



DOORS

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

ESPECIFICAÇÃO FUNCIONAL

Especificações funcionais: aspectos
que serão apresentados ao usuário;

Especificações não funcionais:
aspectos internos do sistema.



Identificação	Descrição	Prioridade
RF-01	O sistema deve prover informações sobre empresas	Alta
RF-02	O sistema deve prover acesso prático às informações de oportunidades disponíveis e suas qualificações mínima	Alta
RF-03	O sistema deve possuir um mecanismo de busca simples	Média
RF-04	O sistema deve prover filtros para maior precisão de vaga desejada	Alta
RF-05	O sistema deve possuir API que possibilite a atualização dos dados, assim como requisição dos dados	Alta
RF-06	O sistema deve possibilitar contato direto com contratante	Média
RF-07	O sistema deve possibilitar o usuário "favoritar" oportunidades	Média
RF-08	O sistema deve prover mentorias no processo de elaboração do primeiro currículo	Baixa
RF-09	O sistema deve conter um formulário de registro e de login	Alta
RF-10	O sistema deve agrupar vagas abertas e facilitar o acesso a elas	Alta
RF-11	Página de recomendação de vagas (recommendation system)	Alta
RF-12	Implementação de currículo a partir de importação de PDF (information extraction)	Alta

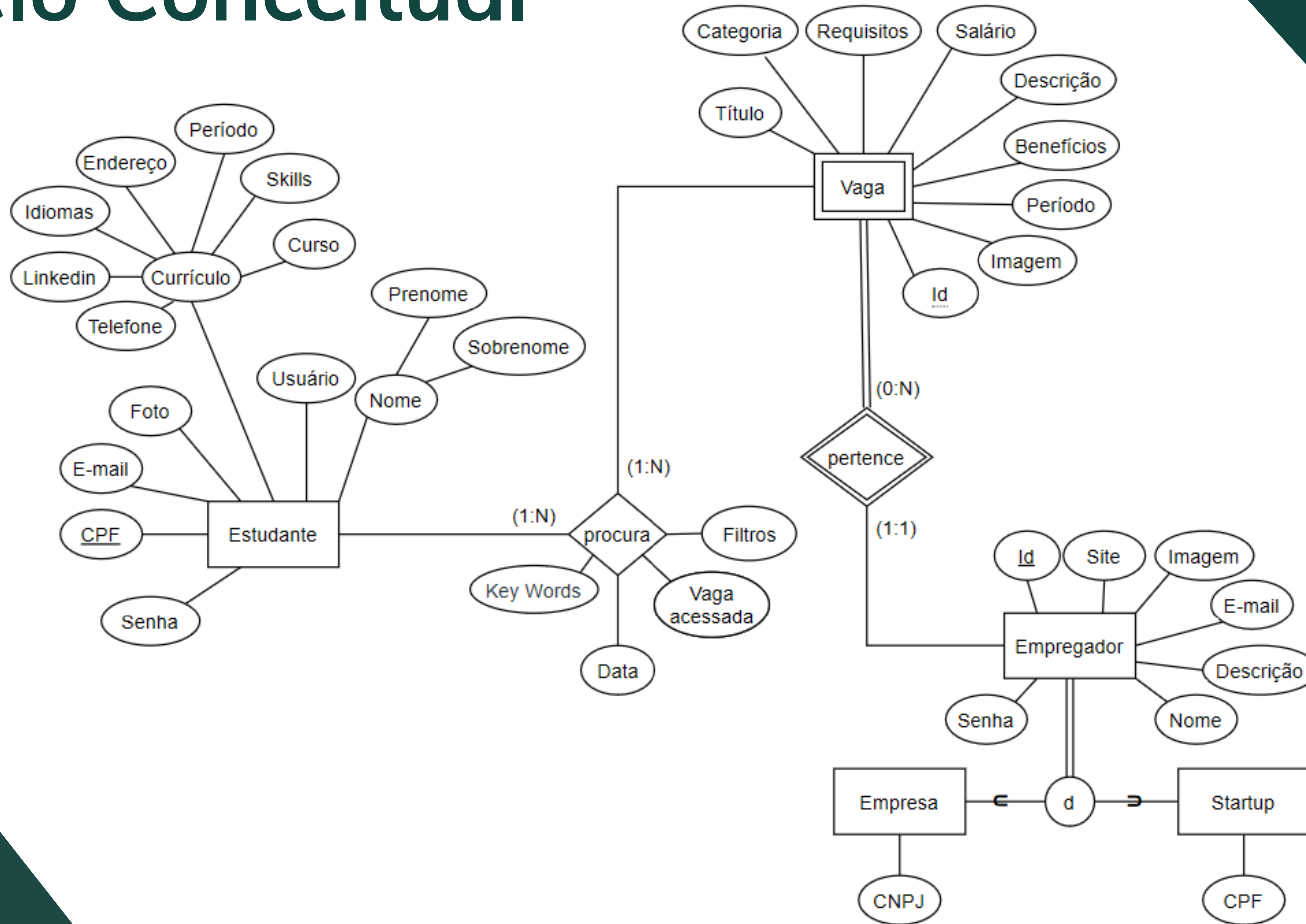
Identificação	Descrição	Prioridade
RNF-01	Desenvolvimento de plataforma que seja responsiva para diferentes tamanhos de layout (Desktop, Tablet e Celular)	Alta
RNF-02	Plataforma compatível com diferentes <i>Browsers</i>	Alta
RNF-03	Design apelativo para o cliente, que facilite compreensão de dados e utilização da plataforma	Média
RNF-04	Priorizar vagas mais adequadas ao perfil do usuário durante as pesquisas	Alta
RNF-05	Facilitar a busca e o acesso às vagas de emprego	Alta
RNF-06	Possuir pacotes de assinatura para as empresas	Média
RNF-07	Operar em conjunto com um banco de dados	Alta

Identificação	Descrição
RE-01	O projeto deverá estar totalmente finalizado antes do fim do semestre letivo.
RE-02	Toda documentação produzida pela equipe deverá ser autoral, podendo apenas usar outras documentações como referência, e quando utilizadas essas devem ser propriamente identificadas.

Banco de Dados



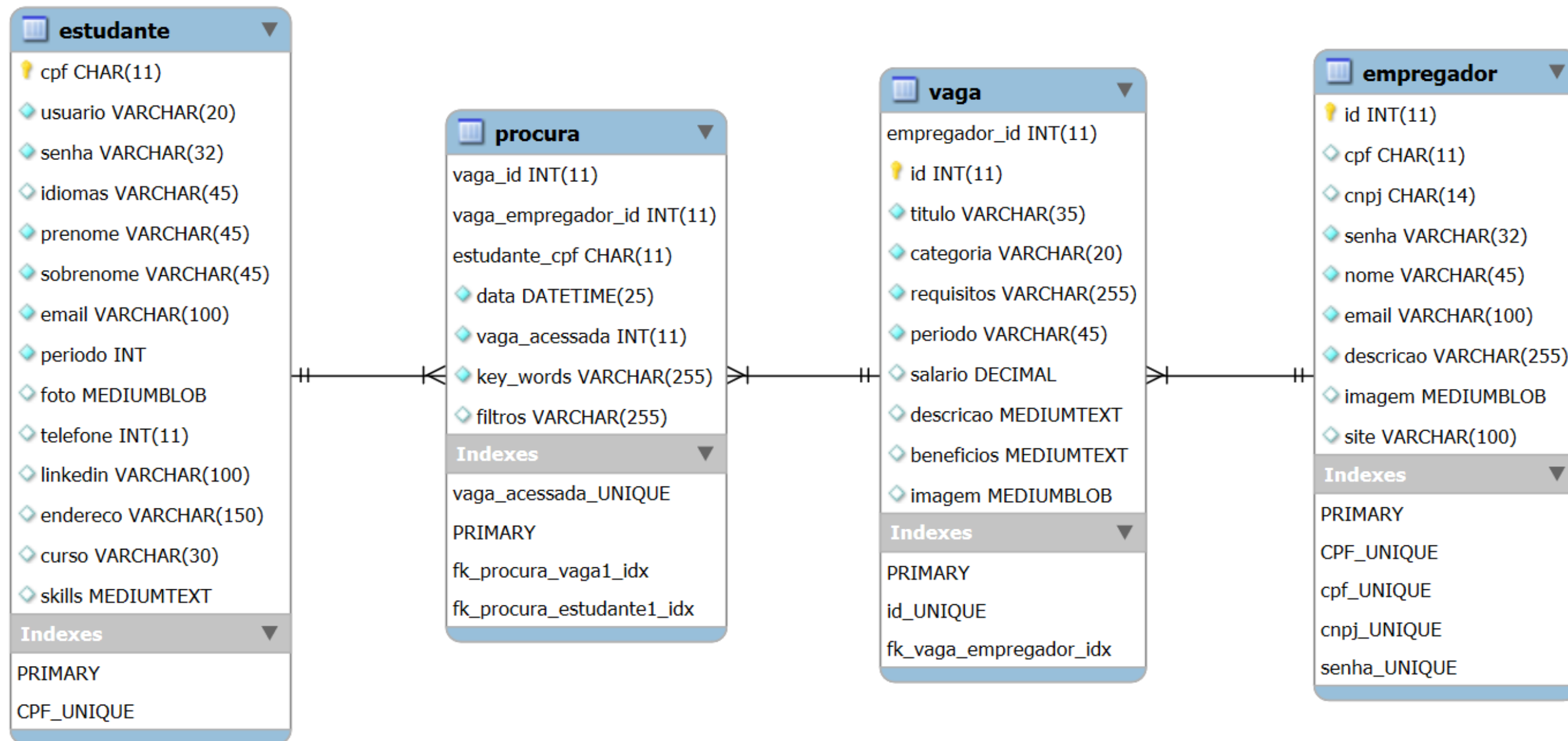
Modelo Conceitual



Modelo Lógico



Modelo de Implementação



Script de Criação

```
-----  
-- Table `mydb`.`vaga`  
-----
```

```
DROP TABLE IF EXISTS `mydb`.`vaga` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`vaga` (  
  `empregador_id` INT(11) NOT NULL,  
  `id` INT(11) NOT NULL,  
  `titulo` VARCHAR(35) NOT NULL,  
  `categoria` VARCHAR(20) NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `requisitos` VARCHAR(255) NOT NULL,  
  `periodo` VARCHAR(45) NOT NULL,  
  `salario` DECIMAL NULL,  
  `descricao` MEDIUMTEXT NULL,  
  `beneficios` MEDIUMTEXT NULL,  
  `imagem` MEDIUMBLOB NULL,  
  PRIMARY KEY (`empregador_id`, `id`),  
  CONSTRAINT `fk_vaga_empregador`  
  FOREIGN KEY (`empregador_id`)  
  REFERENCES `mydb`.`empregador` (`id`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION);
```

```
CREATE UNIQUE INDEX `id_UNIQUE` ON `mydb`.`vaga` (`id` ASC) VISIBLE;
```

```
CREATE INDEX `fk_vaga_empregador_idx` ON `mydb`.`vaga` (`empregador_id` ASC) VISIBLE;
```

```
-----  
-- Table `mydb`.`empregador`  
-----
```

```
DROP TABLE IF EXISTS `mydb`.`empregador` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`empregador` (  
  `id` INT(11) NOT NULL,  
  `cpf` CHAR(11) NULL,  
  `cnpj` CHAR(14) NULL,  
  `senha` VARCHAR(32) NOT NULL,  
  `nome` VARCHAR(45) NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `email` VARCHAR(100) NOT NULL,  
  `descricao` VARCHAR(255) NOT NULL,  
  `imagem` MEDIUMBLOB NULL,  
  `site` VARCHAR(100) NULL,  
  PRIMARY KEY (`id`));
```

```
CREATE UNIQUE INDEX `CPF_UNIQUE` ON `mydb`.`empregador` (`id` ASC) VISIBLE;
```

```
CREATE UNIQUE INDEX `cpf_UNIQUE` ON `mydb`.`empregador` (`cpf` ASC) VISIBLE;
```

```
CREATE UNIQUE INDEX `cnpj_UNIQUE` ON `mydb`.`empregador` (`cnpj` ASC) VISIBLE;
```

```
CREATE UNIQUE INDEX `senha_UNIQUE` ON `mydb`.`empregador` (`senha` ASC) VISIBLE;
```

Script de Criação

```
-----  
-- Table `mydb`.`estudante`  
-----
```

```
DROP TABLE IF EXISTS `mydb`.`estudante` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`estudante` (  
  `cpf` CHAR(11) NOT NULL,  
  `usuario` VARCHAR(20) NOT NULL,  
  `senha` VARCHAR(32) NOT NULL,  
  `idiomas` VARCHAR(45) NULL,  
  `prenome` VARCHAR(45) NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `sobrenome` VARCHAR(45) NOT NULL,  
  `email` VARCHAR(100) NOT NULL,  
  `periodo` INT NOT NULL,  
  `foto` MEDIUMBLOB NULL,  
  `telefone` INT(11) NULL,  
  `linkedin` VARCHAR(100) NULL,  
  `endereco` VARCHAR(150) NULL,  
  `curso` VARCHAR(30) NULL,  
  `skills` MEDIUMTEXT NULL,  
  PRIMARY KEY (`cpf`));
```

```
CREATE UNIQUE INDEX `CPF_UNIQUE` ON `mydb`.`estudante` (`cpf` ASC) VISIBLE;
```

```
-----  
-- Table `mydb`.`procura`  
-----
```

```
DROP TABLE IF EXISTS `mydb`.`procura` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`procura` (  
  `vaga_id` INT(11) NOT NULL,  
  `vaga_empregador_id` INT(11) NOT NULL,  
  `estudante_cpf` CHAR(11) NOT NULL,  
  `data` DATETIME(25) NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `vaga_acessada` INT(11) NOT NULL,  
  `key_words` VARCHAR(255) NOT NULL,  
  `filtros` VARCHAR(255) NULL,  
  PRIMARY KEY (`vaga_id`, `vaga_empregador_id`, `estudante_cpf`),  
  CONSTRAINT `fk_procura_vaga1`  
    FOREIGN KEY (`vaga_empregador_id`, `vaga_id`)  
    REFERENCES `mydb`.`vaga` (`empregador_id`, `id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_procura_estudante1`  
    FOREIGN KEY (`estudante_cpf`)  
    REFERENCES `mydb`.`estudante` (`cpf`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

```
CREATE UNIQUE INDEX `vaga_acessada_UNIQUE` ON `mydb`.`procura` (`vaga_acessada` ASC)  
VISIBLE;
```

```
CREATE INDEX `fk_procura_vaga1_idx` ON `mydb`.`procura` (`vaga_empregador_id` ASC,  
`vaga_id` ASC) VISIBLE;
```

```
CREATE INDEX `fk_procura_estudante1_idx` ON `mydb`.`procura` (`estudante_cpf` ASC)  
VISIBLE;
```

FRONT-END





Nome não
definido

Opções

Editar informações

Editar login

Usuário Comum

Nome:

Nome não definido

CPF:

undefined

Idiomas:

undefined

LinkedIn:

undefined

Skills:

undefined

Curriculo:

Descrição não definida

E-mail:

Email não definido

Telefone:

undefined

Periodo:

undefined

Curso:

undefined

Salvar

Importar PDF

```

    ${(accountType == "userdata") ?

`<div class="col-12 col-md-6">
    <label for="idiomasUser">Idiomas:</label>
    <input type="text" id="idiomasUser" name="titleForm" autocomplete="on"
        placeholder="${personalInfo[0].idiomas}" />
</div>
<div class="col-12 col-md-6">
    <label for="periodoUser">Periodo:</label>
    <input type="text" id="periodoUser" name="titleForm" autocomplete="on"
        placeholder="${personalInfo[0].periodo}" />
</div>
<div class="col-12 col-md-6">
    <label for="linkedinUser">LinkedIn:</label>
    <input type="text" id="linkedinUser" name="titleForm" autocomplete="on"
        placeholder="${personalInfo[0].linkedin}" />
</div>
<div class="col-12 col-md-6">
    <label for="cursoUser">Curso:</label>
    <input type="text" id="cursoUser" name="titleForm" autocomplete="on"
        placeholder="${personalInfo[0].curso}" />
</div>
<div class="col-12 col-md-6">
    <label for="skillsUser">Skills:</label>
    <input type="text" id="skillsUser" name="titleForm" autocomplete="on"
        placeholder="${personalInfo[0].skills}" />
</div>` : ''

}

<div class="col-12">
    <label for="descUser">${(accountType == "userdata") ? "Curriculo" : "Sobre a empresa"}:</label>
    <textarea id="descUser" name="requisForm" placeholder="">${personalInfo[0].desc}</textarea>
</div>
<button type="button" class="btn btn-success" onclick="postUserData()">Salvar</button>

${(accountType == "userdata") ?
`<button type="button" class="btn btn-success" onclick="postUserData()">Importar PDF</button>` : '' }
`

```



Nome não
definido

Opções

Editar informações
Adicionar vagas
Minhas vagas
Editar login

Usuário Empresarial

Nome:

Nome não definido

E-mail:

Email não definido

CNPJ:

undefined

Telefone:

undefined

Sobre a empresa:

Descrição não definida

Salvar

```
$("#profile").html(`<div class="col-12 col-md-6">
    <label for="nomeUser">Nome:</label>
    <input type="text" id="nomeUser" name="titleForm" autocomplete="off"
        placeholder="${personalInfo[0].nome}" />
</div>
<div class="col-12 col-md-6">
    <label for="emailUser">E-mail:</label>
    <input type="text" id="emailUser" name="titleForm" autocomplete="on"
        placeholder="${personalInfo[0].email}" />
</div>

<div class="col-12 col-md-6">
    <label for="cpfUser">${(accountType == "userdata") ? "CPF" : "CNPJ"}:</label>
    <input type="text" id="cpfUser" name="titleForm" autocomplete="on"
        placeholder="${personalInfo[0].cpf}" />
</div>
<div class="col-12 col-md-6">
    <label for="telefoneUser">Telefone:</label>
    <input type="text" id="telefoneUser" name="titleForm" autocomplete="on"
        placeholder="${personalInfo[0].telefone}" />
</div>
```




Categorias Todas



Estágio em engenharia de produção



Somos uma empresa consolidada na área de produção de calçados. Estamos interessado em um estagiário que esteja disponível para começar no próximo mês e que tenha interesse na área de otimização da linha de produção.

[Ver mais](#)



Estágio em recurso humanos



Precisamos de um estagiário para aumentar o controle interno do shopping. Com as mudanças de logística trazidas pela pandemia tivemos que atualizar nosso sistema, portanto o quão antes o candidato puder começar melhor

[Ver mais](#)



Procura-se estagiário de desenvolvimento



Procura-se estagiário URGENTEMENTE, que tenha experiência com HTML5, CSS3, Javascript e que

Oportunidades
Favoritas

```
function newsFeedBox(img, title, desc, id) {

    let dados = JSON.parse(localStorage.getItem("Favoritos"));
    let pos = 0;

    for (j = 0; j < dados.length; j++) {
        if (dados[j].userId == accountId) {
            pos = j;
            break;
        }else if (j == dados.length - 1) {
            pos = dados.length;
            dados.push(new fav(accountId, []));
        }
    }

    let icone = (dados[pos].favs.indexOf(id.toString(10)) < 0) ? "far" : "fas";
    let box = `<div class="card mb-3 col-12">
        <div class="row no-gutters">
            <div class="col-md-4">
                
            </div>
            <div class="col-md-8">
                <div class="card-body">
                    <p style="display:none;">${id}</p>
                    <h5 class="card-title">${title}</h5>
                    <p class="card-text">${desc}</p>
                    <a href="../src/esqueleto.html?id=${id}">Ver mais</a>
                    <i class="${icone} fa-heart favIcon"></i>
                </div>
            </div>
        </div>
    </div>`

    return box;
}
```



OBRIGADO(A)!

—