

# Plano de Projeto 1

## Histórico da Revisão

Data	Versão	Descrição	Autor
20/03/2024	1.0	Criação das seções de 1 a 4 do documento.	Gabriel Merlin
05/04/2024	1.1	Edição após separação das disciplinas	Artur de Vlieger
17/04/2024	1.2	Elaboração das seções 5, 6, 7 e 9	Pedro Augusto Monteiro
24/04/2024	1.3	Elaboração da seção 10	Gustavo Gabriel
05/06/2024	1.4	Elaboração da seção 8	Pedro Augusto Monteiro

### 1. Identificação

- PETamigos
- Equipe 9 (turma A)
- 20/03/2024

### 2. Introdução

Este arquivo visa documentar os objetivos do projeto, seu desenvolvimento e eventuais dificuldades encontradas. Ele se destina a todos os interessados em compreender todo o processo por trás dele, ao invés de apenas o resultado. O nosso propósito é desenvolver uma plataforma para adoção de animais domésticos, como cães e gatos, voltada para que diversas ONGs possam cadastrar seus animais e encontrar possíveis adotantes.

### 3. Escopo do projeto

- No escopo:
  - Desenvolvimento de uma plataforma para adoção de animais.
  - ONGs podem se cadastrar no site para ter acesso a suas ferramentas.
  - ONGs podem cadastrar seus animais para adoção.
  - Usuários podem buscar por animais usando filtros como idade e raça.
  - Usuários podem encontrar formas de contatar as ONGs e enviar dúvidas.
- Fora do escopo:
  - Chat dentro do site.
  - Responsabilidade legal sobre o processo de adoção.
  - Doação financeira para ONGs.
  - Doação de animais para a ONG.

### 4. Equipe e infraestrutura

- a. Equipe:
  - i. Product Owner: Gabriel da Costa Merlin
  - ii. Scrum Master: Pedro Augusto Monteiro Delgado e Artur De Vlieger Lima.
  - iii. Developers: Enzo Nunes Sedenho, Gustavo Gabriel Ribeiro.
- b. Equipamentos, ferramentas, software de apoio, materiais:
  - i. Trello: organização, acompanhamento e gestão dos sprints de projeto e de desenvolvimento;
  - ii. Figma: criação dos protótipos digitais de maneira colaborativa;
  - iii. Github: controle de versões do código fonte e documentação do projeto;
  - iv. Google Meet: reuniões com o cliente;
  - v. Discord: reuniões de desenvolvimento da equipe;
  - vi. Google Drive: organização dos artefatos produzidos;

## 5. Acompanhamento do projeto

O projeto será realizado durante os meses de abril, maio e junho, seguindo o cronograma a ser apresentado na próxima seção. O desenvolvimento baseia-se na divisão de sprints, segundo o Modelo Ágil. A cada sprint, realizar-se-á uma reunião para revisão das tarefas realizadas e para planejamento das tarefas a serem realizadas no futuro. Além disso, serão realizadas reuniões de acompanhamento do andamento do projeto durante os sprints utilizando o Discord (*daily*s). Ao longo do processo, o cliente será constantemente acionado para coleta de feedbacks e validação de requisitos, por meio de reuniões via Google Meet. Por fim, haverá o acompanhamento das tarefas por meio do quadro no Trello, reuniões no Discord e versionamento no Github.

## 6. Cronograma e Marcos do projeto

Marcos importantes e artefatos a serem produzidos	01/04 08/04	08/04 17/04	18/04 28/04	29/04 13/05	14/05 27/05	28/05 16/06	16/06 24/06
Plano de projeto	X						
Extração de requisitos	X						
Estórias do cliente	X						
Protótipo de baixa fidelidade		X					
Casos de uso		X					
Protótipo de alta fidelidade		X					

Cenários de teste			X				
Diagramas para modelagem da base de dados			X				
Início do desenvolvimento Web				X			
Desenvolvimento do produto					X	X	
Finalização do projeto (entrega e apresentação do produto)							X

## 7. Gerência de Riscos

### Riscos de mercado

1. *ONGs não aderirem:* Como nossa aplicação se baseia em um "catálogo" de animais para adoção disponíveis aos nossos usuários, a aplicação inteiramente dependente das ONGs para o cadastramentos de tais animais, ou seja, sem a adesão das ONGs não é possível oferecer ao usuário uma quantidade satisfatória e diversificada de animais.
2. *Usuários não se interessarem:* Outro fator importante é a adesão por parte dos usuários. Uma vez que ONGs disponibilizem seus pets em nosso site, é necessário um retorno positivo por parte dos usuários para que nosso serviço seja benéfico para as ONGs, somente assim elas continuarão como nossas parceiras.

### Riscos técnicos

1. *Os dados dos pets são preenchidos de forma errônea pelas ONGs:* Dado que a única forma dos usuários se interessarem pelos pets e verificarem se são compatíveis com seus modos de vida, espaços de casa, disponibilidade de tempo e financeira, são pelos dados preenchidos pelas ONGs para o dado pet. Assim, dados errados podem levar a conflitos e desagrados, o que é extremamente indesejado para nós.
2. *Pets adotados não são atualizados como tal pela ONG:* A ONG deve manter a situações de seus pets cadastrados sempre atualizada, visto que um pet adotado, ainda visível no site, pode despertar o interesse de um usuário e levar a uma confusão e decepção futura.

*Mitigação proposta para os riscos 1 e 2:* Adicionaremos a opção dos usuários realizarem denúncias nas páginas de ONGs ou PETs de modo a informar a ONG responsável que um certo dado fora preenchido erroneamente ou que deve ser atualizado. Dessa forma, nossos usuários poderão colaborar retroativamente para que as informações em nosso site sempre se mantenham atualizadas e verídicas.

### Riscos de projeto:

1. *Implementação superficial do sistema de matchs:* Devido ao curto tempo, nós como um time faremos o que estiver em nossa mão para desenvolver um sistema de match

verídico e satisfatório, no entanto, dificuldades na implementação e lógica podem surgir, que poderá atrasar ou até descartar a feature.

*Mitigação proposta:* Implementaremos um sistema de labels nos pets cadastrados (dados providos pela ONG) que, em conjunto com as respostas fornecidas pelos usuários na fase de login, serão utilizados para nossa lógica de match. Dessa forma, essa feature se torna mais simples e facilmente implementável, sem prejudicar sua utilidade.

## 8. Testes do produto (ou sistema)

O projeto de teste do sistema será dividido em 3 etapas, sendo cada uma delas descrita abaixo:

- Testes das interfaces do sistema (*Front End*): verificação da responsividade do sistema e também validação dos requisitos não funcionais.
  - Teste da responsividade para telas grandes (PC) e pequenas (Mobile).
- Testes das funcionalidades do banco de dados (*Back End*): verificação da criação, remoção, edição e exclusão de dados de PETs, ONGs e Usuários.
  - Inserção de dados completos, incompletos e incorretos.
- Testes da integração do sistema: verificação do funcionamento geral do sistema, garantindo a qualidade do software.
  - Simulação do uso das funcionalidades por meio de casos de teste.

## 9. Gerenciamento de configuração de software

A padronização da nomenclatura associada aos itens de configuração a serem mantidos neste projeto, via Github, seguem as especificações tabela abaixo:

Item de configuração	Nomenclatura	Dependências	Numeração padronizada
Código fonte	CF_X.X.X	Casos de uso, Modelo de BD	1ºX: Versionamento 2ºX: Adição de funcionalidade 3ºX: Revisão e/ou alterações
Casos de uso	CDU_X.X	Código fonte, Modelo de BD	1ºX: Versionamento 2ºX: Alteração
Requisitos	R_X.X	Código fonte, Casos de uso, Modelo de BD	1ºX: Versionamento 2ºX: Alterações e/ou revisões
Modelo de BD	BD_X.X	Código fonte	1ºX: Versionamento 2ºX: Alteração

Antes da entrega das versões do produto ao cliente, a tabela deve ser consultada de modo a todos os itens (código fonte, documento de casos de uso, documento de requisitos e modelo de banco de dados) seguirem a nomenclatura adequada. É de extrema importância que o versionamento siga o padrão estabelecido para melhor compreensão da evolução do projeto.

## 10. Plano de Manutenção de software

O cliente, analisando o software apresentado, vê que são necessárias mudanças e, portanto, faz um pedido (change request - CR). Tendo esse requerimento em mãos, o PO faz a análise do CR, verificando os custos da mudança e sua possibilidade técnica de implementação. Tudo isso engloba a fase de suporte ao cliente.

Após isso, caso o CR não seja aprovado, ele é arquivado e o processo é reiniciado. Já se for aprovado começa a fase de desenvolvimento, na qual ele é implementado e testado iterativamente até passar em todos os casos criados. Quando isso terminar, fazemos a incorporação das mudanças feitas, fechamos o CR e reiniciamos o processo.

