### [Microservicios con Spring Boot, Cloud y Docker](https://www.udemy.com/course/microservicios-spring-boot-cloud/)

**19. Desarrollo de un microservicio de contactos I**

Este es un proyecto SPRING, utiliza una BD MYSQL y hace un CRUD sobre la tabla contactos, se crea un ENTITY a partir de la conexión a la BD, se ejecuta con run as /Spring Boot app y se consulta en <http://localhost:8080/contactos> se hacen las operaciones de alta, baja, cambio y actualizacion desde POSTMAN.

La bd se llama agenda2 y la tabla se llama contactos, con esta estructura

CREATE TABLE `contactos` (

`idContacto` int NOT NULL AUTO\_INCREMENT,

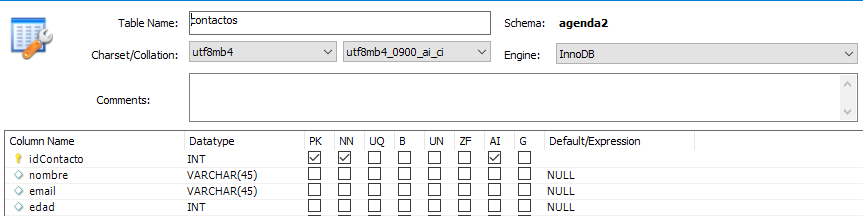
`nombre` varchar(45) DEFAULT NULL,

`email` varchar(45) DEFAULT NULL,

`edad` int DEFAULT NULL,

PRIMARY KEY (`idContacto`)

) ENGINE=InnoDB AUTO\_INCREMENT=510 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;



Se crea 04\_microservicio\_contactos

File New/ Spring starter Project :

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Se debe seleccionar la versión spring boot 2.7.7 ya que es compatible con java 1.8

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Se selecionan los starters:

* Spring Data JPA
* MySql Driver
* Spring Web

Activar JPA tools

Properties / Project Facets / JPA

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Si marca error es porque no esta selecionado JAVA, si aun asi marca error, dar clic en el error y en la siguiente pantalla en Type seleccionar: Disable Library Configuration

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ahora sobre el proyecto clic derecho y ya aparece la opción de JPA Tools

Al agregar una conexión no me aparecía la bd Mysql, así que descargue DTP toos de la pagina

<https://www.eclipse.org/datatools/downloads.php>

Una ves que lo descomprimi, lo instale cono Software desde eclipse y después de instalar ya me aparecía la conexión a MySql,

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Configuro la conexión a MySql

Si no aparece

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ya me deja seleccionar el schema y la tabla contactos

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Voya hacer una Entity, a partir de la tabla contactos, una vez selecionada, clic en next

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En esta pantalla se selecciona identity ya que la llave la debe generar con el campo que genere que es autonumerico, se pone el paquete model y listo

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Al pulsar finish paquete model nos genero la entity contacto

@Entity

@Table(name="contactos")

@NamedQuery(name="Contacto.findAll", query="SELECT c FROM Contacto c")

public class Contacto implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

private int idcontacto;

private int edad;

private String email;

private String nombre;

public Contacto() {

}

…

aquí van los geter y seters...

Ahora se crea el paquete dao

* Se crea la interface JPA AgendaJpaSpring
* Se crea la interface AgendaDao
* Se crea la implementacion AgendaDaoImpl

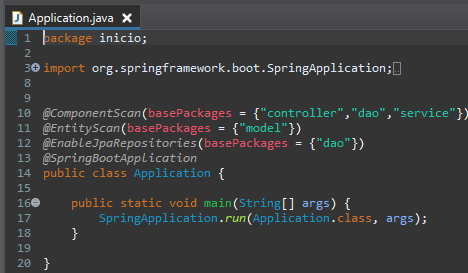
Se crea el paquete service

* Se crea la interface AgendaService
* Se crea la implementacion AgendaServiceImpl

Se crea el paquete controller

* Se crea el controler ContactosController

Se agrega ComponentScan en la clase Application.java



Se agregan las propiedades para conectar la bd en resources/application.properties

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/agenda2?serverTimezone=UTC

spring.datasource.username=root

spring.datasource.password=admin

#propiedades para que hibernate cree adecuadamente las instrucciones SQL

spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyLegacyJpaImpl

spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

#Mostrar sql en consola

spring.jpa.show-sql=true

#spring.jpa.properties.hibernate.format\_sql=true

Finalmente se ejecuta la aplicación con Run As / Spring Boot Application

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Se prueba la aplicación en postman

localhost:8080/contactos

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Actualizar contactos: localhost:8080/contactos

{

    "idcontacto": 489,

    "edad": 66,

    "email": "jersey@gmail.com",

    "nombre": "post"

}

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

Obtener un contacto: localhost:8080/contactos/476

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Borrar un contacto: localhost:8080/contactos/476

Captura de pantalla de un celular

Descripción generada automáticamente

Agregar un contacto: localhost:8080/contactos

{

        "idcontacto": 487,

        "edad": 30,

        "email": "nuevoy@gmail.com",

        "nombre": "postman"

    }

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente