**REPÚBLICA BOLIVARIANA DE VENEZUELA**

**UNIVERSIDAD PRIVADA DR. RAFAEL BELLOSO CHACÍN**

**FACULTAD DE HUMANIDADES Y EDUCACIÓN**

**ESCUELA DE INGENIERIA**

**MENCIÓN COMPUTACIÓN**



**CATÁLOGO DE PIEZAS GRUPO VAS**

**BASES DE DATOS I**

**INFORME DEL PROYECTO FINAL**

**PRESENTADO POR:**

**CARLOS RUGGIERO**

**ÁNGEL CALMEN**

**ASESORADOS POR:**

**RAFAEL ESCALONA**

**MARACAIBO, JUNIO DE 2023**

**INTRODUCCIÓN**

El Grupo Vas se dedica principalmente a la fabricación y distribución de componentes para automóviles, tanto para vehículos particulares como para vehículos comerciales y de transporte de carga.

El mismo tiene presencia en más de 70 países de todo el mundo, y cuenta con una amplia red de distribuidores y tiendas de servicio en Europa, América del Sur y Asia.

La compañía es un importante proveedor de piezas y componentes para varias marcas de automóviles, incluyendo BMW, Daimler, Ford, GM, Renault y Volkswagen, entre otros. Esta tiene una fuerte cultura de innovación y desarrollo de productos, y ha realizado importantes inversiones en investigación y desarrollo en los últimos años para desarrollar nuevas tecnologías en la automoción y mejorar sus productos existentes.

Por lo tanto, es imprescindible contar con una Base de Datos de alta calidad y orden, para que la fabricación y el manejo de los componentes sea eficiente.

A continuación, se propone una Base de Datos que satisface los requerimientos de la compañía (además de mejorar algunos aspectos del sistema de gestión en la misma) que promete un control de las piezas muy ordenado y fácil de implementar.

**DESARROLLO**

**1.- Modelo de análisis**

Como se mencionó, este proyecto busca crear un catálogo de piezas que permita a los empleados de Grupo VAS manipular de manera sencilla y eficiente los datos que se manejan en dicha empresa, para esto, se hizo un análisis para definir lo que este catálogo debe incluir, cosa que se describirá a detalle a lo largo de este informe.

Ahora, en el proyecto consiste en la creación de una base de datos y el desarrollo de una aplicación para manipularla utilizando las herramientas MySql y Python con su librería Tkinter.

Acerca de la base de datos, se decidió utilizar un enfoque relacional para poder definir tablas y operaciones, las cuales nos van a ayudar a buscar información de manera ordenada y estructurada. Estas son las tablas que se definieron:

* **Marca**: recibe las marcas existentes dentro de la compañía, dentro de estas marcas habrían modelos:
* **Modelo**: recibe los modelos del vehículo. Por ejemplo: Corolla 1.8 2030.
* **Vehículo**: recibe los vehículos. Es importante definir esta tabla, ya que se debe definir el tipo de vehículo, y cada vehículo tiene un Nro. de identificación único. En el caso de los carros sería el VIN.
* **Componentes**: recibe los componentes de los vehículos para poder identificar a donde pertenecen las piezas. Por ejemplo: el sistema del motor tiene x cantidad de piezas.
* **Pieza**: recibe las piezas de los vehículos. Por ejemplo: Cigüeñal, cauchos, etc.

**2.- Diccionario de datos:**

**2.1.- Nombre de las tablas**

|  |  |  |
| --- | --- | --- |
| Nombre de la entidad | Número de columnas | Descripción |
| marca | 3 | recibe las mar**c**as  para poder identificar  las piezas |
| modelo | 4 | recibe los modelos  para poder identificar  las piezas |
| vehiculo | 3 | recibe los vehículos  para poder identificar  las piezas |
| componentes | 4 | recibe los componentes de los vehículos  para poder identificar  las piezas |
| pieza | 3 | recibe las piezas de los vehículos |
|  |  |  |

**2.2.- Atributos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre del atributo | Tipo de datos | Pertenece a | Descripción | Restricción |
| idMarca | INT | marcas | sirve para identificar  las marcas | PRIMARY KEY |
| Nombre\_Marca | VARCHAR(45) | marcas | nombre de las marcas | NOT NULL |
| idModelo | INT | modelo | sirve para identificar  los modelos | PRIMARY KEY |
| Nombre\_Modelo | TEXT | modelo | nombre de los modelos | NOT NULL |
| Marca\_idMarca | INT | modelo | para asignar una marca a un modelo | FOREIGN KEY |
| idVehiculo | INT | vehiculo | para tener identificadores únicos para cada vehículo | PRIMARY KEY |
| Tipo\_Vehiculo | VARCHAR(100) | vehiculo | para saber qué clase de vehículo es | NOT NULL |
| Modelo\_idModelo | INT | vehiculo | para asignar un modelo a un vehículo | FOREIGN KEY |
| Año | YEAR | vehiculo | para definir el año de los vehículos | NOT NULL |
| id\_Componentes | INT | componentes | los ids unicos a cada componente | PRIMARY KEY |
| Nombre\_Componente | VARCHAR(100) | componentes | los nombres que recibirá cada componente | NOT NULL |
| Vehiculo\_idVehiculo | INT | componentes | para asignar un componente especifico a un vehículo especifico | NOT NULL |
| Vehiculo\_Modelo\_idModelo | INT | componentes | para asignar un componente especifico a un Modelo especifico | NOT NULL |
| id\_Piezas | INT | piezas | id específico para cada pieza | PRIMARY KEY |
| Nombre\_Pieza | VARCHAR(100) | piezas | nombre para cada pieza | NOT NULL |
| descripcion | TEXT | piezas | para colocar una breve descripción de sobre las piezas | NOT NULL |
| id\_Componentes | INT | piezas | para asignar una pieza a componente especifico | FOREIGN KEY |

**3.- Ingeniería de requisitos**

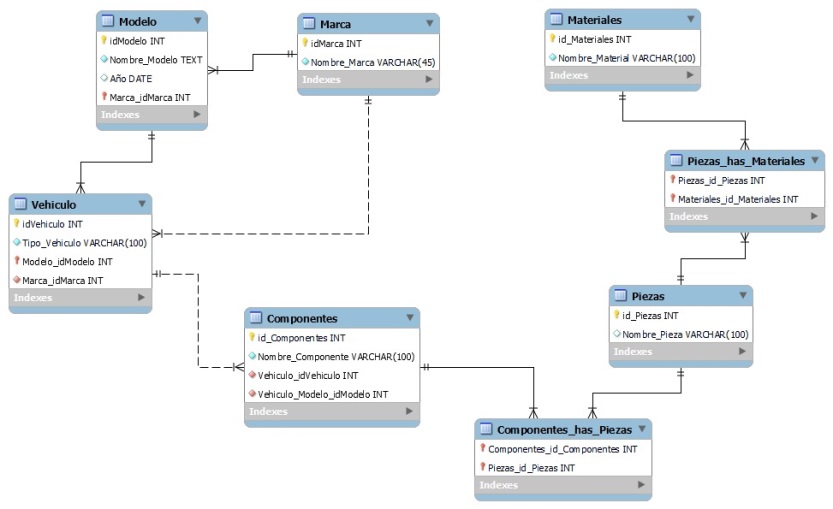
Se hizo un análisis exhaustivo de qué características y funcionalidades debería tener esta aplicación, que a continuación describimos:

* **Ingresar las piezas:** el sistema le debe permitir al usuario poder ingresar los datos relacionados a las piezas como el nombre el nombre y el id, adicional para una búsqueda más efectiva hay otros atributos como lo que podría ser la marca del carro, el modelo y su año.
* **Buscar las piezas:** El sistema debe permitir al usuario el buscar las piezas que desee con el input o información introducida en la base de datos.
* **Modificar las tablas:** el sistema debe permitir al usuario el manipular los datos de las tablas, sea añadir un registro, editarlo o eliminarlo.

Teniendo esto como base, se desarrolló una aplicación que permite acceder a las tablas de manera sencilla y ordenada, y manipularla muy fácilmente con botones, campos de entrada, imágenes, etc.

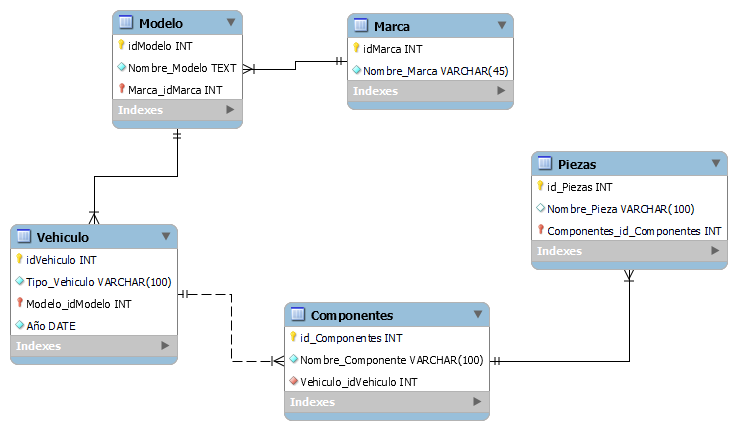
**4.- Normalización**

**4.1.- Modelo sin normalizar**



Tras haber consultado con el profesor, y haber visto los errores en este modelo lógico, entre los que está una tabla de más (materiales), una relación de más entre marca y vehículo, una cardinalidad errónea, etc., se decidió simplificar el modelo lógico de manera que quede más fácil de implementar y más efectivo.

**4.2.- Modelo normalizado**

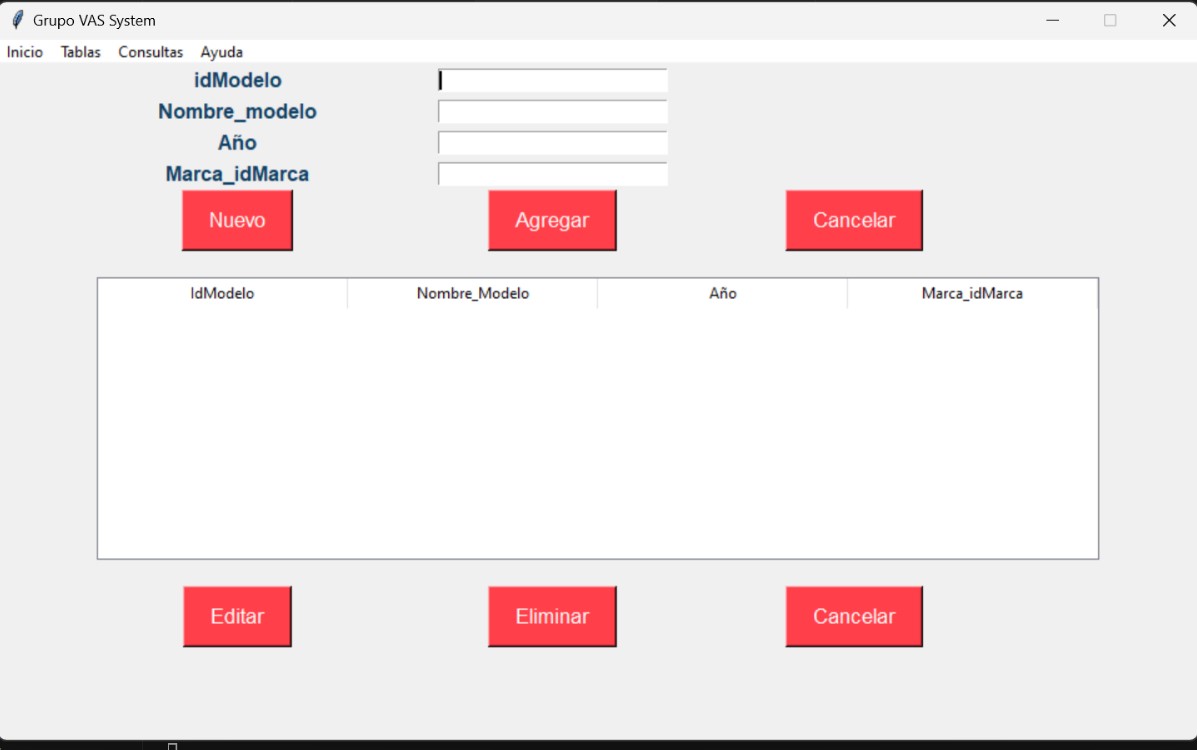


**5.- Diseño de la aplicación**

Como se mencionó anteriormente, se usó la librería Tkinter de Python para desarrollar la interfaz de la aplicación.

5.1.- Diseño de formularios

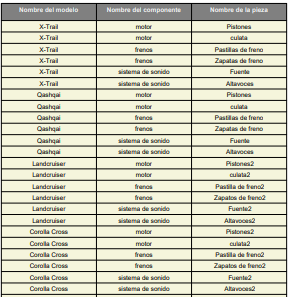
Se utilizaron los widgets Entry, y la función Entry.get() de Tkinter para permitir al usuario ingresar datos en la tabla en cuestión, aquí un ejemplo:



5.2.- Diseño de Reportes

La aplicación tendrá una función para extraer reportes en Pdf.

Aquí un ejemplo:



5.3.- Interfaz de usuario

Aquí se puede ver la pantalla inicial, y los botones de arriba del programa, que es donde se accede a las diferentes secciones del mismo.



Este será el menú para acceder a las tablas:

