**Projeto**

**Porque escolhemos o campo de mercado de trabalho?**

Atualmente no Brasil, menos de 1% dos deficientes intelectuais atuam no mercado de trabalho. Nós pensamos que para cada pessoa conseguir sua autonomia, o primeiro passo é ingressar no mercado de trabalho, pois além de trazer a indepência financeira, também desenvolve outros aspectos, como responsabilidade e o convívio social.

**Porque escolhemos esse projeto?**

Não existe no mercado atualmente uma plataforma de empregos voltada para pessoas com deficiência intelectual. As que existem especificamente para pessoas com tipo de deficiência possuem a maioria dos problemas de qualquer outra: muitos textos e propagandas, o que atrapalha a usabilidade de uma pessoa com DI. Pensando no demasiado tempo gasto para encontrar um emprego na internet, e mesmo quando encontrado não possuir a garantia de que aquele é realmente o melhor que esteja cadastrado na plataforma, decidimos adicionar a funcionalidade de match, para que as vagas compatíveis com o candidato sejam tragas diretamente para ele. Pensando também na autonomia, toda a plataforma possui em todas as telas imagens e ajudas com áudio e vídeo, para pessoas semi-analfabetas.

**Como vamos monetizar?**

A monetização ocorrerá quando uma empresa decidir que deseja ofertar vagas na nossa plataforma. Para que ela possa realizar esta ação, ela deve contratar um plano que tem seu valor baseado no número de vagas que ele deseja ofertar.

**Por que login com QR Code?**

Após uma visita à APAE, nos foi informado que muitos dos deficientes intelectuais possuem dificuldade em lembrar suas credenciais (emails e senhas), a partir deste ponto, decidimos desenvolver o login por QR Code afim de facilitar essa validação do usuário.

**Por que utilizar GIT?**

O GIT é um versionador que permite que várias pessoas trabalhem no mesmo projeto ao mesmo tempo. Por ser um versionador, ele mantém um histórico de cada arquivo, com informações como: quem fez, porquê fez e como fez. Dessa forma, é facilitada a organização do projeto, pois nada é perdido.

**Por que opção de gravar vídeo na descrição do currículo?**

Após uma visita à APAE, nos foi informado que deficientes intelectuais possuem muita dificuldade em lidar com longos textos, pois confundem e eles perdem um longo tempo tentando entender o que está sendo passado. Pensando nisso, habilitamos a opção de gravar vídeo para aqueles que se sentem mais à vontade falando do que escrevendo.

**Front-End**

**Por que desenvolver o mobile/app primeiro?**

Após uma visita à APAE, nos foi informado que a maioria dos deficientes intelectuais não possuem acesso a um computador em casa, porém a maioria possuem celulares e sentem uma maior facilidade ao lidar com aplicativos.

**Por que utilizar sass?**

Sass é uma linguagem baseada em CSS, que após ser compilada, gera o CSS que os navegadores entendem. Ele agiliza e facilita a construção do front-end, pois dispõe de vantagens como a criação de variáveis para propriedades repetidas, como uma cor padrão; também dispõe de mixins, que permitem reutilizar um trecho de código várias vezes sem precisar repeti-lo, além da possibilidade de aninhar o código, deixando a hierarquia mais visível e fácil de compreender.

**Qual foi o critério para a escolha de cores?**

Nossa plataforma tem o layout baseado nas cores azul e branco. De acordo com preceitos UX, é ideal que um aplicativo disponibilize de 3 a 4 cores, para que a visualização seja simples e limpa. Segundo a psicologia das cores, a cor azul representa calma, confiança e segurança, o que é exatamente o que queremos transmitir na nossa plataforma. Os ícones coloridos foram escolhidos para chamar a atenção e serem amigáveis.

**Por que separar o cadastro em etapas?**

Separar o cadastro em etapas permite simplificar um processo longo, como o cadastro do currículo por exemplo, além de ajudar o usuário a focar em uma coisa por vez. Isso diminui as chances de o usuário preencher algo incorretamente e torna o processo de cadastro menos cansativo.

**Por que utilizar tantas imagens?**

Em uma visita à APAE, nos foi informado que pessoas com deficiência intelectual possuem grande dificuldade ao se deparar com textos longos, e que, quando há imagens ou ícones acompanhando esses textos há uma associação do termo com a figura, e assim se torna mais simples o processo de aprender a utilizar um aplicativo, ou qualquer plataforma.

**Back-End**

**Por que usar php?**

O PHP é uma das linguagens mais utilizadas pelo mercado atualmente, assim, há bastante comunidades e fóruns na internet, o que auxilia quem está desenvolvendo.

Em questões de desenvolvimento, é uma linguagem madura e bem difundida, com várias funções prontas para uso, possui uma sintaxe fácil e, por ser fracamente tipada, proporciona uma liberdade maior para quem está programando. Em questão de conexão com banco de dados, ela dispõe de uma grande compatibilidade: Sybase, MySQL e MS-SQL são alguns exemplos.

Além disso, dois integrantes do grupo possuem conhecimento prévio na linguagem, o que nos proporcionou maior confiança e agilidade no desenvolvimento do projeto.

**Por que usar um framework?**

Ele oferece uma base sólida a um projeto, implementando ferramentas e uma estrutura (a maioria MVC) que permite ao desenvolvedor ir direto à programação, ao invés de perder muito tempo criando essa mesma estrutura facilmente oferecida por um framework, além de uma grande estabilidade. Ele também se fez necessário devido ao curto período de tempo que nos foi dado para desenvolvimento.

Seus benefícios se estendem também à segurança, pois eles trazem configurações pré-definidas que protegem seu projeto de ataques comuns, como SQL Injection, por exemplo.

**Por que o Laravel?**

O Laravel, além de trazer todas as vantagens padrões que uma linguagem PHP e um framework disponibilizam, também traz algumas particulares:

**Blades**

Um compilador que separa o html do php, tornando o código mais organizado. Ele também possui heranças e sessões, o que facilita o reuso de código.

**Eloquent ORM e QueryBuilder**

Uma interação direita entre o PHP e o banco de dados, tornando possível manipular os dados do banco como se fossem uma classe do projeto.

Além disso, o queryBuilder é um “construtor” de querys, o que organiza seu código e te protege de ataques por meio de SQL Injection.

**Helpers**

Uma classe onde é possível criar uma função e acessá-la de qualquer parte do seu projeto, o que evita a duplicação de código.

**Migrations**

Uma pasta que é usada para guardar toda a estrutura do banco de dados, assim, não é preciso guardar um backup e depois rodar por um SGBD. Basta apenas executar um comando e o banco de dados é criado.

**Sistema de rotas**

Realizam um mapeamento de URLs digitadas no navegador para alguma ação dentro da plataforma.

**Como o projeto está organizado?**

A organização tem como base o modelo MVC. No projeto, há uma estrutura organizada de pastas, essas são:

**Resources**

Contém: diretório JS que possui todos os arquivos JavaScript não compilados; lang que possui todas as línguas que podem ser traduzidas do inglês; Sass que possui todos os arquivos sass não compilados.

**Public**

Conta com as pastas: *CSS* e *JS* que salvam os arquivos JS e SASS compilados pelo npm; *image* que possui todas as imagens do sistema, como as fotos de perfil dos candidatos e os componentes do sistema;

**App**

Representa as funcionalidades Back-End. Nesse diretório, as classes modelos se encontram na raiz, elas representam cada tabela existente no banco de dados.

**Http**

Nesta pasta temos o Controllers, que contém as classes php responsáveis pela lógica da aplicação. Elas realizam validações, funções de CRUD, redirecionamento e o que mais for necessário para o tratamento daquele componente.

**Database**

Responsável por armazenar os arquivos php que gerenciam o banco de dados. Dentro da pasta migration você pode encontrar arquivos que geram as tabelas e, dentro da pasta seeds, o arquivo que irá popular as tabelas.

**Routes**

Armazena os arquivos que realizam os redirecionamentos. Dentro do arquivo web.php contém todas as rotas do projeto que utilizam o protocolo HTTP para ir a um Controller. Já o arquivo api.php faz as referencias às API’s utilizadas no projeto.

**Quais são as funcionalidades do projeto?**

Um candidato pode realizar um cadastro esse cadastro valida se o email e o nome de usuário ja existem, se a senha contem 8 dígitos e se ela foi confirmada.

Um candidato pode atualizar suas informações de se desejar.

Um usuário que não tem cadastro não tem acesso a paginas que um usuário com login teria e um usuário que tenha login não tem acesso paginas como login e cadastro por exemplo.

Um candidato pode cadastrar um currículo que por sua vez pode ser editado.

E as vagas só aparecem se o candidato supre as necessidades da vaga

**Quais são medidas de segurança do sistema?**

Como estamos utilizado Laravel, toda a segurança do sistema é proporcionada automaticamente pelo framework, através de autenticações e validações de dados inseridos pelo usuário. Além disso, senhas de usuário são criptografadas pelo método hash password\_bcrypt, o que protege esse dado sensível caso um possível vazamento ocorra, além de não deixá-lo exposto a quem possui acesso ao banco de dados.

**Banco de Dados**

**Por que utilizar MySQL?**

O MySQL possui uma alta escalabidade e flexibilidade, o que permite armazenar uma grande quantidade de dados sem nenhum problema e, por ser open-source, a customização de funcionalidades personalizadas para o servidor de banco de dados. Além disso, uma das suas vantagens é seu alto desempenho, sendo capaz de lidar com um grande volume de dados sem perder a qualidade, para conseguir isso, ele conta com um ótimo mecanismo de consulta e a capacidade de inserção de dados extremamente rápida.

**Como funciona o match?**

A maioria das informações que uma empresa cadastra em uma vaga, o usuário cadastra também em seu currículo. Dessa forma, o sistema compara e busca quais vagas tem informações compatíveis com as informações de um currículo.

**Rede Local?**

Como estamos utilizado a linguagem php, precisamos de um servidor que possa interpretar o código e compilar o código php para html portando necessitamos de um servidor apache para que possamos visualizar as telas. Alem de usar o servidor apache o Laravel conta com um servidor local que abre uma porta 8000 que indica aonde a aplicação esta sendo utilizada, para acessar a aplicação basta acessar a url <http://localhost:8000>, caso deseje que um outro dispositivo de forma remota basta conectar a rede wifi local e acessar o endereço de ip com a porta do host. Com dois simples comandos podem ser inicializadas ambos servidores