



DOORS

Grupo: Diogo Casal, Gabriel Alves,
Luana Takeishi e Matheus Aguilar

PROJETO DOORS

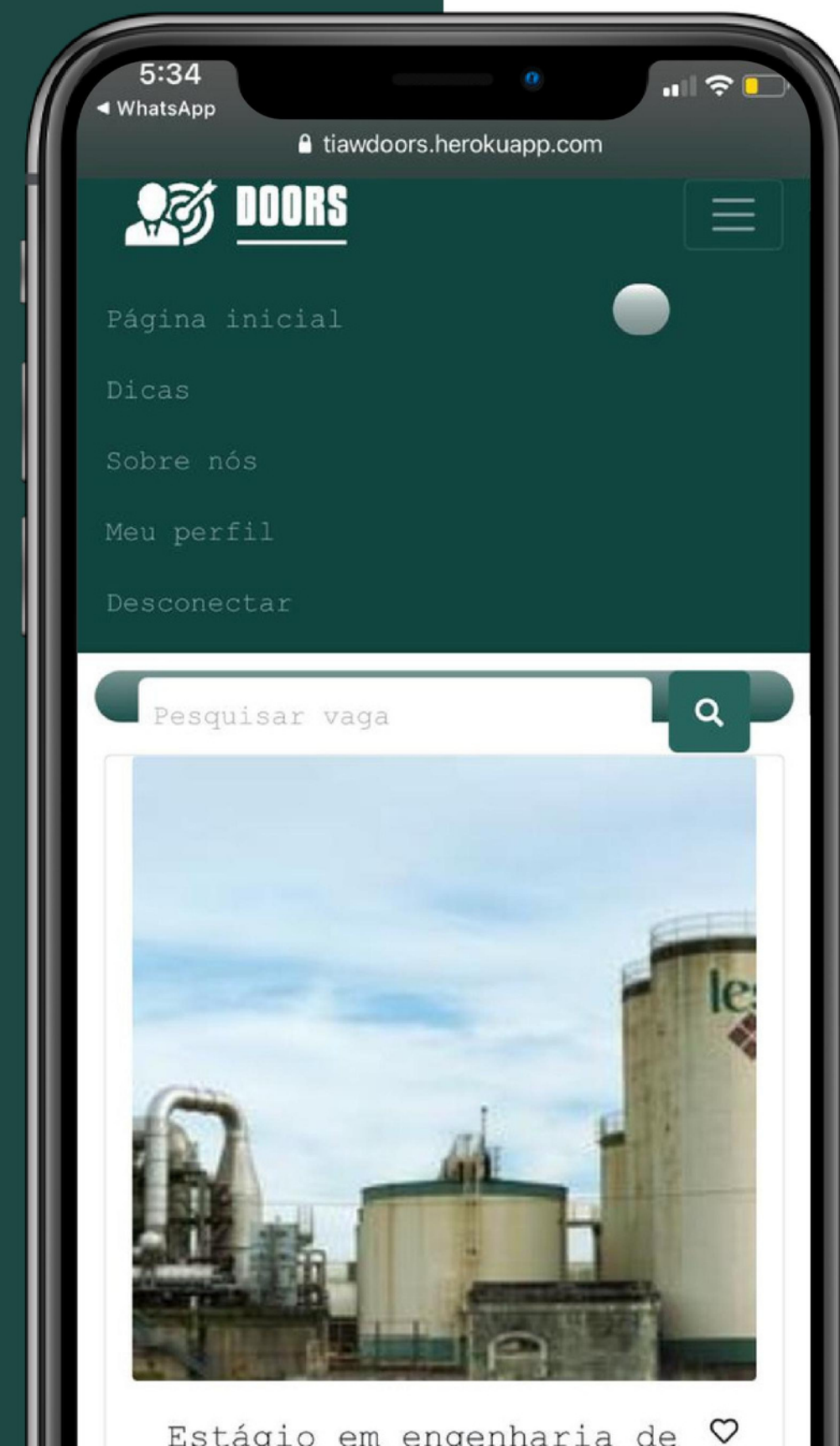
Plataforma para conectar estudantes universitários e empresas

Moldar as ofertas de estágio disponíveis para o perfil do usuário

Facilita a busca e o acesso às oportunidades de emprego

OBJETIVOS

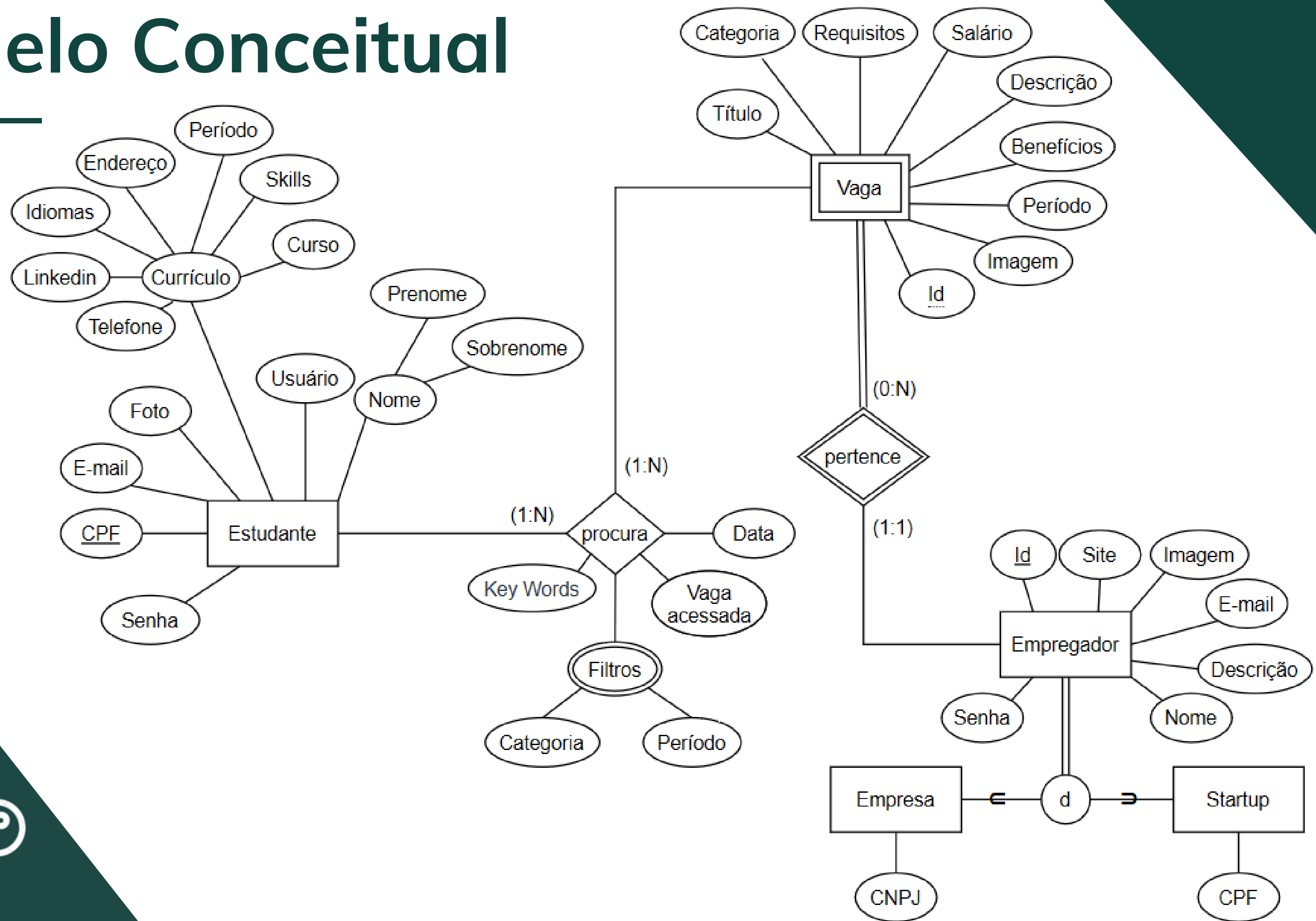
Facilitar a comunicação entre estudante-empresa, auxiliando os estudantes na sua inserção no meio emoesarial.



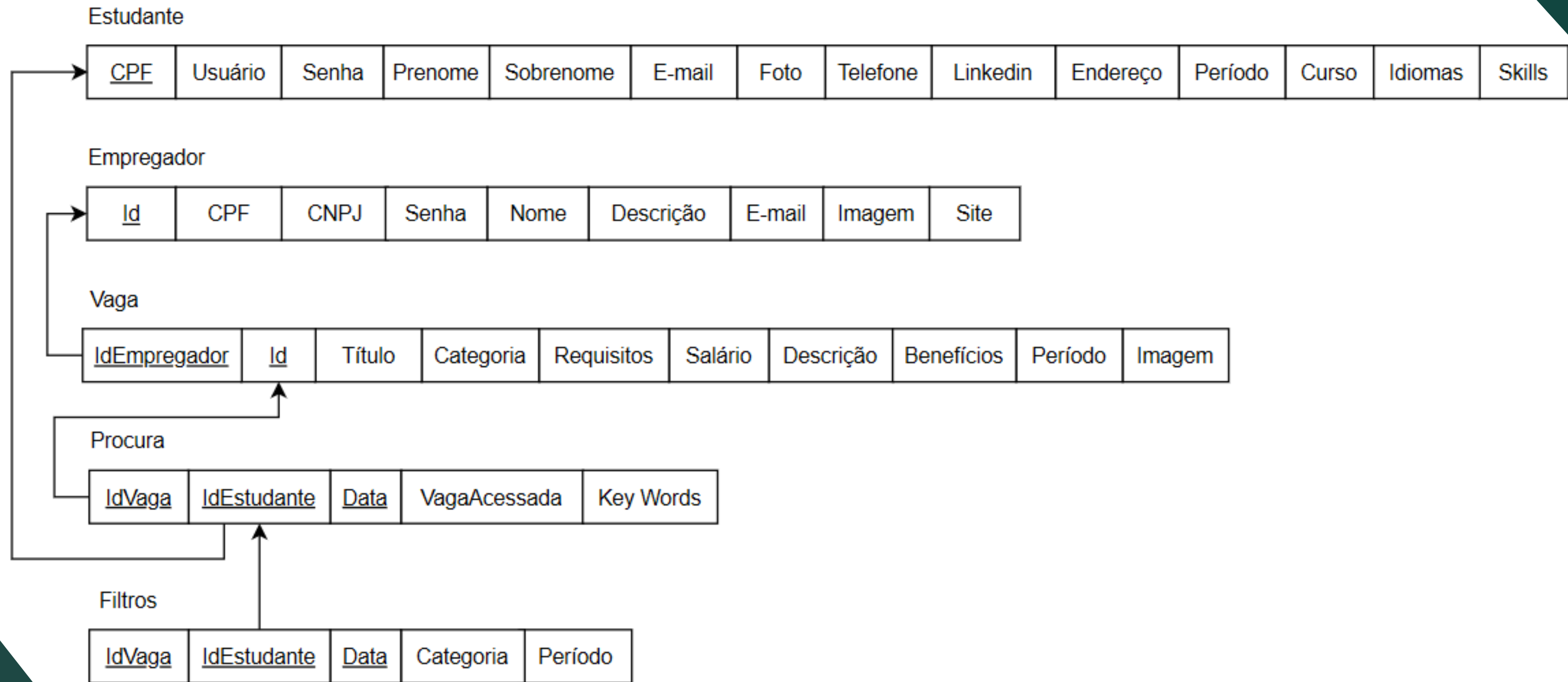
Banco de Dados



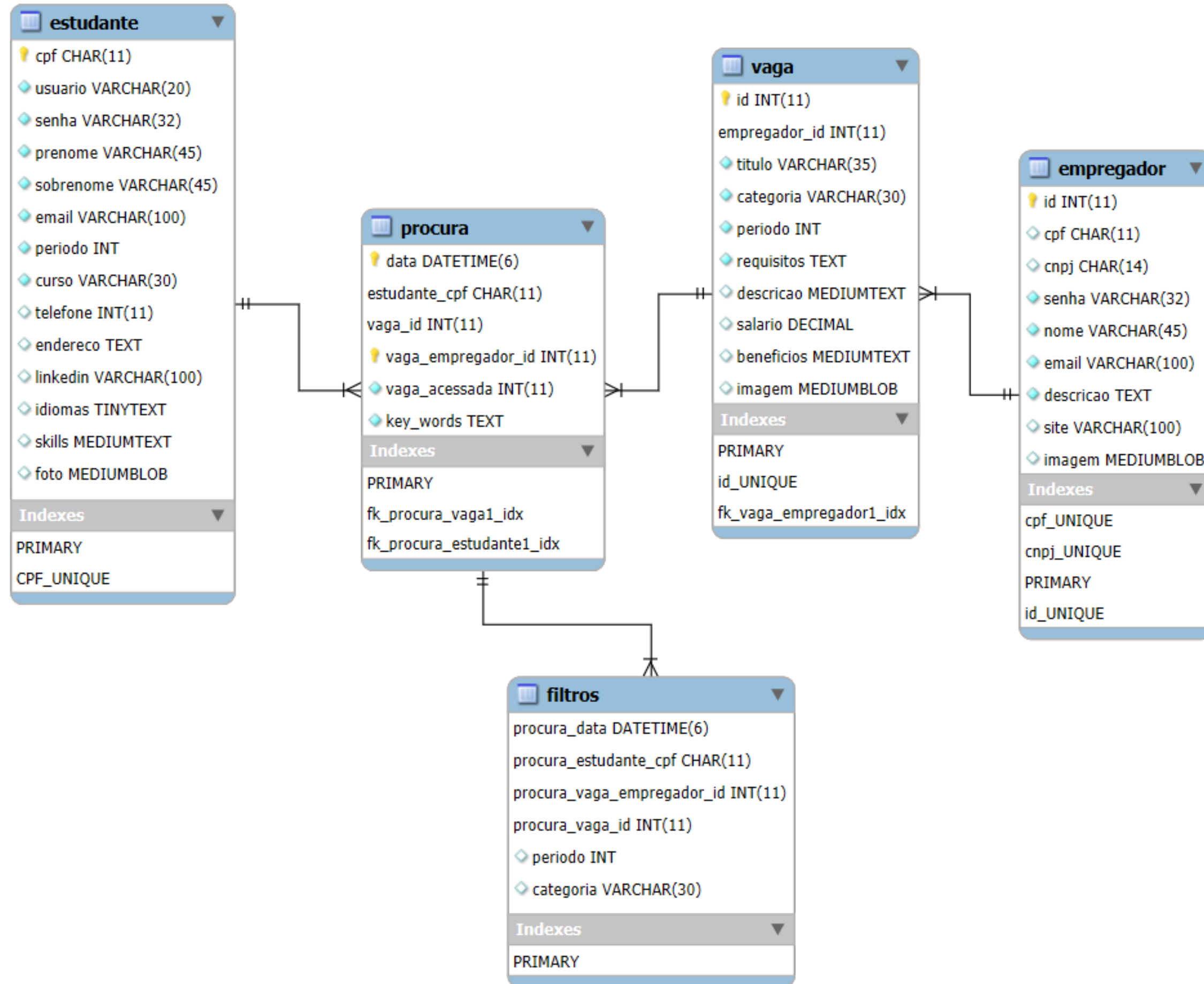
Modelo Conceitual



Modelo Lógico



Modelo de Implementação



Script de Criação

```
--
-- Name: empregador; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.empregador (
    id numeric(11,0) NOT NULL,
    cpf character(11),
    cnpj character(14),
    senha character varying(32) NOT NULL,
    nome character varying(45) NOT NULL,
    email character varying(100) NOT NULL,
    descricao text NOT NULL,
    site character varying(100) NOT NULL,
    imagem bit varying,
    CONSTRAINT "tipoEmpregador" CHECK (((cpf <> NULL::bpchar) OR (cnpj
<> NULL::bpchar)))
);

--
-- Name: estudante; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.estudante (
    cpf character(11) NOT NULL,
    usuario character varying(20) NOT NULL,
    senha character varying(32) NOT NULL,
    prenome character varying(45) NOT NULL,
    sobrenome character varying(45) NOT NULL,
    email character varying(100) NOT NULL,
    periodo integer NOT NULL,
    curso character varying(30) NOT NULL,
    telefone numeric(11,0),
    endereco text,
    linkedin character varying(100),
    idiomas text,
    skills text,
    foto bit varying
);
```


Script de Criação

```
--  
-- Name: filtros; Type: TABLE; Schema: public; Owner: postgres  
--  
  
CREATE TABLE public.filtros (  
    procura_data date NOT NULL,  
    procura_estudante_cpf character(11) NOT NULL,  
    procura_vaga_empregador_id numeric(11,0) NOT NULL,  
    procura_vaga_id numeric(11,0) NOT NULL,  
    periodo integer,  
    categoria character varying(30)  
);  
  
  
--  
-- Name: procura; Type: TABLE; Schema: public; Owner: postgres  
--  
  
CREATE TABLE public.procura (  
    data date NOT NULL,  
    vaga_acessada numeric(11,0) NOT NULL,  
    key_words text NOT NULL,  
    estudante_cpf character(11) NOT NULL,  
    vaga_id numeric(11,0) NOT NULL,  
    vaga_empregador_id numeric(11,0) NOT NULL  
);
```

Banco de Dados

- ▼ public
 - ▼ Tables
 - > empregador
 - > estudante
 - > filtros
 - > procura
 - > vaga
 - > Views
 - > Materialized Views
 - > Indexes
 - > Functions
 - > Sequences
 - > Data types
 - > Aggregate functions

40K

24K

8K

16K

24K

▼ Columns

123 id (numeric(11))

ABC cpf (bpchar(11))

ABC cnpj (bpchar(14))

ABC senha (varchar(32))

ABC nome (varchar(45))

ABC email (varchar(100))

ABC descricao (text)

ABC site (varchar(100))

123 imagem (varbit)

FRONT-END



FRONT-END

▼ resources

> assets

> imagens

▼ scripts

JS app.js

{} composer.json

{} composer.lock

JS esqueleto.js

JS estudante.js

JS feed.js

JS index.js

JS login.js

{} package-lock.json

JS perfil.js

▼ styles

about.css

esqueleto.css

estilos.css

feed.css

login.css

perfil.css

style.css

tips.css

> vendor

🔗 about.html

🔗 esqueleto.html

🔗 feed.html

🏠 home.php

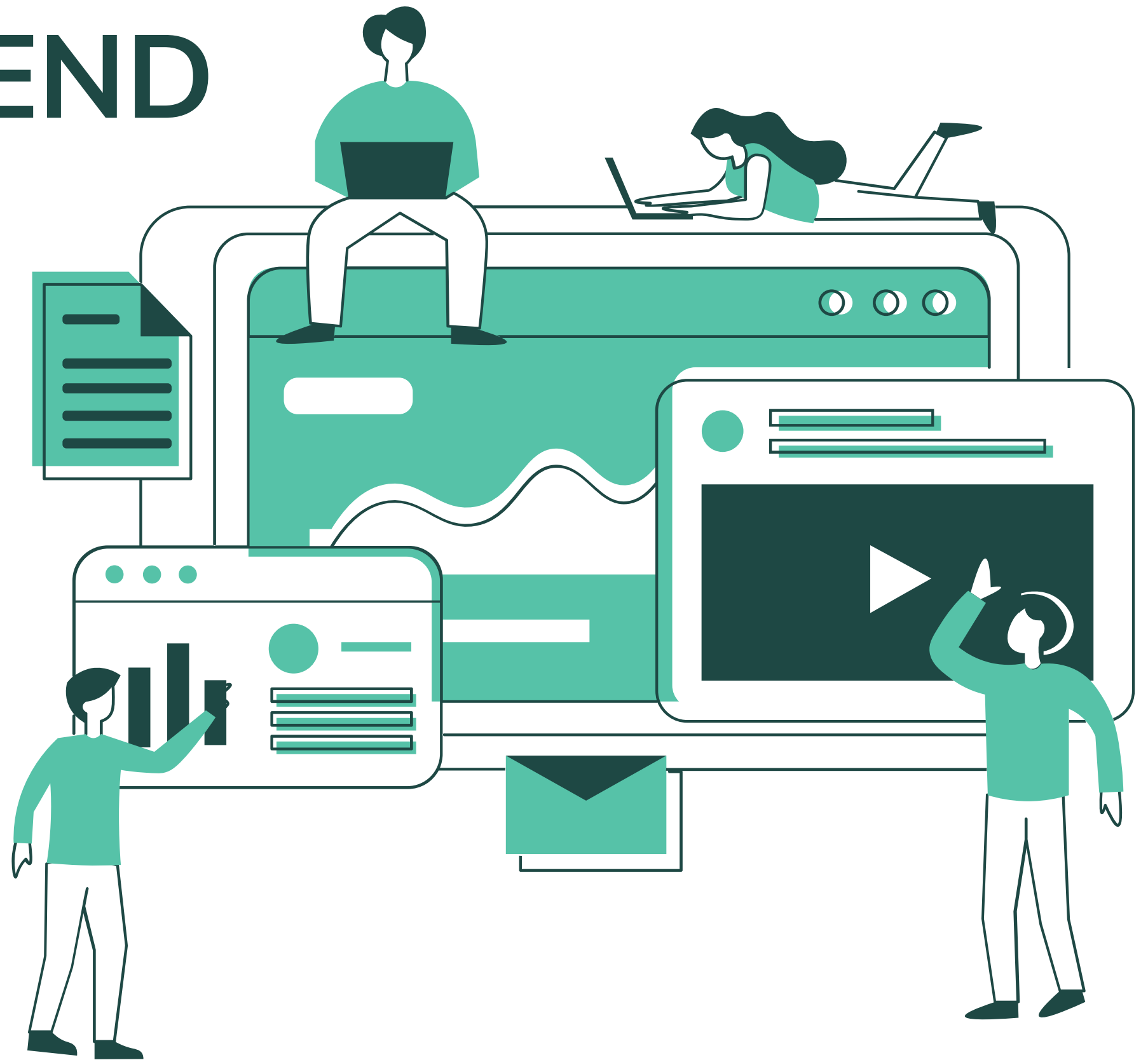
🔗 index.html

🔗 login.html

🔗 perfil.html

🔗 tips.html

BACK-END

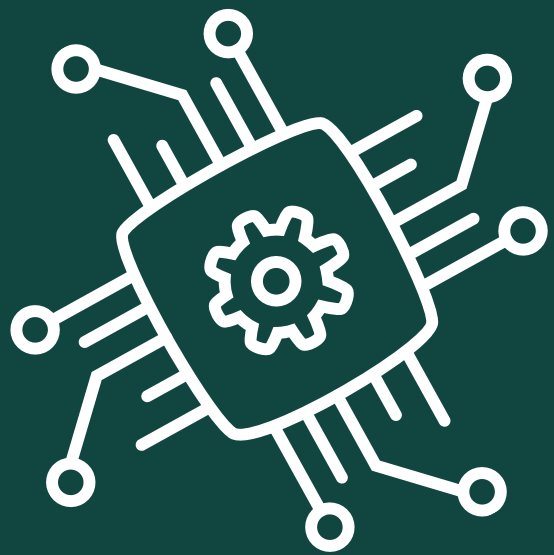


BACK-END

- ▼ java
 - ▼ app
 - 📄 Main.java
 - ▼ dao
 - 📄 CredenciaisDB.java
 - 📄 DAOEmpregador.java
 - 📄 DAOEstudante.java
 - 📄 DAOFiltros.java
 - 📄 DAOProcura.java
 - 📄 DAOVagas.java
 - ▼ model
 - 📄 Empregador.java
 - 📄 Estudante.java
 - 📄 Filtros.java
 - 📄 JsonFormatter.java
 - 📄 Procura.java
 - 📄 Vagas.java
 - ▼ service
 - 📄 EmpregadorService.java
 - 📄 EstudanteService.java
 - 📄 FiltrosService.java
 - 📄 ProcuraService.java
 - 📄 VagasService.java



Intelligent System Canvas



Ferramenta de IA

- A técnica para a extração e subsequente importação dos dados do currículo em PDF será baseada em OCR (reconhecimento óptico de caracteres)
- O software para OCR será uma API REST de leitura da Pesquisa Visual Computacional desenvolvida pela Azure
- Ferramenta de pesquisa inteligente por meio de recomendações baseadas em pesquisas de usuário (machine learning) com o recomendador SVD no designer de Azure Machine Learning

Entradas

- Leitura de PDF (baseada em uma API) a partir de palavras chaves presentes em um banco de dados
- Filtragem de pesquisa a partir do aprendizado da máquina pelo histórico de busca e palavras chaves

Pesquisa
inteligente

PDF

Filtros

Saídas

- Preenchimento automático do currículo no perfil do usuário após a leitura do PDF
- Tela com informações filtradas e importantes para o usuário em si

Proposição de valor

- Os principais valores que iremos levar aos clientes é confiança no armazenamento e privacidade de dados, garantia de facilidade e eficiência no preenchimento de informações para que a única preocupação do usuário seja se será contatado ou não

Papéis da Equipe

- Garantir que a integração entre a API de leitura de PDF e o site seja eficiente e funcional
- Garantir os registros no Banco de dados e utilizar do SGBD para fazer pesquisas eficientes

Stakeholders Chaves

Os tomadores de decisão do projeto serão os integrantes da equipe de desenvolvimento do DOORS se baseando no feedback dos usuários (fazer uma área de feedback?)

Clientes

- CLIENTES INTERNOS: Equipe de desenvolvimento do site DOORS
- CLIENTES EXTERNOS: Estudantes (com enfoque em universitários em busca de estágio) e Empresas (que buscam estagiários)

Custo

- Custos de divulgação da plataforma por anúncios em redes sociais ou em sites
- Custo de manutenção antes do fechamento de parcerias.
- Possível custo da API caso haja necessidade
- Host de servidor

Receitas

- O modo para monetização e eventual captação de receita será por meio de anúncios do Google AdSense, parcerias com empresas para que possam anunciar suas vagas no site DOORS e doações contribuintes.

Ferramenta de IA

- A técnica para a extração e subsequente importação dos dados do currículo em PDF será baseada em OCR (reconhecimento óptico de caracteres)
- O software para OCR será uma API REST de leitura da Pesquisa Visual Computacional desenvolvida pela Azure
- Ferramenta de pesquisa inteligente por meio de recomendações baseadas em pesquisas de usuário (machine learning) com o recomendador SVD no designer de Azure Machine Learning

Entradas

- Leitura de PDF (baseada em uma API) a partir de palavras chaves presentes em um banco de dados
- Filtragem de pesquisa a partir do aprendizado da máquina pelo histórico de busca e palavras chaves

Pesquisa
inteligente

PDF

Filtros

Saídas

- Preenchimento automático do currículo no perfil do usuário após a leitura do PDF
- Tela com informações filtradas e importantes para o usuário em si

Proposição de valor

- Os principais valores que iremos levar aos clientes é confiança no armazenamento e privacidade de dados, garantia de facilidade e eficiência no preenchimento de informações para que a única preocupação do usuário seja se será contatado ou não

Papéis da Equipe

- Garantir que a integração entre a API de leitura de PDF e o site seja eficiente e funcional
- Garantir os registros no Banco de dados e utilizar do SGBD para fazer pesquisas eficientes

Stakeholders Chaves

Os tomadores de decisão do projeto serão os integrantes da equipe de desenvolvimento do DOORS se baseando no feedback dos usuários (fazer uma área de feedback?)

Clientes

- CLIENTES INTERNOS:
Equipe de desenvolvimento do site DOORS
- CLIENTES EXTERNOS:
Estudantes (com enfoque em universitários em busca de estágio) e Empresas (que buscam estagiários)

Custo

- Custos de divulgação da plataforma por anúncios em redes sociais ou em sites
- Custo de manutenção antes do fechamento de parcerias.
- Possível custo da API caso haja necessidade
- Host de servidor

Receitas

- O modo para monetização e eventual captação de receita será por meio de anúncios do Google AdSense, parcerias com empresas para que possam anunciar suas vagas no site DOORS e doações contribuintes.



OBRIGADO(A)!

—