PROPERTIES

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

spring.datasource.url=jdbc:h2:file:~/test

spring.datasource.username=sa

spring.datasource.password=

spring.datasource.driver-class-name=org.h2.Driver

spring.jpa.show-sql=true

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

spring.jpa.defer-datasource-initialization=true

PRIORIDADE

package com.docinel.os.domain.enuns;

public enum Prioridade {

    BAIXA (0,"BAIXA"),

    MEDIA (1, "MEDIA"),

    ALTA (2, "ALTA");

    private Integer cod;

    private String descricao;

    private Prioridade(Integer cod, String descricao) {

        this.cod = cod;

        this.descricao = descricao;

    }

    public Integer getCod() {

        return cod;

    }

    public String getDescricao() {

        return descricao;

    }

    public static Prioridade toEnum(Integer cod) {

        if(cod == null) {

            return null;

        }

        for(Prioridade x : Prioridade.values()) {

            if(cod.equals(x.getCod())) {

                return x;

            }

        }

        throw new IllegalArgumentException("Prioridade inválida!" + cod);

    }

}

STATUS

package com.docinel.os.domain.enuns;

public enum Status {

    ABERTO (0,"ABERTO"),

    ANDAMENTO (1, "ANDAMENTO"),

    ENCERRADO (2, "ENCERRADO");

    private Integer cod;

    private String descricao;

    private Status(Integer cod, String descricao) {

        this.cod = cod;

        this.descricao = descricao;

    }

    public Integer getCod() {

        return cod;

    }

    public String getDescricao() {

        return descricao;

    }

    public static Status toEnum(Integer cod) {

        if(cod == null) {

            return null;

        }

        for(Status x : Status.values()) {

            if(cod.equals(x.getCod())) {

                return x;

            }

        }

        throw new IllegalArgumentException("Status inválido!" + cod);

    }

}

CLIENTE

package com.docinel.os.domain;

import java.io.Serializable;

import java.util.ArrayList;

import java.util.List;

import jakarta.persistence.OneToMany;

public class Cliente extends Pessoa implements Serializable{

    private static final long serialVersionUID = 1L;

    @OneToMany(mappedBy = "cliente")

    private List<OS> list = new ArrayList<>();

    public Cliente() {

        super();

    }

    public Cliente(Integer id, String nome, String cpf, String telefone) {

        super(id, nome, cpf, telefone);

    }

    public List<OS> getList() {

        return list;

    }

    public void setList(List<OS> list) {

        this.list = list;

    }

}

OS

package com.docinel.os.domain;

import java.time.LocalDateTime;

import java.util.Objects;

import com.docinel.os.domain.enuns.Prioridade;

import com.docinel.os.domain.enuns.Status;

import com.fasterxml.jackson.annotation.JsonFormat;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.ManyToOne;

@Entity

public class OS {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Integer id;

    @JsonFormat(pattern = "dd/MM/yyyy HH:mm")

    private LocalDateTime dataAbertura;

    @JsonFormat(pattern = "dd/MM/yyyy HH:mm")

    private LocalDateTime dataFechamento;

    private Integer prioridade;

    private String observacoes;

    private Integer status;

    @ManyToOne

    @JoinColumn(name = "cliente\_id")

    private Tecnico tecnico;

    @ManyToOne

    @JoinColumn(name = "tecnico\_id")

    private Cliente cliente;

    public OS() {

        super();

        this.setDataAbertura(LocalDateTime.now());

        this.setPrioridade(Prioridade.BAIXA);

        this.setStatus(Status.ABERTO);

    }

    public OS(Integer id, Prioridade prioridade,

            String observacoes, Status status, Tecnico tecnico, Cliente cliente) {

        super();

        this.id = id;

        this.setDataAbertura(LocalDateTime.now());

        this.prioridade = (prioridade == null) ? 0 : prioridade.getCod();

        this.observacoes = observacoes;

        this.status = (status == null) ? 0 : status.getCod();

        this.tecnico = tecnico;

        this.cliente = cliente;

    }

    public Integer getId() {

        return id;

    }

    public void setId(Integer id) {

        this.id = id;

    }

    public LocalDateTime getDataAbertura() {

        return dataAbertura;

    }

    public void setDataAbertura(LocalDateTime dataAbertura) {

        this.dataAbertura = dataAbertura;

    }

    public LocalDateTime getDataFechamento() {

        return dataFechamento;

    }

    public void setDataFechamento(LocalDateTime dataFechamento) {

        this.dataFechamento = dataFechamento;

    }

    public Prioridade getPrioridade() {

        return Prioridade.toEnum(this.prioridade);

    }

    public void setPrioridade(Prioridade prioridade) {

        this.prioridade = prioridade.getCod();

    }

    public String getObservacoes() {

        return observacoes;

    }

    public void setObservacoes(String observacoes) {

        this.observacoes = observacoes;

    }

    public Status getStatus() {

        return Status.toEnum(this.status);

    }

    public void setStatus(Status status) {

        this.status = status.getCod();

    }

    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;

    }

    @Override

    public int hashCode() {

        return Objects.hash(id);

    }

    @Override

    public boolean equals(Object obj) {

        if (this == obj)

            return true;

        if (obj == null)

            return false;

        if (getClass() != obj.getClass())

            return false;

        OS other = (OS) obj;

        return Objects.equals(id, other.id);

    }

}

PESSOA

package com.docinel.os.domain;

import java.util.Objects;

import org.hibernate.validator.constraints.br.CPF;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public abstract class Pessoa {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Integer id;

    private String nome;

    @CPF

    private String cpf;

    private String telefone;

    public Pessoa() {

        super();

        // TODO Auto-generated constructor stub

    }

    public Pessoa(Integer id, String nome, String cpf, String telefone) {

        super();

        this.id = id;

        this.nome = nome;

        this.cpf = cpf;

        this.telefone = telefone;

    }

    public Integer getId() {

        return id;

    }

    public void setId(Integer id) {

        this.id = id;

    }

    public String getNome() {

        return nome;

    }

    public void setNome(String nome) {

        this.nome = nome;

    }

    public String getCpf() {

        return cpf;

    }

    public void setCpf(String cpf) {

        this.cpf = cpf;

    }

    public String getTelefone() {

        return telefone;

    }

    public void setTelefone(String telefone) {

        this.telefone = telefone;

    }

    @Override

    public int hashCode() {

        return Objects.hash(cpf, id);

    }

    @Override

    public boolean equals(Object obj) {

        if (this == obj)

            return true;

        if (obj == null)

            return false;

        if (getClass() != obj.getClass())

            return false;

        Pessoa other = (Pessoa) obj;

        return Objects.equals(cpf, other.cpf) && Objects.equals(id, other.id);

    }

}

TECNICO

package com.docinel.os.domain;

import java.io.Serializable;

import java.util.ArrayList;

import java.util.List;

import jakarta.persistence.Entity;

import jakarta.persistence.OneToMany;

@Entity

public class Tecnico extends Pessoa implements Serializable {

    private static final long serialVersionUID = 1L;

    @OneToMany(mappedBy = "tecnico")

    private List<OS> list = new ArrayList<>();

    public Tecnico() {

        super();

    }

    public Tecnico(Integer id, String nome, String cpf, String telefone) {

        super(id, nome, cpf, telefone);

    }

    public List<OS> getList() {

        return list;

    }

    public void setList(List<OS> list) {

        this.list = list;

    }

}