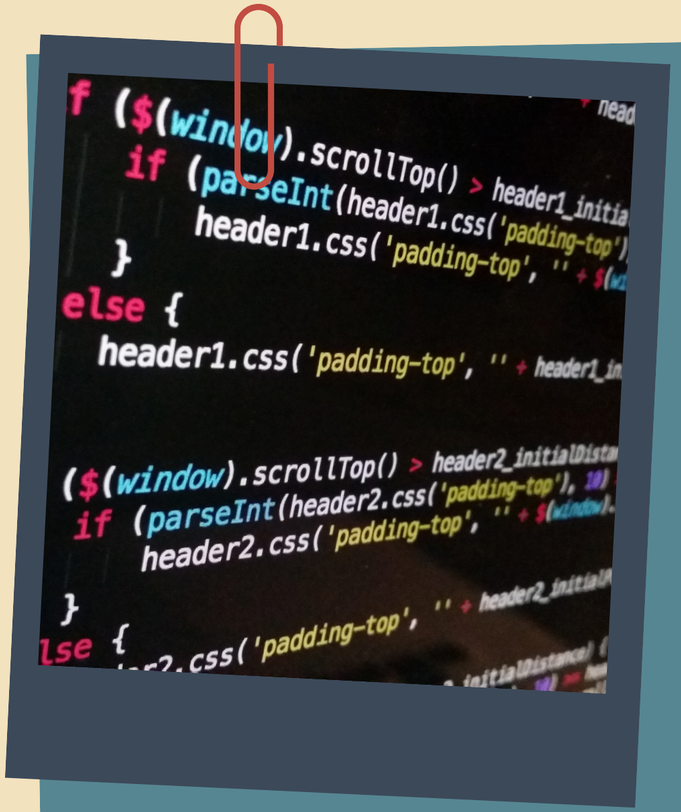
CSBH

CONEXÃO SOLIDÁRIA

BELO HORIZONTE

SPRINT 3

DAVI GODOI GRILO
FILIPI PEREIRA
JULIO CESAR THUROW
PAULA NOGUEIRA


if ($(window).scrollTop() > header1_initia
if (parseInt(header1.css('padding-top')
header1.css('padding-top', '' + $(wi
}
else {
header1.css('padding-top', '' + header1_in
($(window).scrollTop() > header2_initialDista
if (parseInt(header2.css('padding-top'), 10)
header2.css('padding-top', '' + $(window).
}
lse {

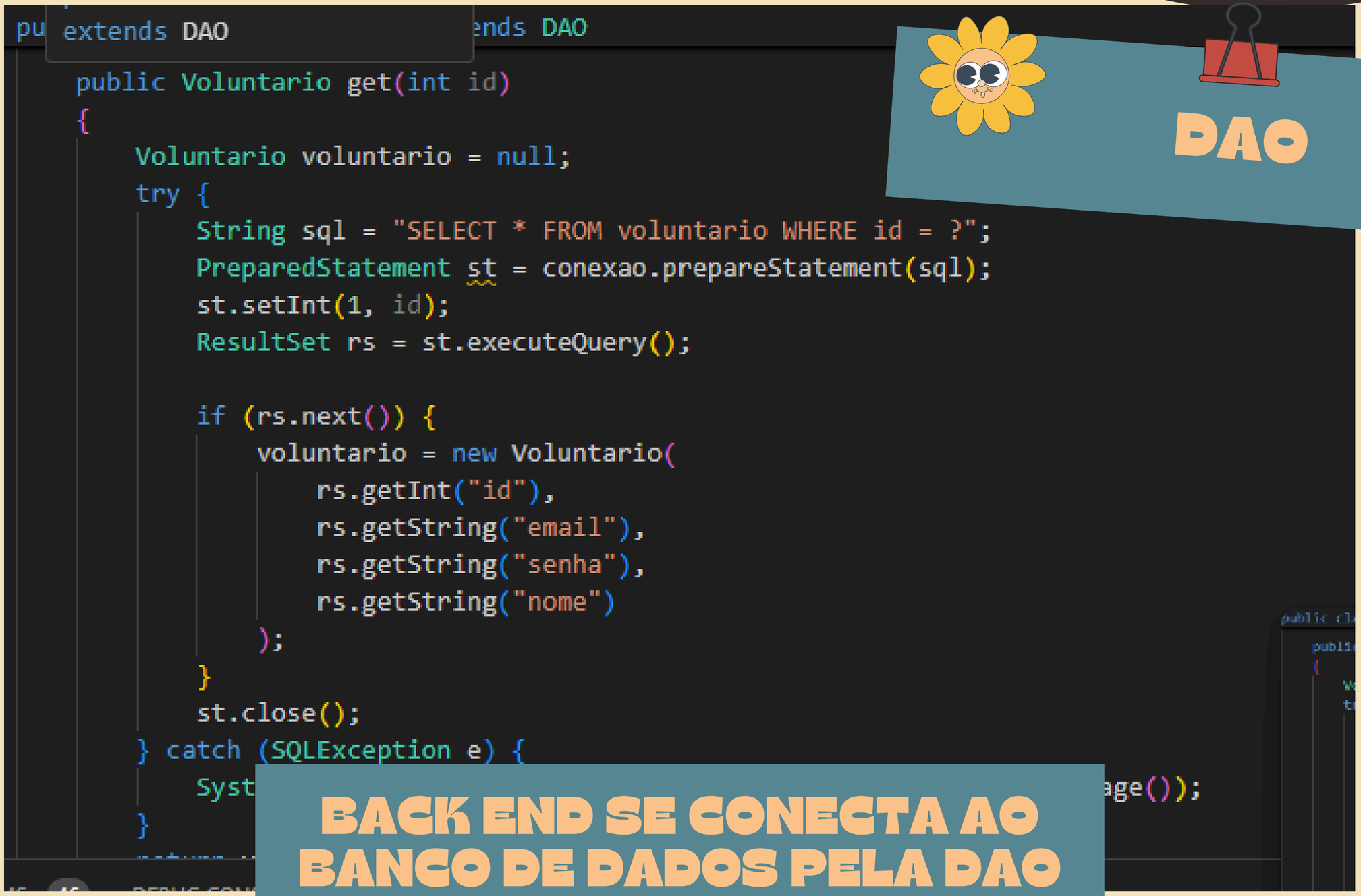
BACK END

ROTAS

```
public class Aplicacao {
    public static void main(String[] args) {

        get("/projeto/delete/:id", (request, response) -> projetoServi

        //anfitriao
        post("/anfitriao/insert", (request, response) -> anfitriaoService.insert(request, response));

        get("/anfitriao/list", (request, response) -> anfitriaoService.getAll(request, response));

        get("/anfitriao/:id", (request, response) -> anfitriaoService.get(request, response));

        post("/anfitriao/update/:id", (request, response) -> anfitriaoService.update(request, response));

        get("/anfitriao/delete/:id", (request, response) -> anfitriaoService.delete(request, response));

        //voluntario
        post("/voluntario/insert", (request, response) -> voluntarioService.insert(request, response));

        get("/voluntario/list", (request, response) -> voluntarioService.getAll(request, response));

        get("/voluntario/:id", (request, response) -> voluntarioService.get(request, response));
```

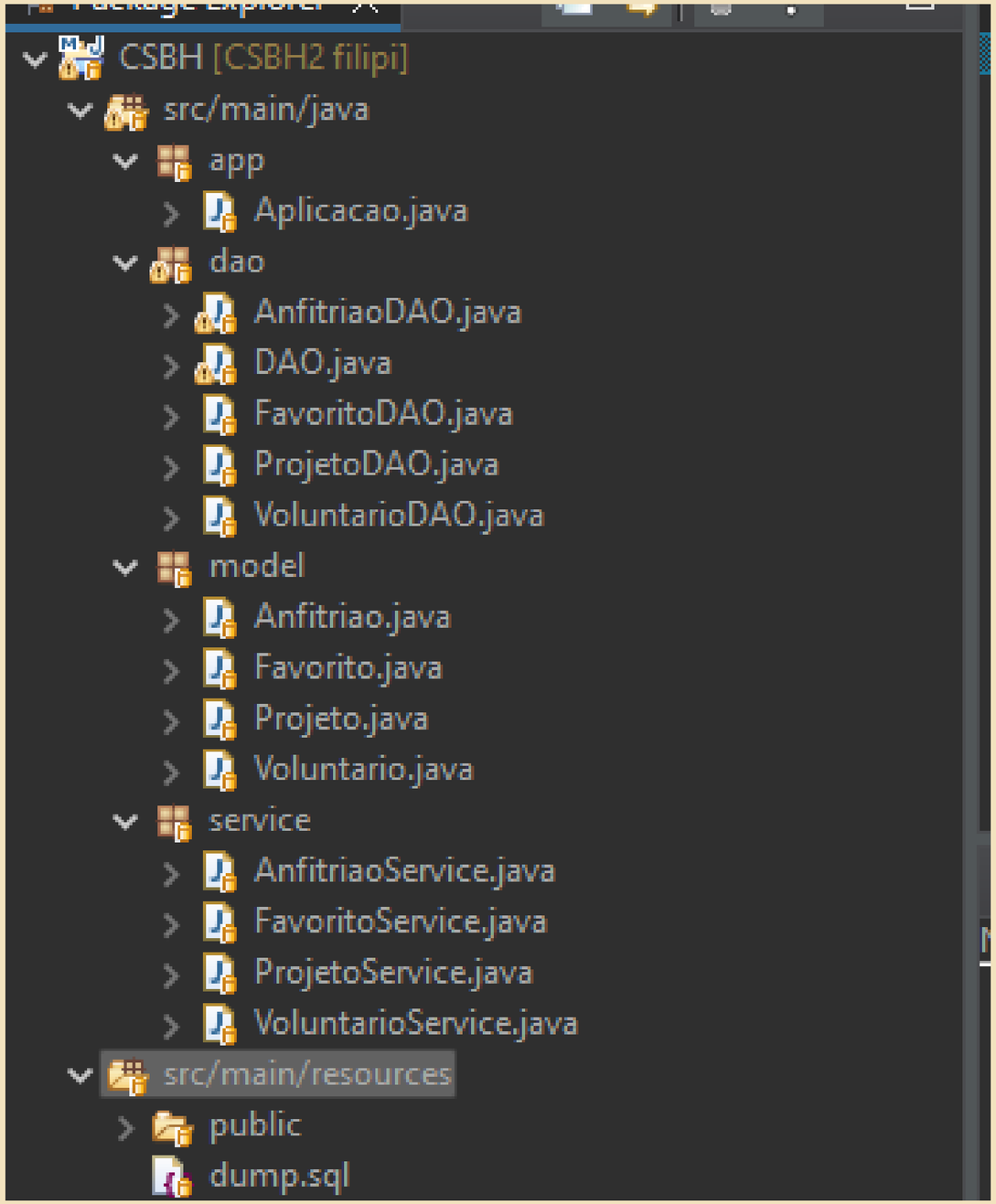
```
BLEMS  DEBUG CONSOLE  TERMINAL  PORTS 5  GITLENS  POSTMAN CONSOLE   bash - Aplicacao
ipi@LAPTOP-Q3CVH73A:/mnt
```

FRONT END SE CONECTA AO BACK END PELAS ROTAS



DAO

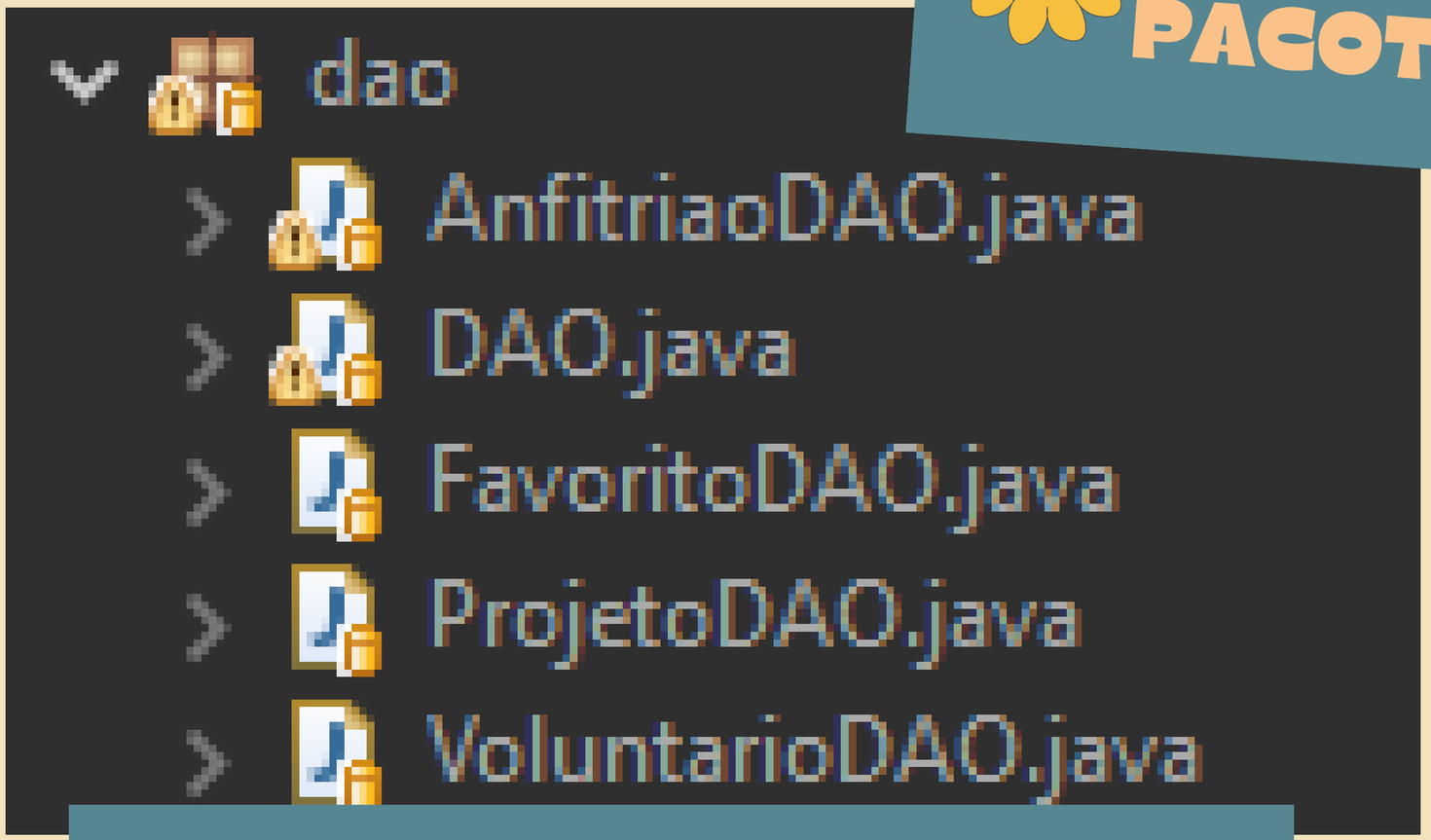
BACK END SE CONECTA AO BANCO DE DADOS PELA DAO



ORGANIZAÇÃO

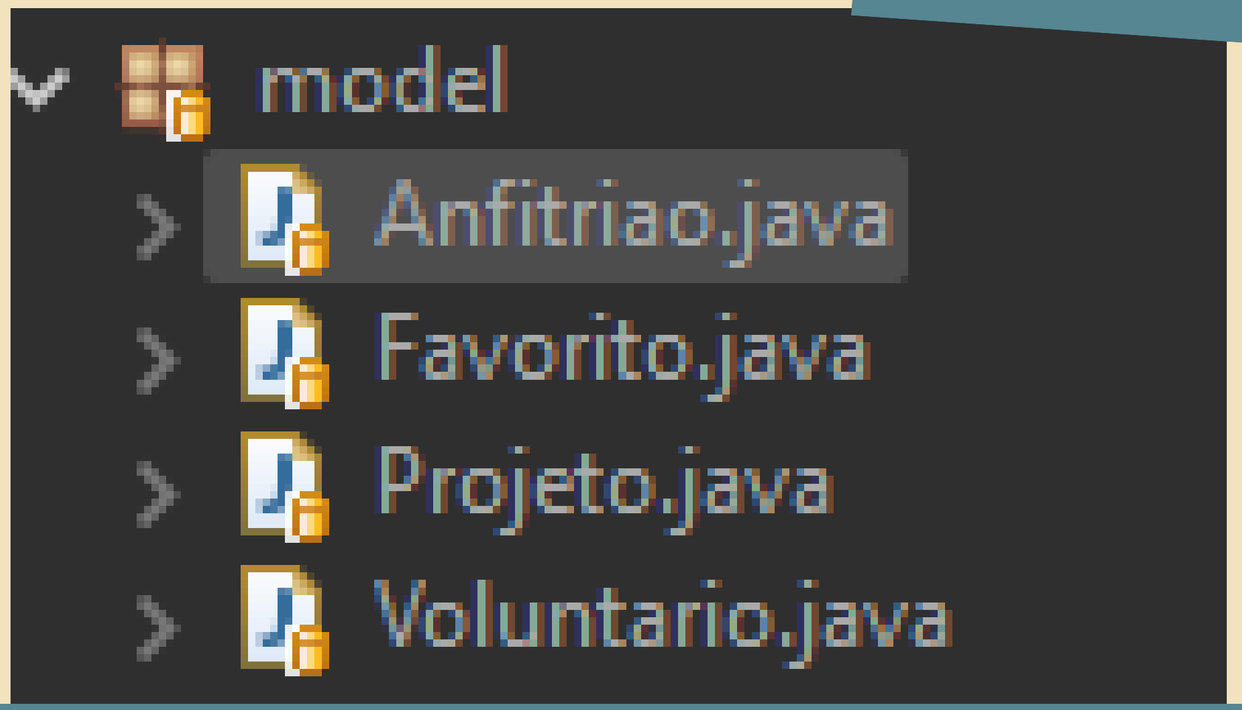
ESTRUTURA DE DIRETORIOS

PACOTE DAO



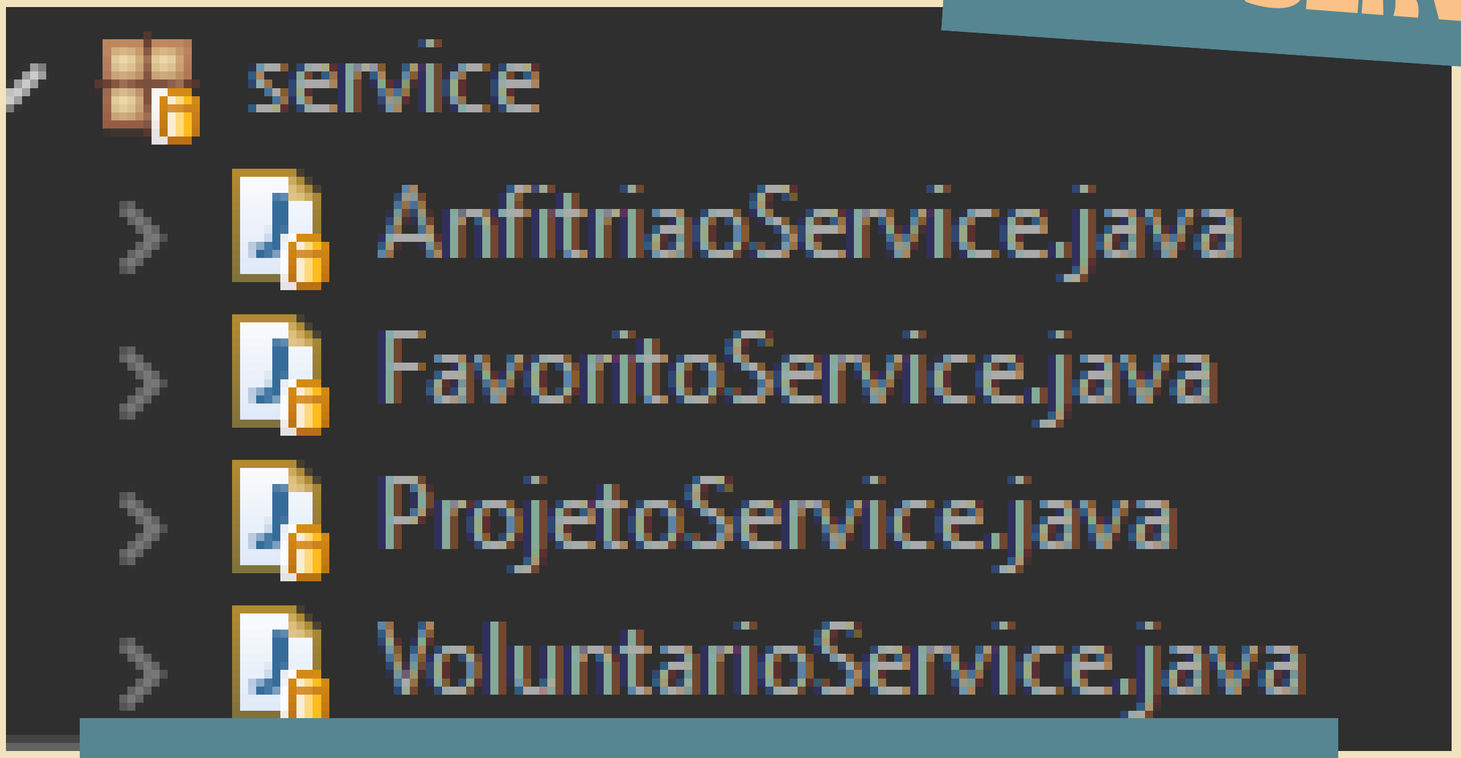
COMPLETAS, POSSUINDO TODAS AS FUNCOES DO CRUD

PACOTE MODEL

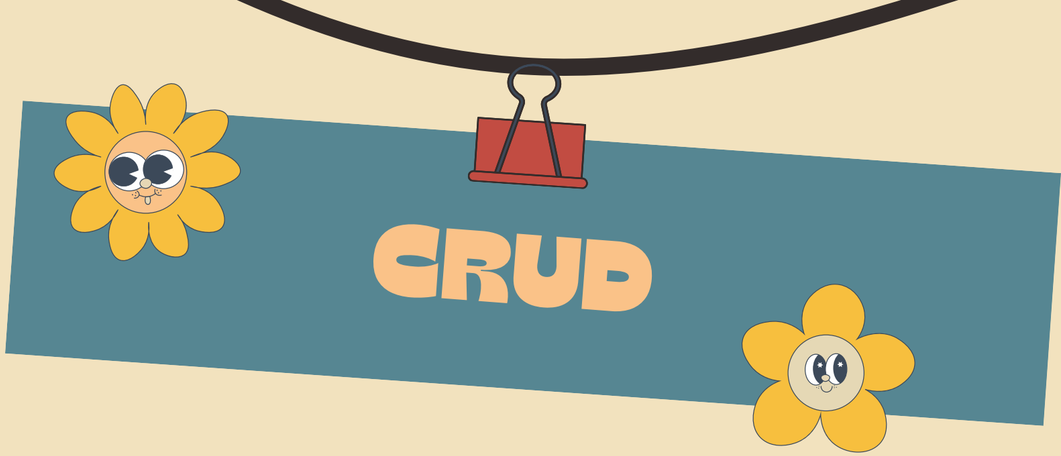


ATRIBUTOS CORRESPONDEM AO BANCO DE DADOS. METODOS GETTERS E SETTERS

PACOTE SERVICE

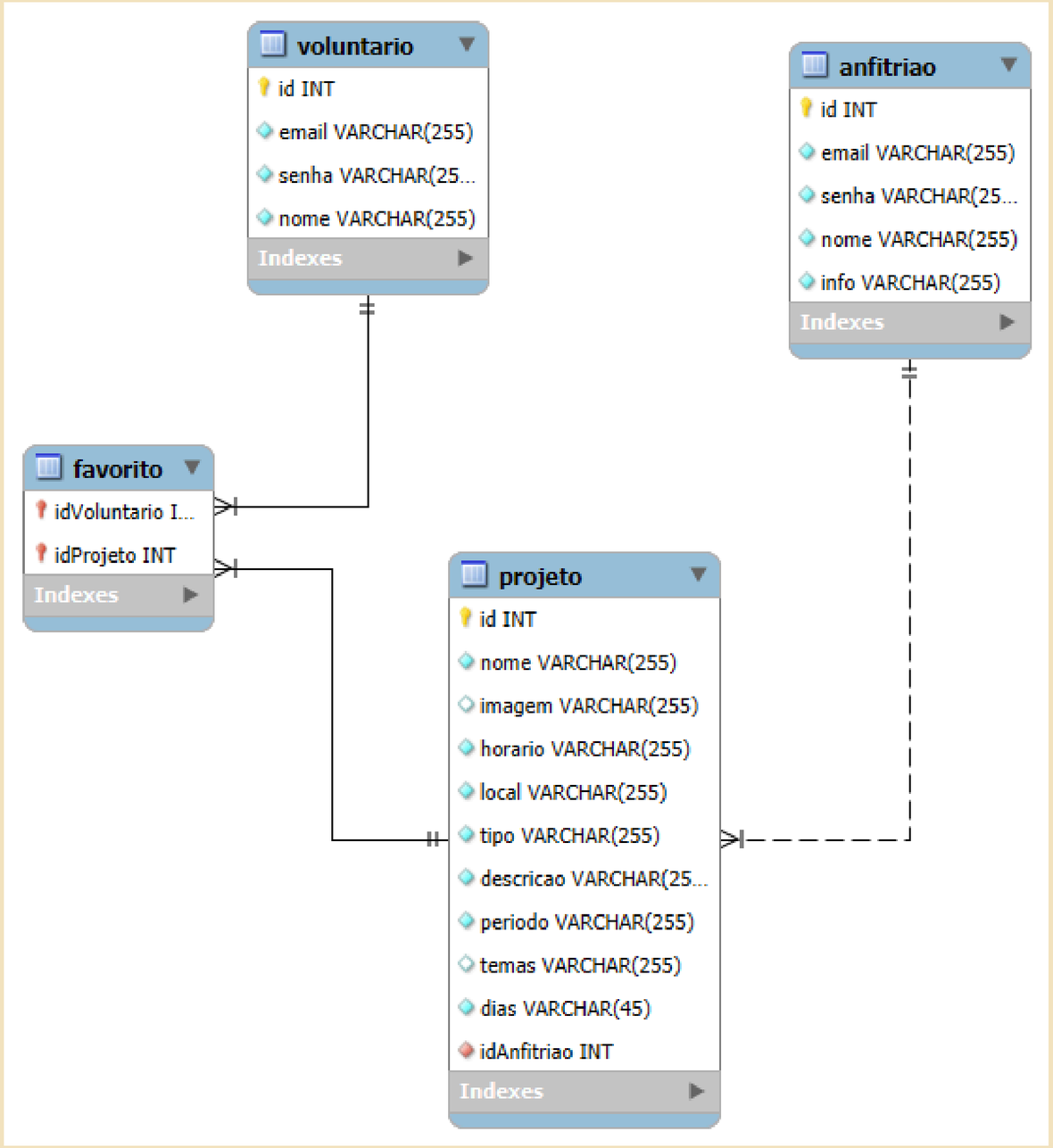


COMPLETAS, POSSUINDO TODAS AS FUNÇÕES DO CRUD




```
try {
    const response = await fetch('http://localhost:6789/voluntario/insert', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(novoUsuario)
    });
```

FRONT END INVOCA CADA UM DOS CRUDS



DIAGRAMA

CORREÇÕES:

Dúvidas se tornou estático.
Adicionamos a tabela favorito para modelar o relacionamento N:N entre voluntários e projetos.


```
-- ----------------------------------------------------
-- Tabela: anfitriao
-- ----------------------------------------------------
CREATE TABLE IF NOT EXISTS anfitriao (
  id SERIAL PRIMARY KEY,
  email VARCHAR(255) NOT NULL,
  senha VARCHAR(255) NOT NULL,
  nome VARCHAR(255) NOT NULL,
  info VARCHAR(255) NOT NULL,
  anfitriaocol VARCHAR(45)
);
-- ----------------------------------------------------
-- Tabela: voluntario
-- ----------------------------------------------------
CREATE TABLE IF NOT EXISTS voluntario (
  id SERIAL PRIMARY KEY,
  email VARCHAR(255) NOT NULL,
  senha VARCHAR(255) NOT NULL,
  nome VARCHAR(255) NOT NULL
);
```

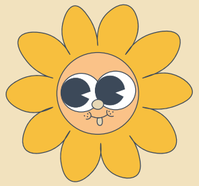
```
-- ----------------------------------------------------
-- Tabela: projeto
-- ----------------------------------------------------
CREATE TABLE IF NOT EXISTS projeto (
  id SERIAL PRIMARY KEY,
  nome VARCHAR(255) NOT NULL,
  imagem VARCHAR(255),
  horario VARCHAR(255) NOT NULL,
  local VARCHAR(255) NOT NULL,
  tipo VARCHAR(255) NOT NULL,
  descricao VARCHAR(255) NOT NULL,
  periodo VARCHAR(255) NOT NULL,
  temas VARCHAR(255),
  dias VARCHAR(45) NOT NULL,
  idAnfitriao INT NOT NULL,
  CONSTRAINT fk_projeto_anfitriao FOREIGN KEY (idAnfitriao)
    REFERENCES anfitriao (id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
-- Índice para a chave estrangeira idAnfitriao
CREATE INDEX IF NOT EXISTS fk_projeto_anfitriao_idx
  ON projeto (idAnfitriao);
```

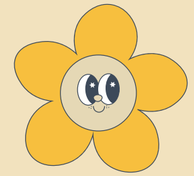
```
-- ----------------------------------------------------
-- Tabela: favorito
-- ----------------------------------------------------
CREATE TABLE IF NOT EXISTS favorito (
  idVoluntario INT NOT NULL,
  idProjeto INT NOT NULL,
  PRIMARY KEY (idVoluntario, idProjeto),
  CONSTRAINT fk_favorito_voluntario FOREIGN KEY (idVoluntario)
    REFERENCES voluntario (id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT fk_favorito_projeto FOREIGN KEY (idProjeto)
    REFERENCES projeto (id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
-- Índices para as chaves estrangeiras em favorito
CREATE INDEX IF NOT EXISTS fk_voluntario_idx
  ON favorito (idVoluntario);
CREATE INDEX IF NOT EXISTS fk_projeto_idx
  ON favorito (idProjeto);
```

SISTEMA INTELIGENTE

FACE ID

- Usuário pode realizar login com o FaceID
- Reconhece padrões faciais do usuário para autenticação
- Autenticação rápida e menos esforço para o usuário
- Aprendizado offline e online

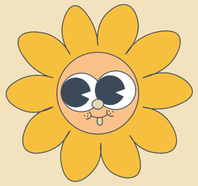


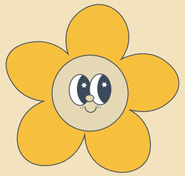


FACE ID

Entradas do modelo

-Imagens/vídeos da face
-Dados biométricos pré-cadastrados





Saídas do modelo:

-Conjunto de dados representando as características únicas do rosto
-Pontuação de similaridade
-Decisão binária

Microsoft Azure

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.
Registro de Acesso: Logs de tentativas de login, incluindo data, hora e resultados.

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.
Acessibilidade: Funcionalidade em múltiplos dispositivos, incluindo smartphones e desktops.

EQUIPE

Todos os alunos do grupo participarão 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 (Microsoft Azure).
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

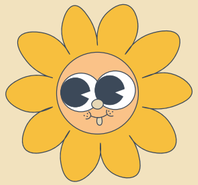
Patrocínios:
- Parcerias com empresas ou organizações que possam estar interessadas em apoiar o projeto em troca de visibilidade ou reconhecimento.

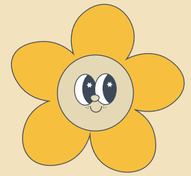
Contribuições Voluntárias:
- Doações ou contribuições de membros da comunidade acadêmica e de profissionais que apoiam a iniciativa.

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

WORKFLOW

TO DO	DOING	DONE
Apresentação Sprint 4	Inteligência Artificial (FACE ID)	Escolha do tema
	Apresentação Sprint 3	Objetivo do projeto
		Fluxo do Usuário (Userflow)
		Determinando as entidades, atributos, relacionamentos
		Front-End
		Definir Sistemas Inteligentes
		Scripts para criação do Banco de Dados
		Desenvolvimento dos Diagramas
		Front-End integrando com Back-End
		Back-End integrando com Banco de Dados





OBRIGADO