

# **PRÁTICAS DE ENGENHARIA DE SOFTWARE**

**UNIRITTER LAUREATE INTERNATIONAL UNIVERSITIES**

AIM1907 Prática de engenharia de software FAPA-N1

Profº: Jean Paul Lopes

## **Integrantes:**

Eduardo Vaz(2020111574), Mauricio Henrique(2018111134), Pedro Goulart (202012393), Ricieri Pandolfo(201612919) e Roger Almeida Neto Teixeira(201718795)

Porto Alegre, 30 de Agosto de 2021

## **AminaPet:**

\_\_\_\_ Sabemos que medicamentos, comida de qualidade e serviços básicos para animais de estimação não são baratos e com a economia na situação atual fica ainda mais difícil arcar com esse tipo de custo, então vimos a necessidade de criação da Amina Pet, que tem como ideia principal reunir pessoas dispostas a ajudar pets prestando algum serviço ou fornecendo algum equipamento/comida de forma gratuita. Um exemplo de serviço que pode ser prestado seria um banho e tosa no animal, cuidados médicos ou até mesmo doação de remédios ou alimento.

- **Quem?** O sistema será utilizado por qualquer pessoa que necessite de algum item citado acima mas não tenha condição de arcar com os custos.

- **Como?** Os usuários irão se conectar por um aplicativo, onde pessoas dispostas a oferecer serviços ou medicamento/comida de forma gratuita irão informar onde e quando podem realizar a ação.

## **Relação de Stakeholders importantes:**

- Veterinários/Veterinárias
- Membros de ONGs que ajudam animais
- Doadores, voluntários
- Usuários com necessidade de doação

## **Levantamento de requisitos:**

Para realizarmos o levantamento com os usuários, a área de negócio terá que eleger o seu Stakeholder / PO / Keyuser, pois será o principal usuário que terá o contato com a área de desenvolvimento do sistema. O Eleito SH, será uma pessoa que tem bastante conhecimento pela área e entende muito do negócio, pois ela é uma peça fundamental entre o desenvolvedor x cliente.

A área de negócio vai se juntar aos usuários escolhidos e desenvolver as suas “User Stories”, ou seja, histórias de usuário, que basicamente eles vão detalhar quais as funcionalidades que o sistema deve atender. Para isso, podemos começar fazendo três perguntas para eles:

- Quem? **Usuário;**
- O quê? **Necessidade;**
- Por que? **Benefícios.**

## **Utilizando os exemplos abaixo:**

**Usuário:** Dono do pet;

**Funcionalidade ou Necessidade:** Buscar serviços;

**Objetivo ou Benefícios:** Conseguir categorizar esses serviços por medicamento, comida ou prestação de serviço (como banho, tosa).

### **Requisitos Funcionais:**

- O sistema deve permitir o cadastramento de um animal (gato, cachorro, etc);
- O sistema deve permitir o cadastramento de um voluntário;
  - Animal :
    - O sistema deverá conter uma área de login para dono do animal;
    - O sistema deverá permitir que o dono do animal possa ver a lista de serviços
    - O sistema deverá permitir que o dono do animal possa filtrar a lista por serviços, medicamentos e comida;
    - O sistema deverá conter uma agenda de serviços marcados pelo dono do animal;
    - O sistema deverá permitir que os itens da agenda sejam cancelados;
    - O sistema deverá notificar o dono do usuário sempre que houver alguma atualização em relação ao serviço marcado;
  - Voluntário:
    - O sistema deverá conter uma área de login para o voluntário;
    - O sistema deverá permitir o cadastro de serviço por parte do voluntário
    - O sistema deverá conter uma agenda de serviços marcados para o voluntário
    - O sistema deverá permitir que o voluntário cancele algum serviço marcado;
    - O voluntário deverá confirmar sempre que algum serviço que ele forneça for solicitado;
    - O sistema deverá notificar o voluntário sempre que alguém solicitar o serviço

### **Requisitos Não Funcionais:**

- O sistema deve ser responsivo (podendo se ajustar em dispositivos móveis ou monitores);
- O sistema deverá ter uma alta disponibilidade, funcionando 24/7
- A interação entre cliente e servidor não deverá ser maior que 5 segundos;

- O sistema deverá conter uma autenticação de 2 fatores no primeiro login;
- Todas as rotas do backend deverão ser documentados com swagger Open API 3.0;
- O front end deverá ser construído utilizando react;
- O backend deverá ser construído utilizando nodeJS;
- O backend deverá rodar em um cluster Kubernetes;

### **Classificação de Requisitos:**

#### **- Must Have:**

- Cadastro de Voluntários e Animais
- Doação de serviços e remédios
- Responsividade (podendo se ajustar em dispositivos móveis ou monitores)
- Alta disponibilidade, funcionando 24/7

#### **- Should Have:**

- Agenda para organização, marcação e cancelamento de serviços
- Notificações de marcação de serviço
- Filtro e visualização de todos os serviços

#### **- Could Have:**

### **Ideias pra entrevista:**

**(Doação de remédios é legal, não há lei que impeça)**

### **Questionario:**

- 1) Você possui animais de estimação ?
- 2) Você já teve gastos com veterinário/medicamento ?
- 3) O que você costuma fazer com medicamentos veterinários que sobram após o tratamento do animal ?
- 4) Você utiliza ou já utilizou algum site de doação de insumos veterinários ? Se sim qual ?
- 5) Você já prestou algum serviço comunitário relacionado a pet, como por exemplo banho e tosa ?
- 6) Você já usufruiu de algum serviço comunitário relacionado a pet ? Se sim qual ?

- 7) **Você realizaria o cadastro do seu pet em sistema que se propõe a fornecer serviços/medicamentos de forma gratuita ?**
- 8) **Você tem algum conhecimento sobre legislação em relação a doação de remédios para animais ? Se sim poderia descrever melhor quais são os possíveis fatores que impedem essa prática ?**
- 9) **Aceitam doações em dinheiro?**
- 10) **Como é feita as parcerias com empresas do ramo pet?**
- 11) **Além de medicamentos, quais outros produtos podem ser doados?**