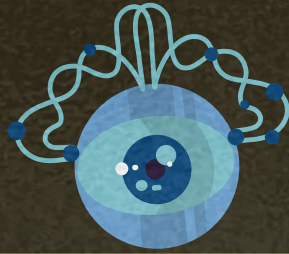


# Crud Java Spring

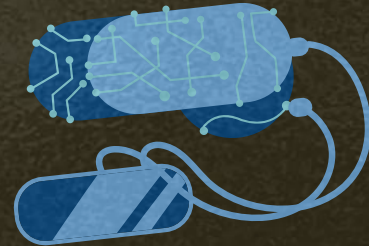


**Professor:** Welliton Cunha

**Alunos:** Gustavo Cunha e  
Luiz Guimarães



# Sistema de Cadastro de Doadores de Sangue

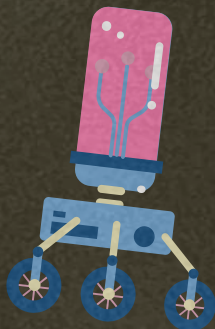
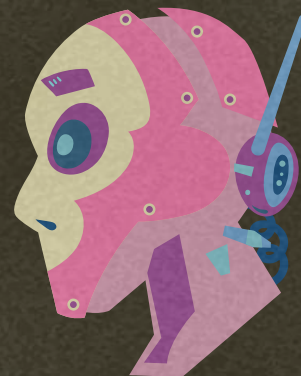




# Sistema

## Descrição

Um sistema que permite a gestão de doadores de sangue, facilitando a conexão entre bancos de sangue e doadores.



## Funcionalidades CRUD

Cadastro de doadores, atualização de informação de doadores, listagem e remoção



# Tecnologias Utilizadas

Visual Studio Code

Postman

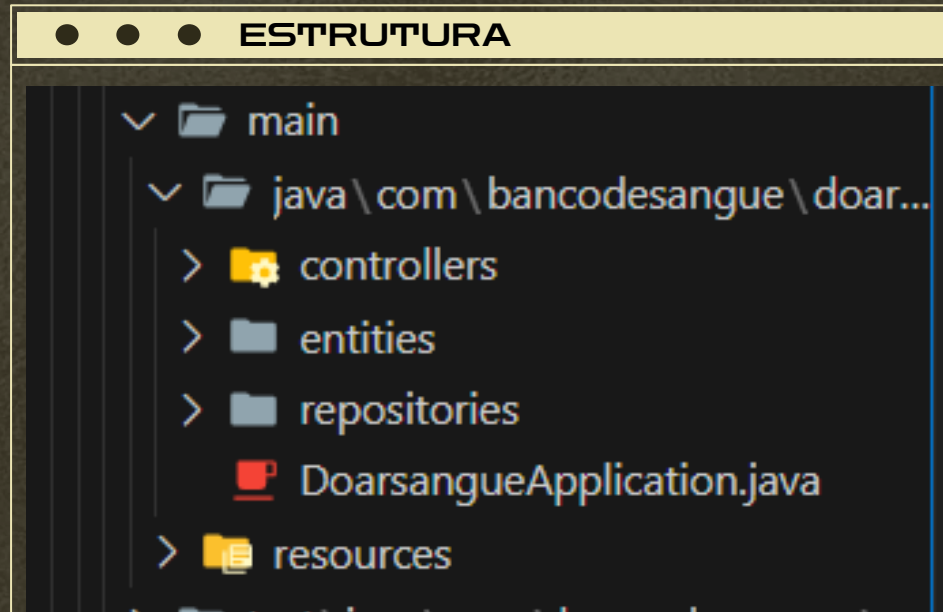
JPA

H2

Github



# Códigos



# Códigos

## CONTROLES

```
@RestController
@RequestMapping(value = "/doadores")
public class DoadorController {

    @Autowired //instanciar automaticamente
    private DoadorRepository doadorRepository;

    @GetMapping
    public List<Doador> findAll(){
        return doadorRepository.findAll();
    }

    @GetMapping(value =("/{id}")
    public ResponseEntity<Doador> findById(@PathVariable Long id){
        Optional<Doador> doador = doadorRepository.findById(id);
        return doador.map(response -> ResponseEntity.ok().body(response)).orElse(ResponseEntity.notFound().build());
    }

    @PostMapping
    public Doador insert(@RequestBody Doador doador){
        return doadorRepository.save(doador);
    }

    @PutMapping("/{id}")
    public ResponseEntity<Doador> update(@PathVariable Long id, @RequestBody Doador novosDoador){
        Optional<Doador> doadorOptional = doadorRepository.findById(id);

        if (doadorOptional.isPresent()) {
            Doador doador = doadorOptional.get();
            doador.setName(novosDoador.getName());
            doador.setCpf(novosDoador.getCpf());
            doador.setIdade(novosDoador.getIdade());
            doador.setPeso(novosDoador.getPeso());
            doador.setTiposanguineo(novosDoador.getTiposanguineo());
            doador.setLocaldoacao(novosDoador.getLocaldoacao());
            Doador resultado = doadorRepository.save(doador);
            return ResponseEntity.ok(resultado);
        } else{
            return ResponseEntity.notFound().build();
        }
    }
}
```

```
        doador.setIdade(novosDoador.getIdade());
        doador.setPeso(novosDoador.getPeso());
        doador.setTiposanguineo(novosDoador.getTiposanguineo());
        doador.setLocaldoacao(novosDoador.getLocaldoacao());
        Doador resultado = doadorRepository.save(doador);
        return ResponseEntity.ok(resultado);
    } else{
        return ResponseEntity.notFound().build();
    }
}

@DeleteMapping("/{id}")
public ResponseEntity<Doador> delete(@PathVariable Long id){
    Optional<Doador> doadorOptional = doadorRepository.findById(id);

    if (doadorOptional.isPresent()) {
        doadorRepository.deleteById(id);
        return ResponseEntity.ok().build();
    } else{
        return ResponseEntity.notFound().build();
    }
}
```

# Códigos

## ENTITIES

```
import jakarta.persistence.Id;
// import jakarta.persistence.JoinColumn;
// import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;

@Entity
@Table(name = "doador")
public class Doador {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nome;
    private String cpf;
    private int idade;
    private String peso;
    private String tiposanguineo;
    private String localdoacao;

    // @ManyToOne
    // @JoinColumn(name = "bancodesangue_id")
    // private Bancodesangue bancodesangue;

    public Long getId() {
        return id;
    }
    public void setId(Long 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) {
```

```
    }
    public void setCpf(String cpf) {
        this.cpf = cpf;
    }
    public int getIdade() {
        return idade;
    }
    public void setIdade(int idade) {
        this.idade = idade;
    }
    public String getPeso() {
        return peso;
    }
    public void setPeso(String peso) {
        this.peso = peso;
    }
    public String getTiposanguineo() {
        return tiposanguineo;
    }
    public void setTiposanguineo(String tiposanguineo) {
        this.tiposanguineo = tiposanguineo;
    }
    public String getLocaldoacao() {
        return localdoacao;
    }
    public void setLocaldoacao(String localdoacao) {
        this.localdoacao = localdoacao;
    }
}
```

# Códigos

## REPOSITORIES E MAIN

```
package com.bancodesangue.doarsangue.repositories;

import org.springframework.data.jpa.repository.JpaRepository;

import com.bancodesangue.doarsangue.entities.Doador;

public interface DoadorRepository extends JpaRepository<Doador,
Long>{
}
```

```
package com.bancodesangue.doarsangue;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DoarsangueApplication {

    public static void main(String[] args) {
        SpringApplication.run(primarySource:DoarsangueApplication.
class, args);
    }
}
```





# GITHUB

[CLIQUE NESTE TEXTO PARA ABRIR O  
DIRETÓRIO DO PROJETO NO GITHUB](#)