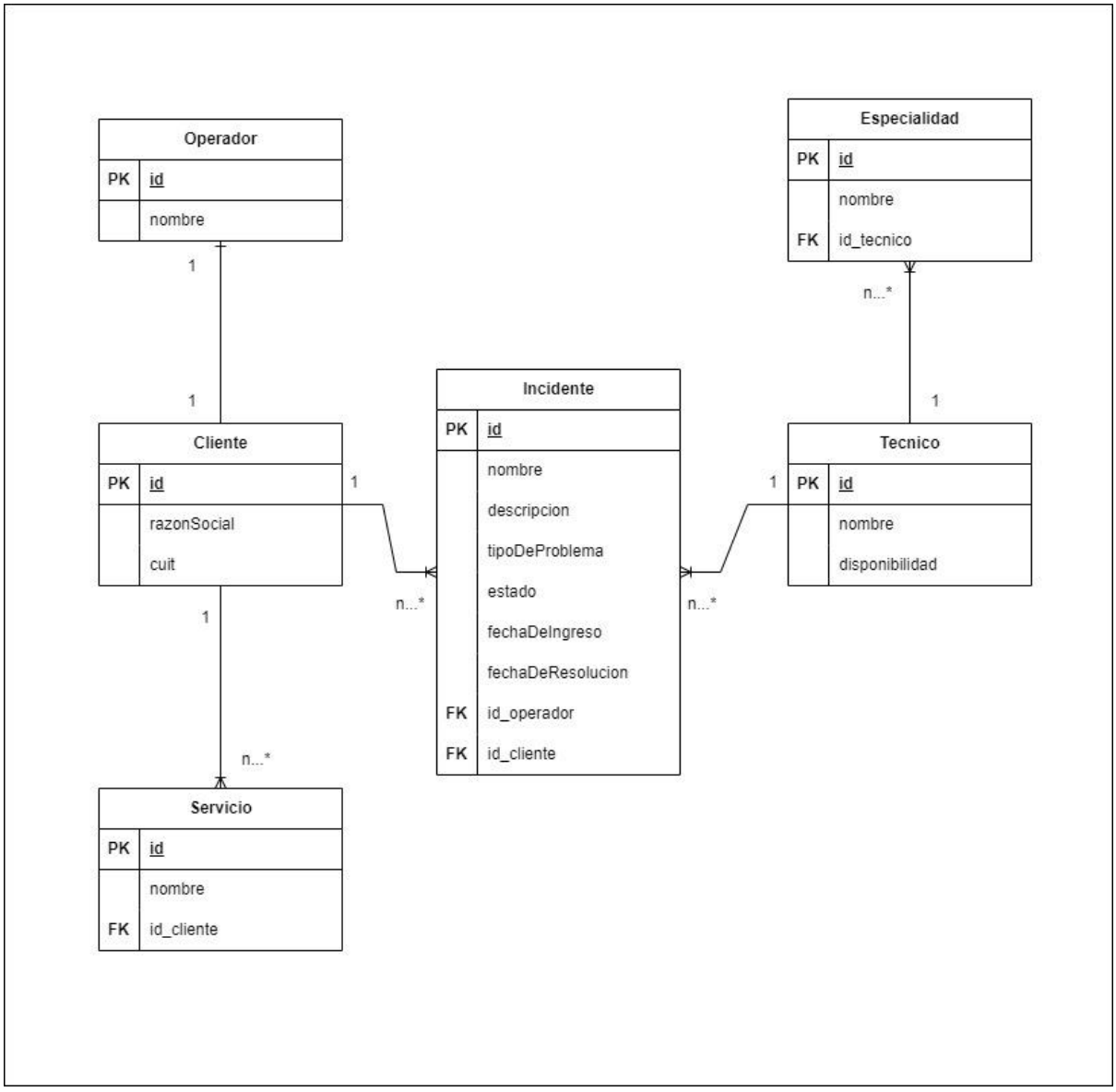
***GRUPO 1**: Baigorria, Belen - Sanchez, Facundo - Torres, Mariana*

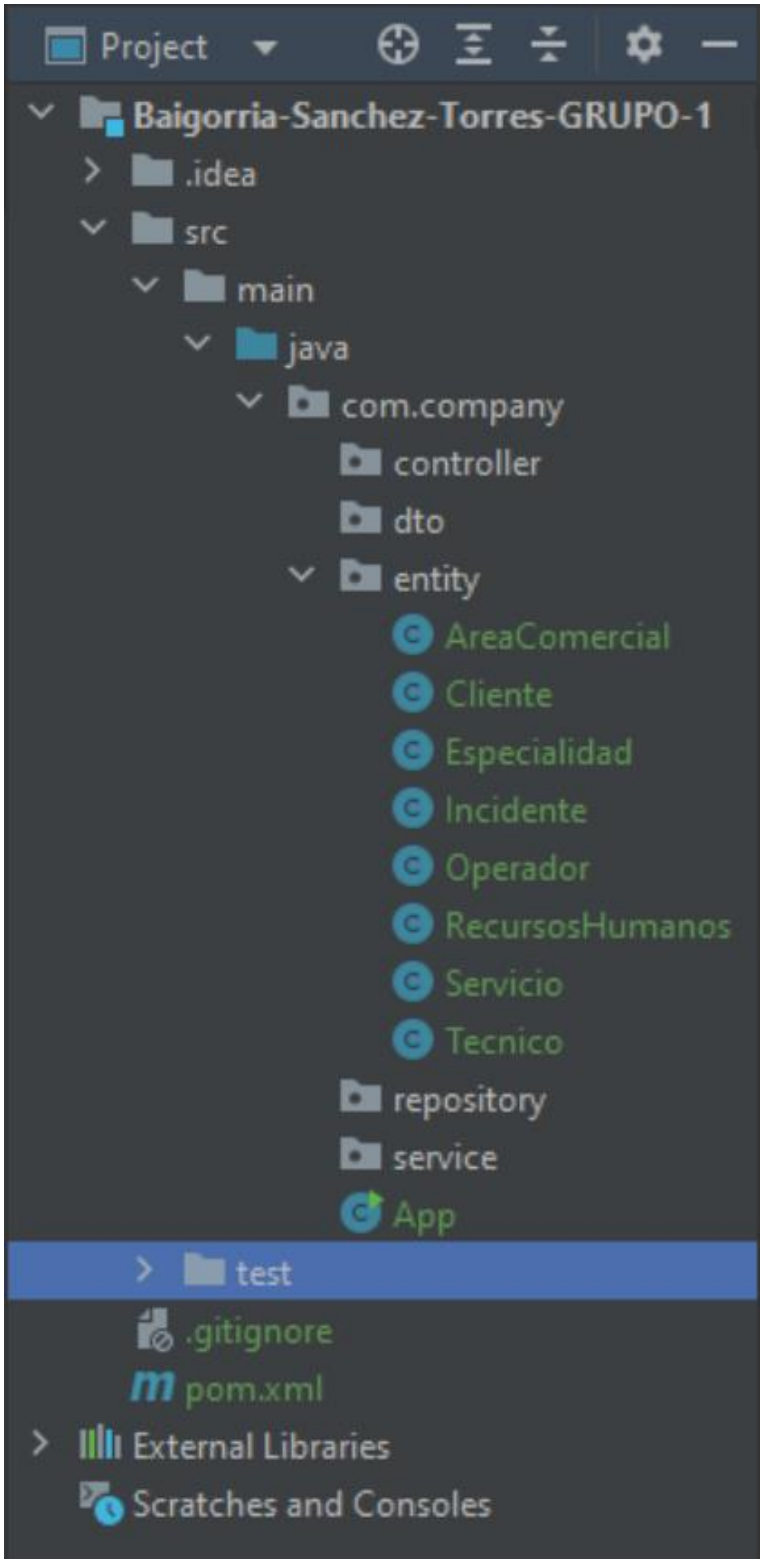
# Trabajo Práctico Integrador

## *Entrega 1*

DER – Diagrama Entidad-Relación



Estructura del proyecto



Clase Tecnico:

```
package com.company.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;
import jakarta.persistence.*;

import java.util.HashSet;
import java.util.Set;

@Entity
@Table(name = "tecnicos")
public class Tecnico {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;
    private String nombre;
    private boolean disponibilidad;

    @OneToMany(mappedBy = "tecnicos", fetch = FetchType.LAZY)
    @JsonIgnore
    private Set<Especialidad> especialidades = new HashSet<>();

    @OneToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private Incidente incidente;


    // Constructor
    public Tecnico() {
    }

    public Tecnico(String nombre, boolean disponibilidad,
Set<Especialidad> especialidades, Incidente incidente) {
        this.nombre = nombre;
        this.disponibilidad = disponibilidad;
        this.especialidades = especialidades;
        this.incidente = incidente;
    }


    // Setter y getter
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public boolean isDisponibilidad() {
        return disponibilidad;
    }

    public void setDisponibilidad(boolean disponibilidad) {
        this.disponibilidad = disponibilidad;
```

```
    }

    public Set<Especialidad> getEspecialidades() {
        return especialidades;
    }

    public void setEspecialidades(Set<Especialidad> especialidades) {
        this.especialidades = especialidades;
    }

    public Incidente getIncidente() {
        return incidente;
    }

    public void setIncidente(Incidente incidente) {
        this.incidente = incidente;
    }


    // ToString
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```

Clase Cliente:

```
package com.company.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;
import jakarta.persistence.*;

import java.util.HashSet;
import java.util.Set;

@Entity
@Table(name = "clientes")
public class Cliente {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;
    private String razonSocial;
    private int cuit;

    @OneToMany(mappedBy = "cliente", fetch = FetchType.LAZY)
    @JsonIgnore
    private Set<Servicio> servicios = new HashSet<>();

    @OneToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private AreaComercial areaComercial;
```

```
    @OneToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private Operador operador;

    @OneToMany(mappedBy = "cliente", fetch = FetchType.LAZY)
    @JsonIgnore
    private Set<Incidente> incidentes = new HashSet<>();


    // Constructor
    public Cliente() {
    }

    public Cliente(String razonSocial, int cuit, Set<Servicio>
servicios, AreaComercial areaComercial, Operador operador,
Set<Incidente> incidentes) {
        this.razonSocial = razonSocial;
        this.cuit = cuit;
        this.servicios = servicios;
        this.areaComercial = areaComercial;
        this.operador = operador;
        this.incidentes = incidentes;
    }

    // Setter y getter
    public String getRazonSocial() {
        return razonSocial;
    }

    public void setRazonSocial(String razonSocial) {
        this.razonSocial = razonSocial;
    }

    public int getCuit() {
        return cuit;
    }

    public void setCuit(int cuit) {
        this.cuit = cuit;
    }

    public Set<Servicio> getServicios() {
        return servicios;
    }

    public void setServicios(Set<Servicio> servicios) {
        this.servicios = servicios;
    }

    public AreaComercial getAreaComercial() {
        return areaComercial;
    }

    public void setAreaComercial(AreaComercial areaComercial) {
        this.areaComercial = areaComercial;
    }

    public Operador getOperador() {
        return operador;
    }
```

```
    public void setOperador(Operador operador) {
        this.operador = operador;
    }


    // ToString
    @Override
    public String toString() {
        return "Cliente{" +
                "razonSocial='" + razonSocial + '\'' +
                ", cuit=" + cuit +
                ", servicios=" + servicios +
                ", areaComercial=" + areaComercial +
                ", operador=" + operador +
                '}';
    }
}
```

Clase Incidente:

```
package com.company.entity;

import jakarta.persistence.*;

import java.time.LocalDate;

@Entity
@Table(name = "incidentes")
public class Incidente {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;
    private String nombre;
    private String descripcion;
    private String tipoDeProblema;
    private String estado;
    private LocalDate fechaDeingreso;
    private LocalDate fechaDeResolucion;

    @OneToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private Tecnico tecnico;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "id_cliente")
    private Cliente cliente;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "id_operador")
    private Operador operador;


    // Constructor
    public Incidente() {
```

```
    }

    public Incidente(String nombre, String descripcion, String
tipoDeProblema, String estado, LocalDate fechaDeingreso, LocalDate
fechaDeResolucion, Tecnico tecnico, Cliente cliente, Operador
operador) {
        this.nombre = nombre;
        this.descripcion = descripcion;
        this.tipoDeProblema = tipoDeProblema;
        this.estado = estado;
        this.fechaDeingreso = fechaDeingreso;
        this.fechaDeResolucion = fechaDeResolucion;
        this.tecnico = tecnico;
        this.cliente = cliente;
        this.operador = operador;
    }


    // Setter y getter
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public String getTipoDeProblema() {
        return tipoDeProblema;
    }

    public void setTipoDeProblema(String tipoDeProblema) {
        this.tipoDeProblema = tipoDeProblema;
    }

    public String getEstado() {
        return estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public LocalDate getFechaDeingreso() {
        return fechaDeingreso;
    }

    public void setFechaDeingreso(LocalDate fechaDeingreso) {
        this.fechaDeingreso = fechaDeingreso;
    }
```

```
    public LocalDate getFechaDeResolucion() {
        return fechaDeResolucion;
    }

    public void setFechaDeResolucion(LocalDate fechaDeResolucion) {
        this.fechaDeResolucion = fechaDeResolucion;
    }

    public Tecnico getTecnico() {
        return tecnico;
    }

    public void setTecnico(Tecnico tecnico) {
        this.tecnico = tecnico;
    }

    public Cliente getCliente() {
        return cliente;
    }

    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }

    public Operador getOperador() {
        return operador;
    }

    public void setOperador(Operador operador) {
        this.operador = operador;
    }



    // ToString
    @Override
    public String toString() {
        return "Incidente{" +
                "nombre='" + nombre + '\'' +
                ", descripcion='" + descripcion + '\'' +
                ", tipoDeProblema='" + tipoDeProblema + '\'' +
                ", estado='" + estado + '\'' +
                ", fechaDeingreso=" + fechaDeingreso +
                ", fechaDeResolucion=" + fechaDeResolucion +
                ", tecnico=" + tecnico +
                ", cliente=" + cliente +
                ", operador=" + operador +
                '}';
    }
}
```

Clase Especialidad:

```java
package com.company.entity;

import jakarta.persistence.*;

@Entity
@Table(name = "especialidades")
public class Especialidad {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;
    private String nombre;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "id_tecnico")
    private Tecnico tecnico;


    // Constructor
    public Especialidad() {
    }

    public Especialidad(String nombre, Tecnico tecnico) {
        this.nombre = nombre;
        this.tecnico = tecnico;
    }


    // Setter y getter
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public Tecnico getTecnico() {
        return tecnico;
    }

    public void setTecnico(Tecnico tecnico) {
        this.tecnico = tecnico;
    }


    // ToString
    @Override
    public String toString() {
        return "Especialidad{" +
                "nombre='" + nombre + '\'' +
                ", tecnico=" + tecnico +
                '}';
    }
}
```

Clase Servicio:

```java
package com.company.entity;

import jakarta.persistence.*;

@Entity
@Table(name = "servicios")
public class Servicio {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;
    private String nombre;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "id_cliente")
    private Cliente cliente;


    // Constructor
    public Servicio() {
    }

    public Servicio(String nombre, Cliente cliente) {
        this.nombre = nombre;
        this.cliente = cliente;
    }


    // Setter y getter
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public Cliente getCliente() {
        return cliente;
    }

    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }

    // ToString

    @Override
    public String toString() {
        return "Servicio{" +
                "nombre='" + nombre + '\'' +
                ", cliente=" + cliente +
                '}';
    }
}
```

Clase Operador:

```
package com.company.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;
import jakarta.persistence.*;

import java.util.HashSet;
import java.util.Set;

@Entity
@Table(name = "operadores")
public class Operador {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;
    private String nombre;

    @OneToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private Cliente cliente;

    @OneToMany(mappedBy = "operador", fetch = FetchType.LAZY)
    @JsonIgnore
    private Set<Incidente> incidentes = new HashSet<>();


    // Constructor
    public Operador() {
    }
    public Operador(String nombre, Cliente cliente, Set<Incidente>
incidentes) {
        this.nombre = nombre;
        this.cliente = cliente;
        this.incidentes = incidentes;
    }


    // Setter y getter
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public Cliente getCliente() {
        return cliente;
    }

    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }
```

```
    public Set<Incidente> getIncidentes() {
        return incidentes;
    }

    public void setIncidentes(Set<Incidente> incidentes) {
        this.incidentes = incidentes;
    }

    // ToString
    @Override
    public String toString() {
        return "Operador{" +
                "nombre='" + nombre + '\'' +
                ", cliente=" + cliente +
                ", incidentes=" + incidentes +
                '}';
    }

}
```

Clase Recursos Humanos:

```
package com.company.entity;
import jakarta.persistence.*;

@Entity
public class RecursosHumanos {

    @OneToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private Tecnico tecnico;


    // Constructor
    public RecursosHumanos() {
    }
    public RecursosHumanos(Tecnico tecnico) {
        this.tecnico = tecnico;
    }

    // Setter y getter
    public Tecnico getTecnico() {
        return tecnico;
    }
    public void setTecnico(Tecnico tecnico) {
        this.tecnico = tecnico;
    }

    // ToString
    @Override
    public String toString() {
        return "RecursosHumanos{" +
                "tecnico=" + tecnico +
                '}';
    }
}
```

Clase Area Comercial:

```java
package com.company.entity;

import jakarta.persistence.*;


@Entity
public class AreaComercial {


    @OneToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private Cliente cliente;




    // Constructor
    public AreaComercial() {
    }

    public AreaComercial(Cliente cliente) {
        this.cliente = cliente;
    }


    // Setter y getter
    public Cliente getCliente() {
        return cliente;
    }

    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }

    // ToString
    @Override
    public String toString() {
        return "AreaComercial{" +
                "clientes=" + cliente +
                '}';
    }
}
```

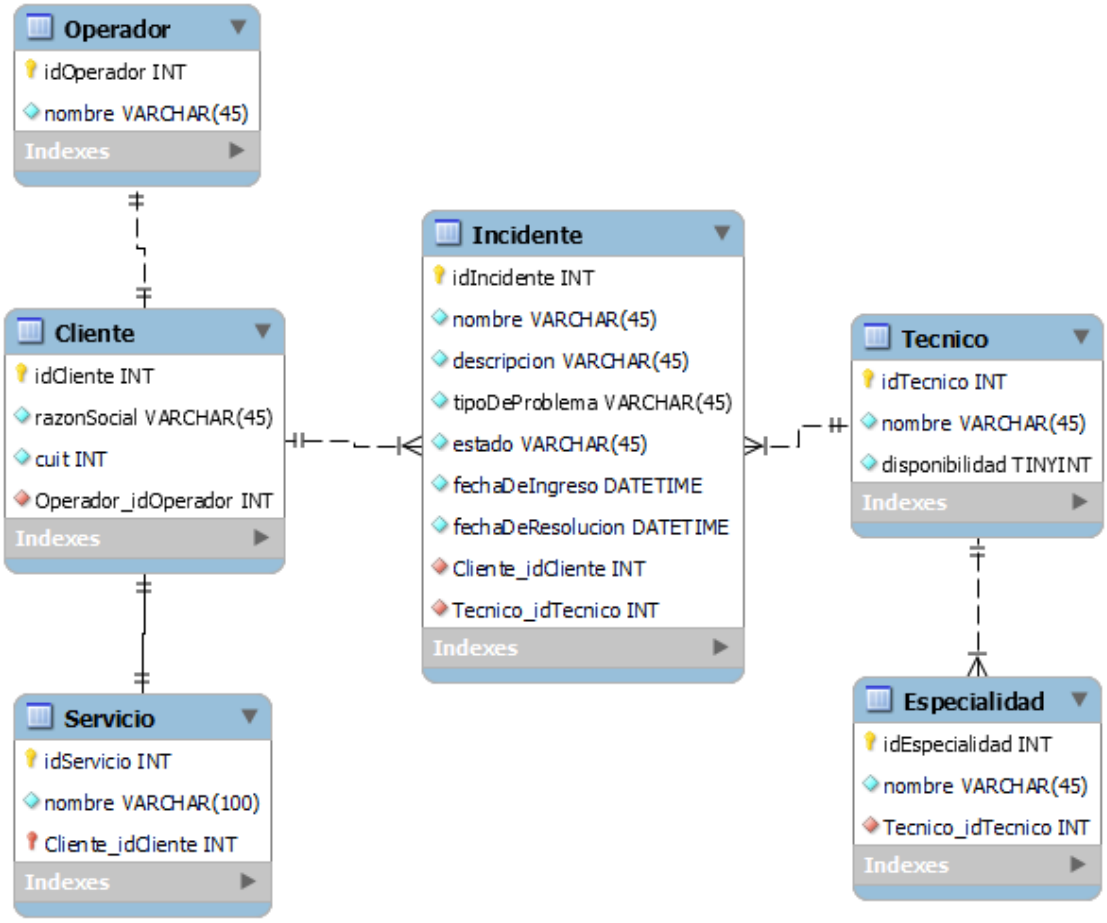
Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.company</groupId>
  <artifactId>Baigorria-Sanchez-Torres-GRUPO-1</artifactId>
  <version>1.0-SNAPSHOT</version>

  <name>Baigorria-Sanchez-Torres-GRUPO-1</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.13.2</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>6.4.0.CR1</version>
    </dependency>

    <!--MySQL-->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.33</version>
    </dependency>

    <!--jakarta-->
    <!--
https://mvnrepository.com/artifact/jakarta.annotation/jakarta.annotati
on-api -->
    <dependency>
      <groupId>jakarta.annotation</groupId>
      <artifactId>jakarta.annotation-api</artifactId>
      <version>2.1.1</version>
    </dependency>
```

```
    <!--LOG4J-->
    <dependency>
      <groupId>org.apache.logging.log4j</groupId>
      <artifactId>log4j-api</artifactId>
      <version>2.21.1</version>
    </dependency>

    <dependency>
      <groupId>org.apache.logging.log4j</groupId>
      <artifactId>log4j-core</artifactId>
      <version>2.21.1</version>
    </dependency>
      <dependency>
          <groupId>com.fasterxml.jackson.core</groupId>
          <artifactId>jackson-annotations</artifactId>
          <version>2.15.3</version>
      </dependency>


  </dependencies>

  <build>
    <pluginManagement><!-- lock down plugins versions to avoid using
Maven defaults (may be moved to parent pom) -->
      <plugins>
        <!-- clean lifecycle, see
https://maven.apache.org/ref/current/maven-
core/lifecycles.html#clean_Lifecycle -->
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
        </plugin>
        <!-- default lifecycle, jar packaging: see
https://maven.apache.org/ref/current/maven-core/default-
bindings.html#Plugin_bindings_for_jar_packaging -->
        <plugin>
          <artifactId>maven-resources-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.8.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-surefire-plugin</artifactId>
          <version>2.22.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-jar-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-install-plugin</artifactId>
          <version>2.5.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-deploy-plugin</artifactId>
          <version>2.8.2</version>
        </plugin>
```

```
        <!-- site lifecycle, see
https://maven.apache.org/ref/current/maven-
core/lifecycles.html#site_Lifecycle -->
        <plugin>
          <artifactId>maven-site-plugin</artifactId>
          <version>3.7.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-project-info-reports-plugin</artifactId>
          <version>3.0.0</version>
        </plugin>
      </plugins>
    </pluginManagement>
      <plugins>
          <plugin>
              <groupId>org.apache.maven.plugins</groupId>
              <artifactId>maven-compiler-plugin</artifactId>
              <configuration>
                  <source>8</source>
                  <target>8</target>
              </configuration>
          </plugin>
      </plugins>
  </build>
</project>
```

Creación de Base de datos:




-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

```
-- -----------------------------------------------------
-- Schema Sanchez-Baigorria-Torres-BBDD
-- -----------------------------------------------------


-- -----------------------------------------------------
-- Schema Sanchez-Baigorria-Torres-BBDD
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `Sanchez-Baigorria-Torres-BBDD` DEFAULT CHARACTER SET utf8 ;
USE `Sanchez-Baigorria-Torres-BBDD` ;


-- -----------------------------------------------------
-- Table `Sanchez-Baigorria-Torres-BBDD`.`Operador`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Sanchez-Baigorria-Torres-BBDD`.`Operador` (
  `idOperador` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idOperador`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Sanchez-Baigorria-Torres-BBDD`.`Cliente`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Sanchez-Baigorria-Torres-BBDD`.`Cliente` (
  `idCliente` INT NOT NULL AUTO_INCREMENT,
  `razonSocial` VARCHAR(45) NOT NULL,
```

```
  `cuit` INT NOT NULL,
  `Operador_idOperador` INT NOT NULL,
  PRIMARY KEY (`idCliente`),
  INDEX `fk_Cliente_Operador1_idx` (`Operador_idOperador` ASC) VISIBLE,
  CONSTRAINT `fk_Cliente_Operador1`
    FOREIGN KEY (`Operador_idOperador`)
    REFERENCES `Sanchez-Baigorria-Torres-BBDD`.`Operador` (`idOperador`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Sanchez-Baigorria-Torres-BBDD`.`Servicio`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Sanchez-Baigorria-Torres-BBDD`.`Servicio` (
  `idServicio` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(100) NOT NULL,
  `Cliente_idCliente` INT NOT NULL,
  PRIMARY KEY (`idServicio`, `Cliente_idCliente`),
  INDEX `fk_Servicio_Cliente1_idx` (`Cliente_idCliente` ASC) VISIBLE,
  CONSTRAINT `fk_Servicio_Cliente1`
    FOREIGN KEY (`Cliente_idCliente`)
    REFERENCES `Sanchez-Baigorria-Torres-BBDD`.`Cliente` (`idCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `Sanchez-Baigorria-Torres-BBDD`.`Tecnico`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Sanchez-Baigorria-Torres-BBDD`.`Tecnico` (
  `idTecnico` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(45) NOT NULL,
  `disponibilidad` TINYINT NOT NULL,
  PRIMARY KEY (`idTecnico`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Sanchez-Baigorria-Torres-BBDD`.`Incidente`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Sanchez-Baigorria-Torres-BBDD`.`Incidente` (
  `idIncidente` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(45) NOT NULL,
  `descripcion` VARCHAR(45) NOT NULL,
  `tipoDeProblema` VARCHAR(45) NOT NULL,
  `estado` VARCHAR(45) NOT NULL,
  `fechaDeIngreso` DATETIME NOT NULL,
  `fechaDeResolucion` DATETIME NOT NULL,
  `Cliente_idCliente` INT NOT NULL,
  `Tecnico_idTecnico` INT NOT NULL,
  PRIMARY KEY (`idIncidente`),
  INDEX `fk_Incidente_Cliente_idx` (`Cliente_idCliente` ASC) VISIBLE,
  INDEX `fk_Incidente_Tecnico1_idx` (`Tecnico_idTecnico` ASC) VISIBLE,
```

```
  CONSTRAINT `fk_Incidente_Cliente`
    FOREIGN KEY (`Cliente_idCliente`)
    REFERENCES `Sanchez-Baigorria-Torres-BBDD`.`Cliente` (`idCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Incidente_Tecnico1`
    FOREIGN KEY (`Tecnico_idTecnico`)
    REFERENCES `Sanchez-Baigorria-Torres-BBDD`.`Tecnico` (`idTecnico`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `Sanchez-Baigorria-Torres-BBDD`.`Especialidad`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Sanchez-Baigorria-Torres-BBDD`.`Especialidad` (
  `idEspecialidad` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(45) NOT NULL,
  `Tecnico_idTecnico` INT NOT NULL,
  PRIMARY KEY (`idEspecialidad`),
  INDEX `fk_Especialidad_Tecnico1_idx` (`Tecnico_idTecnico` ASC) VISIBLE,
  CONSTRAINT `fk_Especialidad_Tecnico1`
    FOREIGN KEY (`Tecnico_idTecnico`)
    REFERENCES `Sanchez-Baigorria-Torres-BBDD`.`Tecnico` (`idTecnico`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

GRUPO 1:

Baigorria, Belen

Sanchez, Facundo

Torres, Mariana