



# UniverSitter

## Boas Prática de Engenharia de Software

**Semestre:** 2024.1

**Professora:** Antonia Diana Braga Nogueira

**Alunos:**

Luis Felipe Morais de Lima — 538605

Marcelo Mikael Pinheiro Lessa Peres — 536011

Renan Victor de Almeida Silva — 538428

# Descrição do Projeto

O Brasil ocupa o 3º lugar no ranking mundial de países com mais pets com um total de 149,6 milhões de animais de estimação, ficando atrás apenas da Argentina e do México. Os dados são do censo Pet do IDP (Instituto Pet Brasil). Uma nova pesquisa do Censo QuintoAndar de Moradia, realizado em parceria com o Datafolha, mostra que 69% dos brasileiros gostam de cultivar [plantas](#) e 57% de enfeitar a casa com [flores](#). Ou seja, sete em cada dez brasileiros curtem ter ou cuidar de um [jardim](#) próprio. Isso, claro, também se enquadra para o cenário universitário.

No entanto, uma preocupação constante para muitos é o bem-estar de seus queridos animais de estimação e plantas quando precisam se ausentar de casa por períodos prolongados. Seja por viagens de trabalho, férias ou compromissos inesperados, encontrar alguém confiável e dedicado para cuidar de seus amados companheiros pode ser um desafio.

Desta forma nasce o UniverSitter, uma plataforma que se propõe a conectar donos preocupados com cuidadores apaixonados, oferecendo uma alternativa segura e eficaz para o cuidado temporário de seus queridos companheiros.

## Posicionamento do Problema

O Problema de:	Não ter amparo para pets/plantas em períodos não letivos
Afeta:	Universitários locais
Seu impacto é:	Obstáculo para adoção de animais, universitários cada vez mais sozinhos.
Uma boa solução seria:	Uma plataforma que conecta possíveis cuidadores a estes universitários.

## Posicionamento do Produto

Para:	Universitários locais
Que:	Que constantemente passam pela incessante busca de um bom cuidador para seus dependentes em períodos não letivos.
O:	UniverSitter
é um:	Rede social web para conexão de pessoas dentro do nicho alvo
Que:	Conecta os donos aos cuidadores de modo efetivo, claro e rápido, oportunizando um âmbito de livre acordo entre eles.

Ao contrário de:	DogHero e/ou TrustedHouseSitter
Nossa solução:	Estamos focados em conectar pessoas e na troca de favores, enquanto as outras plataformas estão oferecendo serviços terceirizados.

## Boas Práticas Aplicadas ao Projeto

Durante o desenvolvimento do projeto, foram analisados e adotados padrões e convenções com foco nas boas práticas de Engenharia de Software e em um desenvolvimento otimizado.

A justificação de nossas escolhas são baseadas em familiaridades com as metodologias e ferramentas, de forma que facilitaria o trabalho dos membros, além de ser ferramentas modernas e atuais no mundo do trabalho.

Como nossa equipe contém poucos membros, não adotamos papéis específicos dentro da organização do projeto, ou seja, cada membro é livre para transitar nos vários papéis e se ajudar dentro do desenvolvimento do sistema.

## Metodologia de Desenvolvimento

- **Scrum**

Aplicamos o Scrum por ser uma metodologia de conhecimento dos membros da equipe e por ser uma das melhores metodologias ágeis atualmente, permitindo o desenvolvimento eficiente do projeto através de iterações.

Organizamos o desenvolvimento em Sprints com uma semana de duração, com 3 reuniões semanais para conversar sobre o progresso do projeto e possíveis dificuldades encontradas pela equipe.

No começo do desenvolvimento, foi realizada uma reunião para analisar o Product Backlog e assim definir o que deve ser priorizado no a cada iteração.

Foi notado uma leve dificuldade para realizar as reuniões semanais devido ao tempo e rotinas diferentes adotadas pelos membros da equipe, este impasse está sendo analisado e corrigido entre os membros do time, de forma que não afete, nem comprometa a equipe.

- **Kanban**

Após a definição dos requisitos funcionais, dispomos todas as histórias de usuário a serem desenvolvidas em um quadro *Kanban* na plataforma TRELLO. As funcionalidades a serem desenvolvidas foram catalogadas na coluna *Backlog*, organizadas em formato de Cards escritos em português.

Cada card se refere a uma funcionalidade que são divididas a partir das etiquetas e cada membro puxa a responsabilidade do card ao adotá-lo para seu nome.

Os cards tem um prazo a ser seguido e corre através das colunas para mostrar o progresso do desenvolvimento.

## Gerenciamento de Projeto

- **Trello**

Ferramenta utilizada para representar o quadro Kanban do projeto. Utilizamos sua versão Web e os membros têm livre acesso para criar cards e modificá-los como bem entender

- **[Github](#)**

O controle de versionamento e a hospedagem do código-fonte do projeto é realizada na plataforma Github.

Todos os padrões de desenvolvimento, desde padrões de commit a nomenclatura de branches, são aplicados no desenvolvimento e podem ser visualizados no Github do projeto.

## Padrões de Design e Prototipação

- **Protótipo**

Para a concepção do protótipo foi utilizado a ferramenta FIGMA e nosso objetivo é construir um sistema simples de ser utilizado e responsivo. O foco é em construir um sistema agradável para que os usuários sintam-se confortáveis ao utilizar nossa aplicação.

## Boas Práticas de Versionamento (Git)

- **Padrão de Commit**

Os Commits devem descrever o que foi desenvolvido a cada iteração. O objetivo é facilitar o entendimento e melhorar o versionamento por parte dos membros.

- **Branches**

A nomenclatura das branches sempre seguirá o seguinte modelo, com escrita em linguagem mista:

Para feature: "feature/nome-do-card"

Para bugs: "bugfix/nome-do-card"

- **PR**

Os **Pull Requests** tem um título seguindo o seguinte formato e em linguagem mista: "Fixes/Feature NOME CARD"

- **Git Controle**

### **Repositórios Centrais**

Em nosso repositório central (origin), mantemos duas branches fundamentais em todo o ciclo de vida de nossas aplicações:

**Main:** Será sempre nosso código-fonte que está em produção.

**Develop:** Paralelamente à branch main, temos a branch develop, onde são colocadas todas as novas features e fixes que estão sendo desenvolvidos no momento. Após finalizarmos a sprint e todos os testes, criamos a versão que será colocada na MAIN que é a principal.

### **Branches de Suporte**

Além das branches principais (main e develop), utilizamos branches de suporte para diferentes finalidades, como implementar novas funcionalidades, corrigir problemas em produção e criar novos releases. Essas branches têm um tempo de vida limitado e são removidas quando não são mais necessárias. As categorias de branches de suporte incluem:

**Features branches:** Criadas para implementação de novas funcionalidades.

**Hotfix branches:** Criadas para corrigir bugs existentes.

- **Revisão de PRs**

Após submetido, um **Pull Request** será revisado e corrigido, caso necessário, por um desenvolvedor que não foi o responsável por fazer essa submissão.

## **Documentação**

A criação da documentação ficou, em maior parte, para um dos membros que se sente mais confortável em realizar os artefatos, sendo revisada por outros membros da equipe. A cada sprint, às documentações são revisadas com todo o time, para reforçar o entendimento do que está sendo desenvolvido e conversar sobre possíveis alterações para com o projeto ou documentação.

Os documentos do projeto se encontram em uma pasta do Google Drive compartilhada entre os membros da equipe.