

Lo primero que tienen que hacer es configurar el sql server para que les permita entrar en authentication

<https://www.dundas.com/support/learning/documentation/installation/how-to-enable-sql-server-authentication>

en esa pagina dice como configurarlo, traten de ver primero el application.properties;

el nombre de usuario: bases

contraseña: 12345

traten de hacerlo con eso por que si no tendremos que cambiar cada que trabajemos en una nueva version en github, ya que si pone cada quien su nombre de usuario no los dejara conectar con sql server

TENIENDO ESO CONFIGURADO, AHORA PASAN A CONECTAR EL PROYECTO

Yo me base en esta pagina, Y LO UNICO QUE LES FALTARIA HACER SEGUN YO ES CREAR LA BASE DE DATOS;

PONGANLE EL NOMBRE QUE LE PUSE EN EL APPLICATION.PROPERTIES PARA EVITAR LOS MISMOS PEDOS

<https://dzone.com/articles/configuring-spring-boot-for-microsoft-sql-server>

The screenshot shows a web browser window with the URL <https://dzone.com/articles/configuring-spring-boot-for-microsoft-sql-server>. The page is from DZone, Database Zone. The article content includes the following text:

Spring Boot sets default database properties only when you don't. So, when we configure SQL Server for use, Spring Boot won't setup the H2 database anymore.

The following data source configurations are required to configure SQL Server with Spring Boot.

**Application.properties:**

```
1 spring.datasource.url=jdbc:sqlserver://localhost;databaseName=springbootdb
2 spring.datasource.username=sa
3 spring.datasource.password=Projects@123
4 spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver
5 spring.jpa.show-sql=true
6 spring.jpa.hibernate.dialect=org.hibernate.dialect.SQLServer2012Dialect
7 spring.jpa.hibernate.ddl-auto = create-drop
```

As we are using JPA, we need to configure Hibernate for SQL Server too. Line 7 tells Hibernate to recreate the database on startup. This is definitely not the behavior we want if this was actually a production database. You can set this property to the following values: none, validate, update, create-drop.

For a production database, you probably want to use validate.

**JPA Entity**

At the bottom of the article, there is a section titled "9 Steps to Owning Your Code" with a "Download Now" button.

Y SEGUN YO SERIA TODO, EL PROYECTO YA LO CORREN Y LISTO

aHORA DE FUNCIONALIDAD, YA HICE LAS TABLAS DE CHOFER, DUEÑO, VEHICULO, USUARIOS

ESO INCLUYE QUE YA ESTAN SUS RELACIONES Y LAS TABLAS NORMALIZADAS.... OSEA QUE YA PUEDEN VER LAS RELACIONES ENTRE

CHOFER\_VEHICULO: ES ONETOONE OSEA QUE 1 CHOFER ARA 1 CARRO

DUEÑO\_VEHICULO: ES MANYTOONE UN CHOFER TIENE VARIOS CARROS PARA EL

ALGUNOS INPUTS PUEDEN QUEDAR VACIOS COMO EN EL CASO DE DIRECCION EL NUMERO INTERIOR

EN VEHICULOS LO DEL SEGURO PUEDE QUEDAR VACIO, Y EN USUARIO LA HORA DE ENTRADA Y SALIDA

A PARTIR DE AQUI FALTA LA ENTIDAD, INTERFAZ, IMPLEMENT DE MULTA Y VIAJES Y LA LOGICA QUE CONLLEVA

VEAN COMO HICE LAS RELACIONES PARA QUE NO SE COMPLIQUEN, EN VIAJES PARA APLICAR LOS DESCUENTOS PONGAN UN INPUT PARA EL USUARIO Y DE ESA BUSQUEDA TOMEN EL TIPO DE USUARIO QUE ES PARA APLICAR LOS DESCUENTOS

LAS GANANCIAS DEL CHOFER AUN NO LAS ENTIENDO PERO IGUAL ESTAN FACIL SOLO LE AGREGAN ESE CAMPO EN CHOFER Y LO VAN MAPEANDO DESDE LOS VIAJES

LA RELACION DE CHOFER\_VIAJE SERA MANYTOONE Y USUARIO\_VIAJE SERA ONETOONE

LO DE LAS RELACIONES TIENE QUE IR DENTRO DE LA ENTIDAD VIAJE..... CHEQUEN COMO HICE LOS DEMAS

ESTA FACIL NO SE COMPLIQUEN TANTO EL PEDO

YO SUBIRE A GITHUB EL DOCUMENTO DE EL SIGNIFICADO DE CADA TIPO QUE PUSE, DESPUES LE MUEVEN PARA QUE PONGAN LOS DATOS QUE VAYAN AGREGANDO

POR ULTIMO PARA POBLAR LA BASE, SOLO PODRIAMOS AGREGAR EN MULTAS POR QUE ESO ES LO QUE SE TIENE QUE VER

VERGA TAMBIEN FALTARIA EN EL BOTON DE CONSULTAS, CAMBIENLO Y AGREGUEN MAS PARA QUE PUEDA VER LOS CHOFERES REGISTRADOS, DUEÑOS, USUARIOS, VEHICULOS, ESO YA ESTA EN BLUE-RABBIT AYUDENSE DE ESO