**MANUAL TECNICO MODULO SERVICIO WEB PERSONAS**

El requerimiento indica crear un servicio web que permita almacenar información básica de una persona y su núcleo familiar.

1- Se Diseña el siguiente modelo de datos para el requerimiento :



-- base de datos oracle

drop table personas;

create table personas (

id\_persona number(10),

nombre varchar2(100),

correo varchar2(100),

parentesco varchar2(100),

id\_padre number(10)

);

alter table personas add constraint personas\_pk primary key (id\_persona);

alter table personas add constraint personas\_personas\_fk foreign key (id\_padre) references personas(id\_persona);

insert into personas values (1,'juan','juan@qq.com','titular',null);

insert into personas values (2,'pedro','pedro@qq.com','hijo',1);

insert into personas values (3,'ana','ana@qq.com','hija',1);

insert into personas values (4,'alfredo','alfredo@qq.com','nieto',1);

insert into personas values (5,'mariela','mariela@qq.com','nieto',1);

2- Servicio Rest : personas-service

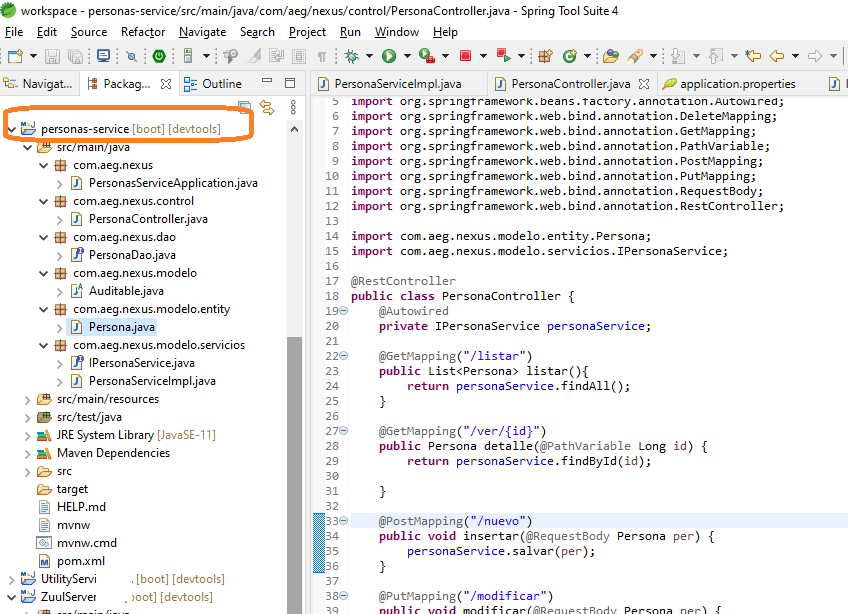
El proyecto se elabora en Spring STS tool. Con el Framework Spring Boot. Se utiliza el servidor embebido Tomcat. Se usa una base de datos Oracle para el modelo de datos y una base de datos Mysql para la auditoria persistente por medio de Zipkin. el servicio implementa las operaciones CRUD (créate, read,update,delete) sobre las entidad propuesta.

a. La arquitectura de la solución esta basada en Microservicios.

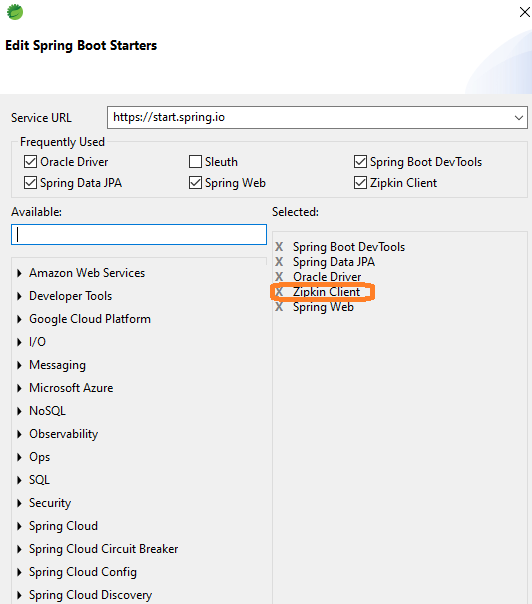
b. Se utilice el framework sprint boot.

c. Se Utilizar la herramienta IDE STS tool.

d. El servicio se desacopla mediante las capas Modelo, Dao, Servicio, Controller como se muestra en la grafica :



e. Se incluye manejo de Excepciones con registro de auditoría con el uso de Zipkin.



f. Se Incluye auditoría de los parámetros de entrada y de salida de cada servicio, es decir se guarda la traza de todas las solicitudes que se realicen a el servicio implementado, incluyendo los parámetros de consulta y la información de respuesta de cada servicio.

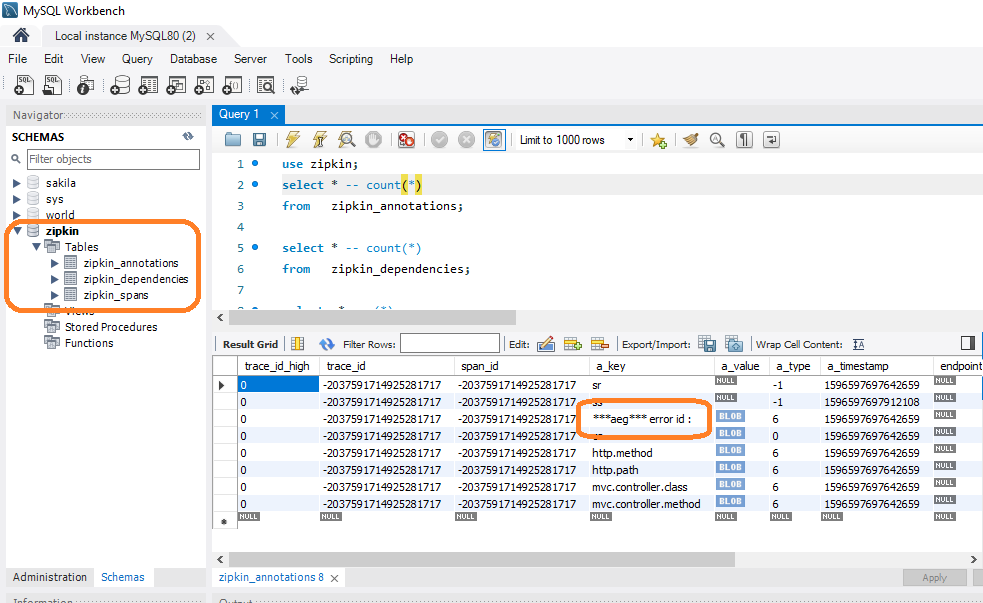
Para esto se crea la base de datos zipkin en Mysql y se crean las tablas de auditoria :



zipkin\_annotations

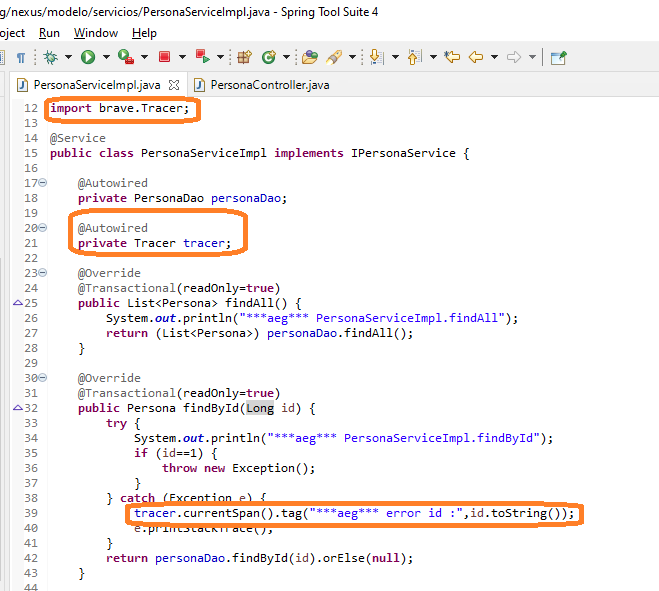
zipkin\_dependencies

zipkin\_spans



VERIFICAR EL MANEJO DE EXCEPCIONES PERSISTENTES

Para las excepciones persistentes se debe importar brave.Tracer. Se debe inyectar la dependencia y se debe invocar el método currentSpan. Como se muestra en la siguiente grafica:



Para la persistencia de las trazas se debe iniciar el servidor de Zipkin con el comando zipkin.cmd:

