
Tabla 9 Tabla de catálogo.....	39
Tabla 10 Pruebas realizadas al sistema.....	46

## ÍNDICE DE FIGURAS

Figura 1 código C++ .....	27
Figura 2 código en Python .....	27
figura 3 Caso de usos administrador .....	36
Figura 4 Casos de usos usuario .....	36
Figura 5 Casos de usos cliente .....	36
Figura 6 Caso de uso modelado 3D .....	37
Figura 7 Modelo relacional .....	37
Figura 8 Página principal .....	40
Figura 9 Página de catálogo .....	40
Figura 10 Página de inicio de sesión.....	41
Figura 11 Página de registro de productos.....	41
Figura 12 Página de listado de productos .....	42
Figura 13 Página de listado de catálogo.....	42
Figura 14 Interfaz Atom.....	43
Figura 15 Interfaz de PostgreSQL .....	44
Figura 16 Módulos del sistema .....	45

## INTRODUCCIÓN

La base de toda empresa comercial es la compra y venta de bienes y servicios; de aquí nace la necesidad del manejo de un sistema por parte de la misma que le permita conocer la disponibilidad de sus recursos. El inventario tiene como propósito fundamental proveer a la empresa de una constante información resumida y analizada sobre sus recursos, para su continuo y regular desenvolvimiento, es decir, el inventario tiene un papel vital para el funcionamiento coherente dentro de las empresas y poder de esta forma responder a la demanda de forma eficiente.

El presente proyecto de investigación está dirigido para la empresa Autofreno Riobamba la cual no cuenta con un sistema automatizado de inventario por lo que se debe realizar este proceso de manera manual provocando retrasos al momento de realizar búsquedas del producto además de que al salir o ingresar mercancía se debe actualizar manual mente los inventarios teniendo pérdidas de tiempo y recursos también existe perdidas económicas ya que al momento de realizar pedidos de mercancía se adquiere producto repetido o innecesario.

Por lo que la implementación de un sistema web de inventario con catálogo 3D ayudara a la empresa Autofreno Riobamba a llevar de una manera correcta y eficaz su ingreso y salida del producto, para el desarrollo del inventario se utilizó un lenguaje de programación Python, framework Django y el gestor de base de datos PostgreSQL, teniendo como finalidad agilizar los procesos de inventario realizados por la empresa ayudando a mejorar su eficiencia y control de productos.

Al conocer los problemas de actualización de inventarios en la empresa Autofreno Riobamba y con ayuda de los gerentes se propuso la realización del proyecto de investigación de creación de inventario de un sistema web el cual se realizará con los conocimientos en el campo de la informática teniendo así los recursos y motivación necesaria para llevarlo a cabo.

El objetivo de la presente investigación es el diseño e implementación de un sistema de inventario con catálogo 3D para la venta de pastillas de frenos automotriz en la empresa Autofreno Riobamba.

Finalmente, este proyecto de investigación consta de 5 capítulos los cuales se detallan a continuación.

En el capítulo I se define el Marco Referencial, en él se describe y evalúa la problemática existente mediante el antecedente, definición del problema, así como los objetivos y la justificación para el desarrollo del proyecto.

En el capítulo II se desarrolla el Marco Teórico, en el que se definen los conceptos sobre las herramientas que se emplearán para el desarrollo del sistema, como lo son el lenguaje de programación Python, el framework Django, el gestor de base de datos PostgreSQL y el editor de código Atom, información que representa la investigación teórica.

En el capítulo III se establece el análisis y diseño del sistema de inventario, el análisis de factibilidad la cual se divide en Técnica, operativa y económica, de igual manera se analizan los requerimientos funcionales, no funcionales y el diseño de la BD completando de esta manera los requerimientos planteados en la fase anterior dando paso a los distintos diagramas y esquemas del sistema.

En el capítulo IV se realiza la implementación del sistema en el cual se define la configuración de las herramientas, la arquitectura del sistema, la definición de módulos y pruebas realizadas al sistema para verificar su correcto funcionamiento, confirmando de esta manera la seguridad de la base de datos, las conexiones al servidor y los datos.

En el capítulo V se encuentran las conclusiones y recomendaciones a las que se llegó en la investigación, de igual manera se encuentra el manual de usuario en el que se detalla cada una de las funciones que se podrán realizar en el sistema y el manual técnico en el que se detalla cómo está desarrollado cada uno de los módulos y funciones con los que cuenta el sistema.

## RESUMEN

El trabajo de investigación titulado diseño e implementación de un sistema de inventario con catálogo 3D para la venta de pastillas de frenos automotriz en la empresa Autofreno Riobamba, elaborado con el lenguaje de programación Python, framework Django y gestor de base de datos PostgreSQL se desarrolló en el periodo 2021. La metodología aplicada para el diseño e implementación del sistema de inventario con catálogo 3D fue FDD (*feature driven development*), en la fase: de desarrollo de un modelo general.- se realizaron entrevistas, se construyó un soporte de un primer modelo del sistema, permitiendo de esta manera identificar las necesidades y el alcance de la empresa; en la fase: construir una lista.- se identificaron las características del sistema, se agruparon y priorizaron; en la fase: planear.- se realizó el cronograma de diseño y se definió que se utilizará el gestor de base de datos PostgreSQL; en la fase: diseño de características.- se utilizará el lenguaje HTML 5 que permitió crear la interfaz gráfica, establecer los colores, textos, iconos y se complementó con Python para la creación de los formularios con ayuda del framework Django lo que permitió el desarrollo del FrontEnd y BackEnd y para la fase: construcción de las características.- se realizó una inspección del código y posteriormente se implementó alojándolo en un servidor web.

Los resultados con la implementación del sistema de inventario con catálogo 3D fueron: módulo de usuario, módulo de proveedor, módulo de productos, módulo de catálogo con sus respectivos formularios: formulario de registro de usuarios, formulario de proveedores, formulario de productos, formularios de catálogo y un método de búsqueda basado en filtros para la búsqueda de productos el cual tiene los siguientes parámetros: filtrado de productos por: código, ubicación, marca y vehículo, reporte de productos en formato xlsx con filtrado por código y ubicación, visualización del catálogo a través de la página web. Con lo que se logrará la reducción de tiempo al buscar un producto, un mejor control del inventario el cual permitirá conocer la cantidad de producto con el que se cuenta y su ubicación, reducción de gastos y mayor alcance de reconocimiento de la empresa.

## SUMMARY

The research work entitled design and implementation of an inventory system with 3D catalog for the sale of automotive brake pads in the company Autofreno Riobamba, developed with the Python programming language, Django framework and PostgreSQL database manager was developed in the period 2021. The methodology applied for the design and implementation of the inventory system with 3D catalog was FDD (feature driven development), in the phase: development of a general model - interviews were conducted, a support of a first model of the system was built, thus allowing to identify the needs and scope of the company; in the phase: build a list - the characteristics of the system were identified, grouped and prioritized; in the phase: plan - the timeline of the project was developed, and in the phase: plan - a schedule for the implementation of the system was developed; in the phase: plan - a schedule for the implementation of the system was developed, and in the phase: plan - a timeline for the implementation of the system was developed. the design chronogram was made and it was defined that the PostgreSQL database manager will be used; in the phase: design features.- the HTML 5 language will be used which allowed creating the graphic interface, establishing the colors, texts, icons and it was complemented with Python for the creation of the forms with the help of the Django framework which allowed the development of the FrontEnd and BackEnd and for the phase: construction of the features.- an inspection of the code was made and later it was implemented by hosting it in a web server.

The results with the implementation of the inventory system with 3D catalog were: user module, supplier module, product module, catalog module with their respective forms: user registration form, supplier form, product form, catalog forms and a search method based on filters for product search which has the following parameters: product filtering by: code, location, brand and vehicle, product report in xlsx format with filtering by code and location, catalog display through the web page. This will reduce time when searching for a product, improve inventory control, which will allow to know the amount of product available and its location, reduce costs and increase the company's recognition.

## **CAPÍTULO I**

### **MARCO REFERENCIAL**

#### **1.1. Antecedentes del problema**

La disponibilidad de nuevas herramientas para el desarrollo de aplicaciones ya sean de ámbito web o de escritorio, ha permitido que las empresas las adquieran para mejorar su desempeño, obteniendo un mejor posicionamiento y alcance frente a su competencia. El sistema de inventarios es de gran importancia ya que ayuda a mantener un balance en las existencias de un almacén y a estar al pendiente de los productos o artículos que tienen mayor demanda, puede reducir costos por que reflejan oportunamente lo que no tiene una rotación saludable y o que debe sustituirse lo más pronto posible, evitando retrasos en los pedidos o que se estropeen los productos por un almacenamiento deficiente o prolongado.

La empresa Autofreno Riobamba con el paso del tiempo ha incrementado la venta de pastillas de frenos, teniendo que almacenarlas en diferentes locaciones dentro de la empresa, lo cual tiene como consecuencia el desconocimiento de la cantidad existente de este producto.

En la empresa Autofreno Riobamba no cuenta con un sistema de inventario óptimo, demostrando que la falta de estos sistemas aumenta el tiempo empleado en la búsqueda de la disponibilidad del producto, así como también, existe el riesgo de pérdida de toda la información almacenada en la única computadora que existe en la empresa.

Con la implementación del sistema propuesto la empresa se beneficiará de varias maneras al poder centralizar toda su información en una base de datos, y dar a conocer los diferentes tipos y marcas de pastillas de frenos a través de internet.

#### **1.2. Definición del problema**

La empresa Autofreno Riobamba actualmente registra su inventario de hojas de papel las cuales pueden llegar a perderse o alterarse, ocasionando al momento de su transcripción ha hojas de Excel la información sea errónea, dificultando el rápido acceso y conocimiento de los productos con los que cuenta en ese momento, además del tiempo y personal que se emplea el conteo de su inventario. Teniendo como resultado el almacenamiento de producto existente por un mayor tiempo afectando su calidad.

### **1.3. Justificación**

La empresa Autofreno Riobamba al adquirir nuevas pastillas de frenos son almacenadas de forma directa en la bodega, contadas e inventariadas de forma manual por parte del personal encargado, lo que implica una gran inversión de tiempo y acumulación de papeles al llevar el registro de ingresos y salida en hojas que se pierden o traspapelan bajando de esta manera la calidad y tiempo de servicio. La información almacenada en la empresa se la realiza por medio de una hoja de Excel que es la transcripción de las hojas de ingreso que llenan los encargados, lo que implica un riesgo de daño o pérdida del archivo, considerando que los datos tomados por el personal se encuentren completos, de manera legible y clara; teniendo como consecuencia el desconocimiento de cantidades exactas del producto que se encuentra almacenado.

Al no tener un inventario actualizado, las pérdidas económicas y de tiempo son incalculables, ya que entre los problemas más frecuentes está que se ha realizado la adquisición de pastillas de frenos con existencia en bodega, deteniendo de esta manera el crecimiento de la empresa.

Con el desarrollo e implementación del presente proyecto se pretende ayudar a la empresa Autofreno Riobamba a mejorar la gestión de su inventario, teniendo como principal beneficio conocer en tiempo real, el lugar y la cantidad en stock de las pastillas de frenos reduciendo de esta manera el tiempo del personal que se empleaba en el conteo y búsqueda de este producto en los diferentes lugares de almacenamiento.

### **1.4. Objetivos**

#### **1.4.1. Objetivo general**

Implementar un sistema de inventario con catálogo 3D para la venta de pastillas de frenos automotriz en la empresa Autofreno Riobamba.

#### **1.4.2. Objetivos específicos**

- Investigar las características que posee el lenguaje de programación Python en el desarrollo de sistemas de catálogo 3D.
- Analizar los requisitos del sistema de inventario que se acoplen a los datos de la empresa Autofreno Riobamba.
- Desarrollar el sistema de inventario con catálogo 3D utilizando el lenguaje de programación Python, framework Django y gestor de base de datos PostgreSQL.
- Realizar pruebas con datos reales, resultados y reportes obtenidos en el sistema, después de ser implementado.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1. Sistema de inventario**

Es un sistema de control de los productos que están disponibles dentro de una empresa o en los diferentes puntos de almacenamiento, garantizando la correcta existencia de dichos productos (Corvo, 2019).

##### **2.1.1. Tipos de inventarios**

###### **a) Sistemas de inventarios perpetuo**

Con un sistema de inventario perpetuo se actualizan continuamente los registros de inventarios, con las adiciones y sustracciones cuando los artículos de inventarios son recibidos, vendidos o trasladados de una ubicación a otra. El sistema de inventarios perpetuo es el preferido para hacer seguimiento al inventario, porque brinda continuamente resultado precisos cuando se administra de forma adecuada. (Pontius, 2020)

###### **b) Sistemas de inventarios periódico**

Un sistema de inventarios periódicos permite que las empresas puedan conocer los niveles iniciales y finales de los inventarios según los diferentes periodos de tiempo, los cuales usan conteos físicos para poder rastrear el inventario. (Corvo, 2019)

##### **2.1.2. Beneficios de un sistema de inventario**

La administración y disposición son los principales factores que afectan en el desempeño y las ganancias que obtiene la empresa, por lo cual es de suma importancia para las compañías contar con un inventario bien gestionado y examinado. (Castro, 2014)

El objetivo es obtener un equilibrio entre oferta y demanda, el sistema de inventario cuenta con múltiples ventajas para la empresa al ofrecer información clara y oportuna en tiempo real que ayudara una mejor planeación y toma de decisiones adecuadas para ser más eficiente. (Castro, 2014)

#### **Beneficios**

- Vigilar la cantidad de productos
- Elevar el nivel de servicio al cliente
- Detectar artículos de lento movimiento
- Reducir costos

- Liberar y optimizar espacio en los almacenes

## **2.2. Catálogo**

Un catálogo se lo puede definir como un listado ordenado con algún criterio libros, documentos u objetos en venta de una persona, empresa o institución que generalmente contiene una descripción del objeto (Asale, 2020).

### **2.2.1. Catálogo 3D**

Un catálogo 3D se puede definir como la visualización de una lista de productos con imágenes con proporciones (alto, ancho y profundidad) permitiendo interactuar con dichas imágenes.

### **2.2.2. 3D**

Se hace alusión a un objeto que está conformado por ancho, altura y profundidad (longitud) de una imagen. Existen aspectos de la visión en 3D que no se pueden conseguir mediante la visión de imágenes planas o 2D, el primero de ellos es conocido como paralaje estéreo el cual consiste en dar la sensación de profundidad a partir de la vista de dos imágenes distintas de la misma escena, otro aspecto es el movimiento en paralaje, el cual consiste en que al movernos de la escena y cambiar el punto de vista tenemos diferentes vistas del mismo objeto. (Prieto, 2020)

#### **2.2.2.1. Usos del modelado 3D**

El modelado 3D se utiliza en una amplia gama de industrias.

**Cine y televisión.** – Los estudios de animación y efectos visuales utilizan programas de modelado para crear elementos y personajes generados por computadora para películas y programas de televisión. (Autodesk, 2022)

**Desarrollo de juegos.** – los desarrolladores de juegos crean entornos y personajes 3D para juegos y cinemáticas mediante herramientas de modelado. (Autodesk, 2022)

**Diseño de productos.** – la industria de manufactura maneja diferentes softwares de modelado 3D el cual es utilizado para la creación de productos, los cuales pueden llegar a ser utilizados en, la ingeniería y la renderización de conceptos. (Floriano, 2021)

**Arquitectura.** – La industria de la arquitectura emplea software de modelado 3D para la representación de los interiores y exteriores de viviendas, edificios y entornos ayudando a representar los proyectos propuestos. (Floriano, 2021)

### 2.2.2.2. Características del modelado 3D

El modelado 3D cuenta con varias características que abarcan todo el campo de aplicación del mismo. (grupokefren, 2021)

- Los diseños 3D o modelado tiene como característica la versatilidad para la visualización de objetos o proyectos de manera virtual, con la ayuda de la conexión a internet se puede llegar a compartir estos diseños o avances de proyectos además de poder acceder desde cualquier parte del mundo. (grupokefren, 2021)
- Ofrece a los profesionales de una empresa que se encuentren en diferentes departamentos de trabajo puedan tener acceso de manera continua en un proyecto de diseño para un mejor entendimiento. (Chávez, 2017)
- Es adaptable a los diferentes sistemas de unidades que se manejan según el sector o lugar de trabajo. (Chávez, 2017)
- El modelado 3d permite que los ingenieros o proyectos de ingeniería puedan ser analizados sin importar el tipo de estructura o pieza realizada sin necesidad de realizar ensayos que puedan llegar a ser destructivos. (grupokefren, 2021)

### 2.2.2.3. Tipos de modelados

**Sólidos.** – Estos modelados definen el volumen del objeto que representan por lo que llegan a ser realistas lo cual dificulta al momento de construirlos. Los modelos solidos normalmente se usan para simulaciones no visuales, como médicas y/o de ingeniería (Tokio, 2021)

**Modelado 3D paramétrico.** – Representa la evolución del modelo 3D de sólidos. (grupokefren, 2021)

**Modelado 3D orgánico.** – Este tipo de modelado llegan a ser más versátiles además de llegar a ser los más empleados en la actualidad. Permitiendo representar diferentes entornos virtuales además de objetos existentes en la realidad como objetos orgánicos o más conocidos como animales, personas, vegetación. (González, 2022)

## 2.3. Lenguaje de programación

Es cualquier lenguaje artificial que nos permita formular ordenes formadas por símbolos, reglas semánticas y sintácticas que especifican su estructura y el significado de sus elementos y expresiones, las cuales podrán ser ejecutadas por maquinas (Ceballos, 2022).

### **2.3.1. Programación**

El termino programación se puede detallar como la creación de programas de computación, los cuales buscan facilitar actividades que requieren una gran cantidad de tiempo (Ceballos, 2022).

### **2.3.2. Lenguaje interpretado**

El lenguaje interpretado se debe traducir para que el procesador pueda llegar a interpretar ya que es diferente al lenguaje de las maquinas (Correa, 2015).

### **2.3.3. Lenguaje compilado**

Un programa escrito en un lenguaje compilado estos programas llegan a ser traducidos por otro programa el cual es conocido como compilador este es un programa anexo el cual es el encargado de crear nuevos archivos sin necesidad de que algún otro programa para ejecutarse. (Correa, 2015)

### **2.3.4. Lenguajes intermediarios**

Algunos lenguajes pertenecen a ambas categorías dado que el programa escrito en estos lenguajes puede soportar una fase de compilación intermedia, en un archivo escrito en un lenguaje confuso y no ejecutable (López, 2021).

### **2.3.5. Técnica**

En programación existen diferentes técnicas que se emplean para obtener un mejor resultado en el desarrollo de un sistema (López, 2021).

#### **a) Corrección**

Si los programas realizan sus funciones de manera correcta y como se establece al momento de realizar su desarrollo por lo que se debe tener en cuenta que es lo que se busca al momento de desarrollar el programa.

#### **b) Claridad**

Los programas deben llegar a ser legibles y llegar a ser claros al momento de realizarlos para que así lleguen a facilitar el desarrollo además de sus futuros mantenimientos.

#### **c) Eficiencia**

Los programas deben realizar de manera correcta los pasos o los objetivos para lo cual fue realizado y además debe llegar a gestionar los recursos que llegue a utilizar de manera asertiva.

#### **d) Portabilidad**

La portabilidad de los programas es cuando son capaces de poder ejecutarse en diferentes plataformas ya sea en hardware o software sin importar en su lugar de elaboración.

### **2.4. Python**

Es un lenguaje de programación de alto nivel que incluye algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, permitiendo realizar algunas tareas complejas con menos líneas de código y legibles (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

#### **2.4.1. Cultura de Python**

Tim Peters un veterano en Python, analiza los principios de diseño de la BDFI para el diseño de Python en 20 aforismos que se deberían seguir cuando se codifica código en Python: (Peters, 2019).

- Hermoso es mejor que feo
- Explicito es mejor que implícito
- Simple es mejor que complejo
- Plano es mejor que anidado
- Disperso es mejor que denso
- El código legible cuenta
- Casos especiales no son lo suficientemente especiales para romper las reglas
- Casi siempre lo practico vence a lo formal
- Los errores nunca deben pasar desapercibidos, a menos que se especifique este comportamiento
- Ante una ambigüedad, descarte la tentación a adivinar
- Debe haber una, y preferentemente una sola, forma lógica de obtener un resultado, así no se alcance a apreciar los resultados finales a menos que seas un genio.
- Nunca es mejor que ahora, pero la mayoría de veces es mejora el ahora que el nunca.
- Si la aplicación de lo que se necesita tiene una difícil explicación, entonces es una mala idea

- Si la aplicación de lo que se necesita tiene una fácil explicación, entonces pudiera ser una buena idea

### 2.4.2. Sintaxis

La sintaxis de Python es muy sencilla tanto que en algunas ocasiones parase pseudocódigo, es interesante observar las diferencias que existen entre el programa Hola Mundo de Python y el de otro lenguaje de alto nivel como C++ (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

**Figura 1** código C++

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World" <<
endl;
    return 0;
}
```

Fuente: (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014)

Y ahora se ve cómo logra esta misma tarea en Python

**Figura 2** código en Python

```
print "Hello World"
```

Fuente: (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014)

### 2.4.3. Librería estándar

Una de las fortalezas de Python y quizás la mayor, es la librería estándar con la que cuenta. Con decenas de módulos cubre la mayoría de las necesidades básicas de un programador, en esta se le da cobertura de forma muy intuitiva a tópicos como (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

- Cadenas
- Estructuras de datos
- Funciones numéricas y matemáticas

- Compresión de datos
- Formatos de archivo
- Criptografía
- Servicios de los sistemas operativos
- Comunicación entre procesos
- Manejo de datos de internet
- Servicio multimedia
- Manejo de excepción

#### **2.4.4. Rendimiento**

Python a diferencia de otros lenguajes interpretados ha implementado toda su librería en el lenguaje C, lo que hace que sus funciones primitivas sean bastante eficientes, además puede compilarse su código a bytecodes, similar al que emplea .Net y Java, optimizando aún más la interpretación (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

#### **2.4.5. Documentación**

Python incluye un sistema para documentar, todas las funciones y clases, poseen una variable expresa que se llama doc conocida como docstring o cadena de documentación. Donde se define el uso de esa función o clases y los parámetros que median en detalle. La sintaxis de estos docstring es muy parecida a la de JavaDoc (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

#### **2.4.6. Licencia**

Python cuenta con su propia licencia llamada Python licence certificada por la ideología código abierto y es compatible con la GPL (GNU Public Licence) de la Free Software Foundation (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

La principal diferencia es que no tiene una restricción copyleft, acercándola más al movimiento OpenSource. Permitiendo codificar programas que se podrán comerciar libremente sin entregar su código fuente. Esto significa que puede usarse tanto para hacer software libre como privado (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

### **2.4.7. Python para la educación**

Fue pensado principalmente como un lenguaje fácil de aprender, su creador ha señalado que la programación se convertirá en una asignatura sustancial como la matemática y la física en la educación media (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

Universidades como la prestigiosa MIT han escogido este lenguaje para realizar cursos de programación e introducción a los algoritmos, la universidad de Chicago lo utiliza para impartir ciencias de la computación 2 y la universidad de Jaume de España para la introducción a la programación (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

### **2.4.8. Python para la ciencia**

Los científicos usan la programación como herramienta para solucionar problemas, requiriendo que las herramientas sean simples y eficientes para que puedan concentrarse en los problemas de sus especialidades (Rossum, 2018).

Python es usado en grandes centros de investigación como el CERN y por científicos en áreas como la bioinformática, neurofisiología, físicas, matemáticas (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

### **2.4.9. Empresas que usan Python**

Empresas como Google han recurrido a Python para elaborar algoritmos de búsqueda para la web. Otras empresas como Lucasfilm lo usa para la fabricación de los motores de representado en 3D para películas (Challenger Pérez, Becerra Garcia, & Díaz Ricardo, 2014).

## **2.5. Framework**

Un framework es un conjunto de componentes que ayudan a desarrollar sitios web de una manera más rápida y eficiente (Muenta, 2020).

## **2.6. Django**

Django nació de programas codificados por un grupo de programadores Web en Lawrence, Kansas. En 2003, los desarrolladores Web del diario Lawrence Journal World, Adrian Holovaty y Simon Willson, usaron Python para escribir sus softwares. En el verano boreal de 2005, luego de desarrollar este framework logrando que gran parte de las páginas World Online lo utilizaran, que contaba con Jacob Kaplan-Moss, se concluyó distribuir el framework como Open Source,

fue liberado en julio de 2005 y lo nombraron Django, por el guitarrista de jazz Django Reinhardt. (Holovaty & Kaplan Moss, 2008)

Resaltando cómo los principios de Django han formado la cultura de su sociedad de código abierto. Considerando que Django fue generado de código existente, no como un ejemplo educativo, está específicamente orientado a solucionar fallos de desarrollo Web encontrados por los programadores de Django y con los que continúan encontrándose. Los creadores del framework tienen un gran interés en disminuir el tiempo a los desarrolladores. (Holovaty & Kaplan Moss, 2008)

## **2.7. PostgreSQL**

Es un potente gestor de datos relacional de objetos de código libre que usa y amplía el lenguaje SQL (PostgreSQL, 2020).

Es conocido actualmente como uno de los administradores de base de datos relacionales más potente del mercado, presenta fácil accesibilidad, es multiplataforma y está disponible para ser utilizado en casi todos los sistemas operativos en la actualidad sin reducir su rendimiento. Igualmente, no fue hasta después de mediados de los años 90's cuando en ese momento llamado "Postgres95", es liberado para el desarrollo en formato abierto, de esta manera varias personas se unieron al proyecto para mejorar y estabilizar el sistema, añadirle funciones y el soporte al lenguaje SQL. (hostingpedia, 2019)

En 1996, cambia su nombre de manera oficial para reflejar las características del lenguaje SQL, lanzando oficialmente la primera versión con el nombre de "PostgreSQL" el cual se mantiene hasta la actualidad. (hostingpedia, 2019)

### **2.7.1. Características**

PostgreSQL cuenta con una variedad de características entre las cuales están (PostgreSQL, 2020):

- Fiabilidad, recuperación ante desastres
- Seguridad
- Extensibilidad

### **2.7.2. Ventajas**

**Instalación y uso gratuito:** PostgreSQL es un gestor de datos de código abierto y gratuito, permitiendo utilizarlo las veces requeridas (hostingpedia, 2019)

**Sistema disponible multiplataforma:** Es compatible con prácticamente todas las tecnologías y SO de la actualidad (hostingpedia, 2019).

**Estabilidad:** Está en constante desarrollo y soporte de versiones, cuenta con una aplicación utilizable cuando los servidores están suspendidos o en mantenimiento permitiendo a los usuarios ingresar en modo de solo lectura si bloquear completamente el sistema (hostingpedia, 2019).

**Escalabilidad y configuración:** Permite configurar por separado los recursos con los que cuenta una computadora, modificando la numeración del CPU y cantidad de RAM disponible para funcionar correctamente (hostingpedia, 2019).

**Herramienta gráfica:** Dispone de una interfaz gráfica para la gestión de la base de datos de manera fácil e intuitiva (hostingpedia, 2019).

**Soporte y ayuda:** Aunque no dispone de soporte técnico, se encuentran una gran cantidad de páginas para nuestra ayuda, contando con una de las comunidades más activas (hostingpedia, 2019).

## CAPÍTULO III

### ANÁLISIS Y DISEÑO

#### 3.1. Recopilación de información

La empresa Autofreno Riobamba se encuentra en la necesidad de tener un control adecuado de su inventario de pastillas de freno, los cuales en la actualidad lo han estado realizando de forma manual dificultando de esta manera conocer la cantidad exacta de su inventario.

Con la implementación del sistema de inventario con catálogo 3D se da solución gracias al uso de herramientas tecnológicas las mismas que cubren las necesidades de la empresa de forma ágil y sencilla.

Las principales funciones del sistema de inventario con catálogo 3D da solución a los siguientes aspectos:

- Permite al usuario registrar sus productos de forma rápida y sencilla.
- Permite promocionar sus productos por medio de un catálogo.
- Permite realizar el seguimiento de su inventario y obtener reportes.
- Permite conocer su inventario desde distintos dispositivos.

#### 3.2. Análisis

##### 3.2.1. Análisis de factibilidad

##### 3.2.2. Técnica

La empresa Autofreno Riobamba cuenta con el equipo software y hardware para el desarrollo e implementación del sistema de inventario con catálogo 3D, todas las herramientas son opensource. De esta manera se concluye que es factible el desarrollo del proyecto.

#### **HARDWARE:**

Los recursos hardware para el desarrollo del proyecto son:

**Tabla 1** Componentes hardware

Cantidad	Recurso
1	Laptop HP, Procesador Core i5-7200U, Memoria RAM 8,00 Gb, Disco duro 1T

*Fuente: (Segovia Eduardo, 2022)*

## **SOFTWARE:**

Los recursos software para el desarrollo del proyecto son:

**Tabla 2** Componentes Software

Windows 10 arquitectura 64 bits
Python versión 3.8.3
Django versión 3.0.7
PostgreSQL versión 12
Hosting SSD3 (Hostinet)

*Fuente: (Segovia Eduardo, 2022)*

### **3.2.3. Operativa**

Para la ejecución de este proyecto se cuenta con el personal necesario para el desarrollo y manipulación del sistema, obteniendo de esta manera un sistema de inventario con catálogo 3D intuitivo, que nos permita controlar de mejor manera el inventario de la empresa haciendo uso de las herramientas anteriormente planteadas.

### **3.2.4. Económica**

Este proyecto será autofinanciado, los softwares empleados en su desarrollo tales como Python, Django y PostgreSQL son de código abierto lo que implica que no se realizarán gastos en licencias, para el alojamiento del proyecto en la nube se hará uso de un hosting con un costo de \$60.00 dólares anuales los cuales serán financiados por la empresa.

### **3.2.5. Análisis de requerimientos**

#### **3.2.5.1. Requerimientos funcionales**

**Tabla 3** Requerimientos funcionales

<b>CÓDIGO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>PRIODIDAD</b>
RF-1	Registro de usuarios	El sistema debe permitir el registro de un usuario y usuario administrativo.	ALTA
RF-2	Inicio de sesión	El sistema debe permitir el ingreso únicamente a los usuarios registrados correctamente.	ALTA
RF-3	Gestión de productos	El sistema debe permitir registrar, editar y eliminar productos.	ALTA
RF-4	Gestión de catálogo 3D	El sistema debe permitir registrar y eliminar imágenes.	ALTA
RF-5	Gestión de proveedores	El sistema debe permitir registrar, editar y eliminar proveedores	ATLA
RF-6	Generar reportes	El sistema debe permitir generar reportes de los productos.	ALTA

Fuente: (Segovia Eduardo, 2022)

### 3.2.5.2. Requerimientos no funcionales

**Tabla 4** Requerimientos no funcionales

CÓDIGO	NOMBRE	DESCRIPCIÓN	PRIODIDAD
RNF-1	Seguridad de acceso	El sistema debe restringir el acceso a determinados módulos y funciones acorde al rol del usuario.	ALTA
RNF-2	Seguridad lógica y de datos	Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador.	ALTA
RF-3	Usabilidad	El sistema debe contar con manuales de usuarios estructurados.	ALTA
RF-4	Eficiencia	Toda la funcionalidad del sistema tiene la capacidad de realizar las funciones requeridas por el usuario.	ALTA
RF-5	Extensibilidad	El sistema tomara en consideración y facilitara su crecimiento en el futuro.	ATLA
RF-6	Portabilidad	El sistema tendrá compatibilidad con hardware y software.	ALTA

Fuente: (Segovia Eduardo, 2022)

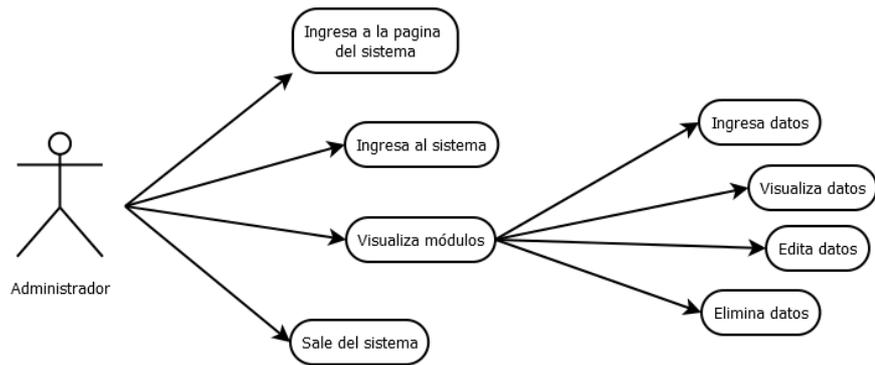
### 3.2.6. Casos de usos

Los actores que interactúan con el sistema se muestran a continuación:

#### **Administrador:**

El actor administrador es el encargado de gestionar los usuarios, productos, catálogo y reportes de productos.

**figura 3** Caso de usos administrador

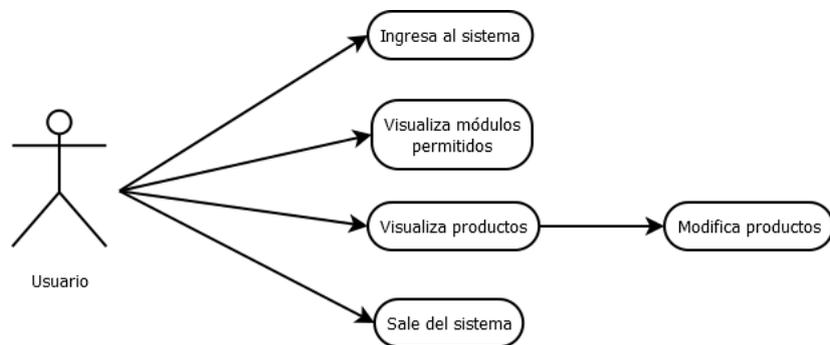


Fuente: (Segovia Eduardo, 2022)

**Usuario:**

El actor usuario es el encargado de buscar los productos para su venta y modificar su existencia.

**Figura 4** Casos de usos usuario

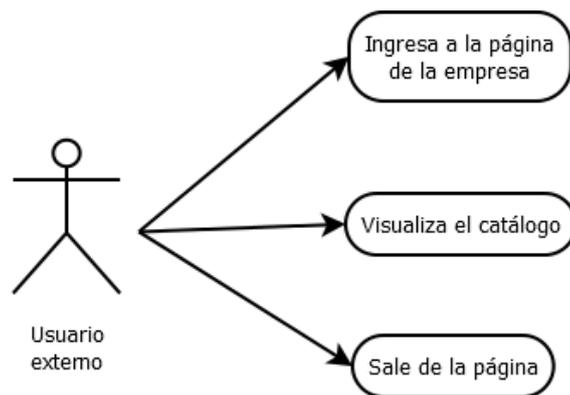


Fuente: (Segovia Eduardo, 2022)

**Usuario externo:**

El actor usuario externo es la persona ajena a la empresa que puede ver el catálogo de los productos.

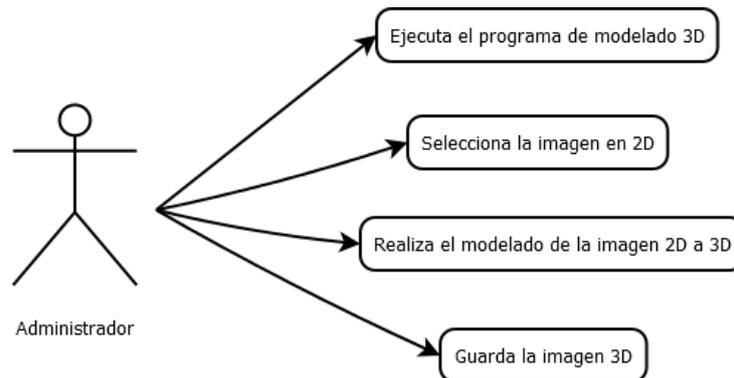
**Figura 5** Casos de usos cliente



Fuente: (Segovia Eduardo, 2022)

## Modelado 3D

Figura 6 Caso de uso modelado 3D



Fuente: (Segovia Eduardo, 2022)

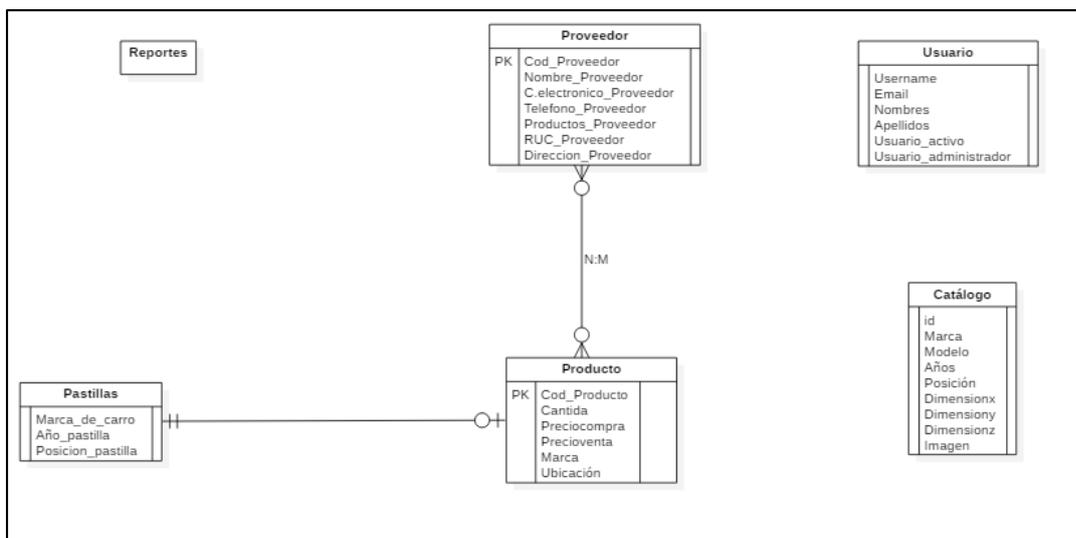
## 3.3. Diseño

### 3.3.1. Diseño conceptual

El diseño conceptual muestra los diagramas técnicos de la base de datos requerida para el desarrollo del sistema de inventario con catálogo 3D teniendo como principal función la administración, el usuario y el catálogo 3D.

## MODELO RELACIONAL

Figura 7 Modelo relacional



Fuente: (Segovia Eduardo, 2022)

### 3.4. Diccionarios de datos

El diccionario de datos representa la estructura lógica de la base de datos del sistema de inventario con catálogo 3D.

#### Significado de las columnas de la Base de Datos

**PK**= Primary key (llave primaria)

**NN**= Not Null (no nulo)

**Boolean**= Dato de tipo booleano (verdadero o falso)

**AI**= Auto incremental (auto incrementable)

#### Tabla Usuario

**Tabla 5** Tabla de usuario

<b>Campo</b>	<b>Tipo</b>	<b>Formato</b>
Cod_Proveedor	PK	Integer(AI)
Username	NN	Character varying(50)
Email	NN	Character varying(254)
Nombres	NN	Character varying(200)
Apellidos	NN	Character varying(200)
Usuario_activo	Boolean	Boolean
Usuario_administrador	Boolean	Boolean

Fuente: (Segovia Eduardo, 2022)

#### Tabla Proveedor

**Tabla 6** Tabla de proveedor

<b>Campo</b>	<b>Tipo</b>	<b>Formato</b>
ID	PK	Integer(AI)
Nombre	NN	Character varying(200)
Email	NN	Character varying(254)
Teléfono	NN	Character varying(10)
Dirección	NN	Character varying(200)

Fuente: (Segovia Eduardo, 2022)

## Tabla producto

**Tabla 7** Tabla de producto

<b>Campo</b>	<b>Tipo</b>	<b>Formato</b>
ID	PK	Integer(AI)
Código	NN	Character varying(20)
Repuesto	NN	Character varying(100)
Cantidad	NN	Integer
Preciocompra	NN	Numeric(5,2)
Precioventa	NN	Numeric(5,2)
Marca	NN	Character varying(50)
Ubicación	NN	Character varying(50)

Fuente: (Segovia Eduardo, 2022)

## Tabla pastilla heredada de producto

**Tabla 8** Tabla pastilla heredada de producto

<b>Campo</b>	<b>Tipo</b>	<b>Formato</b>
Producto_ptr_id	NN	Integer(AI)
Carro	NN	Character varying(200)
Años	NN	Character varying(100)
Posición	NN	Character varying(20)

Fuente: (Segovia Eduardo, 2022)

## Tabla catálogo

**Tabla 9** Tabla de catálogo

<b>Campo</b>	<b>Tipo</b>	<b>Formato</b>
<b>ID</b>	PK	Integer(AI)
<b>Marca</b>	NN	Character varying(40)
<b>Modelo</b>	NN	Character varying(100)
<b>Años</b>	NN	Character varying(100)
<b>Posición</b>	NN	Character varying(200)
<b>Dimensiónx</b>	NN	Numeric(5,2)
<b>Dimensióny</b>	NN	Numeric(5,2)
<b>Dimensiónz</b>	NN	Numeric(5,2)
<b>Imagen</b>	NN	Character varying(200)

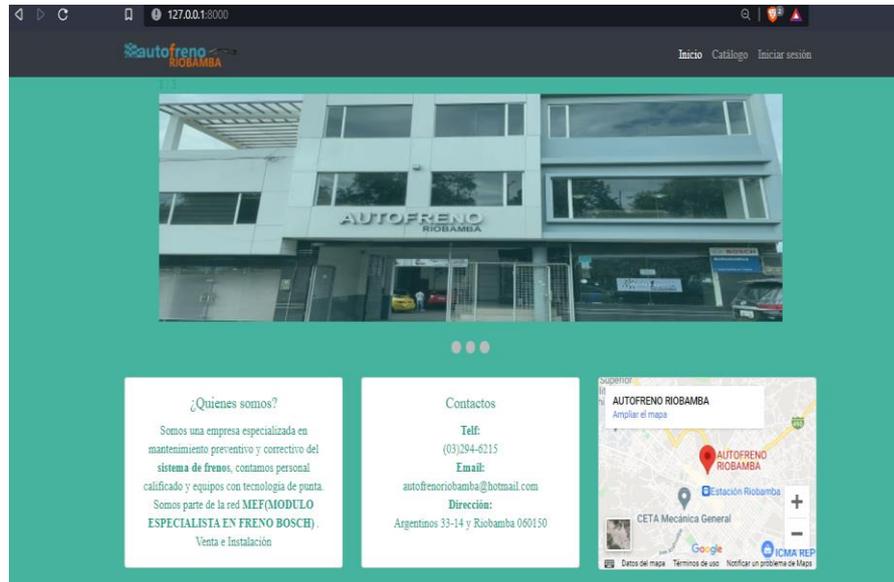
Fuente: (Segovia Eduardo, 2022)

### 3.5. Diseño de interfaces

El diseño de interfaz de usuario permite dar una idea de cómo se visualizará y navegará dentro del espacio de trabajo o del catálogo. A continuación, se muestran las principales interfaces de la aplicación.

- **Página principal**

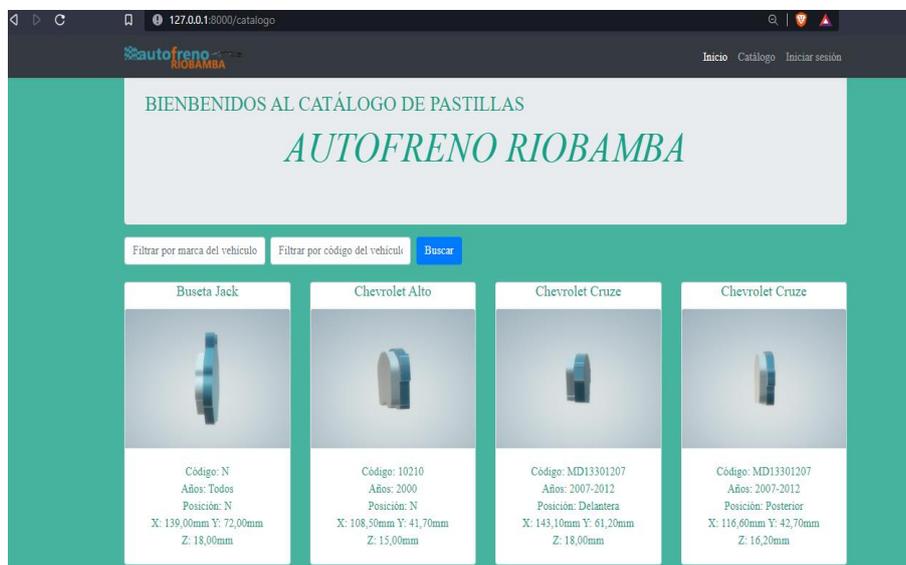
**Figura 8** Página principal



Fuente: (Segovia Eduardo, 2022)

- **Catálogo**

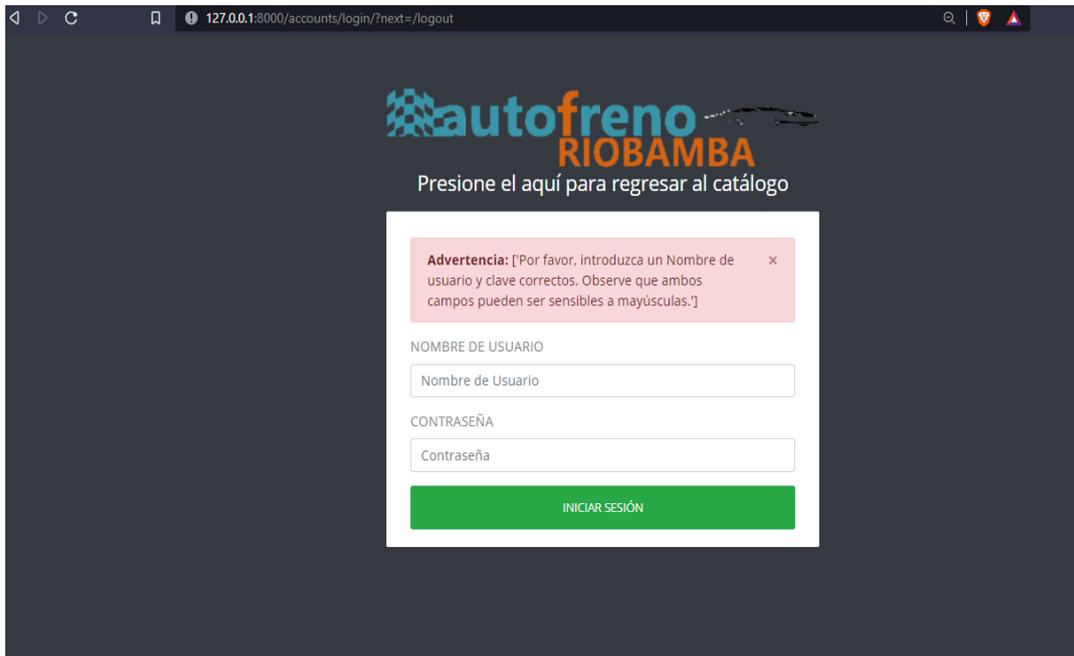
**Figura 9** Página de catálogo



Fuente: (Segovia Eduardo, 2022)

- **Inicio de sesión**

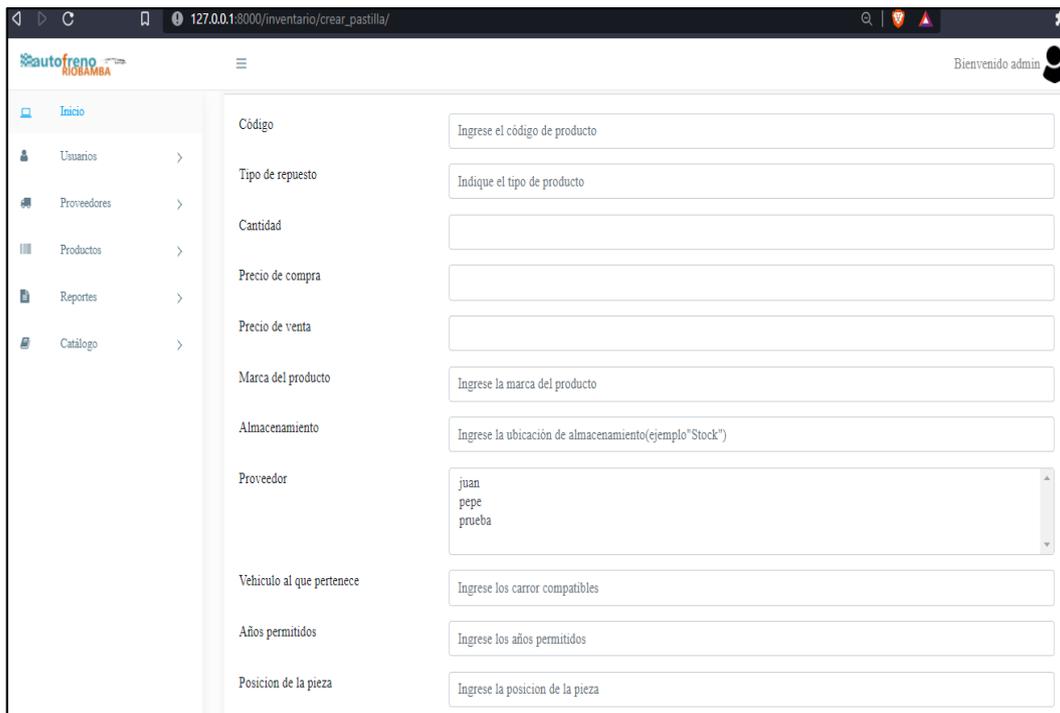
**Figura 10** Página de inicio de sesión



Fuente: (Segovia Eduardo, 2022)

- **Registro de productos**

**Figura 11** Página de registro de productos



Fuente: (Segovia Eduardo, 2022)

- **Listado de productos**

**Figura 12** Página de listado de productos

autofreno RIOBAMBA

Bienvenido admin

Filtrar por código    Filtrar por ubicación    Filtrar por marca    Filtrar por carro    **Buscar**

#	Repuesto	Cant.	P.compra	P.venta	Marca	Carro	Años	Posición	Ubicación	Proveedor	Opciones
x-1-s	pastilla	4	\$0.03	\$0.03	Hond	y	y6	qwe	Stock	• juan	
x-2-s	freno	5	\$9,00	\$7,00	y	w	w	q	Bodega2	• juan	
x-3-s	pastilla	1	\$12,00	\$123,00	a	1	98	kj	bodega3	• pepe	
x-4-s	aaa	3	\$2,00	\$2,00	L	1	2	k	Stock	• prueba	
x-5-s	pastilla	2	\$4,00	\$6,00	Chevrolet	chevrolet	123	delantera	bodega1	• juan	

página 1 de 1

Fuente: (Segovia Eduardo, 2022)

- **Listado catálogo**

**Figura 13** Página de listado de catálogo

autofreno RIOBAMBA

Bienvenido admin

Filtrar por marca del vehículo    Filtrar por código del vehículo    **Buscar**

#	Marca	Código	Años	Posición	Imagen	Opciones
11	Buseta Jack	N	Todos	N	catalog/busetajack_DXz4CZY.gif	
8	Chevrolet Alto	10210	2000	N	catalog/chevroletalto.gif	
2	Chevrolet Cruze	MD13301207	2007-2012	Delantera	catalog/chevroletcruzedelantera.gif	
7	Chevrolet Cruze	MD13301207	2007-2012	Posterior	catalog/chevroletcruzeposterior_azlhYOc.gif	
13	Ford Eco Sport	D1175	2004-2009	N	catalog/FordEcoSport_bllxJLZ.gif	
14	Ford Escape	D843	2002	N	catalog/FordEscape_hsQir17.gif	
19	Ford Expedition	D935	Todos	Posterior	catalog/FordExpedition_FEueq37.gif	
10	Great Wall Hover H5	D580	Todos	Posterior	catalog/Great_FHthNwe.gif	
18	Honda Civic 79	D177	79	N	catalog/HondaCivic79_Rw2fytI.gif	
16	Honda Civic 92	D564	1992	Posterior	catalog/HondaCivic92_RDVyPhb.gif	
22	Honda CRV	D537	2003	Posterior	catalog/hondacr2.gif	

Fuente: (Segovia Eduardo, 2022)

## CAPÍTULO IV

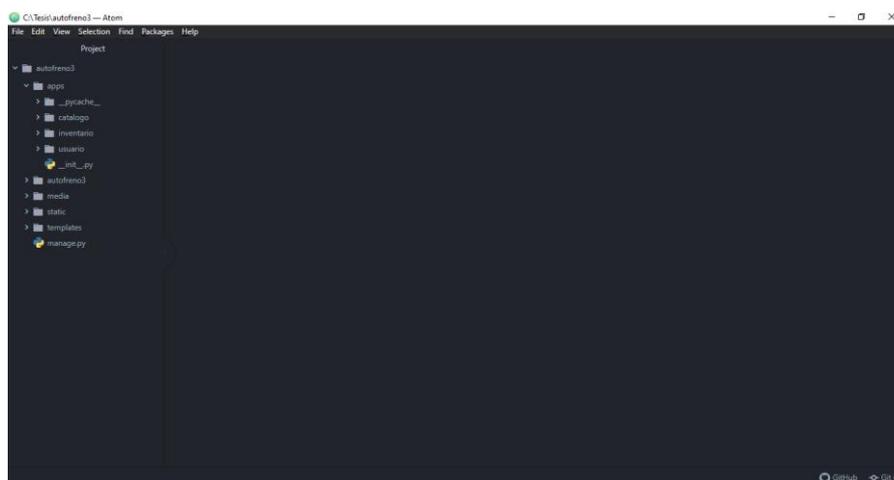
### IMPLEMENTACIÓN DE SISTEMA

#### 4.1. Configuración de las herramientas de desarrollo

##### ATOM

Para el desarrollo del sistema de inventario con catálogo 3D se utilizó el editor de código fuente Atom de código abierto el cual se procedió a instalar en nuestro equipo luego de haberlo descargado de su sitio oficial (<https://atom.io/>).

**Figura 14** Interfaz Atom



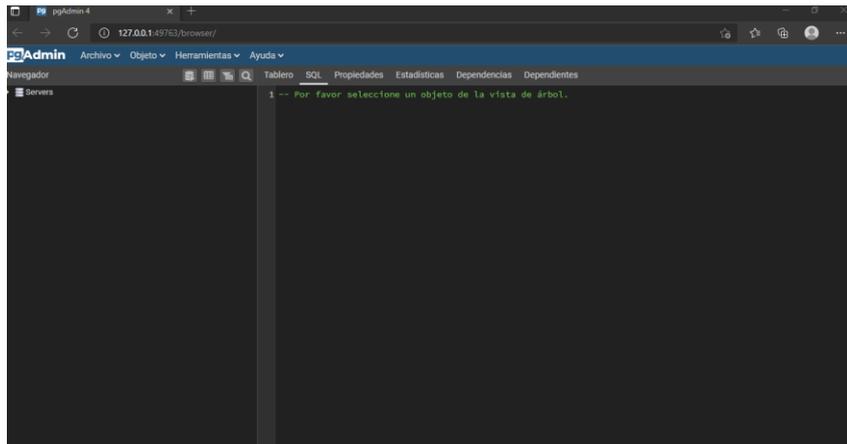
Fuente: (Segovia Eduardo, 2022)

##### PostgreSQL

El sistema gestor de base de datos empleado para el desarrollo del proyecto es PostgreSQL el cual es un gestor de base de datos relacional orientado a objetos y de código abierto.

El cual se procedió a instalar luego de su descarga en su página oficial (<https://www.postgresql.org/download/>)

**Figura 15** Interfaz de PostgreSQL



Fuente: (Segovia Eduardo, 2022)

## **Python**

El lenguaje de programación utilizado en el desarrollo del sistema fue Python el cual es un lenguaje de programación interpretado cuya filosofía es la legibilidad de su código además de ser de código abierto. El cual fue instalado previo a su descarga en su página oficial (<https://www.python.org/>).

## **Django**

El framework empleado en el desarrollo del sistema fue Django de código abierto escrito en Python. El cual fue instalado por medio de la terminal (CMD) los pasos se describen a continuación:

- Ejecutamos nuestro terminal CMD
- Una vez ejecutado la terminal CMD procedemos a digitar el siguiente comando: `pip install Django==3.2.6`
- Se ejecutará el comando el cual procederá a descargar e instalar Django de forma automática.

## **4.2. Arquitectura del sistema**

La arquitectura que se empleó en el desarrollo del sistema de inventario con catálogo 3D es cliente-servidor, el cual permite que el sistema pueda ser visualizado desde cualquier dispositivo electrónico, siempre y cuando este dispositivo cuente con conexión a internet y a un navegador que será necesario para ingresar la dirección web.

### 4.3. Implementación del sistema

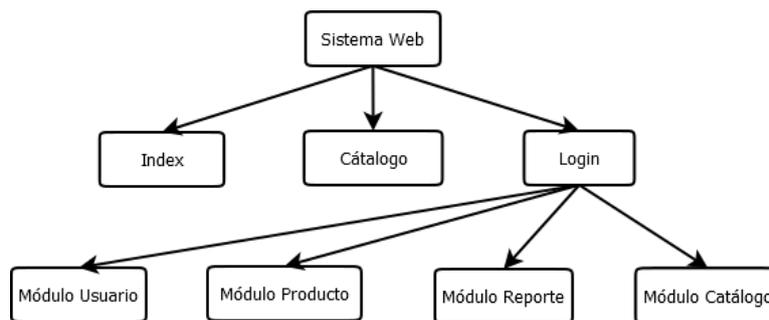
Para la implementación del sistema de inventario con catálogo 3D, es necesario el previo conocimiento de las herramientas con las que se desarrollará el sistema, de igual manera de contar con un análisis que permitirá establecer el desarrollo de cada requerimiento mediante módulos, así como sus posibles modificaciones que pueden presentarse en el transcurso del desarrollo del sistema.

Para realizar la implementación del sistema en el hosting, se procede a ingresar a la página web del proveedor hosting (<https://heroku.com>) y a registrarse, una vez iniciada la sesión se crea la aplicación con el nombre con el que se encontrara en la web, se ingresan las credenciales de la base de datos otorgada por el hosting, se realizar la conexión con los servicios de GitHub para subir los archivos del sistema.

#### 4.3.1. Definición de módulos del sistema

El sistema de inventario con catálogo 3D consta de varios módulos que fueron establecidos en los requerimientos funcionales del sistema, los cuales serán empleados en el desarrollo e implementación.

**Figura 16** Módulos del sistema



Fuente: (Segovia Eduardo, 2022)

#### 4.3.2. Desarrollo del sistema

Una vez establecidas las necesidades y módulos con los que contara el sistema de inventario con catálogo 3D, se procede a su desarrollo cumpliendo con las necesidades de usabilidad y optimización de tareas dentro de la empresa Autofreno Riobamba. El sistema se desarrolló de manera dinámica, de fácil acceso permitiendo a los usuarios ingresen desde cualquier dispositivo que cuente con acceso a internet y un navegador.

### 4.3.3. Implementación del sistema

La implementación del sistema de inventario con catálogo 3D se realizó por en un hosting que fue adquirido para el aplicativo con la siguiente URL <https://autofrenorio.herokuapp.com/>, al mismo que tienen acceso el personal de la empresa para su utilización.

### 4.4. Pruebas

**Tabla 10** Pruebas realizadas al sistema

<b>ACTIVIDAD</b>	<b>FUNCIONAL</b>	<b>NO FUNCIONAL</b>
Presenta página web	✓	
Acceso a la página web mediante URL	✓	
Visualización de catálogo 3D	✓	
Login	✓	
Navegación por módulos	✓	
Filtros	✓	
Emisión de reportes	✓	
Mensajes de error	✓	
Compatibilidad con navegadores	✓	
Responsive	✓	
Gestión de usuarios	✓	
Gestión de productos	✓	
Gestión de catálogo	✓	
Permisos de usuarios	✓	

Fuente: (Segovia Eduardo, 2022)

### 4.5. Capacitación al personal

Para la capacitación del manejo y funcionamiento del sistema se tomó en cuenta los siguiente:

#### Temas de capacitación

- Ingreso y autenticación del sistema
- Gestión de producto
- Gestión de catálogo 3D
- Permisos de usuario
- Generación de reportes

## **Equipos usados**

- Computadora
- Proyector

## **4.6. Mantenimiento**

El mantenimiento al sistema con el que contará la empresa Autofreno Riobamba tendrá una duración de 1 año, en el que desarrollador se compromete a brindar soporte técnico, además estará sujeto a posibles cambios e incrementos de módulos que requiera la empresa garantizando la funcionalidad y calidad del sistema de inventario con catálogo 3D.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **CONCLUSIONES**

- ❖ Con la previa investigación de las características del lenguaje de programación Python se logró conocer tanto el alcance como su funcionalidad para el desarrollo de sistemas de inventario con catálogo 3D, permitiendo de esta manera definirlos como herramientas ágiles y eficaces.
- ❖ Se analizaron los requisitos del sistema de inventario acorde a las necesidades que presentaba la empresa Autofreno Riobamba, las cuales se obtuvieron mediante una entrevista realizada a los integrantes de dicha empresa.
- ❖ Se desarrolló el sistema de inventario con catálogo 3D en la empresa Autofreno Riobamba, permitiendo obtener en tiempo real la existencia del inventario reduciendo así el tiempo y personal empleado en su conteo, además de reducir gastos en adquisición de producto existentes.
- ❖ Se analizaron las pruebas de los datos, resultados y reportes del sistema verificando así su correcto funcionamiento al ser implementado.

## **RECOMENDACIONES**

- ❖ La capacitación al personal que usara el sistema es de gran importancia ya que permitirá obtener un mayor provecho de cada una de las funciones del sistema.
- ❖ Realizar entrevistas o cuestionarios al personal de la empresa para identificar nuevos requerimientos y necesidades para futuras actualizaciones del sistema.
- ❖ Se propone a la empresa Autofreno Riobamba realizar mantenimientos anuales de hosting para el correcto funcionamiento del sistema.

## BIBLIOGRAFÍA

- Asale, R. . (01 de 07 de 2020). «*Diccionario de la lengua española*» - Edición del *Tricentenario*. Obtenido de catálogo | Diccionario de la lengua española: <https://dle.rae.es/cat%C3%A1logo+>
- Autodesk. (13 de 02 de 2022). *autodesk*. Obtenido de <https://www.autodesk.mx/solutions/3d-modeling-software>
- Castro, J. (21 de octubre de 2014). *Corponet*. Obtenido de <https://blog.corponet.com.mx/beneficios-de-un-sistema-de-control-de-inventarios>
- Ceballos. (01 de 01 de 2022). *Lenguaje de programación (informática) - EcuRed*. Obtenido de [https://www.ecured.cu/Lenguaje\\_de\\_programaci%C3%B3n\\_\(inform%C3%A1tica\)](https://www.ecured.cu/Lenguaje_de_programaci%C3%B3n_(inform%C3%A1tica))
- Challenger Pérez, I., Becerra Garcia, R., & Díaz Ricardo, Y. (2014). EL LENGUAJE DE PROGRAMACIÓN PYTHON. *Ciencias Holguín*, 1-13.
- Chávez. (29 de 07 de 2017). *prezi.com*. Obtenido de <https://prezi.com/it3zlp1wurtw/caracteristicas-de-modelado-3d/>
- Correa, P. (3 de octubre de 2015). *itm201511.webnode.es*. Obtenido de <https://itm201511.webnode.es/archivos-del-sistema/lenguajes/lenguaje-compilado-e-interpretado/>
- Corvo, H. (14 de 07 de 2019). *Sistema de inventarios: tipos, métodos y ejemplos*. Obtenido de Liferder: <https://www.liferder.com/sistema-de-inventarios/>
- Floriano. (3 de 10 de 2021). *Blog de DSIGNO*. Obtenido de <https://www.dsigno.es/blog/disenografico/que-es-el-modelado-en-3d-y-que-usos-tiene>
- Gonzáles. (25 de 04 de 2022). *CICE*. Obtenido de [https://www.cice.es/blog/articulos/modelado-organico-resolviendo-tus-dudas/#:%7E:text=El%20modelado%203D%20es%20un,CAD%20\(Computer%20Aided%20Design\).](https://www.cice.es/blog/articulos/modelado-organico-resolviendo-tus-dudas/#:%7E:text=El%20modelado%203D%20es%20un,CAD%20(Computer%20Aided%20Design).)
- grupokefren. (21 de 05 de 2021). *grupokefren.com*. Obtenido de <https://grupokefren.com/disen/modelado-3d/>

Holovaty. (29 de 09 de 2019). *developer.mozilla.org*. Obtenido de <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>

Holovaty, A., & Kaplan Moss, J. (2008). La historia de Django. En A. Holovaty, & J. Kaplan Moss, *El libro de Django* (págs. 3-4).

hostingpedia. (07 de febrero de 2019). *hostingpedia*. Obtenido de <https://hostingpedia.net/postgresql.html>

kuchling, A. (1 de noviembre de 2018). *linuxjournal*. Obtenido de <https://www.linuxjournal.com/article/2959>

López. (04 de 02 de 2021). *CCM*. Obtenido de <https://es.ccm.net/contents/304-lenguajes-de-programacion>

Muente. (08 de 01 de 2020). *rockcontent.com*. Obtenido de <https://rockcontent.com/es/blog/framework/>

Peters, T. (19 de agosto de 2019). *Python.org*. Obtenido de PEP 20 -- The Zen of Python: <https://www.python.org/dev/peps/pep-0020/>

Pontius, N. (23 de junio de 2020). *camcode.com*. Obtenido de <https://www.camcode.com/asset-tags/inventory-control-systems-types/>

PostgreSQL. (25 de junio de 2020). *PostgreSQL: About*. Obtenido de <https://www.postgresql.org/about/>

Prieto, J. (12 de 02 de 2020). <https://es.calameo.com/>. Obtenido de <https://es.calameo.com/books/00423425303608485d189>

Rossum. (22 de 02 de 2018). *python.org*. Obtenido de <https://www.python.org/doc/essays/cp4e/>

Segovia. (30 de 08 de 2018). *todopostgresql.com*. Obtenido de <https://www.todopostgresql.com/ventajas-y-desventajas-de-postgresql/>

Tokio. (2 de 07 de 2021). *Nebrinox*. Obtenido de <https://www.nebrinox.com/tipos-proceso-modelado-3d/>

tutorial.django. (s.f.). *tutorial.djangogirls.org*. Obtenido de <https://tutorial.djangogirls.org/es/django/>

Vezjak, I. (07 de 06 de 2020). *Python.org*. Obtenido de <https://www.python.org/success-stories/elementary-school-education-is-it-love-or-just-python/>

## **ANEXOS**

# **MANUAL TÉCNICO**



8 de septiembre del

# Manual técnico

## Índice

Introducción .....	58
Objetivos .....	58
Objetivo general .....	58
Objetivos específicos.....	58
A quien va dirigido y que debe saber .....	58
Creación base de datos .....	58
Creación de tablas y relaciones .....	59
Conexión a la base de datos .....	60
Creación de vistas.....	61
Creación de URLS .....	63
Creación de formulario.....	63
Gestión de módulos y archivos estáticos.....	64
Como realizar cambios o actualización del código de manera local.....	66
Configuración archivo HTML para mostrar de datos .....	66
Recomendaciones.....	68

## Índice de imágenes

Imagen 1 Creación de Base de datos PostgreSQL .....	59
Imagen 2 Creación tabla Producto y relación .....	59
Imagen 3 Conexión BD local .....	60
Imagen 4 Conexión BD producción.....	61
Imagen 5 Creación vista TemplateView .....	61
Imagen 6 Creación vista ListView .....	62
Imagen 7 Creación vista CreateView.....	62
Imagen 8 Creación de URLs.....	63
Imagen 9 Creacion de Forms.....	63
Imagen 10 Instalación de apps .....	64
Imagen 11 Configuración archivos estaticos.....	65
Imagen 12 Configuracion Static Url .....	65
Imagen 13 Configuración local en wsgi.py.....	66
Imagen 14 Configuración local en manage.py .....	66
Imagen 15 Configuración Html.....	67

## **Introducción**

El sistema de inventario con catálogo 3D, fue creado utilizando el lenguaje de programación Python 3.8.3, el gestor de base de datos PostgreSQL 4.23 y el Framework Django 3.2 el cual permite la creación de páginas web dinámicas y profesionales.

El manual presentado muestra la conexión a la base de datos en producción y local, gestión de módulos y archivos estáticos.

## **Objetivos**

### **Objetivo general**

Instruir al técnico informático como navegar por las diferentes carpetas y código del sistema de inventario para realizar el mantenimiento o actualización del mismo.

### **Objetivos específicos**

Crear un manual indicando los pasos para la gestión del sistema de inventario.

Reconocer las necesidades del hardware y software que se necesitaran para la gestión del sistema de inventario.

### **A quien va dirigido y que debe saber**

Este manual va dirigido al técnico informático, el cual deberá tener conocimientos sobre el framework Django 3.2, gestión de base de datos PostgreSQL y en el lenguaje de programación Python 3.8.3.

### **Creación base de datos**

Para la creación de la base de datos se instaló el programa PostgreSQL, en el cual se realizó las diferentes configuraciones para dar inicio a la programación de la base de datos.

```
Tablero SQL Propiedades Estadísticas Dependencias Dependientes
1 -- Database: autofreno1
2
3 -- DROP DATABASE autofreno1;
4
5 CREATE DATABASE autofreno1
6     WITH
7     OWNER = postgres
8     ENCODING = 'UTF8'
9     LC_COLLATE = 'Spanish_Spain.1252'
10    LC_CTYPE = 'Spanish_Spain.1252'
11    TABLESPACE = pg_default
12    CONNECTION LIMIT = -1;
```

*Imagen 1 Creación de Base de datos PostgreSQL*

En la imagen 1 se describe como se creó de la base de datos en PostgreSQL en la que se detalla las propiedades de la mismas: OWNER (Usuario de la base de datos), ENCODING (configuración de caracteres), LC\_COLLATE (idioma de la base de datos), LC\_CTYPE(capitalización), TABLESPACE (ubicación en el sistema de archivos), CONNECTION LIMIT (límite de conexiones).

### Creación de tablas y relaciones

La creación de las tablas y relaciones con las que contara la base de datos se las realizo mediante código empleando el framework Django y el lenguaje de programación Python.

```
class Producto(models.Model):
    id = models.AutoField(primary_key=True)
    codigo = models.CharField('Codigo del producto', max_length=20, blank=False, null=False)
    repuesto = models.CharField('Tipo de repuesto',max_length=100, blank=False, null=False)
    cantidad = models.IntegerField('Cantidad de producto',blank=False, null=False)
    scant = models.IntegerField('Sumar cantidad', blank=False, null=False)
    rcant = models.IntegerField('Restar cantidad', blank=False, null=False)
    precioc = models.DecimalField('Precio de compra',max_digits=5, decimal_places=2, blank=False, null=False)
    preciov = models.DecimalField('Precio de venta',max_digits=5, decimal_places=2, blank=False, null=False)
    marca = models.CharField('Marca del repuesto',max_length=50, blank=True, null=True)
    ubicacion = models.CharField('Ubicación',max_length=50, blank=False, null=False)
    proveedor_codigo = models.ManyToManyField(Proveedor)
    class Meta:
        db_table = 'Producto'
```

*Imagen 2 Creación tabla Producto y relación*

La imagen 2 describe la estructura y relación de la tabla Producto: id (clave de producto), código (código del producto), repuesto (tipo de repuesto de producto), cantidad (cantidad de producto), scant (cantidad a sumar de producto), rcant (cantidad a restar de producto), precioc (precio de

compra de producto), precio (precio de venta de producto), marca (marca del producto), ubicación (ubicación de almacenamiento del producto), proveedor\_codigo (relación con la tabla Proveedor), models.AutoField (generación de id automático), models.CharField (cadena de caracteres alfanuméricos), models.IntegerField (caracteres enteros), models.DecimalField (caracteres decimales), models.ManyToManyField (relación de tabla de muchos a muchos), max\_length (número de caracteres permitidos), blank (permite o no campos en blanco), null (permite o no campos vacío), max\_digits (máximo de dígitos numéricos), decimal\_places (número de decimales).

### Conexión a la base de datos

La base de datos se encuentra ubicada en dos archivos .py para permitir la actualización o mantenimiento del código sin que requiera detener el servicio en el servidor, la ubicación de estos archivos es autofreno1/autofreno1/settings/local.py.

```
DEBUG = True
ALLOWED_HOSTS = []
# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'autofreno4',
        'USER': 'postgres',
        'PASSWORD': 'autofrenorio',
        'HOST': 'localhost',
        'PORT': 5432
    }
}
```

Imagen 3 Conexión BD local

La imagen 3 muestra cómo se realiza la conexión a la base de datos de forma local. El primer punto a tener en cuenta es DEBUG que es True el cual permite identificar y corregir errores de programación, ALLOWED\_HOST (host permitidos) el cual se ingresa al subir el proyecto a un servidor web, DATABASE (base de datos), default (por defecto), ENGINE (motor o gestor de base de datos que se va a emplear) se agrega después de backends el gestor de base de datos, NAME (nombre de la base de datos), USER (nombre del usuario de la base de datos), PASSWORD (contraseña de la base de datos), HOST (servidor empleado), PORT (puerto con el que trabaja la base de datos)

```
ALLOWED_HOSTS = ['autofrenorio.herokuapp.com']
# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'ddg0rhk1fbr1ob',
        'USER': 'ioiwndhycktmx',
        'PASSWORD': '1d3d5ed1e6f25ec9e2c2a152ac113b7bf14e6aae675854551f5227b0756158cf',
        'HOST': 'ec2-54-204-241-136.compute-1.amazonaws.com',
        'PORT': 5432
    }
}
```

Imagen 4 Conexión BD producción

La imagen 4 muestra la conexión a la base de datos en producción. Los parámetros que se modifican son DEBUG en producción cambia a FALSE para evitar mostrar las direcciones url al presentarse algún error al manipular el sistema, NAME, USER, PASSWORD y HOST se ingresan de acuerdo a los que otorgue el servidor al cual se hospedo el sistema.

### Creación de vistas

La creación de vistas permite indicar como se va a navegar entre los archivos .html y que función va a realizar acorde a lo requerido.

```
class Inicio(TemplateView):
    template_name = 'index.html'
```

Imagen 5 Creación vista TemplateView

La imagen 5 describe como se crea la vista Inicio la cual será la primera pestaña que se mostrara al ingresar a la página web, empleando el framework Django que cuenta con un módulo de vista basadas en clases. La sintaxis inicia con class (clase) seguido del nombre de la vista, entre paréntesis la clase de vista en este caso TemplateView (genera una vista específica a un archivo .html), continua con template\_name (nombre del archivo .html) y finalmente entre comillas simples el nombre del archivo .html.

```

class ListarProveedor(ListView):
    model = Proveedor
    template_name = 'inventario/proveedor/listar_proveedor.html'
    queryset = Proveedor.objects.all()
    paginate_by = 12

    def get_queryset(self, *args, **kwargs):
        qs = super().get_queryset(*args, **kwargs)
        bnombre = self.request.GET.get('filtrarn')
        if bnombre:
            return qs.filter(nombre__icontains = bnombre)
        return qs

```

Imagen 6 Creación vista ListView

La imagen 6 describe como se crea la vista ListarProveedor. La sintaxis inicia con class (clase) el nombre de la vista, entre paréntesis la clase de vista ListView (genera una vista que permite listar objetos), continua con model (indica el modelo o tabla con la que se trabajara) seguido del nombre del modelo, template\_name (dirección del archivo .html), queryset (genera una petición a la base de datos), nombre del modelo.objects (indica que se trabaja con objetos).all (indica que se trabajara con todos los objetos), paginte\_by (número de objetos que se mostraran en una pestaña), def (función) get\_queryset(devuelve un valor) entre paréntesis self (referencia al nombre del objeto), \*args (argumentos), \*kwars (diccionario), qs (variable), super() (da acceso a métodos y propiedades), bnombre (variable), get (método que encapsula datos), nombre del filtro, if (condición), bnombre, return (regresa un valor), qs.filter(filtro), entre paréntesis el parámetro por el que se va a filtrar seguido de \_\_icontains (que el parámetro no se exacto al de la base de datos), variable bnombre y finalmente un return con la variable qs.

```

class CrearProveedor(CreateView):
    model = Proveedor
    form_class = ProveedorForm
    template_name = 'inventario/proveedor/crear_proveedor.html'
    success_url = reverse_lazy('inventario:listar_proveedor')

```

Imagen 7 Creación vista CreateView

La imagen 7 describe como se crea la vista CrearProveedor. La sintaxis inicia con class (clase) nombre de la vista, entre paréntesis la clase de vista CreateView (genera una vista para crear y guardar objetos), model (modelo o tabla con la que se trabajara), nombre del modelo, form\_class (formulario), nombre del form, template\_name (dirección del archivo .html), success\_url (redirige si el formulario procede correctamente), reverse\_lazy (proporciona una URL invertida) entre paréntesis dirección a la que se quiere regresar.

## Creación de URLs

La creación las urls va a permitir establecer cómo se va a mostrar y como se va a direccionar la página web.

```
urlpatterns = [
    path('listar_proveedor/', login_required(ListarProveedor.as_view()), name = 'listar_proveedor'),
    path('crear_proveedor/', login_required(CrearProveedor.as_view()), name = 'crear_proveedor'),
    path('editar_proveedor/<int:pk>/', login_required(ActualizarProveedor.as_view()), name = 'editar_proveedor'),
    path('eliminar_proveedor/<int:pk>/', login_required(EliminarProveedor.as_view()), name = 'eliminar_proveedor'),
    path('listar_pastilla/', login_required(ListarPastillas.as_view()), name = 'listar_pastilla'),
    path('crear_pastilla/', login_required(CrearPastillas.as_view()), name = 'crear_pastilla'),
    path('editar_pastilla/<int:pk>/', login_required(ActualizarPastillas.as_view()), name = 'editar_pastilla'),
    path('eliminar_pastilla/<int:pk>/', login_required(EliminarPastilla.as_view()), name = 'eliminar_pastilla'),
    path('reportes_productos/', login_required(Reportes_P.as_view()), name = 'reportes_productos'),
    path('reporte/', login_required(Report_P.as_view()), name = 'reporte')
]
```

Imagen 8 Creación de URLs

La imagen 8 indica cómo se crean las urls que permitirían navegar entre los distintos módulos. La sintaxis inicia con urlpatterns (tupla de Python) entre corchetes path (devuelve un elemento), nombre con el que mostrara la url, login\_required (indica que requiere iniciar sesión para dirigirse a esa url), nombre de la vista.as\_view (método que devuelve una función), name (nombre del archivo .html)

## Creación de formulario

La creación de formularios permite establecer como se visualizarán los campos para la inserción de datos, así como restringir tipos de datos y mostrar mensajes de alerta.

```
class ProveedorForm(forms.ModelForm):
    class Meta:
        model = Proveedor
        fields = ['nombre', 'correo', 'telefono', 'productos', 'ruc', 'direccion']
        labels = {
            'nombre': 'Nombre',
            'correo': 'Email',
            'telefono': 'Teléfono/Celular',
            'productos': 'Productos que oferta',
            'ruc': 'RUC',
            'direccion': 'Dirección',
        }
        widgets = {
            'nombre': forms.TextInput(attrs = {
                'class': 'form-control',
                'placeholder': 'Ingrese un nombre ej.(Marco)',
                'pattern': '[A-Z a-z]*',
                'title': 'No se permiten números'
            })
        }
```

Imagen 9 Creación de Forms

La imagen 9 muestra cómo se crean formularios empleando Django para evitar escribir código html y css para el diseño de cada campo en el que se digitara. La sintaxis empieza con class (clase) nombre del formulario, forms (llama al método forms) ModelForms (indica el modelo del formulario), class seguido de Meta (adosa metadatos), model (llama al modelo) nombre del modelo, fields (tipos de datos) seguido de los datos de las tablas en orden, labels (rendiriza el campo en html), entre comillas simples el primer field seguido del nombre con el que se mostrara en la plantilla html, widgets (argumentos específicos para field), nombre del field seguido del tipos de dato a ingresar, attrs (atributos), class (tipo de form que requiere), placeholder (texto de marcado), patter (expresión regular que verifica el input) [A-Z a-z]\* (condición a validar), title (mensaje al ingresar un carácter incorrecto al input).

### Gestión de módulos y archivos estáticos

Para la gestión de módulos o apps cómo se las conocen en Django, hay que dirigirse al archivo base.py ubicado en autofreno1/autofreno1/settings/.

En el archivo base.py en la línea código INSTALLED\_APPS se encuentran todas las aplicaciones definidas por Django y las agregadas por el desarrollador.

```
31 # Application definitions
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'import_export',
41     'ckeditor',
42     'apps.catalogo',
43     'apps.inventario',
44     'apps.usuario',
```

*Imagen 10 Instalación de apps*

La imagen 10 muestra cómo se instalan o registran las aplicaciones para poder ser empleadas y gestionadas. Entre las aplicaciones que se encuentran en INSTALLED\_APPS (aplicaciones instaladas) se instalan por defecto las provenientes de Django al crear un proyecto, para registrar las aplicaciones que se han creado se debe tener en cuenta la ubicación en que se encuentran ya que puede afectar que sean reconocidas en otras partes del código, la sintaxis es carpeta en la encuentra apps. nombre de la aplicación.

En el mismo archivo base.py en la línea de código TEMPLATES y en el parámetro 'DIRS' se definirá cuáles son los archivos estáticos que se emplearan los cuales pueden ser html, css, js e imágenes.

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ['templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

*Imagen 11 Configuración archivos estáticos*

La imagen 11 indica donde se realizará el reconocimiento de los archivos estáticos para poder emplearlos en el proyecto, el parámetro que se editara es DIRS ya que este indica la dirección en que se encuentran.

De igual manera al final del archivo base.py se define la url estática y la dirección de los archivos estáticos.

```
112
113 # Static files (CSS, JavaScript, Images)
114 # https://docs.djangoproject.com/en/3.2/howto/static-files/
115
116 STATIC_URL = '/static/'
117 STATIC_URL = '/static/'
118 STATICFILES_DIRS = (BASE_DIR, 'static')
119 MEDIA_URL = '/media/'
120 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
121
```

*Imagen 12 Configuración Static Url*

La imagen 12 muestra cómo se direccionarán los archivos estáticos, STATIC\_URL (es el prefijo que se antepone a los archivos estáticos), STATICFILES\_DIRS (dirección de los archivos estáticos) BASE\_DIR(directorio base), esta línea indica que buscare en el directorio base la carpeta static. De igual manera para los archivos media MEDIA\_URL (es el prefijo que se antepone a los archivos media), MEDIA\_ROOT (contiene la ruta donde se cargan los archivos media) os.path.join(BASE\_DIR, 'media') buscará la carpeta media para almacenar los archivos media, si no encuentra la carpeta procederá a crearla automáticamente.

## Como realizar cambios o actualización del código de manera local

Para realizar actualizaciones o actualización del código se deberá definir la conexión a la base de datos local en el archivo wsgi.py en cual se encuentra en autofreno1/autofreno1 y el archivo manage.py el cual se encuentra en autofreno1.

### Archivo wsgi.py

Se realiza el cambio de 'autofreno1.settings.production' por 'autofreno1.settings.local' el cual hace referencia al archivo local.py en el cual se encuentra la conexión a la base de datos local.

```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'autofreno1.settings.production')
```

*Imagen 13 Configuración local en wsgi.py*

### Archivo manage.py

Se realiza el cambio de 'autofreno1.settings.production' por 'autofreno1.settings.local' el cual hace referencia al archivo local.py en el cual se encuentra la conexión a la base de datos local.

```
def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'autofreno1.settings.production')
    try:
        from django.core.management import execute_from_command_line
```

*Imagen 14 Configuración local en manage.py*

Finalizada la configuración para trabajar con la base de datos local, se pueden verificar los cambios o actualización del código mediante Django el cual ofrece un servidor para trabajar localmente, para iniciar el servidor se ejecuta CDM, se ingresa la ubicación del sistema de inventario C:/...../autofreno1 y se escribe el comando (python manage.py runserver) el cual iniciara el servidor y mostrara la URL local con la que se podrá visualizar la página web.

## Configuración archivo HTML para mostrar de datos

Para mostrar los datos almacenados en un archivo html se empleará un ciclo for con una condición a la tabla de la base de datos.

```

</tbody>
...{% for Pastilla in object_list %}-
...<tr>-
...<td>{{Pastilla.codigo}}</td>-
...<td>{{Pastilla.repuesto}}</td>-
...<td>{{Pastilla.cantidad}}</td>-
...{% if request.user.is_staff %}-
...<td>${{Pastilla.precioc}}</td>-
...{% endif %}-
...<td>${{Pastilla.preciov}}</td>-
...<td>{{Pastilla.marca}}</td>-
...<td>{{Pastilla.carro}}</td>-
...<td>{{Pastilla.años}}</td>-
...<td>{{Pastilla.posicion}}</td>-
...<td>{{Pastilla.ubicacion}}</td>-
...<td class = "text-center">-
...<ul>-
...<li>{{Proveedor.nombre}}</li>-
...</ul>-
...{% endfor %}-
...</td>-

```

*Imagen 15 Configuración HTML*

La imagen 15 muestra cómo se llaman a los datos almacenados para ser mostrados en el archivo html. La sintaxis es: se inicia con llave y porcentaje para que se puede escribir código de Python en html, se genera un ciclo for seguido del nombre de la tabla requerida in object\_list (lista de objetos), se usa object\_list ya que Python reconoce los datos almacenados como objetos, seguido se abren dos llaves, se escribe el nombre de la tabla y separado por un punto el campo que se quiere mostrar, de la misma manera se procedería con el resto de datos.

## **Recomendaciones**

Hacer uso del manual técnico para conocer la funcionalidad de las herramientas empleadas.

No permitir el acceso a personas ajenas al personal administrativos de la empresa.

# **MANUAL DE USUARIO**



# Manual de usuario

## Índice

Introducción .....	73
Objetivos .....	73
Objetivo general .....	73
Objetivos específicos.....	73
Como usar el sistema de inventario.....	73
Catálogo 3D.....	74
Login administrativo .....	74
Registro de usuario.....	76
Listado de usuarios.....	76
Edición de usuario.....	77
Eliminación de usuario.....	78
Registro de proveedores .....	78
Lista de proveedores.....	79
Edición de proveedor.....	79
Eliminación de proveedor .....	80
Registro de productos.....	80
Listado de productos .....	81
Editar producto.....	81
Eliminar producto.....	82
Registro de producto en catálogo .....	82
Listado del catálogo .....	83
Edición de producto del catálogo .....	83
Eliminar producto del catálogo .....	84
Login Usuario.....	84
Recomendaciones.....	86

## Índice de imágenes

Imagen 1	Página principal.....	73
Imagen 2	Página de catálogo.....	74
Imagen 3	Página de login .....	75
Imagen 4	Administración del sistema .....	75
Imagen 5	Añadir usuario .....	76
Imagen 6	Listado de usuarios.....	77
Imagen 7	Edición de usuario.....	77
Imagen 8	Eliminación de usuario.....	78
Imagen 9	Registro de proveedores .....	78
Imagen 10	Listado de proveedores.....	79
Imagen 11	Edición de proveedor.....	79
Imagen 12	Eliminación de proveedor .....	80
Imagen 13	Registro de productos.....	80
Imagen 14	Listado de productos .....	81
Imagen 15	Edición de productos.....	81
Imagen 16	Eliminación de producto .....	82
Imagen 17	Registro de producto en catálogo .....	82
Imagen 18	Listado de catálogo.....	83
Imagen 19	Edición de producto del catálogo .....	84
Imagen 20	Eliminar producto del catálogo .....	84
Imagen 21	Sesión de usuario.....	85

## Introducción

El sistema de inventario con catálogo 3D no permite visualizar y gestionar los productos con los que cuenta la empresa Autofreno Riobamba. Este proyecto permitirá que la empresa tenga un mejor control de su inventario.

## Objetivos

### Objetivo general

Instruir al personal de la empresa Autofreno Riobamba en el manejo del sistema de inventario.

### Objetivos específicos

Crear un manual indicando los pasos para el manejo del sistema de inventario.

Mostrar al personal de empresa como generar reportes.

### Como usar el sistema de inventario

Abrir un navegador de preferencia y en la barra de direcciones URL ingresar la dirección <https://autofrenorio.herokuapp.com/>

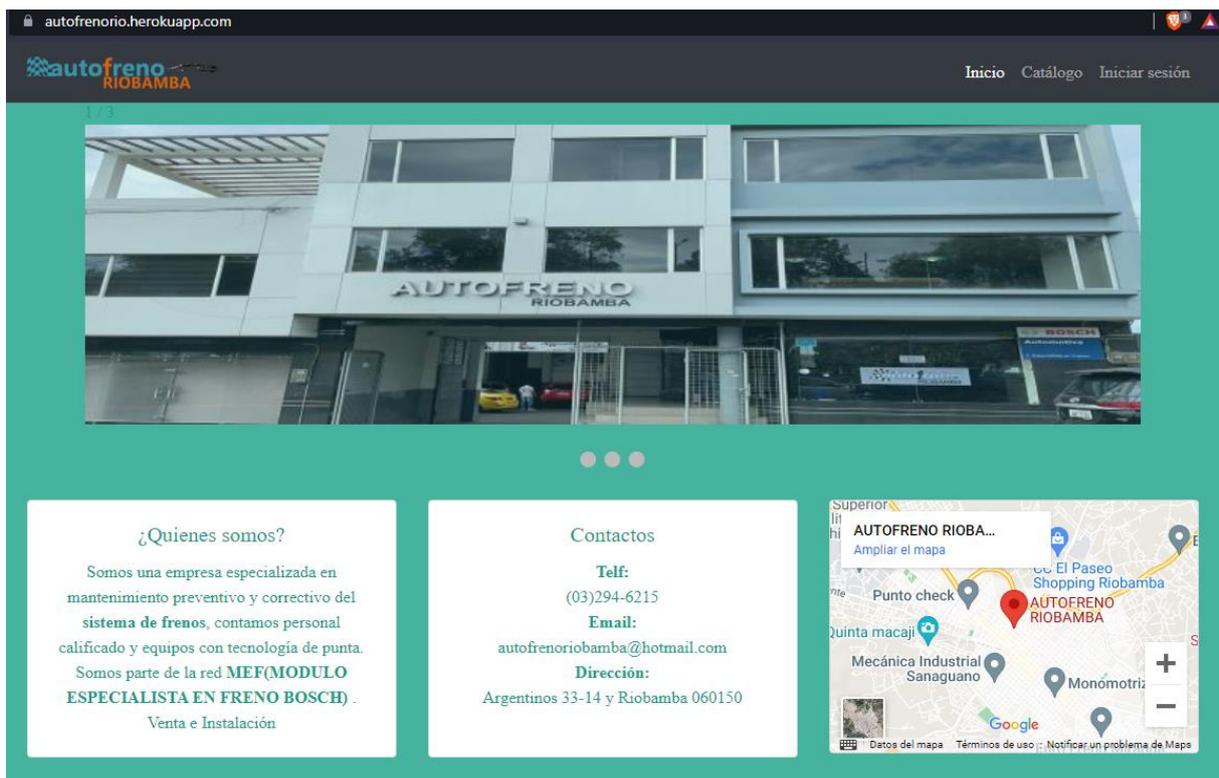


Imagen 1 Página principal

La imagen 1 muestra página web de la empresa Autofreno Riobamba, se mostrará la página de Inicio en la cual da información sobre la empresa. En la esquina superior derecha se mostraba una barra de navegación las cuales nos dirigirán al catálogo y al login.

### Catálogo 3D

Para ingresar al catálogo 3D se deberá presionar sobre la pestaña Catálogo que se encuentra en la esquina superior derecha.

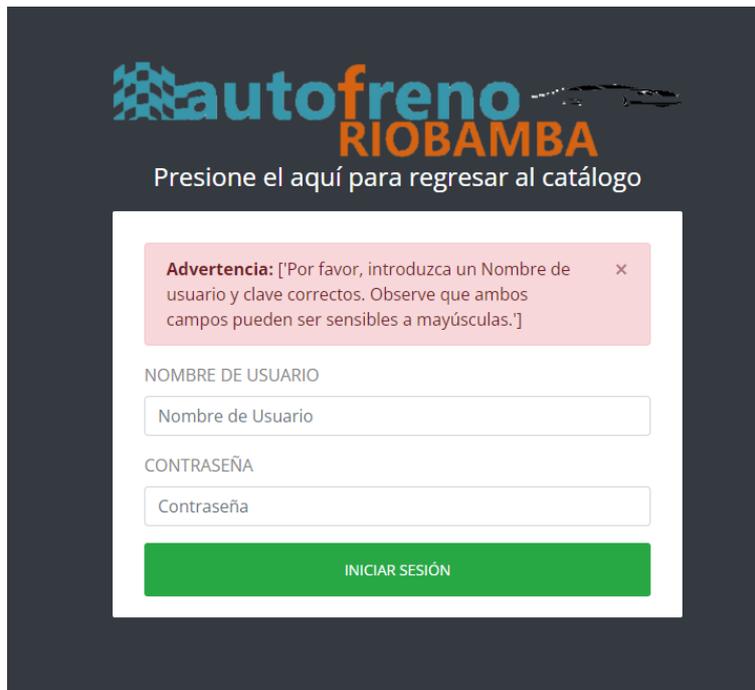


Imagen 2 Página de catálogo

La imagen 2 muestra el catálogo 3D de la empresa Autofreno Riobamba, en la cual se puede visualizar los productos con sus respectivos datos, de igual manera se puede realizar el filtrado de productos por la marca del vehículo o código al ingresarlos en los recuadros y presionando en buscar.

### Login administrativo

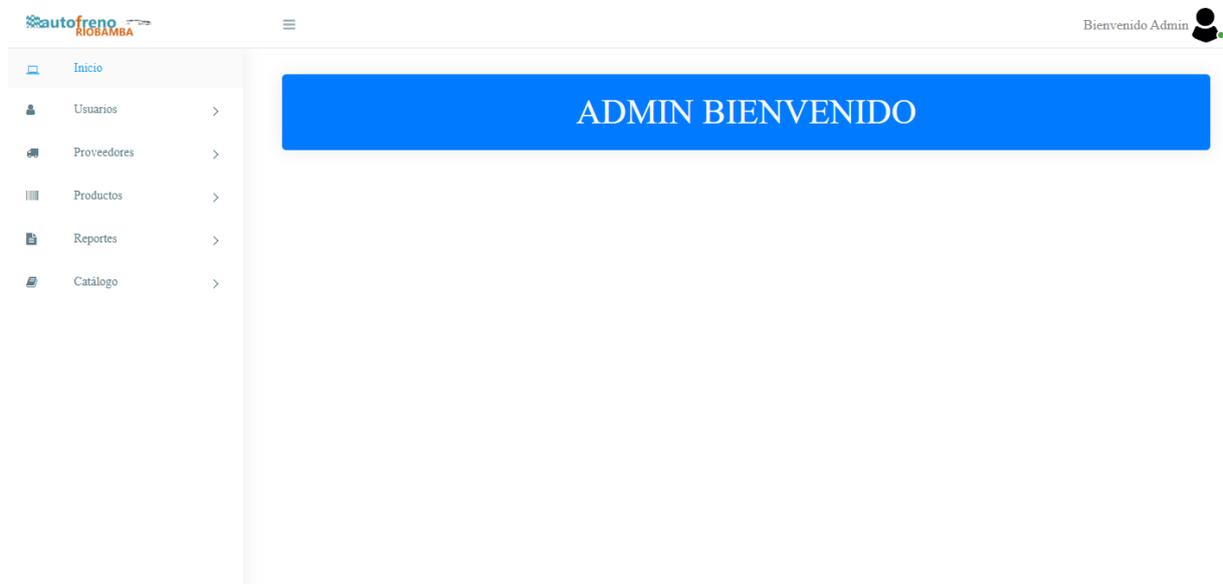
Para ingresar al sistema de administración deberá presionar en iniciar sesión ubicado en la esquina superior derecha.



*Imagen 3 Página de login*

La imagen 3 muestra el login, en la cual se visualiza el logo de la empresa el cual permite regresar al catálogo presionando sobre él, más abajo se muestra un recuadro con una advertencia la cual indica que se ingrese los datos correctos para iniciar sesión, abajo del mensaje se encuentran dos campos que piden que se ingrese el nombre de usuario y contraseña además de un botón para iniciar sesión.

Una vez iniciada la sesión se mostrará la página de administración del sistema de inventario.

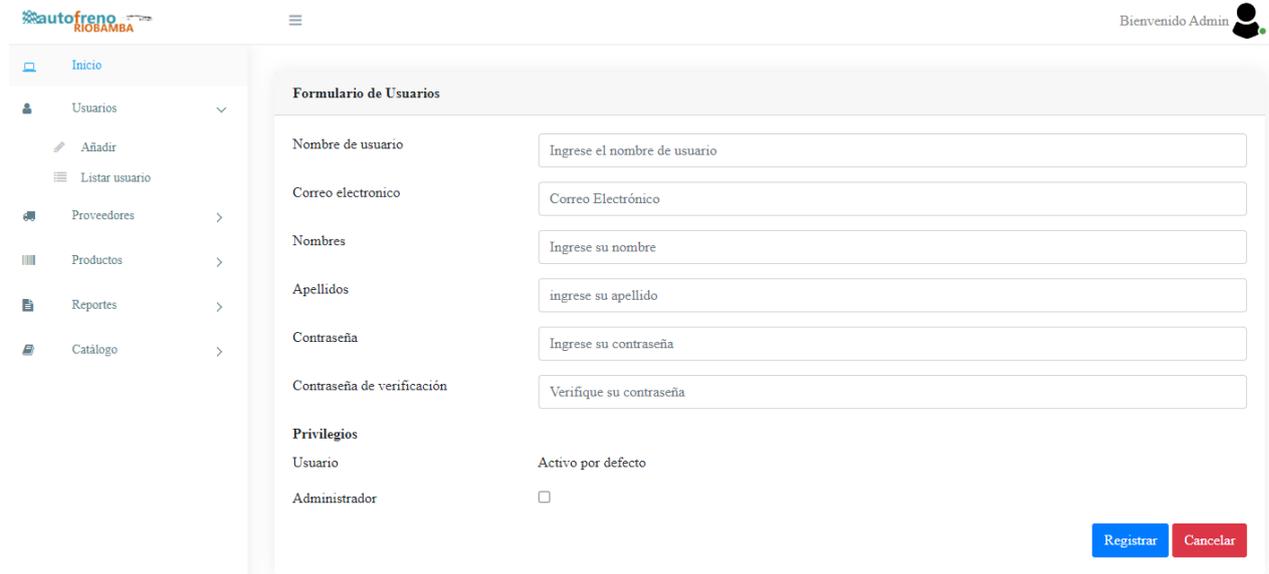


*Imagen 4 Administración del sistema*

La imagen 4 muestra la página principal de administración del inventario, donde se puede ver al lado izquierdo los módulos con los que cuenta como son: Usuarios, Proveedores, Productos, Reportes y Catálogo.

## Registro de usuario

Para registrar un nuevo usuario se debe presionar sobre Usuarios para desplegar las opciones y presionar añadir.



*Imagen 5 Añadir usuario*

En la imagen se muestra el formulario de usuarios, en el cual se podrán registrar nuevos usuarios para gestionar el sistema de inventario. Para realizar el registro se tendrá que ingresar un nombre de usuario el cual permitirá iniciar sesión, un correo electrónico, nombres y apellidos en los cuales no se aceptan números, contraseñas las cuales deben coincidir en los dos campos y por último se solicitará que marque la casilla de administrador si requiere que el nuevo usuario tenga todos los privilegios caso contrario se deja desmarcada, ya ingresados los datos se presiona en registrar y se dirigirá automáticamente a la lista de usuarios.

## Listado de usuarios

Para revisar los usuarios registrados deberá presionar sobre Usuarios para desplegar las opciones y presionar sobre listar usuarios.



*Imagen 6 Listado de usuarios*

La imagen 6 muestra el listado de usuarios, donde se podrán filtrar de acuerdo al nombre de usuario, nombre o apellido y presionando en el botón buscar, también se visualizarán dos botones en la celda de Opciones los cuales permitirán editar o eliminar un usuario.

### Edición de usuario

Para editar un usuario deberá ubicarse en la lista de usuarios y presionar sobre el icono de lápiz que se encuentra en la celda de Opciones.

*Imagen 7 Edición de usuario*

La imagen 7 muestra el formulario del usuario el cual se editará, una vez editados los campos requeridos se deberá ingresar nueva mente una contraseña, y presionar registrar, caso contrario si desea descartar los cambios presiona cancelar y se dirigirá automáticamente al listado de usuarios.

## Eliminación de usuario

Para la eliminación de usuario deberá ubicarse en la lista de usuarios y presionar sobre el icono de basurero que se encuentra en la celda de Opciones.

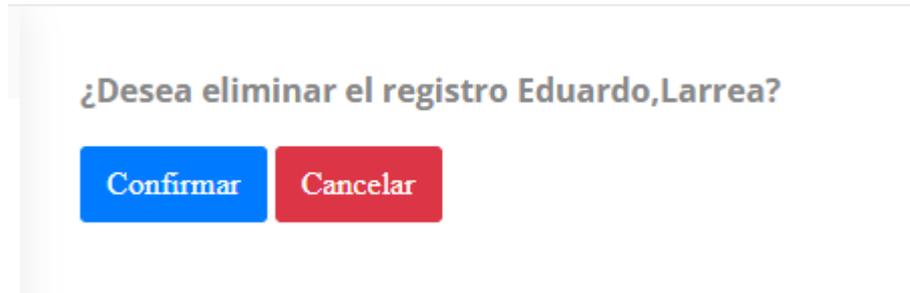


Imagen 8 Eliminación de usuario

La imagen 8 muestra un mensaje que indicara si desea eliminar el registro de usuario que tiene el nombre y apellido, tendrá la opción de confirmar la eliminación del usuario o cancelar en cuyo caso será dirigido a la lista de usuarios sin eliminar el usuario.

## Registro de proveedores

Para realizar el registro de proveedores deberá presionar sobre Proveedores para mostrar las opciones y presionar en añadir.

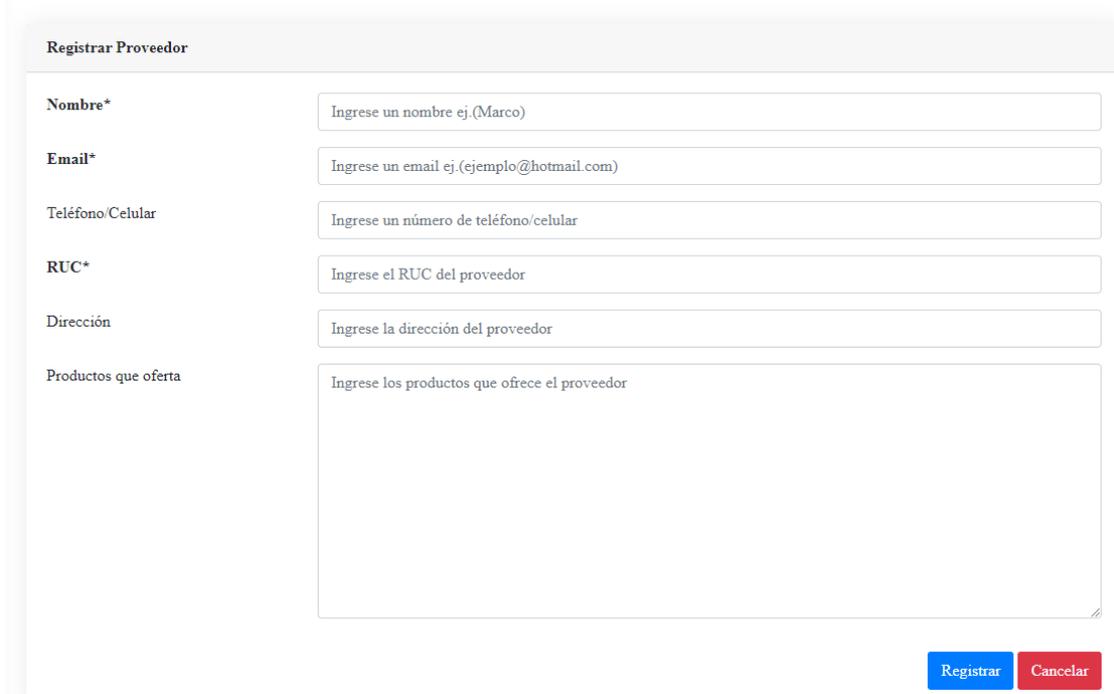
Un formulario titulado "Registrar Proveedor". Tiene los siguientes campos: "Nombre\*" con el texto "Ingrese un nombre ej.(Marco)", "Email\*" con "Ingrese un email ej.(ejemplo@hotmail.com)", "Teléfono/Celular" con "Ingrese un número de teléfono/celular", "RUC\*" con "Ingrese el RUC del proveedor", "Dirección" con "Ingrese la dirección del proveedor", y "Productos que oferta" con "Ingrese los productos que ofrece el proveedor". En la parte inferior derecha hay dos botones: "Registrar" (azul) y "Cancelar" (rojo).

Imagen 9 Registro de proveedores

La imagen 9 muestra el formulario de registro de proveedor, en el cual se le pedirá que ingrese el nombre del proveedor, email y ruc como campos obligatorios, un teléfono o celular, una

dirección, y productos que oferta, finalmente presiona el botón registrar y será dirigido automáticamente a la lista de proveedores.

## Lista de proveedores

Para ver el listado de proveedores deberá presionar en Proveedores para mostrar las opciones y presionar en listar proveedores.



#	Nombre	Correo	Teléfono	Productos	Ruc	Dirección	Opciones
1	IMPORFRENOS	imporfrenos@outlook.com	0997590420	chebrolet	111111111111	Av 10 de Agosto N58-115 y Luis Tufiño	 

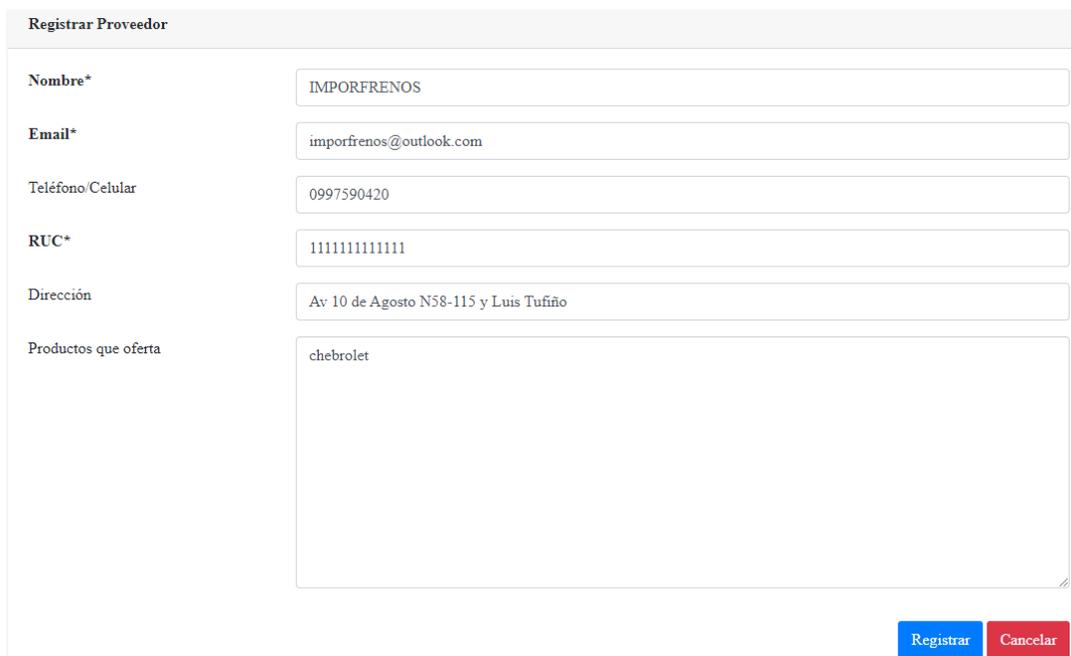
página 1 de 1

*Imagen 10 Listado de proveedores*

La imagen 10 muestra el listado de los proveedores, los cuales se pueden filtrar por su nombre, de igual manera se muestra una celda de opciones en la cual puede editar y eliminar un proveedor.

## Edición de proveedor

Para editar un proveedor deberá dirigirse a la lista de proveedores y presionar el icono de lápiz para abrir el formulario de registro de proveedores y editarlo.



Registrar Proveedor

Nombre\* IMPORFRENOS

Email\* imporfrenos@outlook.com

Teléfono/Celular 0997590420

RUC\* 111111111111

Dirección Av 10 de Agosto N58-115 y Luis Tufiño

Productos que oferta chebrolet

Registrar Cancelar

*Imagen 11 Edición de proveedor*

La imagen 11 muestra el formulario de registro de proveedor en el cual se podrá editar los campos requeridos para posteriormente presionar el botón registrar para guardar los cambios o cancelar y ser direccionado a la lista de proveedores.

### Eliminación de proveedor

Para eliminar un proveedor deberá ubicarse en la lista de proveedores y presionar en el icono de basurero.

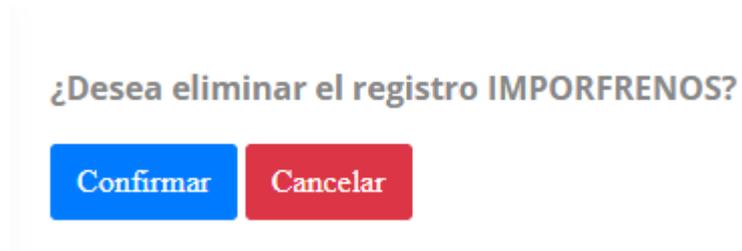


Imagen 12 Eliminación de proveedor

La imagen 12 muestra un mensaje que indicará si desea eliminar el registro del proveedor de nombre, tendrá la opción de confirmar la eliminación o cancelar y ser dirigido a la lista de proveedores.

### Registro de productos

Para el registro de un producto deberá presionar en Productos para mostrar más las opciones y presionar añadir.

Formulario de Productos	
Código	<input type="text" value="Ingrese el código de producto"/>
Tipo de repuesto	<input type="text" value="Indique el tipo de producto"/>
Cantidad	<input type="text"/>
Sumar cant	<input type="text"/>
Restar cant	<input type="text"/>
Precio de compra	<input type="text"/>
Precio de venta	<input type="text"/>
Marca del producto	<input type="text" value="Ingrese la marca del producto"/>
Almacenamiento	<input stock")"="" type="text" value="Ingrese la ubicación de almacenamiento(ejemplo"/>
Proveedor	<input type="text" value="IMPORFRENOS"/>
Vehículo al que pertenece	<input type="text" value="Ingrese los carror compatibles"/>
Años permitidos	<input type="text" value="Ingrese los años permitidos"/>
Posicion de la pieza	<input type="text" value="Ingrese la posicion de la pieza"/>

Imagen 13 Registro de productos

La imagen 13 muestra el formulario de registro de productos, en el cual se le pedirá que ingrese el código del producto, tipo de repuesto, cantidad, precio de compra, precio de venta, almacenamiento, seleccionar un proveedor, vehículo al que pertenece, años permitidos, posición, también se le pedirá los campos sumar cant y restar cant en los cuales deberá ingresar la cantidad de 0 para el registro de un nuevo producto y presionar el botón registrar.

## Listado de productos

Para ver la lista de productos deberá presionar en Productos para mostrar más opciones y presionar en listar productos.

The screenshot shows a product list interface. At the top, there are four filter input fields: 'Filtrar por ubicación', 'Filtrar por marca', 'Filtrar por carro', and 'Filtrar por año', followed by a blue 'Buscar' button. Below the filters is a table with the following data:

#	Repuesto	Cant.	P.compra	P.venta	Marca	Carro	Años	Posición	Ubicación	Proveedor	Opciones
10210	pastilla	4	\$25,00	\$35,00	Mansuera	Chevrolet alto	2000	N	Stock	• IMPORFRENOS	 

*Imagen 14 Listado de productos*

La imagen 14 muestra la lista de productos, los cuales se pueden filtrar por ubicación de almacenamiento, marca del producto, vehículo y año, de igual manera se muestra una celda de opciones para editar o eliminar un producto.

## Editar producto

Para editar un producto deberá ubicarse en la lista de productos y presionar el icono de lápiz para abrir el formulario de registro de productos.

The screenshot shows the 'Formulario de Productos' (Product Form) with the following fields and values:

- Código: 10210
- Tipo de repuesto: pastilla
- Cantidad: 4
- Sumar cant: 0
- Restar cant: 0
- Precio de compra: 25,00
- Precio de venta: 35,00
- Marca del producto: Mansuera
- Almacenamiento: Stock
- Proveedor: IMPORFRENOS
- Vehículo al que pertenece: Chevrolet alto
- Años permitidos: 2000
- Posición de la pieza: N

At the bottom right, there are two buttons: 'Registrar' (blue) and 'Cancelar' (red).

*Imagen 15 Edición de productos*

La imagen 15 muestra el formulario de registro de productos en el cual se podrán editar los campos requeridos, en caso de aumentar o restar la cantidad se deberá ingresar dicha cantidad en los campos sumar cant o restar cant y presionar el botón registrar para guardar los cambios.

### Eliminar producto

Para eliminar un producto deberá ubicarse en la lista de productos y presionar en el icono de basurero.

¿Desea eliminar el registro 10210?



Imagen 16 Eliminación de producto

La imagen 16 muestra un mensaje indicando si desea eliminar el registro del producto con el código, tendrá la opción de confirmar o cancelar la eliminación del producto y ser dirigido automáticamente a la lista de productos.

### Registro de producto en catálogo

Para registrar un producto en el catálogo deberá presionar el Catálogo para mostrar más opciones y presionar en registrar.

Formulario de Catálogo	
Marca	<input type="text" value="Ingrese la marca del vehículo"/>
Modelo	<input type="text" value="Ingrese el modelo del vehiculo"/>
Años	<input type="text" value="Ingrese los años permitidos"/>
Posicion	<input type="text" value="Ingrese la posición de la pieza"/>
Valor x	<input type="text" value="ingrese el valor de x"/>
Valor y	<input type="text" value="ingrese el valor de y"/>
Valor z	<input type="text" value="ingrese el valor de z"/>
Imagen	<input type="button" value="Seleccionar archivo"/> Ninguno archivo selec.

Imagen 17 Registro de producto en catálogo

La imagen 17 muestra el formulario de registro de catálogo, en el cual se pedirá que ingrese la marca del vehículo, modelo del vehículo, años de los vehículos permitidos, posición del producto, valor de x que representa el ancho, valor de y que representa el valor de alto y valor de z que representa el grosor del producto, la imagen 3D previamente diseñada empleando FreeCAD y guardado en formato git y presionar en registrar.

### Listado del catálogo

Para mostrar el listado del catálogo deberá presionar en Catálogo para mostrar más opciones y presionar en listar catálogo.

Filtrar por marca del vehículo

Filtrar por código del vehículo

Buscar

#	Marca	Código	Años	Posición	Imagen	Opciones
11	Buseta Jack	N	Todos	N	catalog/busetajack.gif	
8	Chevrolet Alto	10210	2000	N	catalog/chevroletalto.gif	
2	Chevrolet Cruze	MD13301207	2007-2012	Delantera	catalog/chevroletcruzedelantera.gif	
7	Chevrolet Cruze	MD13301207	2007-2012	Posterior	catalog/chevroletcruzeposterior.gif	
13	Ford Eco Sport	D1175	2004-2009	N	catalog/FordEcoSport.gif	
14	Ford Escape	D843	2002	N	catalog/FordEscape.gif	
19	Ford Expedition	D935	Todos	Posterior	catalog/FordExpedition.gif	
10	Great Wall Hover H5	D580	Todos	Posterior	catalog/Great.gif	
18	Honda Civic 79	D177	79	N	catalog/HondaCivic79.gif	

*Imagen 18 Listado de catálogo*

La imagen 18 muestra el listado del catálogo, en el cual se podrá filtrar por la marca del vehículo y por código del vehículo, también tendrá una celda con opciones para editar o eliminar un producto del catálogo.

### Edición de producto del catálogo

Para editar un producto del catálogo deberá ubicarse en la lista del catálogo y presionar el icono del lápiz de la celda de opciones.

**Formulario de Catálogo**

Marca	<input type="text" value="Buseta Jack"/>
Modelo	<input type="text" value="N"/>
Años	<input type="text" value="Todos"/>
Posicion	<input type="text" value="N"/>
Valor x	<input type="text" value="139,00"/>
Valor y	<input type="text" value="72,00"/>
Valor z	<input type="text" value="18,00"/>
Imagen	Actualmente: catalog/busetajack.gif Modificar: <input type="button" value="Seleccionar archivo"/> Ninguno archivo selec.

*Imagen 19 Edición de producto del catálogo*

La imagen 19 muestra el registro de producto del catálogo, en el cual se podrá editar los campos requeridos y presionar en el botan registrar para guardar los cambios y ser dirigido a la lista del catálogo.

### **Eliminar producto del catálogo**

Para eliminar un producto del catálogo deberá ubicarse en la lista de productos del catálogo y presionar en el icono del basurero.

**¿Desea eliminar el registro Buseta Jack?**

*Imagen 20 Eliminar producto del catálogo*

La imagen muestra un mensaje que indica si desea eliminar el registro de producto del catálogo con la marca del vehículo, tendrá la opción de confirmar o cancelar la eliminación del producto y ser dirigido a la lista de productos del catálogo.

### **Login Usuario**

Para ingresa al sistema de administración como usuario deberá iniciar sesión con una cuenta sin permisos de administrador.



*Imagen 21 Sesión de usuario*

La imagen muestra la página principal del usuario, donde se visualizan los módulos de proveedor el cual solo podrá visualizar la lista de proveedores y el módulo de productos en el cual podrá visualizar la lista de productos y editar las cantidades a sumar o restar de un producto.

## **Recomendaciones**

Registrar un usuario y contraseña fácil de recordar

Recurrir al manual de usuario para despejar dudas sobre el manejo del sistema