

**Documentação de um**

**Produto de Software**

**Dog Walker**

**Eric Batista de Santana**

**Gabriel Ramalho Berti**

**Gabriel Maireno de Lima**

**Manoel Victor Castor Simões**

**Rafael Francisco Fortes Lacerda**

**2020**

**ÍNDICE DETALHADO**

**1.** **Introdução 3**

1.1. Tema 3

1.2. Objetivos a serem alcançados 3

1.3. Escopo principal 3

**2.** **Definição do Modelo de Processo 4**

**3.** **Requisitos do Sistema de Software 4**

3.1. Requisitos Funcionais 4

3.2. Requisitos Não-Funcionais 4

**4.** **Projeto 5**

4.1. Arquitetura Lógica 5

4.2. Arquitetura Física 5

**5.** **Protótipo de Interface 6**

**6.** **Critérios de Qualidade de Software 7**

**7. Testes 8**

7.1. Plano de Testes 8

7.2. Roteiro de Testes 8

**Anexo I 9**

1. **Introdução**
   1. **Tema**

O tema do projeto é a criação de um aplicativo que visa facilitar os passeios com cachorros.

* 1. **Objetivos a serem alcançados**

O principal objetivo a ser alcançado com o desenvolvimento deste software, é facilitar a vida de pessoas que, por conta de fatores exteriores, acabam não tendo um tempo livre para passear com seu cachorro. O software busca atingir todos os públicos, desde recém-adultos a pessoas mais velhas. Nós buscamos a praticidade no uso, por esse motivo, a plataforma de desenvolvimento será voltada para o mobile, tendo em vista que a maioria da população possui um dispositivo celular e está sempre presente. O aplicativo contará com a API do Google Maps, visando à facilitação do uso, tendo em vista que é uma API de fácil entendimento e por sua vez, acaba garantindo uma alta acessibilidade.

* 1. **Escopo principal**

O projeto visa garantir um aplicativo funcional e de fácil manuseio para quem busca uma ajuda quando o assunto é passear com seu cachorro. O aplicativo contém uma interface simples e de fácil entendimento, tendo em vista que um dos principais objetivos é garantir a acessibilidade para todos os públicos.

1. **Definição do Modelo de Processo**

Neste item deve-se descrever o modelo de processo a ser utilizado para realização do projeto, como por exemplo: cascata, incremental, SCRUM, entre outros.

Referência: UC Modelos, métodos e técnicas da engenharia de software

1. **Requisitos do Sistema de Software**
   1. **Requisitos Funcionais**

|  |  |  |
| --- | --- | --- |
| Identificador | Descrição | Prioridade |
| RF01 | O sistema deve prover uma interface para cadastrar cachorros | Alta |
| RF02 | O sistema deve possuir uma tela de Perfil do usuário | Alta |
| RF03 | O sistema deve permitir alterações nas informações pessoais | Alta |
| RF04 | O sistema deve possuir um mapa que mostrará em tempo real os Andadores na região | Alta |
| RF05 | O sistema deve possuir uma tela com o histórico de passeios dos cachorros | Média |
| RF06 | O sistema deve definir uma rota de ida e volta como trajeto | Alta |
| RF07 | O sistema deve possuir um wallet virtual | Alta |
| RF08 | O sistema deve possuir avaliação de andador | Alta |
| RF09 | O sistema deve ter um chat para que o cliente se comunique com o andador | Alta |
| RF10 | O sistema deve incluir uma opção de denunciar o andador | Alta |
| RF11 | O sistema deve incluir opções de pagamento com cartão de débito e crédito | Alta |
| RF12 | O sistema deve calcular o valor do passeio de acordo com o trajeto | Alta |

* 1. **Requisitos Não-Funcionais**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador | Descrição | Categoria | Escopo | Prioridade |
| RNF01 | O sistema deve atender a lei de proteção de uso de dados | Segurança de dados | Sistema | Alta |
| RNF02 | O sistema só pode ser acessado com um usuário previamente cadastrado | Segurança de acesso | Sistema | Alta |
| RNF03 | O sistema deve ter uma fácil usabilidade | Facilidade de Operação | Funcionalidade | Alta |
| RNF04 | O sistema deve possuir boa segurança | Segurança de dados | Sistema | Alta |
| RNF05 | O sistema deve ser programado na Java | Desenvolvimento | Sistema | Média |
| RNF06 | O sistema deve possuir Acessibilidade para pessoas deficientes | Acessibilidade | Funcionalidade | Alta |
| RNF07 | O sistema deve utilizar a API do google maps para a geolocalização | Desenvolvimento | Funcionalidade | Alta |

* 1. **Regras de Negócio**

|  |  |  |
| --- | --- | --- |
| Identificador | Descrição | Prioridade |
| RN01 | O limite de cachorros por passeio deverá ser menor ou igual a 5. | Alta |
| RN02 | O sistema deverá criar um trajeto de ida e volta com base nos endereços preenchidos nos campos “Casa” e “Destino” selecionados pelo usuário. | Alta |
| RN03 | O valor de cada passeio será calculado com base no porte do(s) cachorro(s) e na distância total do trajeto selecionado pelo usuário. | Alta |
| RN04 | Os cartões cadastrados deverão ser verificados pelo sistema toda vez que forem selecionados como forma de pagamento. | Alta |
| RN05 | O pagamento será realizado após o término do passeio. | Alta |
| RN06 | O pagamento será realizado caso o cliente solicite o cancelamento do passeio, uma vez que o andador já iniciou o processo. | Alta |
| RN07 | Caso o andador esteja a caminho do ponto de partida para buscar o(s) cachorro(s), e o cliente solicite o cancelamento do passeio, uma taxa será cobrada do cliente. | Alta |
| RN08 | Caso o cliente queira denunciar o andador, ele será direcionado a um site no qual preencherá um formulário que será enviado a equipe de suporte do aplicativo, sendo assim possível tomar as medidas necessárias para solucionar o problema. | Alta |

1. **Projeto**
   1. **Arquitetura Lógica**

Neste item deve ser apresentada a arquitetura lógica de implementação, descrever arquitetura em camadas, padrão de projeto, linguagem de programação, banco de dados, componentes externos, ....

Referência: UC Modelos, métodos e técnicas da engenharia de software

* 1. **Arquitetura Física**

Neste item deve ser apresentada a arquitetura de infraestrutura do sistema, demonstrando o tipo de arquitetura física, a configuração de hardware, de rede... Para a representação da arquitetura de infraestrutura pode-se utilizar o diagrama de implantação.

Referência: UC Modelos, métodos e técnicas da engenharia de software

1. **Protótipo de Interface**

Neste item deve ser apresentado o protótipo do projeto. O protótipo é um recurso que deve ser adotado como estratégia para levantamento, detalhamento, validação de requisitos e modelagem de interface com o usuário (usabilidade).

Referência: UC Modelos, métodos e técnicas da engenharia de software

1. **Critérios de Qualidade de Software**

Neste item devem ser listados e descritos os critérios de garantia da qualidade do processo que serão considerados no Projeto.

Referência: UC Gestão e Qualidade de software

**7. Testes**

**7.1. Plano de Testes**

Neste item deve ser criado o plano de testes do sistema, permitindo a validação do sistema por parte do desenvolvedor, através da verificação dos requisitos do sistema desenvolvido.

Referência: UC Gestão e Qualidade de software

**7.2. Roteiro de Testes**

Neste item devem ser registrados os testes realizados no sistema tendo como base o Plano de Testes do Sistema. O roteiro de testes deve ser elaborado com base nos casos de uso ou user stories (cartão de histórias) de forma manual ou automatizada.

Referência: UC Gestão e Qualidade de software

**Anexo I**

Neste item deve ser anexado o roteiro de entrevista ou questionário respondido, caso tenha sido aplicado.

Referência: UC Modelos, métodos e técnicas da engenharia de software