CSBH
CONEXÃO SOLIDÁRIA
BELO HORIZONTE
SPRINT 4

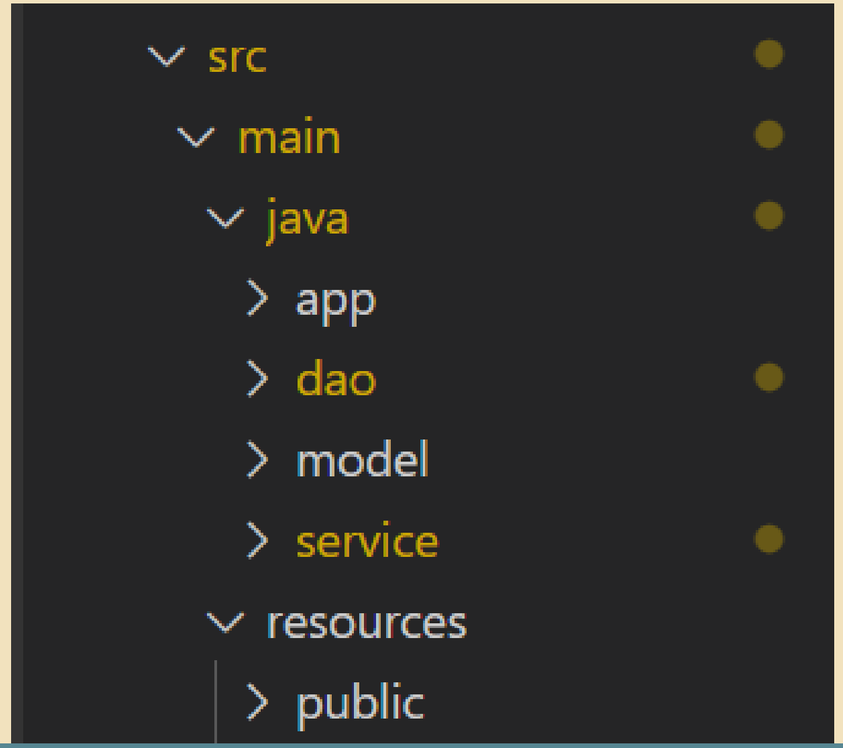
DAVI GODOI GRILO
FILIPI PEREIRA
JULIO CESAR THUROW
PAULA NOGUEIRA

REQUISITOS FUNCIONAIS

RF-001 O sistema deve permitir que o voluntário pesquise projetos por palavras-chaves.
RF-002 O sistema deve permitir que o voluntário filtre projetos, com filtros de diferentes temas.
RF-003 O sistema deve apresentar detalhes específicos de cada projeto, quando o usuário selecioná-lo.
RF-004 O voluntário pode favoritar projetos que o interessem.
RF-005 O sistema deve mostrar os projetos favoritados por um voluntário.
RF-006 Cada projeto de vagas de voluntariado ou de doação de material deve mostrar uma imagem, informações do anfitrião; uma descrição, a localização (mapa) e o horário do projeto.
RF-007 Cada projeto de doação de dinheiro deve mostrar uma imagem, informações do anfitrião, descrição do projeto e as informações bancárias para a doação.
RF-008 O sistema deve apresentar soluções para dúvidas frequentes.
RF-009 Um usuário deve poder realizar cadastro no site criando login nome de usuário e senha.
RF-010 Um usuário deve poder realizar login usando seu nome de usuário e senha cadastrados.
RF-011 Um usuário deve conseguir se tornar anfitrião (parceiro) do site.
RF-012 O sistema deve mostrar as informações cadastradas do usuário e permitir alterações.
RF-013 O sistema deve permitir que um anfitrião cadastre novos projetos.
RF-014 O sistema deve mostrar os projetos cadastrados por um anfitrião.

TODOS OS REQUISITOS FORAM CUMPRIDOS E ADICIONAMOS DOIS: O SISTEMA DEVE PERMITIR QUE ANFITRIÃO ATUALIZE PROJETOS E O SISTEMA DEVE PERMITIR QUE OS USUÁRIOS LOGUEM COM FACEID.

MELHORES PRÁTICAS



ORGANIZAÇÃO DOS DIRETÓRIOS

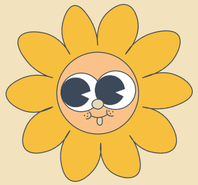
BANCO DE DADOS

```java
public boolean conectar() {
    String driverName = "org.postgresql.Driver"; // Nome do driver do PostgreSQL
    String serverName = "projetocsbh.postgres.database.azure.com";              // Endereço do servidor
    String mydatabase = "postgres";                     // Nome do banco de dados
    int porta = 5432;                                  // Porta do banco de dados
    String url = "jdbc:postgresql://" + serverName + ":" + porta +"/" + mydatabase; // URL de conexão
    String username = "adm";                    // Nome de usuário do banco de dados
    String password = "postgres";                       // Senha do banco de dados
    boolean status = false;                             // Status da conexão
```

BANCO DE DADOS HOSPEDADO NA MICROSOFT AZURE

SISTEMA INTELIGENTE

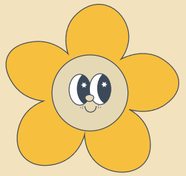
FACE ID



Face API JS

Link da API

API JS



Login com FACE ID através da câmera

OBS Caso não tenha câmera :
Login com FACE ID através de Upload de Foto

INTELLIGENT SYSTEM CANVAS

FERRAMENTA DE IA

Reconhecimento Facial: Algoritmos de aprendizado de máquina (ex: redes neurais convolucionais) para identificar e autenticar usuários com base em suas características faciais.
Processamento de Imagem: Algoritmos para capturar e processar a imagem do rosto do usuário em tempo real.

ENTRADAS

Imagens Faciais: Captura de imagens em tempo real a partir da câmera do dispositivo do usuário.
Dados do Usuário: Informações como nome, e-mail e dados biométricos (imagens faciais previamente registradas).

SAIDAS

Autenticação do Usuário: Resultado da validação da identidade do usuário (autenticado ou não).
Feedback ao Usuário: Mensagens de sucesso ou falha no login.

PROPOSIÇÃO DE VALOR

Facilidade de Uso: Login rápido e conveniente sem a necessidade de senhas.
Segurança Aprimorada: Redução de riscos de fraudes e acessos não autorizados.
Experiência do Usuário: Interface intuitiva e feedback em tempo real durante o processo de login.

EQUIPE

Todos os alunos do grupo participaram ativamente em todas as etapas do desenvolvimento do sistema de Face ID, incluindo:
Desenvolvimento Frontend e Backend.
Implementação de Algoritmos de IA.
Design da Interface do Usuário.
Gerenciamento do Projeto.
Testes e Validação.

STAKEHOLDERS CHAVES

Usuários Finais: Pessoas que utilizarão o sistema para login.
Parceiros de Tecnologia: Fornecedores de ferramentas e tecnologias necessárias para o desenvolvimento (FaceAPI.js).
Reguladores: Entidades que garantem conformidade com normas de privacidade e proteção de dados.

CLIENTES

Usuários do Sistema:
Voluntários e Anfitriões, que utilizarão do sistema de Face ID para gerenciar o acesso e garantir a segurança durante seu login.

Administradores:
Membros da equipe que gerenciarão o sistema e monitorarão as autenticações.

CUSTOS

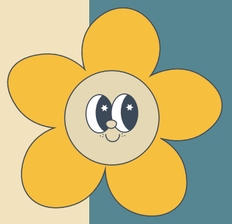
Desenvolvimento: Investimentos em tempo e recursos humanos para o desenvolvimento do sistema.
Infraestrutura: Custos com servidores, armazenamento de dados (banco de dados) e hospedagem.
Licenças de Software: Possíveis custos com bibliotecas de IA e serviços de nuvem.
Manutenção: Custos contínuos de manutenção e atualização do sistema.

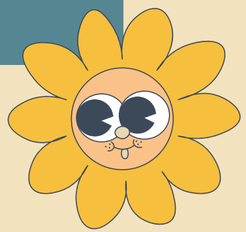
RECEITAS

Possíveis Patrocínios:
- Parcerias com empresas ou organizações que possam estar interessadas em apoiar o projeto em troca de visibilidade ou reconhecimento.

Possíveis Contribuições Voluntárias:
- Doações ou contribuições de membros da comunidade acadêmica e de profissionais que apoiam a iniciativa.

Possíveis Colaborações de Instituições:
- Parcerias com instituições que possam usar o sistema, garantindo suporte financeiro para a manutenção e desenvolvimento contínuo.





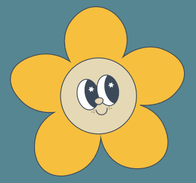
SENHA

FOTO RETIRADA DO BANCO DE DADOS:

email character varying (255)	senha character varying (255)
julia@gmail.com	$2a$10$hT.bQjE9WSg5tsoDKhZZWerM.lHM4x0VnOq0OZnd9Mp3Ytkh/...
fernando@gmail.com	$2a$10$5Kt3aXMxxxXnOX5xvpCTuOBBqT8G89jlQVCZMcCZKQ3laS3dx...
bruno@gmail.com	$2a$10$PU4H23j9ol29e64fKMQjfeGSmiGjYTXB4OJflju.t4D2AIOHBp1m...

CRIPTOGRAFADAS PELO BCRYPT

SQL INJECTION



```java
public boolean delete(int idVoluntario, int idProjeto) {
    boolean status = false;
    try {
        String sql = "DELETE FROM favorito WHERE idvoluntario = ? AND idprojeto = ?";
        PreparedStatement st = conexao.prepareStatement(sql);
        st.setInt(1, idVoluntario); // Define o id do voluntário
        st.setInt(2, idProjeto);    // Define o id do projeto
        st.executeUpdate();
        st.close();
        status = true;
    } catch (SQLException e) {
        System.err.println("Erro ao deletar anfitrião: " + e.getMessage());
    }
    return status;
}
```

Proteção contra ataques

OBRIGADO

Link GitHub