

## **FIAP**

Enzo Trevisan RM94745

Karen Mastrogiacomio Antiquera RM94488

Karina Megumi Sakamoto RM95121

Thiago Martins Prado RM94426

Vinícius Renan Benjamim Aoyagui RM93386

## **Projeto Syncing:**

Challenge IBM

São Paulo, SP

2022

## Capítulo 1

### Objetivos e justificativa da solução de forma objetiva

O projeto Syncing tem como propósito diminuir e resolver problemas frequentes na área de Recursos Humanos por meio de uma plataforma digital. Isso ocorre de forma que, a comparação de capacidades e talentos de um candidato supra aquilo que é necessário para a inscrição para a vaga da empresa. Vale ressaltar que a plataforma tem total foco na área de Tecnologia da Informação (T.I.).

Para que os candidatos e empresas consigam utilizar a nossa plataforma, é necessário que ambos se cadastrem. Após o cadastro e o login, a empresa poderá criar e anunciar vagas. Os candidatos cadastrados poderão ver as vagas disponíveis e se candidatar à uma ou mais delas. É importante ressaltar que apenas candidatos cadastrados poderão ver as vagas.

Salienta-se que Tags serão criadas em nosso site, a fim de descrever melhor e de maneira mais prática as habilidades do candidato e as habilidades necessárias (ou favoráveis) para a vaga. Assim que o candidato inserir suas Tags, poderá se candidatar a vaga. E antes de confirmar a inscrição, aparecerá essa porcentagem de compatibilidade. Assim, o candidato poderá analisar se realmente é apto para vaga. Se o candidato optar por continuar o processo, a empresa também receberá a mesma porcentagem. Sendo assim, não será necessário ler o currículo completo do candidato para ter uma noção geral de suas habilidades.

## Capítulo 2

### Primeiro requisito funcional (Cadastro)


#### Protótipos das telas envolvidas

Tela 1-

SoulCoderz.

Menu principalProcurar vagasLoginRegistrar-se

Abra sua conta para novas oportunidades



Quer se registrar como Empresa?

Candidato

Nome completo

CPF

Email

Celular

Senha

Confirmar senha

Gênero

☐ Masculino

☐ Feminino

☐ Não binário

Cadastrar-se

Tela 2-

SoulCoderz.

Menu principalProcurar vagasLoginRegistrar-se

Encontre os candidatos qualificados que está procurando



Quer se registrar como Candidato?

Empresa

Nome da empresa

CNPJ

Email

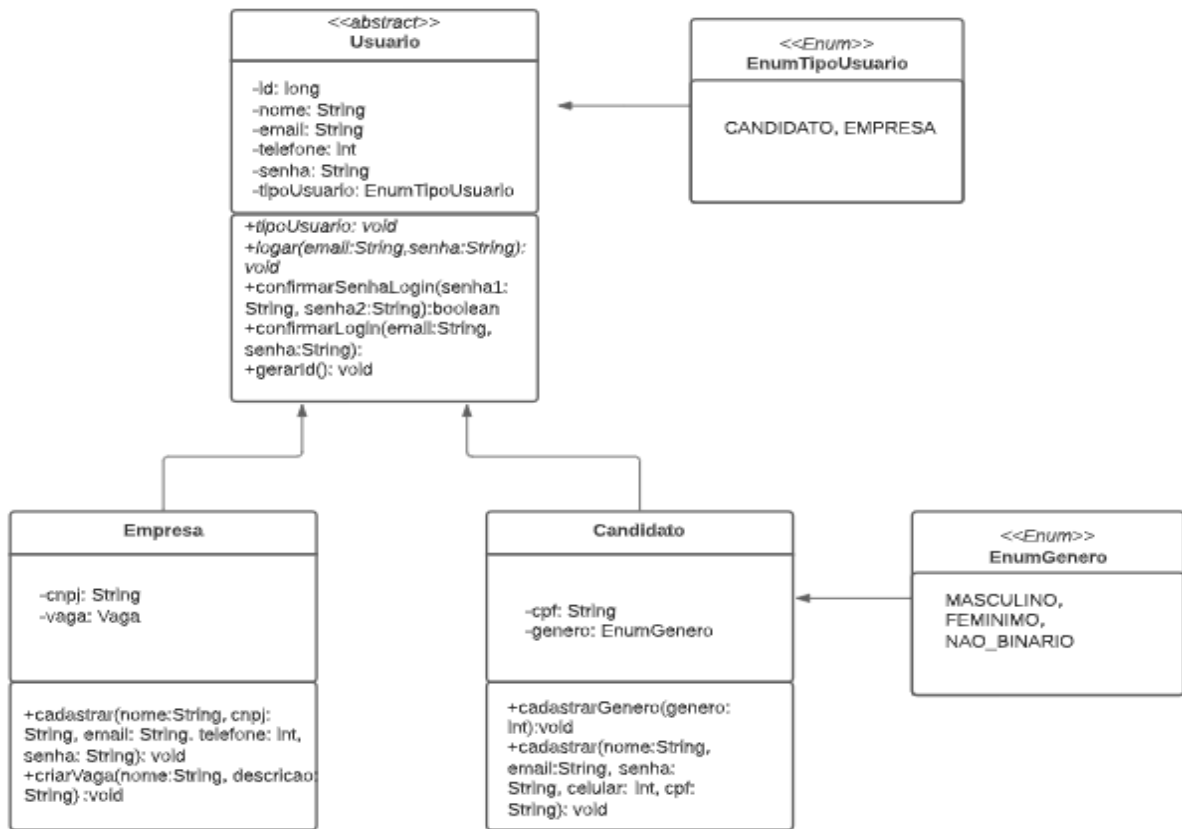
Telefone

Senha

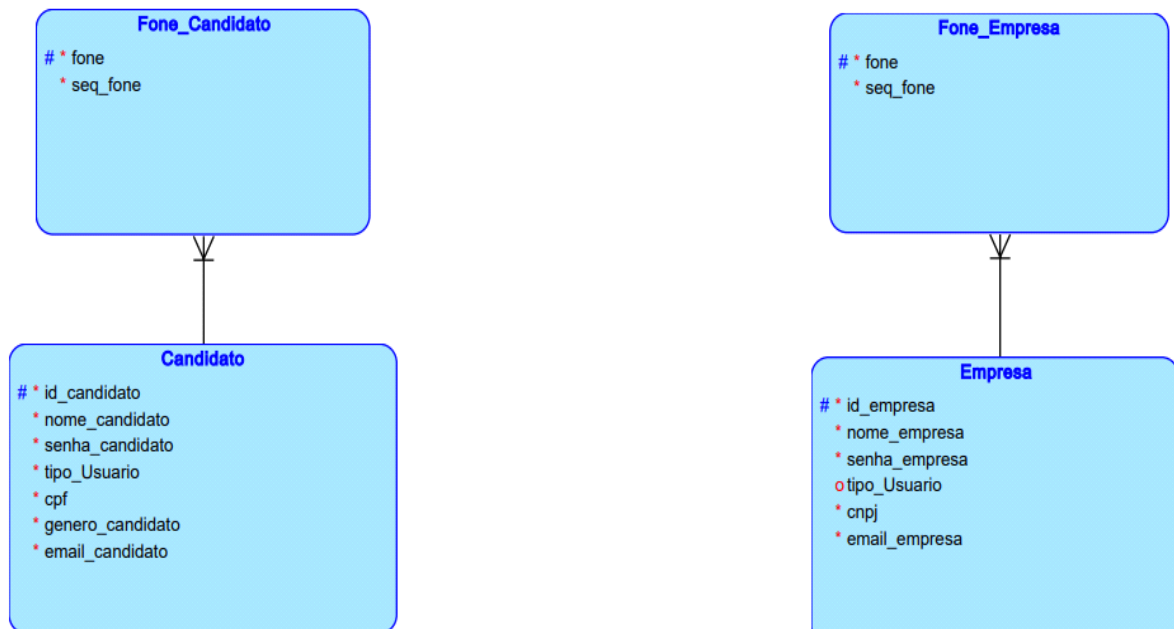
Confirmar senha

Cadastrar-se

Apontamento das classes envolvidas (diagrama de classes)



Modelo do banco de dados



## Segundo Requisito Funcional (Vagas)

Protótipos das telas envolvidas

Tela 1-

Filtragem de vagas

Two input fields for filtering jobs.

Vaga 1

Vaga 2

Vaga 3

This wireframe represents a job filtering interface. It features a title 'Filtragem de vagas' at the top. Below the title are two empty rectangular input fields for search criteria. Further down, there are three stacked rectangular boxes, each labeled 'Vaga 1', 'Vaga 2', and 'Vaga 3' respectively, representing individual job listings.

Tela 2-

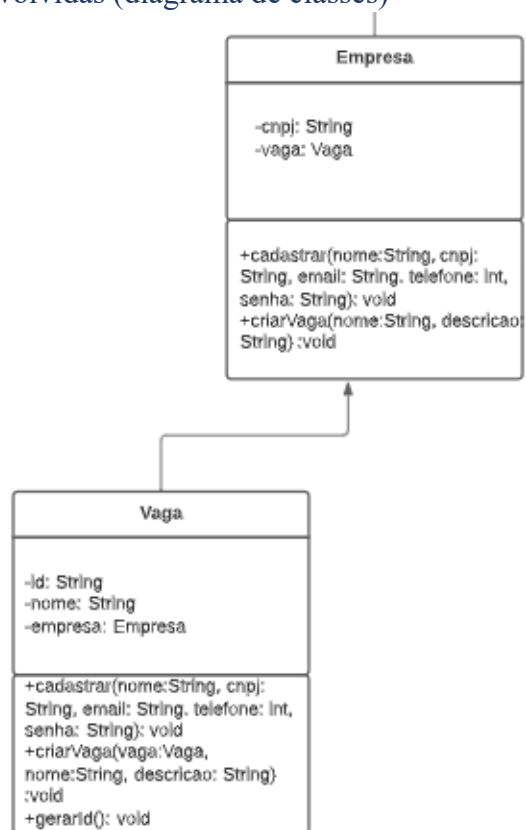
Nome da vaga

Descrição da vaga

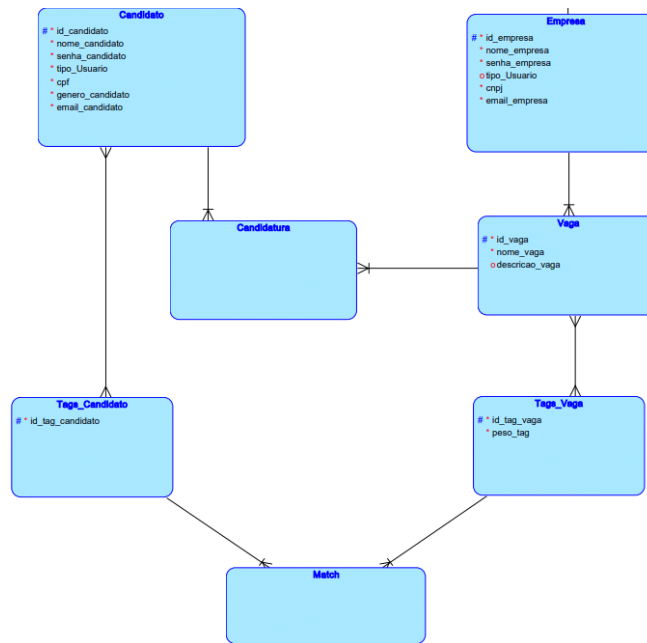
Se candidate

This wireframe represents a job details page. It has a title 'Nome da vaga' at the top. Below it is a large rectangular area labeled 'Descrição da vaga' for the job description. At the bottom right, there is a button labeled 'Se candidate' for applying to the job.

Apontamento das classes envolvidas (diagrama de classes)



Modelo do banco de dados



## Capítulo 3

### Primeiro Requisito Funcional (Cadastro)

#### Títulos

Título da Tela 1: Inserção de dados (candidato)

Título da Tela 2: Inserção de dados (empresa)

#### Funcionalidades

Tela “Inserção de dados(candidato)”

Utilizada para que o usuário candidato consiga se registrar na nossa plataforma

Tela “Inserção de dados (empresa)”

Utilizada para que o usuário empresa consiga se registrar na nossa plataforma

#### Classes Java

Tela “Inserção de dados(candidato)”

```

public void cadastroCandidato() {
    System.out.println("Cadastro do candidato:");
    System.out.println("Nome: ");
    String nomeCandidato = ent.next();
    System.out.println("Cpf:");
    String cpf = ent.next();
    candidato.tipoUsuario();
    cadastroFone(candidato.tipoUsuario());
    System.out.println("Email: ");
    String emailCandidato = ent.next();

    String senhaCandidato = "";
    boolean verificar;

    do {

        System.out.println("Senha: ");
        senhaCandidato = ent.next();
        System.out.println("Confirme sua senha: ");
        String senhaCandidato2 = ent.next();

        verificar = candidato.confirmarSenhaLogin(senhaCandidato,senhaCandidato2);
    }while(verificar==false);

    System.out.println("Gênero:\n1. Masculino\n2. Feminino\n3. Não Binário");
    int genero = ent.nextInt();

    candidato.cadastrarGenero(genero);
    candidato.gerarID();
    candidato.cadastrar(nomeCandidato,senhaCandidato,emailCandidato,cpf);
    try {
        candidatoDAO.Inserir(candidato);
    } catch (SQLException e) {
        e.printStackTrace();
    }

}
}

```



Tela “Inserção de dados (empresa)”

```
public void cadastroEmpresa() {
    System.out.println("Cadastro da empresa:");
    System.out.println("Nome: ");
    String nomeEmpresa = ent.next();
    System.out.println("E-mail:");
    String emailEmpresa = ent.next();
    System.out.println("CNPJ: ");
    String cnpj = ent.next();
    empresa.tipoUsuario();

    cadastroFone(empresa.tipoUsuario());

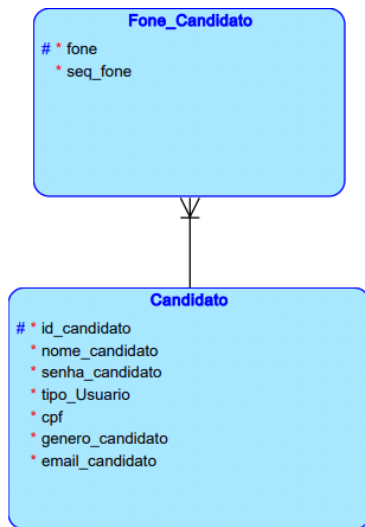
    String senhaEmpresa = "";
    boolean verificar = false;
    do {
        System.out.println("Senha: ");
        senhaEmpresa = ent.next();
        System.out.println("Confirme sua senha: ");
        String senhaEmpresa2 = ent.next();

        verificar = empresa.confirmarSenhaLogin(senhaEmpresa, senhaEmpresa2);
    } while (verificar == false);

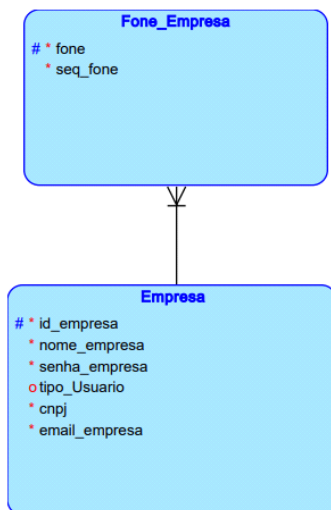
    empresa.gerarID();
    empresa.cadastrar(nomeEmpresa, emailEmpresa, cnpj, senhaEmpresa);
    try {
        empresaDAO.Inserir(empresa);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Tabelas do banco de dados utilizadas

Tela “Inserção de dados(candidato)”



Tela “Inserção de dados (empresa)”



## Segundo Requisito Funcional (Vagas)

Títulos

Tela 1: Pesquisa vaga

Tela 2: Vaga detalhes

Funcionalidades

Tela “Pesquisa vaga”

Utilizada para que o candidato consiga achar a vaga que se adeque para ele

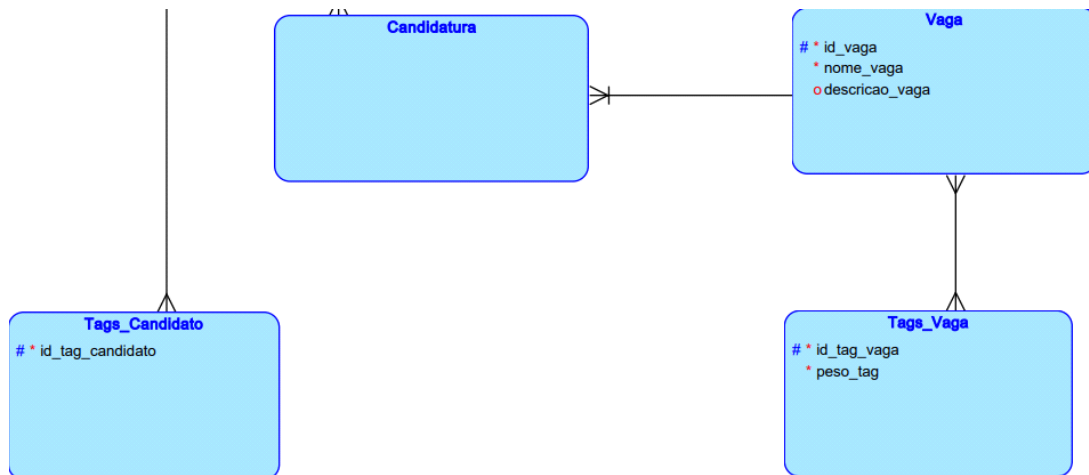
Tela “Vaga detalhes”

Utilizada para que o candidato consiga ter mais informações sobre a vaga prestes a se candidatar

## Classes Java

```
public void cadastroVaga() {  
    System.out.println("Crie sua vaga:");  
  
    System.out.println("Digite o nome da vaga:");  
    String nomeVaga = ent.next();  
    System.out.println("Digite a descrição:");  
    String descricaoVaga = ent.next();  
    empresa.criarVaga(nomeVaga, descricaoVaga);  
    vagaDAO.Inserir(empresa);  
}
```

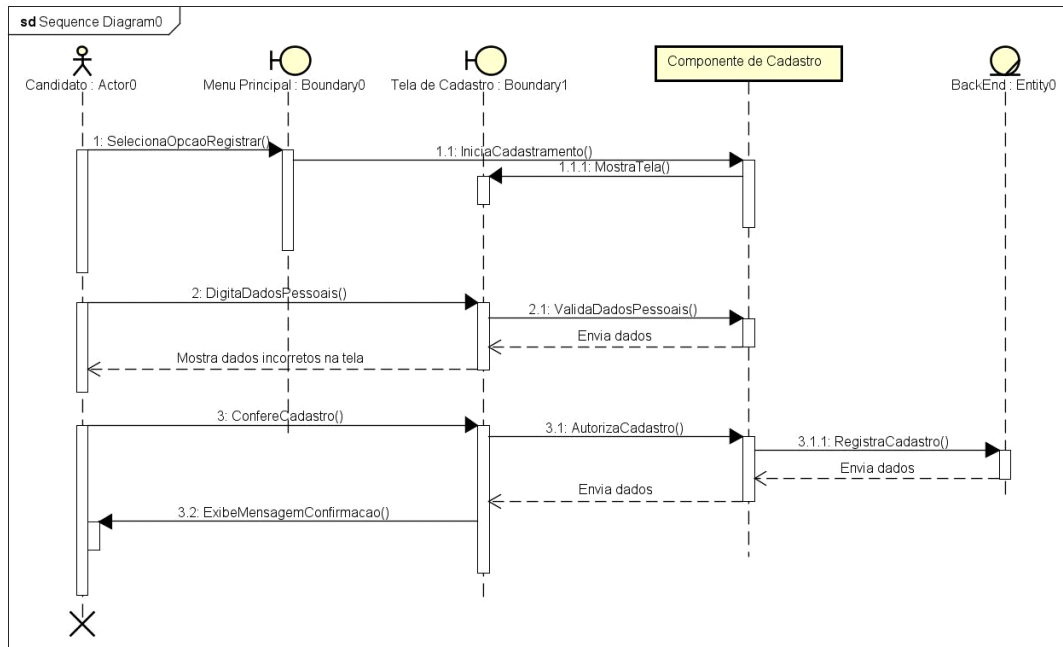
## Tabelas do banco de dados utilizadas



## Capítulo 4

### Cadastro

#### Apresentação dos requisitos funcionais



SeleccionaOpcaoRegistrar() – O usuário deve escolher qual é o tipo de cadastro

IniciaCadastramento() – Após o usuário escolher o tipo de cadastro, o sistema inicia o processo de catalogação

MostraTela()- O Componente de cadastro mostra a tela de cadastro para o usuário

DigitaDadosPessoais() – O usuário digita os dados pessoais na plataforma

ValidaDadosPessoais() – O componente de cadastro verifica se há algum erro no cadastro

ConfereCadastro() – O usuário irá confirmar o seu cadastro

AutorizarCadastro() – O componente de cadastro irá autorizar o cadastro

RegistrarCadastro() – Os dados serão armazenados no banco de dados

ExibirMensagemConfirmação() – Irá enviar uma mensagem, confirmando o cadastro

#### Apresentação dos requisitos não funcionais

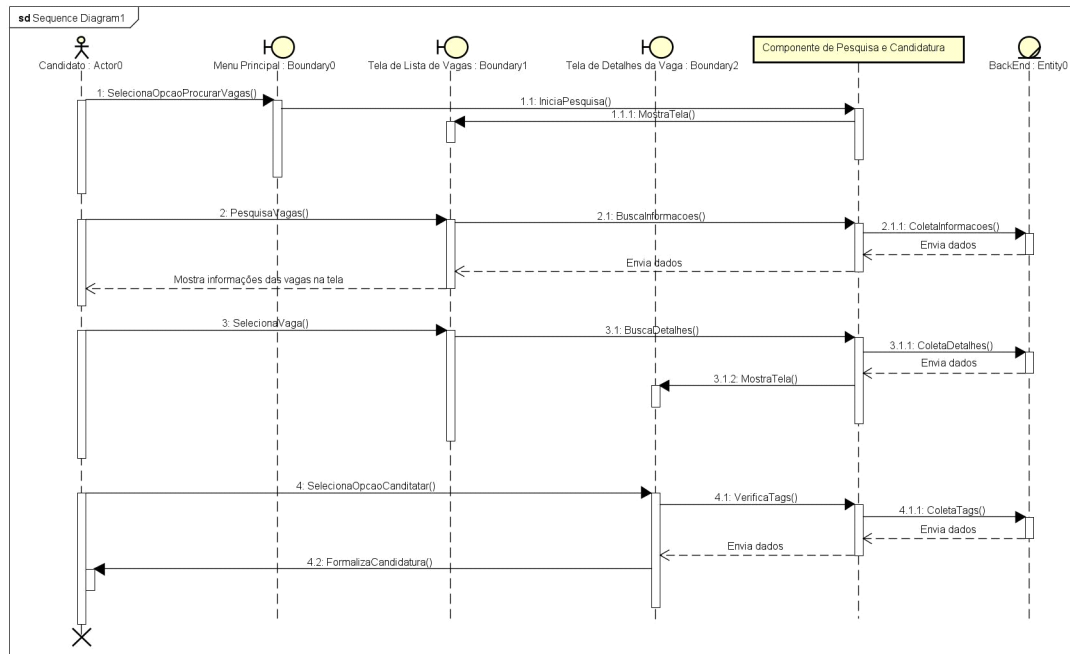
Tempo de resposta por parte do usuário, no caso, podendo durar 5 minutos

#### Apresentação das regras de negócios

Condição que deve ser atendida: possuir um e-mail para que seja possível o cadastro.

## Vagas

### Apresentação dos requisitos funcionais



SelecionaOpcaoProcurarVagas() – O usuário pressiona o botão “Procurar vagas” para que encontre uma vaga

IniciaPesquisa() – A pesquisa é iniciada

MostraTela() – É apresentada ao usuário algumas vagas, sem nenhum filtro

PesquisaVaga- O usuário procura a vaga com os filtros escolhidos

BuscaInformações() -Busca informações no BackEnd

ColetaInformações() – Coleta as informações escolhidas pelo backEnd

SelecionaVaga() – O usuário escolhe uma vaga que foi fornecida a ele

BuscaDetalhes() - Busca os detalhes no BackEnd

ColetaDetalhes() – Coleta os detalhes fornecidos pelo BackEnd

MostraTela() – Mostra na tela do usuário, os detalhes da vaga

SelecionaOpcaoCandidatar() – O usuário pressiona a opção para candidatar a vaga

VerificaTags() – O sistema vai verificar as tags

ColetaTags() – O sistema coleta as Tags fornecidas pelo BackEnd

FormalizaCandidatura() – O sistema irá perguntar ao usuário se pode formalizar a candidatura

### Apresentação dos requisitos não funcionais

Tempo de resposta por parte do usuário, no caso, podendo durar 2 horas, caso ao contrário, irá sofrer Logout automático

### Apresentação das regras de negócios

O usuário deverá ter uma conta registrada na plataforma

O usuário deverá ter tags cadastradas na conta