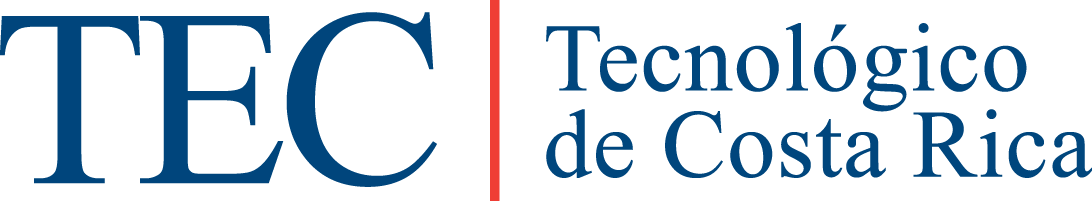
**Instituto Tecnológico de Costa Rica**

Estudiantes:

**Jonathan Calvo Obando**

2016201949

**José Navarro Acuña**

2016254241

**Randall Leonardo Román Montero**

2016115532

**Josue Suarez Campos**

2016089518

Escuela de Ingeniería en Computación

**Bases de Datos II**

Proyecto: Venta de Vehículos

Lunes 17 de junio

I Semestre, 2019

**Tabla de contenido:**

[**Introducción**](#_lbdq3ormqqcb) **3**

[**Descripción del Problema**](#_mdjt8vrugjsf) **4**

[**Diseño de solución**](#_atpieitp1tac) **6**

[Diagrama de paquetes](#_5m88blwfwj4o) 6

[Paquetes de Diseño significativos en la arquitectura](#_ykscpqaggpnv) 6

[Controlador](#_9mpv4nupgv5d) 6

[Vista](#_w59jghe5ot1q) 8

[Modelo](#_dqx1ygeuwbv5) 9

[Diagrama de clases](#_25sxml25sbzl) 10

[Vista de Datos](#_2e2k82yoef0n) 11

[Diccionario de datos](#_i6ebbvn50sqv) 13

[Modelo BrachOffice:](#_bkxy0z9ok176) 13

[Modelo de la fábrica (Parte 1):](#_swagsdj7yhx5) 16

[Modelo de la fábrica (Parte 2):](#_b4uxazp4gt8m) 18

[Modelo de Human Resources:](#_ccafvdbmkjpn) 19

[**Conclusión**](#_bx02szut4ed5) **21**

[**Además se implementan procesos almacenados que aprovechan las características únicas que el motor de búsqueda ofrece y que además agilizará la creación de un front end eficaz y seguro.**](#_ff1ydvkjjxfw) **21**

[**Anexos**](#_lbzw7ivthads) **22**

[Protocolo de seguridad](#_xdh304wnl5t3) 22

[Consideraciones tomadas](#_b7ry2qgl2bez) 22

[Credenciales para la base de datos](#_vzq44zbji1ih) 22

[Backups](#_2fpxwcoq69rk) 22

[Proceso de instalación](#_anm9eohysd8y) 22

## 

# Introducción

El presente documento tiene como objetivo documentar el proyecto que consta en la elaboración de un sistema de bases de datos que facilite el manejo de inventario, facturas y administración de una venta de vehículos, la cual deberá de aplicar las enseñanzas vistas en clase, generando tablas, procedimientos y funciones necesarios para la solución del problema descrito.

El documento contendrá distintos apartados, los cuales contienen los diagramas que se diseñaron para la creación de la Base de Datos, además de un diccionario de datos con la tipificación de los atributos utilizados y apartados con puntos claves para el desarrollo de un proyecto afín.

# Descripción del Problema

La descripción de problema es el siguiente:

Se requiere diseñar un sistema para una empresa de venta de automóviles, esta empresa fabrica diferentes tipos de autos, y entre ellos también los automóviles híbridos y eléctricos. Después de la fabricación se movilizan hacia diferentes sucursales dentro y fuera del país. Cada automóvil tiene N cantidad de características, entre ellas, precio, año, tipo de combustible, color, diferentes tipos de extras, puertas, cantidad de pasajeros , precio por extras, etc. Es necesario poder mostrar diferentes fotos de cada uno de los automóviles.

Cada sucursal tiene su propio inventario de automóviles, puede también recibir automóviles por consignación, donde se maneja una comisión por venta.

Para el funcionamiento de la aplicación se requieren varios niveles de usuarios, el usuario de consulta, el usuario que factura y el usuario administrador.

Hay una parte que es de la fábrica para poder llevar control de los empleados, funcionamiento, pedidos y despachos.

Un usuario administrador puede consultar, actualizar y asignar precios, ingresar nuevos automóviles y cambiar las características, consultar productos y existencias en todos los establecimientos, consultar ventas por sucursal x tipo de automovil x país y/o por fechas. Ventas por tipo de pago por sucursal y por fechas. Usualmente también se desea poder ver cuáles autos son los más vendidos y cuáles son los que no tienen salida, por sucursal o a nivel global.

Otra parte importante a tomar en cuenta, es que la sucursal puede otorgar crédito al comprador, por lo que cuando alguien desea que se les financie un automóvil se debe entonces cobrar un 20% de prima y luego llevar control de los diferentes pagos que debe realizar a una tasa de interés que puede ser variable.

Un usuario facturador es el encargado de realizar la factura a cada cliente. Un cliente recibe un 10% de descuento en la compra del automóvil si es un cliente que ha realizado más de 3 compras de automóviles en los últimos 5 años, el descuento puede ser variable.

Deben tomar en cuenta que si cada pago que se realiza es con tarjeta de crédito, se le retiene al establecimiento el 10% del monto de la compra como parte del impuesto de ventas que se deben entregar a hacienda cada final de mes. La compra se pueden hacer en línea también, por lo que el usuario debe poder enviar y guardar su identificación para validar que es una persona real y mayor a 18 años, no se puede vender un automóvil a personas menores de 18 años.

Si el automóvil deseado no se encuentra disponible en la sucursal, el usuario debe de poder visualizar la sucursal más cercana donde se encuentra el automóvil disponible; en caso de no estar disponible, se debe poder realizar el pedido a la fábrica, y la fábrica debe de indicar cuándo es la fecha más cercana a la entrega.

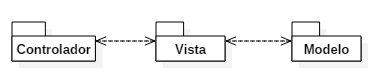
En cada consulta del producto se debe poder visualizar las diferentes fotos y características, además de la distancia del usuario hasta cada una de las sucursales donde se encuentra disponible. El usuario puede ver también los horarios de cada lugar y las personas que trabajan en el establecimiento.

## 

# Diseño de solución

## Diagrama de paquetes

El modelo a utilizar en el sistema a desarrollar es el de Controlador-Vista-Modelo. Java hace uso de tres paquetes principales. Nos pareció la mejor manera de realizarlo ya que es una forma de tener las clases bastante ordenadas y fácil de buscar lo que se necesite.

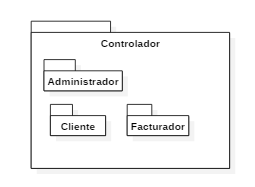


### Paquetes de Diseño significativos en la arquitectura

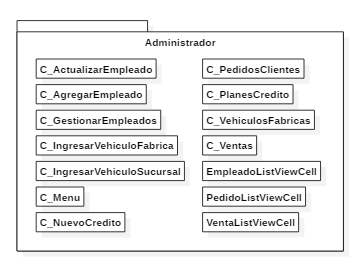
A continuación se dará una descripción de cada paquete y a contener en estos.

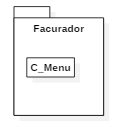
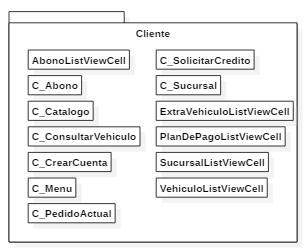
#### Controlador

Es la que se encarga de realizar la lógica del programa y clases que solo tienen métodos. Se divide en tres sub paquetes diferentes.



Los sub paquetes son:

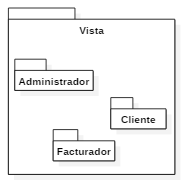




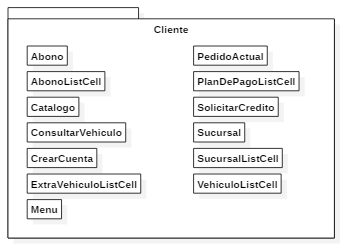
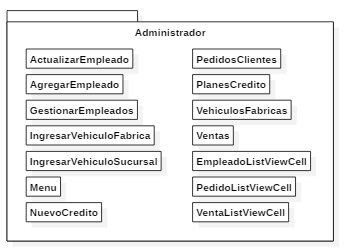
#### 

#### Vista

Este paquete contiene las clases que manejan las interfaz gráfica que se va a mostrar al usuario. Se divide en tres sub paquetes diferentes.

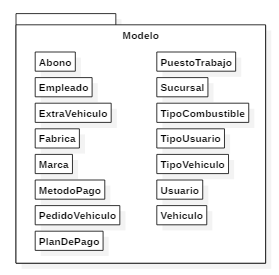


Los sub paquetes son:



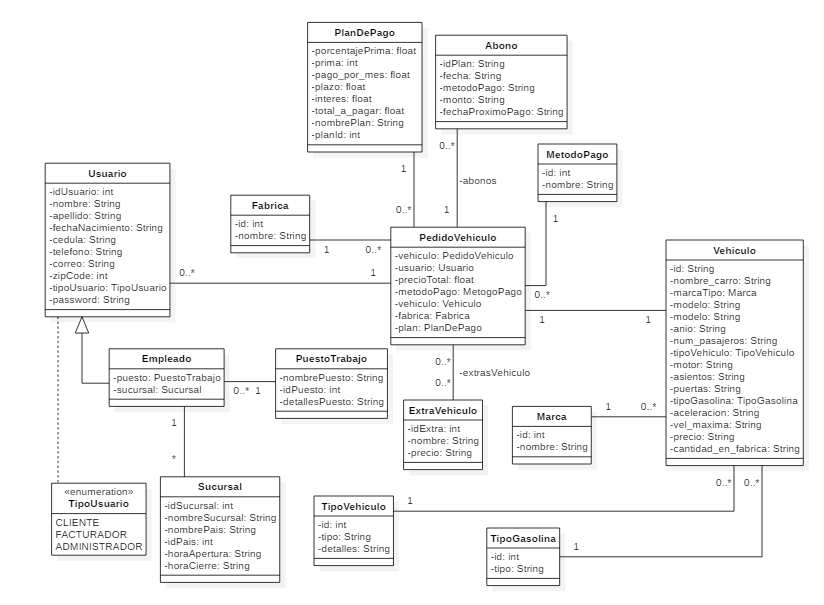
#### Modelo

Tiene las clases que van a representan a los demás datos almacenados en la base de datos. Se utilizarán estos objetos para realizar el comportamiento que el usuario utiliza.



## 

## Diagrama de clases



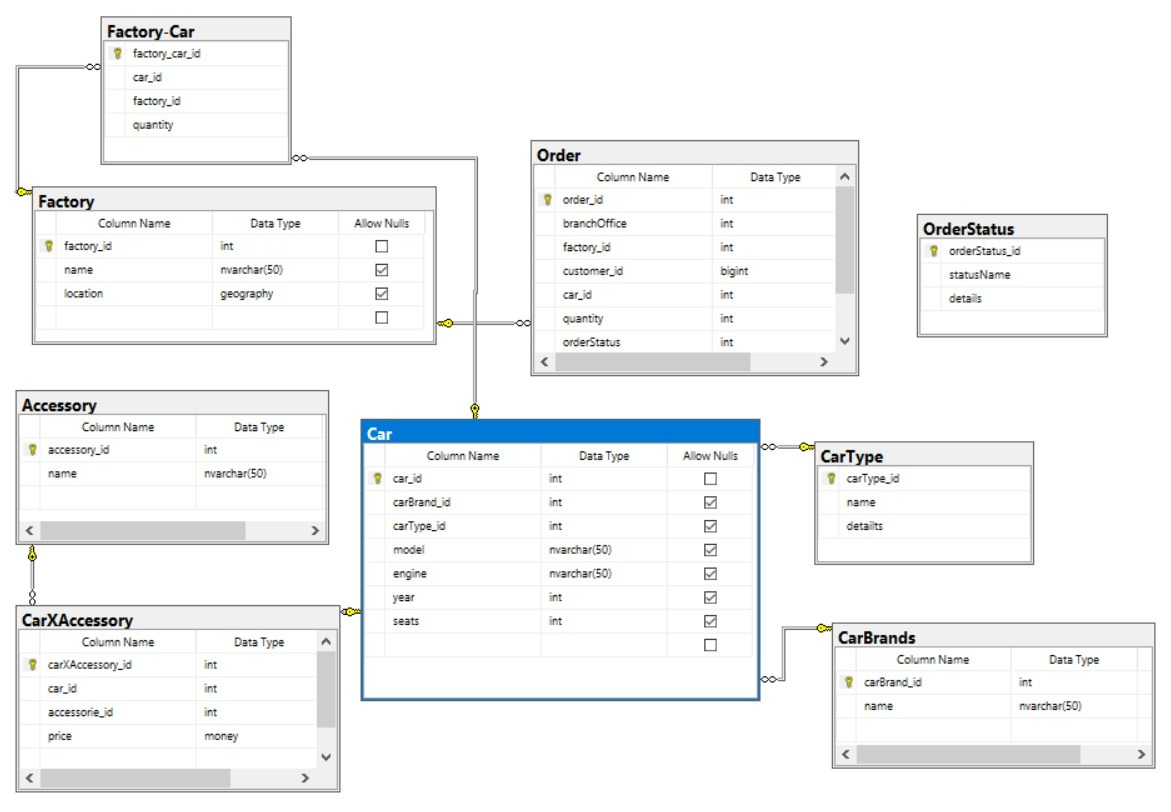
## 

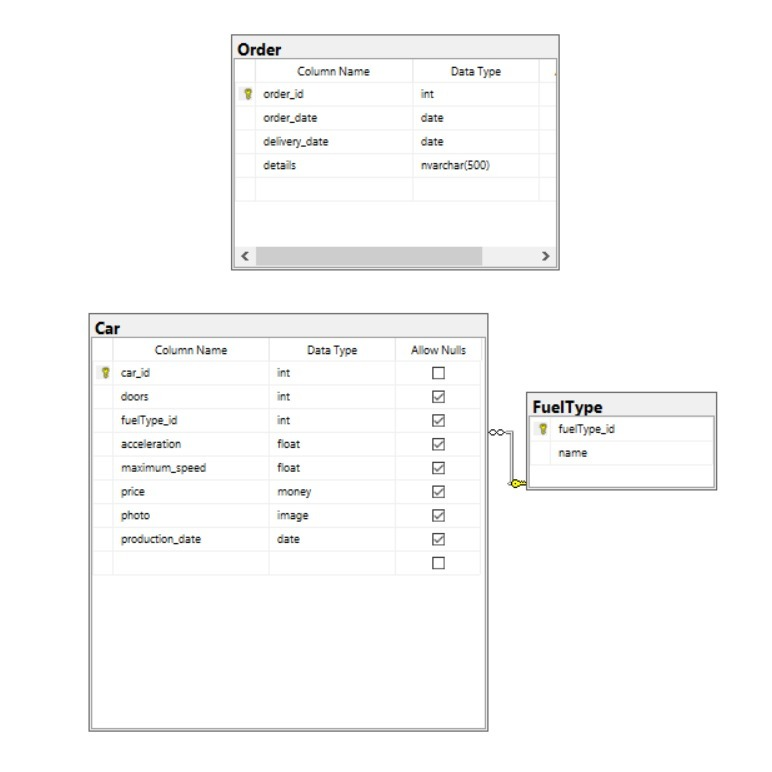
## Vista de Datos

En esta sección la manera mostramos cómo fueron almacenados los datos del sistema en la base de datos. Ya que es un sistema de bases de datos distribuidas si utilizan 4 modelos diferentes para el control de las bases de datos.

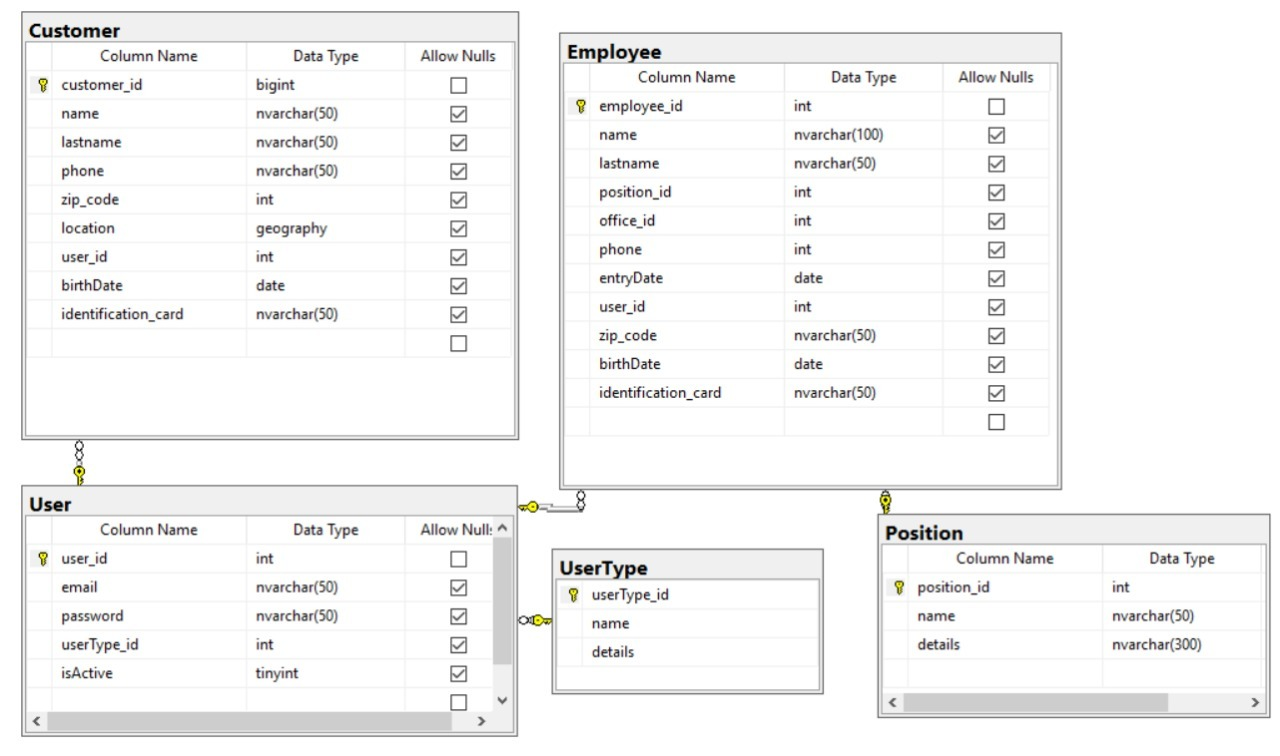
Para las Branch Office:

Para la fábrica:





Y para los recursos humanos:



## Diccionario de datos

### Modelo BrachOffice:

1. Tabla Country:

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| idCountry | NO | int |
| name | NO | varchar |

2. Tabla BranchOffice

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| branchOffice\_id | NO | int |
| name | NO | varchar |
| location | NO | geometry |
| country\_id | NO | int |
| horaApertura | NO | date |
| horaCierre | NO | date |

3. Tabla Stock

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| stock\_id | NO | int |
| office\_id | NO | int |

4. Tabla Car-Stock

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| car\_stock\_id | NO | int |
| car\_id | NO | int |
| stock\_id | NO | int |
| quantity | NO | int |

5. Tabla CarSold

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| car\_sold\_id | NO | int |
| car\_id | NO | int |

6. Tabla CarSold-Accesory

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| car\_sold\_id | NO | int |
| accesory\_id | NO | int |

7. Tabla SalesOrder

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| salesOrder\_id | NO | bigint |
| customer\_id | NO | bigint |
| order\_status | NO | int |
| order\_date | NO | date |
| paymentMethod\_id | NO | int |
| office\_id | NO | int |
| totalPrice | NO | money |
| totalPayment | NO | money |

8. Tabla OrderStatus

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| status\_id | NO | int |
| name | NO | varchar |
| details | NO | varchar |

9. Tabla SalesOrderDetails

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| orderDetails\_id | NO | bigint |
| salesOrder\_id | NO | bigint |
| car\_sold\_id | NO | int |
| quantity | NO | int |
| price | NO | money |

10. Tabla CreditGiven

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| credit\_id | NO | int |
| order\_id | NO | int |
| nextPayment\_date | NO | date |
| creditPlan\_id | NO | int |
| balance | NO | money |
| mensualPayment | NO | float |
| creditStatus | NO | int |

11. Tabla CreditGiven-Payment

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| payment\_id | NO | int |
| credit\_id | NO | int |
| payment | NO | money |
| date | NO | date |
| paymentMethod\_id | NO | int |

12. Tabla PaymentMethod

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| paymentMethod\_id | NO | int |
| name | NO | varchar |

13.Tabla CreditStatus

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| creditStatus\_id | NO | int |
| name | NO | varchar |
| details | SI | varchar |

14. Tabla CreditPlan

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| creditPlan\_id | NO | int |
| prima | NO | float |
| interest | NO | float |
| anualTerm | NO | float |
| planName | NO | varchar |

### Modelo de la fábrica (Parte 1):

1. Tabla Accesory

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| accessory\_id | NO | int |
| name | NO | varchar |

2. Tabla Car

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| car\_id | NO | int |
| carBrand\_id | NO | int |
| carType\_id | NO | int |
| model | NO | varchar |
| engine | NO | varchar |
| year | NO | int |
| seats | NO | int |

3. Tabla CarBrand

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| carBrand\_id | NO | int |
| name | NO | varchar |

4. Tabla CarType

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| carType\_id | NO | int |
| name | NO | varchar |
| details | NO | varchar |

5. Tabla CarCAccesory

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| carXAccessory\_id | NO | int |
| car\_id | NO | int |
| accessory\_id | SI | int |
| price | NO | money |

6. Tabla Factory

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| factory\_id | NO | int |
| name | NO | varchar |
| location | SI | geometry |

7. Tabla Factory-Car

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| factory\_car\_id | NO | int |
| car\_id | NO | int |
| factory\_id | NOI | int |
| quantity | SI | int |

8. Tabla Order

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| order\_id | NO | int |
| branchOffice | NO | int |
| factory\_id | NO | int |
| customer\_id | NO | bigint |
| car\_id | NO | int |
| quantity | NO | int |
| orderStatus | NO | int |

9. Tabla OrderStatus

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| orderStatus\_id | NO | int |
| statusName | NO | varchar |
| details | SI | varchar |

### Modelo de la fábrica (Parte 2):

1. Tabla Car

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| car\_id | NO | int |
| doors | NO | int |
| fuelType\_id | NO | int |
| acceleration | NO | float |
| maximum\_speed | NO | float |
| price | NO | money |
| photo | SI | image |
| production\_date | NO | date |

2. Tabla FuelType

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| fuelType\_id | NO | int |
| name | NO | varchar |

3. Tabla Order

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| order\_id | NO | int |
| order\_date | NO | date |
| delivery\_date | NO | date |
| details | SI | varchar |

### Modelo de Human Resources:

1. Tabla Customer

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| customer\_id | NO | int |
| name | NO | varchar |
| lastName | NO | varchar |
| phone | SI | varchar |
| zip\_code | SI | int |
| location | NO | geography |
| user\_id | NO | int |
| birthDate | NO | date |
| identification\_Card | NO | varchar |

2. Tabla Employee

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| employee\_id | NO | int |
| name | NO | varchar |
| lastName | NO | varchar |
| position\_id | SI | int |
| office\_id | SI | int |
| phone | NO | varchar |
| entryDate | NO | date |
| user\_id | NO | int |
| zip\_code | NO | varchar |
| birthDate | NO | date |
| identification\_card | NO | varchar |

3. Tabla Position

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| position\_id | NO | int |
| name | NO | varchar |
| details | SI | varchar |

4. Tabla User

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| user\_id | NO | int |
| email | NO | varchar |
| password | NO | varchar |
| userType\_id | NO | int |
| isActive | NO | tinyint |

5. Tabla UserType

|  |  |  |
| --- | --- | --- |
| **Nombre de la columna** | **Puede ser NULL?** | **Tipo de Dato** |
| userType\_id | NO | int |
| name | NO | varchar |
| details | SI | varchar |

# Conclusión

El proyecto ha logrado cumplir con los objetivos básicos planteados en el problema y siguiendo las especificaciones solicitadas. Se han desarrollado tablas que permiten tanto el almacenamiento como manejo de la información necesaria para la automatización de los sistemas necesarios para la empresa.

Además se implementan procesos almacenados que aprovechan las características únicas que el motor de búsqueda ofrece y que además agilizará la creación de un front end eficaz y seguro.

# Anexos

## Protocolo de seguridad

### Consideraciones tomadas

#### Credenciales para la base de datos

Considerando que para el proyecto se utilizó distribución de bases de datos se crearon credenciales únicas para las bases de datos creadas, además de que se les proporcionó con los privilegios mínimos necesarios.

#### Backups

Dentro del sistema existe un script de backup, el cual se puede ejecutar o inclusive automatizar para que cree una copia de los datos de la base de dato y del transaction log.

## Proceso de instalación

Los pasos por seguir para poder ejecutar el sistema son los siguientes:

1. Descargar e instalar SQL Server el cual puede ser adquirido mediante el siguiente enlace: <https://www.microsoft.com/es-es/sql-server/sql-serverdownloads>
2. Descargar e instalar la herramienta SQL Server Management Studio (SSMS) para monitorear, controlar y administrar las instancias de SQL. Esta herramienta se puede descargar en el siguiente enlace: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-servermanagement-studio-ssms?view=sql-server-2017>
3. Crear 6 instancias ya que cada una de ellas controla una base diferente del sistema.
4. Restaurar la copia de seguridad de las Bases de Datos del sistema, esta se encuentra en el directorio DataBase del proyecto. Una base en cada instancia,
5. Para poder abrir el proyecto se recomienda utilizar el IDE Intellij IDEA el cual puede ser descargado mediante el siguiente enlace: h[ttps://www.jetbrains.com/idea/download/#section=windows](https://www.jetbrains.com/idea/download/#section=windows)
6. Para que funcione el proyecto correctamente se debe instalar la librería que se encuentra en la carpeta lib, la cual incluye la librería mssql-jdbc7.0.0.jre8.
7. Modificar el enlace de conexión contenido por el atributo url siguiendo la siguiente estructura “jdbc:sqlserver://NombreServidor:1433;databaseName=Autos-Jx3-L;user =usuarioSQL;password=123;" En donde se debe de cambiar el parámetro NombreServidor por el nombre del servidor de su equipo.(Hacer esto para cada instancia)
8. En caso de que no tenga el protocolo TCP activado en su motor de base de datos para cada instanciaSQL diríjase a la SQL Server 2017 Configuration Manager, el cual puede ser accesado mediante el buscador de sistema operativo. Seleccione la pestaña SQL Server Network Configuration -> Protocols for SQLEXPRESS y habilite el protocolo TCP/IP. Seguidamente dé clic derecho sobre el protocolo -> Propiedades -> IP Adresses y baje hasta la sección IPAII donde debe indicar el puerto. Los puertos a ingresar son los siguientes:

* Factory Instance: 51024
* Human Resources Instance: 51171
* Branch Office 1: 50449
* Branch Office 2: 57352
* Branch Office 3: 57348

Bibliografía

Docs.microsoft.com. (2018). Create a Full Database Backup (SQL Server). [online] Available at: <https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/create-a-full-database-backup-sql-server> [Accessed 22 Jan. 2018].

Docs.microsoft.com. (2018). Execute a Stored Procedure. [online] Available at: <https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/execute-a-stored-procedure> [Accessed 22 Jan. 2018].

Medic, M. and &rarr;, V. (2018). Creando usando procedimientos almacenados CRUD - SQL Shack - articles about database auditing, server performance, data recovery, and more. [online] SQL Shack - articles about database auditing, server performance, data recovery, and more. Available at: <https://www.sqlshack.com/es/creando-usando-procedimientos-almacenados-crud/> [Accessed 22 Jan. 2018].

Smartdraw.com. (2018). Entity Relationship Diagram - Everything You Need to Know About ER Diagrams. [online] Available at: <https://www.smartdraw.com/entity-relationship-diagram/> [Accessed 22 Jan. 2018].

W3schools.com. (2018). W3Schools Online Web Tutorials. [online] Available at: <https://www.w3schools.com> [Accessed 22 Jan. 2018].

(2018). Cs.ulb.ac.be. Retrieved 19 June 2018, from <http://cs.ulb.ac.be/public/_media/teaching/infoh415/student_projects/distributed_databases.pdf>