

HEMOVIDAS



Doe sangue.
Doe vidas!

•
EQUIPE: DANIEL DA SILVEIRA
MOREIRA, JULIA VELOSO DIAS
E ERICK GONZAGA

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS - ICEI

The slide features four decorative red circles of different sizes. One large circle is in the top right, another large one is in the bottom left, and two smaller ones are positioned in the top left and middle right areas.

01.

ECLIPSE E

BACK-END

A large, semi-transparent red circle in the top-left corner of the slide.

PASTAS E

CÓDIGOS ECLIPSE

A small, semi-transparent red circle located below the first letter of the word "CÓDIGOS".Two semi-transparent red circles in the bottom-right corner: a smaller one partially overlapping the word "ECLIPSE" and a larger one below it.

ORGANIZAÇÃO ECLIPSE

- ▼ app
 - ▼ Aplicacao.java
 - > Aplicacao
- ▼ DAO
 - ▼ DAO.java
 - > DAO
 - ▼ Principal.java
 - > Principal
 - ▼ ProdutoDAO.java
 - > ProdutoDAO
 - ▼ Usuario.java
 - > Usuario
- ▼ model
 - ▼ Produto.java
 - > Produto
- ▼ service
 - ▼ ProdutoService.java
 - > ProdutoService

- ▼ src/main/resources
 - ▼ public
 - index.html
 - dump.sql
- > src/test/java
- > src/test/resources
- > JRE System Library [JavaSE-11]
- > Maven Dependencies
- > src
- > target
 - form.html
 - pom.xml

CRUDS

```
function finalizarCadastroUsuarioPessoa() {
  let idUsuarioPessoa = proximoIdUsuarioPessoa();
  let strEmail = document.getElementById('email').value;
  let strSenha = document.getElementById('senha').value;
  let strCPF = document.getElementById('cpf').value;
  let strFoto = document.getElementById('foto').value;
  let strNome = document.getElementById('nome_completo').value;
  let strTelefone = document.getElementById('telefone_pessoal').value;
  let strCep = document.getElementById('cep').value;
  let strCidade = document.getElementById('cidade').value;
  let strBairro = document.getElementById('bairro').value;
  let strRua = document.getElementById('rua').value;
  let strNumero = document.getElementById('numero').value;
  let strComplemento = document.getElementById('complemento').value;
  let strTipoSanguineo = pegaTipoSanguineo();

  let novoUsuarioPessoa = {
    idUsuarioPessoa: idUsuarioPessoa,
    email: strEmail,
    senha: strSenha,
    cpf: strCPF,
    foto: strFoto,
    nome_completo: strNome,
    telefone_pessoal: strTelefone,
    cep: strCep,
    cidade: strCidade,
    endereco: { bairro: strBairro, rua: strRua, numero: strNumero, complemento: strComplemento },
    tipo_sanguineo: strTipoSanguineo,
    tipos_sanguineos_compativeis_receber: "0-",
    idAgendamento: null
  };
}
```

GET

```
function incluirContato () {
  // Ler os dados do localStorage
  let objDados = leDados();

  // Incluir um novo contato
  let strUnidade = document.getElementById('campoUnidade').value;
  let strData = document.getElementById('campoData').value;
  let strHora = document.getElementById('campoHora').value;
  let strNome = document.getElementById('campoNome').value;
  let strEmail = document.getElementById('campoEmail').value;
  let strCpf = document.getElementById('campoCpf').value;
  let strNascimento = document.getElementById('campoNascimento').value;
  let strRg = document.getElementById('campoRg').value;
  let strContato = document.getElementById('campoCelular').value;

  let novoContato = {
    unidade: strUnidade,
    data: strData,
    hora: strHora,
    nome: strNome,
    email: strEmail,
    cpf: strCpf,
    nascimento: strNascimento,
    rg: strRg,
    contato: strContato,
  };

  objDados.contatos.push (novoContato);
}
```

VALIDAR

CRUDS

```
public class Produto {  
    private int id;  
    private String email;  
    private String senha;  
    private String cpf;  
    private String nome;  
    private String telefone;  
    private String cep;  
    private String bairro;  
    private String rua;  
    private String numero;  
    private String tpSanguineo;
```

```
    public Produto() {  
        id = -1;  
        email = "";  
        senha = "";  
        cpf = "";  
        nome = "";  
        telefone = "";  
        cep = "";  
        bairro = "";  
        rua = "";  
        numero = "";  
        tpSanguineo = "";
```

model

```
    public int getID() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
  
    public void setEmail(String email) {  
        this.email = email;  
    }
```

```
    public boolean update(Produto produto) {  
        boolean status = false;  
        try {  
            String sql = "UPDATE produto SET email = '" + produto.  
                + "senha = '" + produto.getSenha() + ", "  
                + "cpf = '" + produto.getCpf() ;  
            PreparedStatement st = conexao.prepareStatement(sql);  
  
            st.executeUpdate();  
            st.close();  
            status = true;  
        } catch (SQLException u) {  
            throw new RuntimeException(u);  
        }  
        return status;  
    }
```

DAO

front-end



CADASTRO PARA DOAÇÃO

Informações para Login

Email: * Senha: * Confirme sua senha: * [Mostrar senha](#) ☐

Informações do Usuário

CPF: * Foto (URL): Nome Completo: *

Telefone de Contato: * CEP: Cidade: *

Endereço

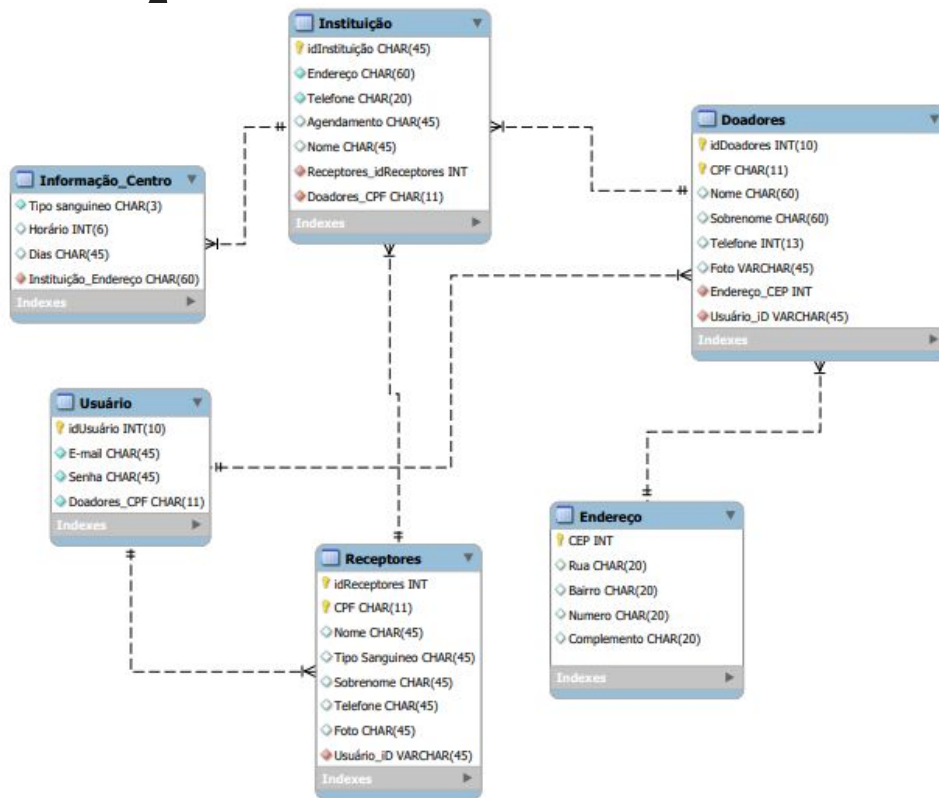
Bairro: * Rua: * Número: * Complemento:

Tipo Sanguíneo *

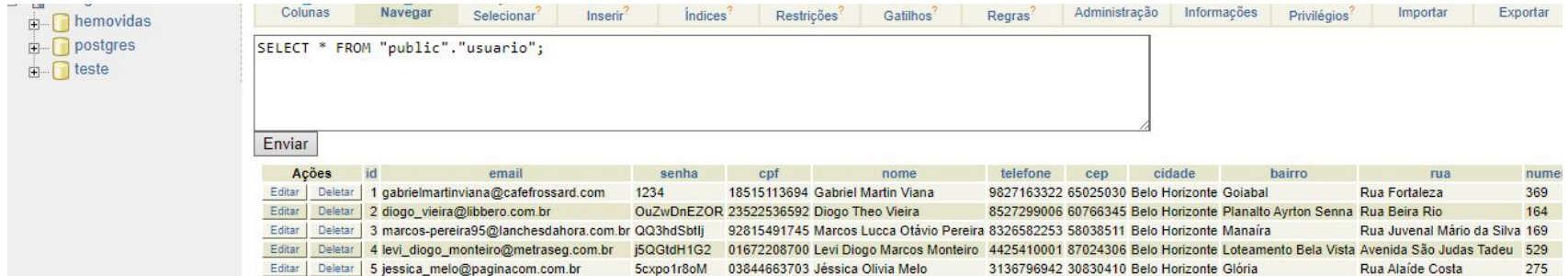
A+ ☐ A- ☐ B+ ☐ B- ☐ AB+ ☐ AB- ☐ O+ ☐ O- ☐

(*)Campos obrigatórios

notação pé de galinha



banco de dados

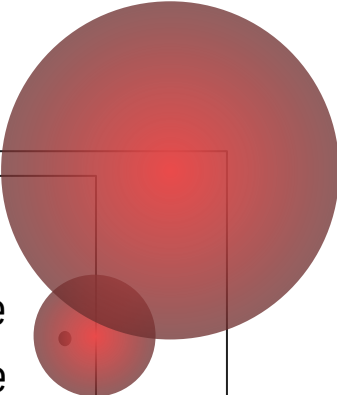


The screenshot shows a database management tool interface. On the left, a sidebar lists three databases: 'hemovidas', 'postgres', and 'teste'. The main area has a top menu bar with options: 'Colunas', 'Navegar', 'Selecionar', 'Inserir', 'Índices', 'Restrições', 'Gatilhos', 'Regras', 'Administração', 'Informações', 'Privilégios', 'Importar', and 'Exportar'. Below the menu is a text input field containing the SQL query: `SELECT * FROM "public"."usuario";`. A button labeled 'Enviar' is positioned below the query field. The result of the query is displayed as a table with 13 columns: 'Ações', 'id', 'email', 'senha', 'cpf', 'nome', 'telefone', 'cep', 'cidade', 'bairro', 'rua', and 'nume'. The table contains 5 rows of user data.


Ações	id	email	senha	cpf	nome	telefone	cep	cidade	bairro	rua	nume
Editar Deletar	1	gabrielmartinviana@cafeafrossard.com	1234	18515113694	Gabriel Martin Viana	9827163322	65025030	Belo Horizonte	Goiabab	Rua Fortaleza	369
Editar Deletar	2	diogo_vieira@libbero.com.br	OuZwDnEZOR	23522536592	Diogo Theo Vieira	8527299006	60766345	Belo Horizonte	Planalto	Ayrton Senna	164
Editar Deletar	3	marcos-pereira95@lanchesdahora.com.br	QQ3hdSbtIj	92815491745	Marcos Lucca Otávio Pereira	8326582253	58038511	Belo Horizonte	Manaira	Rua Juvenal Mário da Silva	169
Editar Deletar	4	levi_diogo_monteiro@metraseg.com.br	j5QGtdH1G2	01672208700	Levi Diogo Marcos Monteiro	4425410001	87024306	Belo Horizonte	Loteamento Bela Vista	Avenida São Judas Tadeu	529
Editar Deletar	5	jessica_melo@paginacom.com.br	5cxpo1r8oM	03844663703	Jéssica Olivia Melo	3136796942	30830410	Belo Horizonte	Glória	Rua Alaíde Costa	275

5 linha(s)

[Expandir](#) | [Inserir](#) | [Atualizar](#)



O centro de doação identificado por principalmente pelo seu id e endereço, adicionado ao nome, telefone e data para coleta, esta que contém dia, horário e tipo sanguíneo necessário. Por meio de um cadastro de contendo, email e senha, é a principal forma de receber doadores e receptores que são identificados em particular com o id, tipo sanguíneo, CPF, nome completo, telefone de contato e cidade. Adicionalmente deve ser incluída uma foto, o CEP e o endereço completo, contendo rua, bairro, número e complemento.





02.

SISTEMA

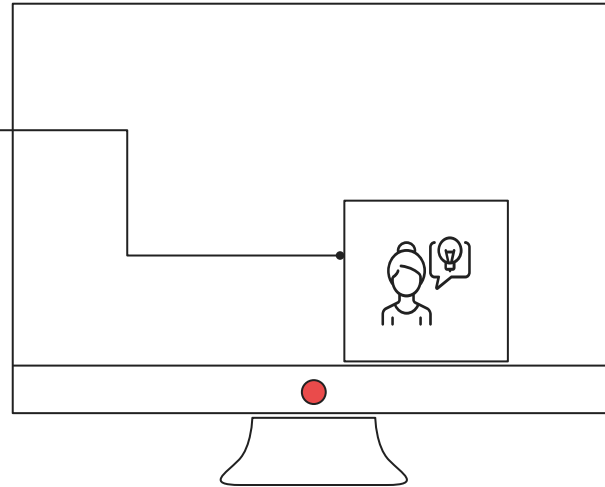
INTELIGENTE

The slide features three decorative red circles of different sizes. One large circle is in the top-left corner, partially overlapping the text. A smaller circle is positioned just below and to the right of it. Another large circle is in the bottom-right corner. A medium-sized circle is located between the middle and bottom lines of text on the right side.

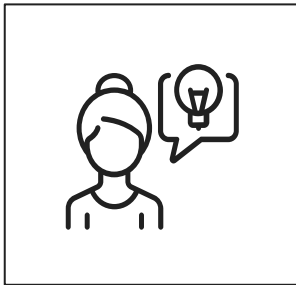
RECURSOS DE INTELIGÊNCIA ARTIFICIAL

CHATBOT

Iremos implementar a IA por meio de um chatbot que facilite para o usuário a consulta de centros de doação dentro do seu bairro e os dias e horários para doação.



BOT QUESTIONS & ANSWER



- O QnA Maker é um serviço de NLP (processamento de idioma natural) baseado em nuvem que permite criar uma camada de conversa natural sobre seus dados. Ele é usado para encontrar a resposta mais apropriada para qualquer entrada da KB (base de dados de conhecimento) personalizada de informações.

PRÓS DO USO DE QnA



**QnA não
armazena
dados do
cliente**



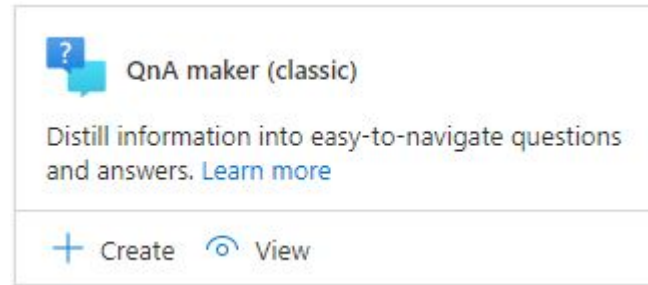
**O QnA Maker
fornece criação,
treinamento e
publicação, junto
com as permissões
de colaboração.**



**Disponível em
53 idiomas**

AZURE COGNITIVE SERVICES

O sistema escolhido para implementação de QnA apresentado foi o serviço Azure. Que por meio da assinatura de estudante, fornece a IA de forma gratuita e com certeza de funcionalidade porém online



OBRIGADO!

Alguma dúvida?