

**INSTITUTO INFNET**  
**ESCOLA SUPERIOR DE TECNOLOGIA**  
**GRADUAÇÃO EM ENGENHARIA DE**  
**SOFTWARE**



**ENGENHARIA DISCIPLINADA DE**  
**SOFTWARE:**  
**PROJETO DE BLOCO - TP3**

**Luiza Mendes**

# Índice

<b>1. Projeto</b>	<b>3</b>
1.1 VenturaHR	3
1.2 Atores	3
1.2.1 Recrutador	3
1.2.2 Candidato	3
1.2.3 Sistema	3
1.2.4 Administrador	3
1.3 Casos de Uso	4
1.3.1 Diagrama de caso de uso	4
1.3.2 Lista e classificação	6
1.4 Requisitos	7
1.4.1 Matriz de requisitos	7
1.5 Modelo de domínio	9
1.6 Diagrama de classes	10
1.6.1 Responsabilidade dos objetos	10
1.6 Diagramas Sequenciais	12
1.6.1 Aplicar para vaga	12
<b>2. Processo de Desenvolvimento de Software</b>	<b>13</b>
2.1 TP1	13
2.2 TP3	13

# 1. Projeto

## 1.1 VenturaHR

VenturaHR é um sistema para auxiliar o recrutamento de talentos, onde os recrutadores das empresas cadastram suas vagas em aberto, os candidatos podem se inscrever nelas e o sistema gradua os candidatos de acordo com suas informações e os requisitos da vaga, processando essas informações em um relatório para a empresa contratante. O sistema também gera relatórios e faz gestão de vagas expiradas. Link para github: <https://github.com/luizamendes/VenturaHR>

## 1.2 Atores

### 1.2.1 Recrutador

É o funcionário da empresa cliente responsável por cadastrar as vagas no sistema, por visualizar as candidaturas recebidas e fechar ou estender a vaga.

### 1.2.2 Candidato

É a pessoa que irá visualizar as vagas no sistema e se candidatar.

### 1.2.3 Sistema

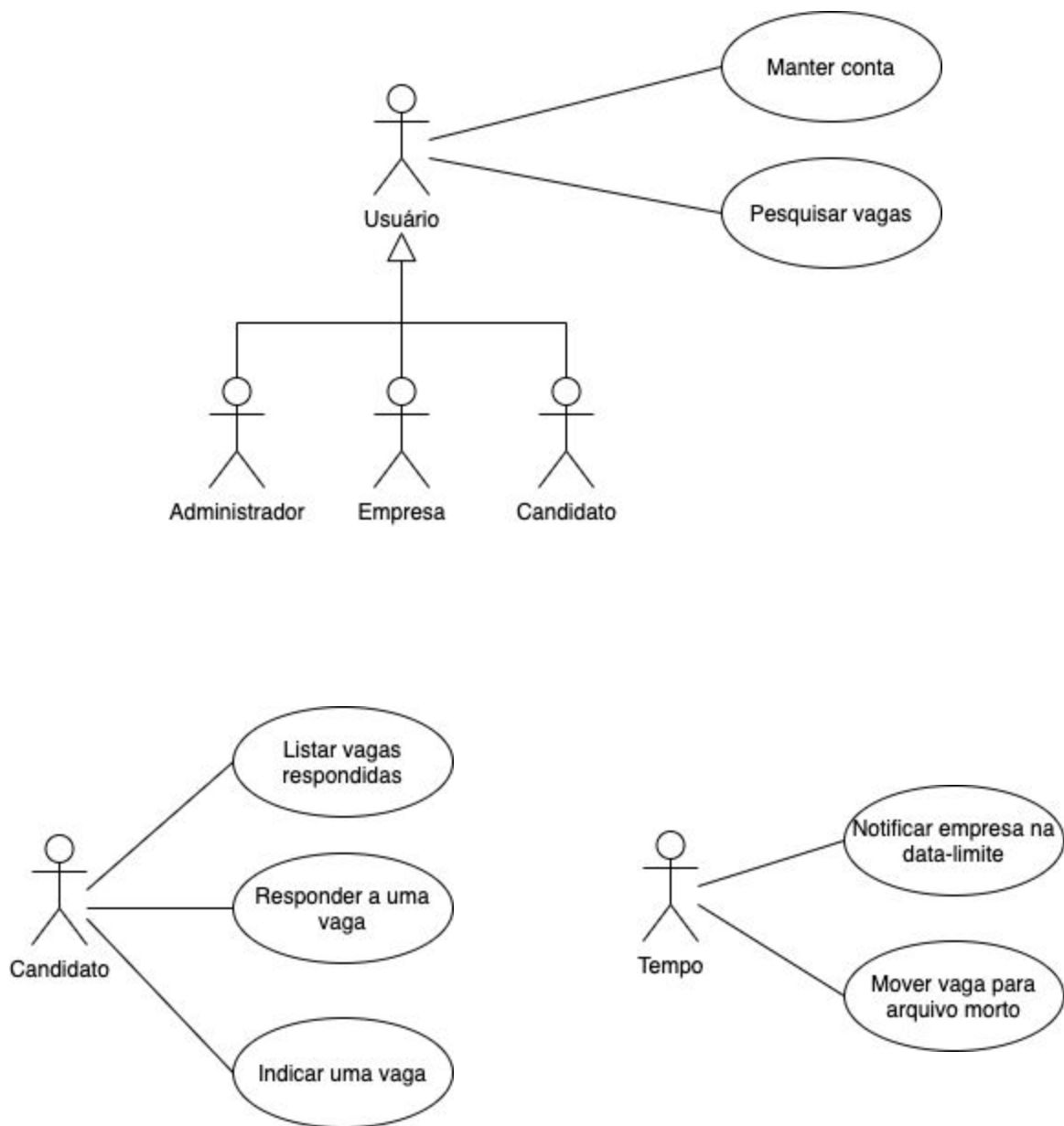
É o sistema onde serão cadastradas as vagas, responsável por controlar o cadastro e acesso de usuários (candidatos e empresas), fazer a manutenção automática das ofertas vencidas, e gerar relatórios básicos de acesso por usuário, empresa e número de ofertas.

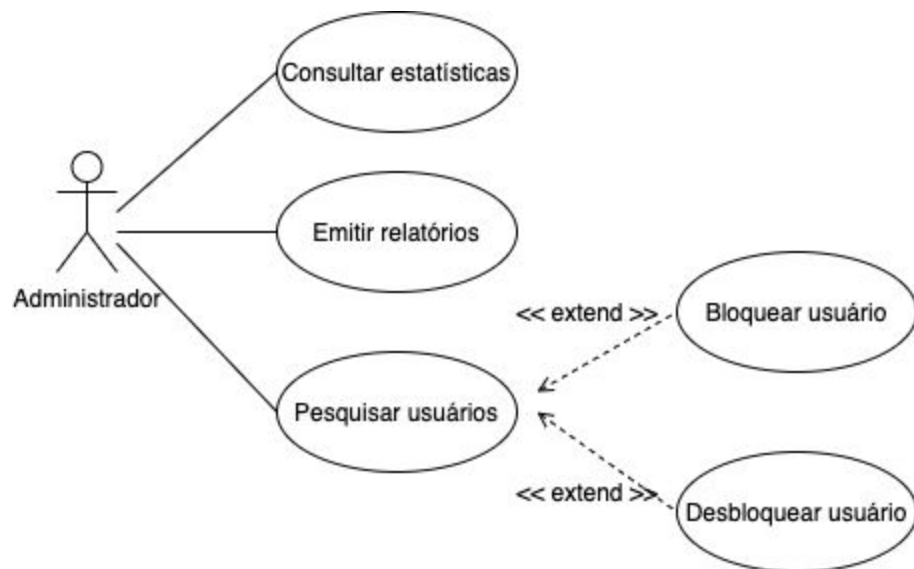
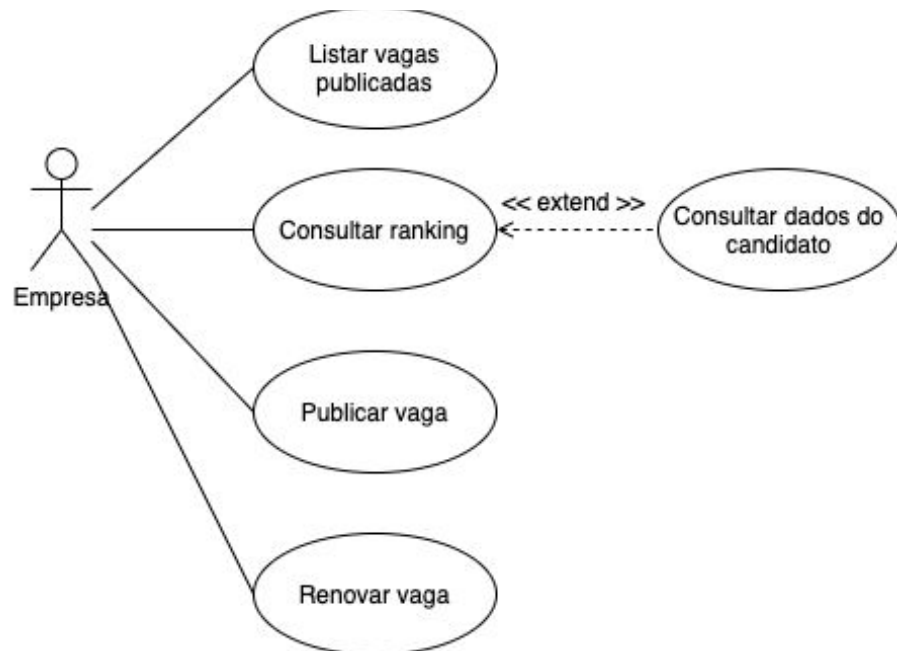
### 1.2.4 Administrador

É a pessoa que tem acesso aos dados de acesso por usuário, por empresa e o número de ofertas

## 1.3 Casos de Uso

### 1.3.1 Diagrama de caso de uso





### 1.3.2 Lista e classificação

Casos de uso em ordem dos mais interessantes para os menos interessantes classificados baseado na complexidade de implementação.

#### **Legenda de classificação:**

1 - Muito fácil, 2 - Fácil, 3 - Médio, 4 - Complexo, 5 - Muito complexo.

- **Publicar vagas** - Complexidade 2
- **Listar vagas publicadas** - Complexidade 2
- **Responder a uma vaga** - Complexidade 3
- **Listas vagas respondidas** - Complexidade 2
- **Pesquisar vagas** - Complexidade 2
- **Manter conta** - Complexidade 3
- **Consultar ranking** - Complexidade 2
- **Consultar dados do candidato** - Complexidade 3
- **Renovar vaga** - Complexidade 3
- **Indicar uma vaga** - Complexidade 1
- **Pesquisar usuários** - Complexidade 2
- **Bloquear usuário** - Complexidade 3
- **Desbloquear usuário** - Complexidade 3
- **Mover vaga para arquivo morto** - Complexidade 3
- **Notificar empresa na data limite** - Complexidade 3
- **Consultar estatísticas** - Complexidade 3
- **Emitir relatórios** - Complexidade 3

## 1.4 Requisitos

### 1.4.1 Matriz de requisitos

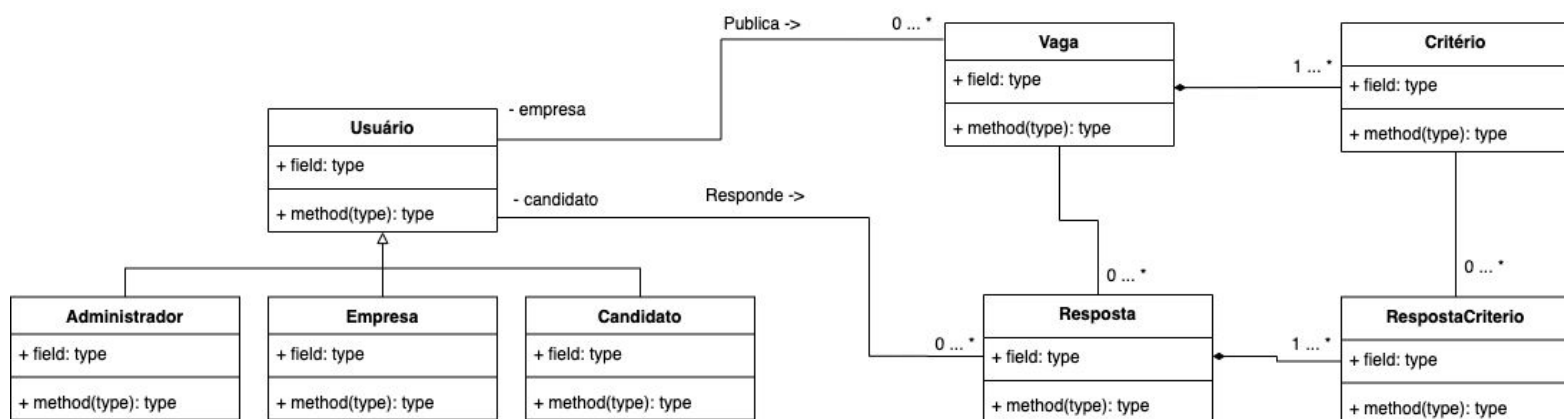
ID	Nome	Descrição	Prioridade	Complexidade
RF-01	Publicar vagas	Os usuários do tipo recrutador devem poder publicar vagas	Essencial	2
RF-02	Listar vaga publicadas	Os usuários do tipo recrutador devem poder listar suas vagas publicadas	Essencial	2
RF-03	Responder a uma vaga	Os usuários do tipo candidato devem poder responder a vagas	Importante	3
RF-04	Listas vagas respondidas	Os usuários do tipo candidato devem poder visualizar as vagas para as quais se candidatou	Essencial	2
RF-05	Pesquisar vagas	Qualquer usuário do sistema deve poder pesquisar vagas	Essencial	2
RF-06	Manter conta	Qualquer usuário do sistema deve poder se cadastrar e possuir uma conta	Desejável	3
RF-07	Consultar ranking	Os usuários do tipo recrutador devem poder consultar o ranking dos candidatos para cada uma das vagas	Essencial	2
RF-08	Consultar dados do candidato	Os usuários do tipo recrutador devem poder visualizar os dados dos candidatos do ranking	Essencial	3

RF-09	Renovar vaga	Os usuários do tipo recrutador devem poder renovar uma vaga para que ela fique mais tempo ativa no sistema	Importante	3
RF-10	Indicar uma vaga	Os usuários do tipo candidato devem poder indicar vagas publicadas para terceiros	Desejável	1
RF-11	Pesquisar usuários	Os usuários do tipo administrador devem poder pesquisar usuários do sistema	Desejável	2
RF-12	Bloquear usuários	Os usuários do tipo administrador devem poder bloquear um usuário do sistema	Desejável	3
RF-13	Desbloquear usuários	Os usuários do tipo administrador devem poder desbloquear um usuário bloqueado do sistema	Desejável	3
RF-14	Mover vaga para arquivo morto	A entidade tempo deve mover as vagas para arquivo morto caso expire	Importante	3
RF-15	Notificar empresa na data limite	A entidade tempo deve mover as vagas para arquivo morto caso expire	Importante	3
RF-16	Consultar estatísticas	Os usuários do tipo administrador devem poder consultar as estatísticas de uso dos usuários	Importante	3
RF-17	Emitir relatórios	Os usuários do tipo administrador devem poder emitir relatórios	Essencial	3

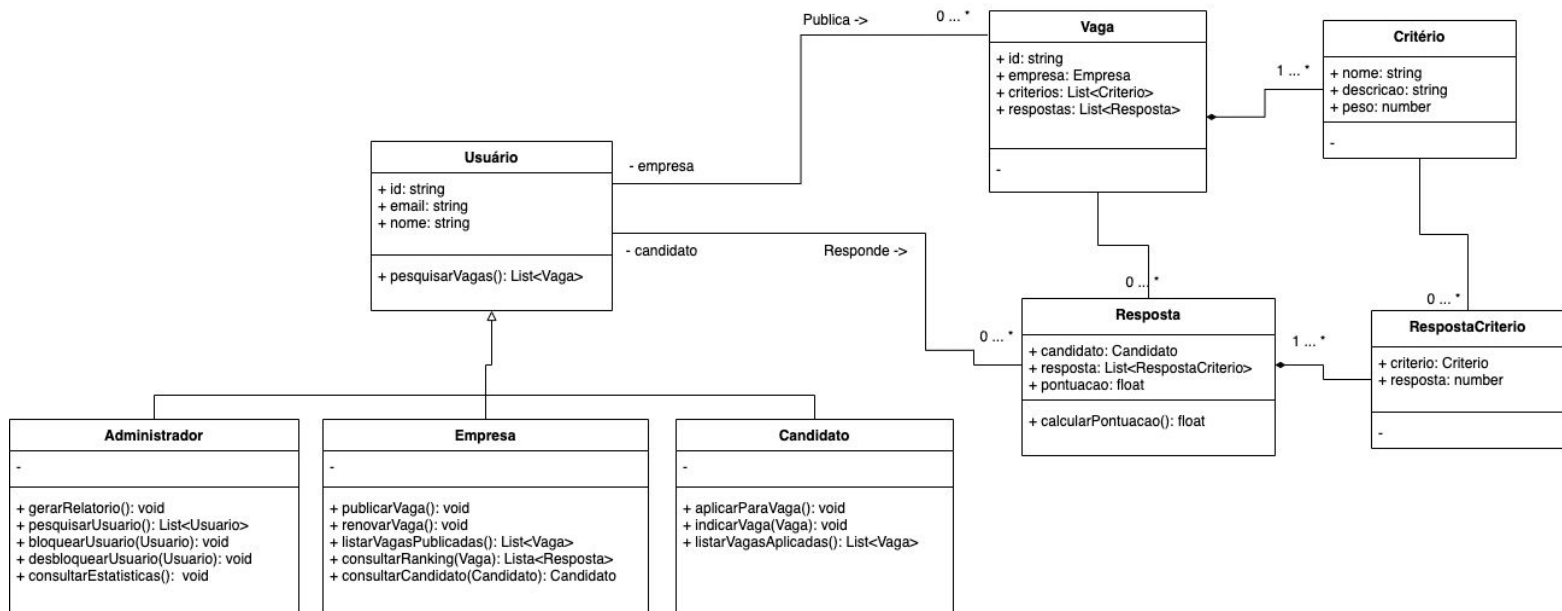


		para os recrutadores		
NF-01	Deve ser acessível via browser	O sistema deve ser uma solução web, disponível na rede para acesso de todos por meio de um navegador	Essencial	1
NF-02	Os dados devem ser trafegados de forma segura	O sistema deve realizar as requisições HTTP de forma segura, utilizando HTTPS	Desejável	2

## 1.5 Modelo de domínio



## 1.6 Diagrama de classes



### 1.6.1 Responsabilidade dos objetos

**Administrador:** É responsável pela gestão da aplicação e dos seus usuários, por gerar relatórios, fazer bloqueio e desbloqueio de usuários, consultas de estatísticas.

**Empresa:** Possui a responsabilidade de gerir as vagas: criação e renovação. Padrão *creator* do GRASP.

**Candidato:** Possui a responsabilidade de fazer as aplicações para as vagas e também podem indicar vagas a terceiros.

**Vaga:** É responsável por ter as informações da vaga, possui uma lista de critérios e uma lista de respostas. Está atrelada a um objeto do tipo Empresa.

**Critério:** Objeto independente, auxiliar à Vaga.

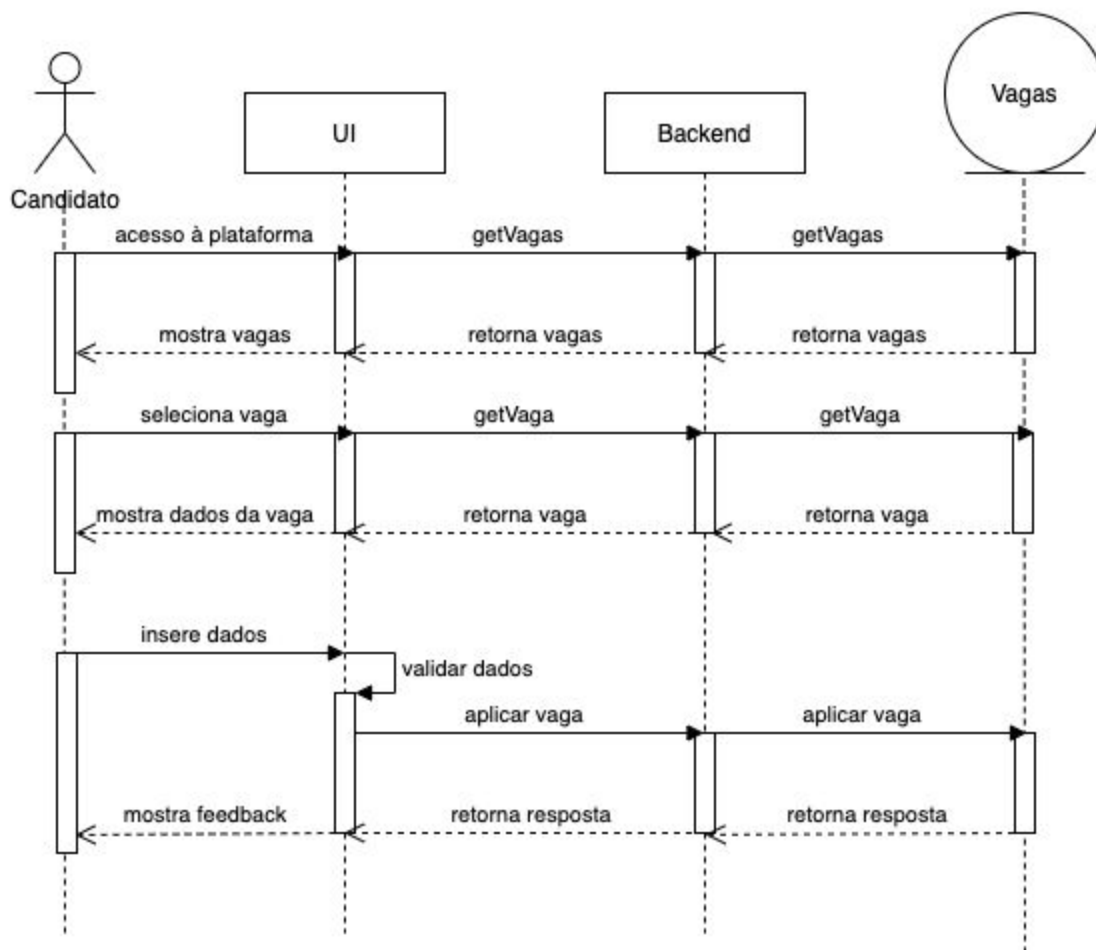
**Resposta:** É responsável por ter informações da resposta do usuário a uma determinada vaga e por fazer o cálculo da pontuação daquela resposta.

**RespostaCritério:** Objeto independente, auxiliar à Resposta.

## 1.6 Diagramas Sequenciais

### 1.6.1 Aplicar para vaga

Aplicar para vaga



## 2. Processo de Desenvolvimento de Software

### 2.1 TP1

1. Considerando o que aprendeu sobre DAD, qual seria o ciclo de vida de software mais adequado para o desenvolvimento desse projeto? Justifique a sua escolha.

O ciclo de vida **ágil**, porque é de simples entendimento e execução e as entregas são planejadas, baseado em três fases, que são a iniciação, construção e transição, que resultam em pequenas entregas de valor a cada ciclo.

2. Considerando que iremos utilizar o DAD no desenvolvimento desse projeto, quais elementos do Processo Unificado serão importados para esse projeto?

Modelo de casos de uso e modelo de diagrama de classes.

3. Quais seriam as diferentes etapas do desenvolvimento desse projeto?

Etapas de concepção, construção e transição.

4. Que elementos de desenvolvimento ágil devem ser utilizados nesse projeto?

Sprint, planning, retrospectiva e weekly.