### Documentação Mentorin

*Codinome: Projeto Carcará*

*Por: Lucas Barbosa Brancalhão*

#### *O que é o Mentorin?:*

O Mentorin consiste em um software de gestão e match entre um possível mentorado e um mentor, para que um possa auxiliar o outro.

O software possibilitará a Fomenta Vale automatizar o processo de combinar mentores e mentorados, que hoje é feito manualmente, além de dar uma visão geral de todos os mentores e mentorados cadastrados, o que pode ser estendido para outras empresas caso o projeto seja comercializado.

#### Tecnologias usadas:

***Backend:***

* *Typescript*
* *Nestjs*
* *Bcrypt*
* *MySql / Xampp*
* *Prisma*
* *Argon2*
* *Passport*
* *Nodemailer*
* *handlebars*

***Dependências Backend:***

* *Class-validator*
* *Reflect-metadata*
* *Uuidv4*
* *Nestjs/typeorm*
* *Passport-jwt*
* *Class-transformer*

***Comentários sobre o Backend:***

O projeto foi produzido em cima do Framework NestJS, já que esse tem uma boa escalabilidade, além de fácil curva de aprendizado. Porém, ao que for trabalhar nesse projeto, é ideal que exita uma familiaridade com o superset Typescript, já que para que o projeto fosse mais seguro e de melhor manejo, foi utilizado a ferramenta.

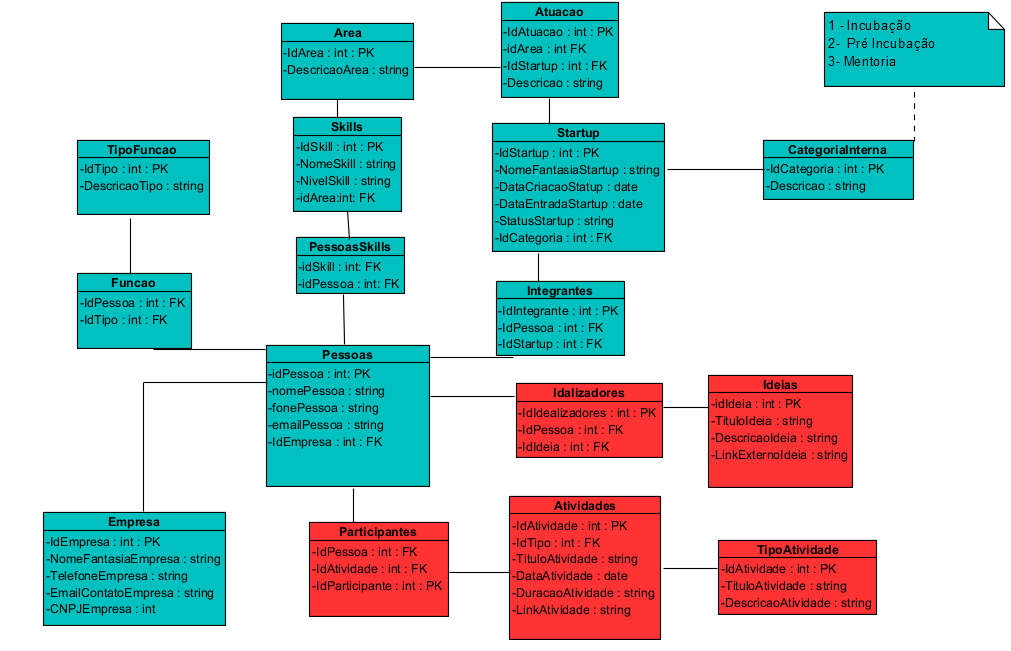
O banco de dados utilizado foi o MySQL, novamente por praticidade de uso, já que o mesmo oferece um ambiente pré-pronto para o fácil manejo e criação da base de dados.

No caso do ORM (Object-Relational-Mapping / Mapeamento de Objeto Relacional) foram feitas várias tentativas de diferentes ORMs, mas o que se viu mais apropriado foi o Prisma, já que o mesmo foi de fácil aprendizagem, boa implementação de relacionamento entre tabelas e boa escalabilidade.

Além disso foram usadas outras tecnologias que serão mencionadas ao longo do documento.

***Tabelas:***

Foi feito uma estrutura prévia das tabelas que seriam usadas no projeto. Segue abaixo o desenho feito.



Como pode ser visto acima, foram prototipadas 16 tabelas, das quais 11 foram coloridas como ***Azuis*** e 5 como ***Vermelhas***.

***Azul - Primeira Fase:***

Tabelas que foram codificadas inicialmente. - ***MVP (Minimum Viable Product)***

***Vermelho - Segunda Fase:***

Tabelas que serão codificadas futuramente.

***Relacionamento:***

**(Ternário)** - Pode ser vazio ou não (Operador “? - Ternário”).

**(1:1)** - Relação “Um para Um”.

**(1:m)** - Relação “Um para Muitos”.

**(m:m)** - Relação “Muitos para Muitos”.

***Primeira Fase:***

**Funcao - TipoFuncao** - 1:1;

**Area - Skills** - 1:m;

**Pessoas - Skills** - m:m; - Como é o caso de muitos para muitos, foi utilizado uma tabela auxiliar PessoasSkills, em que cada tabela tem a relação 1:m, respectivamente.

**Funcao - Pessoas** - 1:m;

**Pessoas - Integrantes** - 1:m;

**Empresa - Pessoas** - 1:m;

**Startup - Integrantes** - 1:m;

**Atuacao - Startup** - 1:m; - Foi utilizada a relação 1:m (Um para Muitos), mas talvez seria interessante usar uma relação m:m (Muitos para Muitos), já que uma Startup pode ter muitas atuações, e muitas atuações podem pertencer a muitas Startups.

**Area - Atuacao** - 1:m - Da mesma forma foi utilizada a relação 1:m (Um para Muitos), mas da mesma forma como o exemplo acima, talvez seja bom usar uma relação m:m (Muitos para Muitos), já que uma área pode pertencer a muitas atuações e muitas áreas podem pertencer a uma atuação.

**CategoriaInterna - Startup** - 1:m - Talvez seria válido fazer uma relação da tabela CategoriaInterna com outras tabelas, como por exemplo Pessoas, já que ela também tem a função de categorizar se alguém é mentor ou Mentorado.

***Segunda Fase:***

As relações da segunda fase de implementação não foram especificadas, já que são de uma fase futura que será codificada, mas pode se imaginar as seguintes relações:

**Idealizadores - Ideias** - 1:m;

**Idealizadores - Pessoas** -1-1;

**Pessoas - Participantes** - 1:1;

**Participantes - Atividades** - 1:m;

**Atividades - TipoAtividade** - 1:m;

Todas as relações entre tabelas funcionam manualmente, porém, para melhor manipulação do sistema, algumas relações foram comentadas para a futura automação da relação, como foi feito no caso das tabelas Usuário e Pessoa.

**Descrição de Tabelas:**

**Pessoa** - A tabela de pessoas registra todos os dados de uma pessoa física.

**Funcao** - Relaciona uma pessoa e uma função atribuída pela empresa/startup onde a mesma trabalha.

**TipoFuncao** - Descreve o tipo de uma determinada função atribuída.

**PesoasSkils** - Tabela intermediária que faz a relação entre a tabela Skills e a tabela Pessoas.

**Skills** - Define uma habilidade relacionada a uma determinada área de atuação que uma pessoa pode possuir.

**Area** - Descreve a área em que uma habilidade/skill pode ser melhor encaixada. A tabela está relacionada com a atuação.

**Atuacao** - Define a atuação em que uma Area se encaixa. Também define a atuação de uma determinada Startup cadastrada.

**CategoriaInterna** - Descreve a situação de uma Startup cadastrada, seja Incubada ou não, ou no caso de uma Pessoa, Mentor ou não.

**Startup** - Descreve uma Startup incubada ou não.

**Integrantes** - Descreve os integrantes de uma Startup e as relaciona com pessoas, já que cada integrante precisa ser obrigatoriamente uma pessoa.

**Empresa** - Define uma empresa cadastrada a quem um mentor ou mentorado pertence.

**Participantes** - Define os participantes do “Match” entre mentores e mentorados. Também pode ser usada para saber quais os participantes de uma atividade(Reunião, Mentoria, etc.).

**Idealizadores** - Define os idealizadores de uma ideia, pessoas mentoradas.

**Ideias** - Define as idéias de pessoas idealizadoras querem implementar / ser mentoradas.

**Atividades** - Define uma atividade a ser realizada por participantes, como Reunião, Mentoria, Etc.

**TipoAtividade** - Define o tipo da Atividade sendo executada (Reunião, Mentoria, etc.)