



Universidade Federal do Ceará

Projeto Integrado em Engenharia de Software I

Ferramentas, tecnologias e processos

Versão 2.0

Projeto: Patas Felizes

Graziella Rodrigues de Lima - 539004

Michael Farias de Lima - 537835

Pedro Emanuel Santana - 537386

Wania Kelly dos Santos Oliveira - 540491

Sumário

1. Introdução.....	3
2. Linguagem de Programação: Java.....	3
3. Banco de Dados: PostgreSQL.....	3
4. Modelagem de Interface: Scenebuilder e JavaFX.....	4
5. Controle de Versão e Armazenamento do Código: Git e GitHub.....	5
6. Ambiente de Desenvolvimento: NetBeans.....	5
7. Prototipação: InVision e Framer.....	6
8. Gerenciamento de Equipe: Scrum.....	6
9. Cronograma.....	6
1.1. Cronograma do Projeto.....	6

1. Introdução

Este documento tem como objetivo fornecer uma visão abrangente das ferramentas e tecnologias selecionadas para o desenvolvimento do projeto "Patas Felizes". A escolha cuidadosa dessas ferramentas é fundamental para o sucesso do projeto, garantindo eficiência, escalabilidade e qualidade na criação do sistema de gerenciamento de animais.

2. Linguagem de Programação: Java

A escolha da linguagem de programação Java para o desenvolvimento do projeto é uma decisão que traz diversos benefícios e vantagens. Abaixo, listamos alguns motivos pelos quais Java é uma escolha adequada:

- I. **Multiplataforma:** Java é conhecida por sua capacidade de ser executada em diversas plataformas sem a necessidade de grandes modificações no código-fonte. Isso significa que iremos desenvolver o programa em Java e executá-lo em diferentes sistemas operacionais, como Windows, macOS e Linux, o que amplia consideravelmente o alcance do nosso software.
- II. **Orientação a Objetos:** Java é uma linguagem orientada a objetos, o que se alinha bem com a modelagem que usaremos no componente curricular "Projeto Detalhado de Software". Esse modelo permite criar representações claras e intuitivas dos objetos envolvidos no sistema, como animais, adotantes e voluntários.
- III. **Ampla Comunidade e Recursos:** Java é uma das linguagens de programação mais populares e amplamente adotadas no mundo, o que significa que existe uma vasta comunidade de desenvolvedores e uma abundância de recursos disponíveis. Recursos maduros e já bem estabelecidos no mercado, com boa segurança e fáceis de usar.
- IV. **Comunidade de Desenvolvedores Ativa:** A comunidade de desenvolvedores Java é ativa e pronta para fornecer suporte e orientação. Fóruns, grupos de discussão e comunidades online estão disponíveis para solucionar dúvidas e problemas que possam surgir durante o desenvolvimento.

3. Banco de Dados: PostgreSQL

O PostgreSQL foi selecionado como o sistema de gerenciamento de banco de dados para armazenar e organizar os dados essenciais do projeto. O PostgreSQL oferece um ambiente robusto para o armazenamento e recuperação de

informações relacionadas aos animais, adotantes, voluntários e outras informações presentes no sistema. Abaixo, alguns motivos dessa escolha:

- I. **Integração com Java:** O PostgreSQL é compatível com uma variedade de linguagens de programação e tecnologias, incluindo Java, o que simplifica a integração com o nosso sistema.
- II. **Maturidade e Histórico:** O PostgreSQL é um sistema de gerenciamento de banco de dados maduro, com um histórico sólido de desenvolvimento e aprimoramento ao longo dos anos. Isso significa que é uma escolha segura para o projeto
- III. **Código Aberto e Comunidade Ativa:** O PostgreSQL é uma tecnologia de código aberto, o que significa que é acessível e não requer custos de licença. Além disso, possui uma comunidade global ativa de desenvolvedores, o que resulta em suporte técnico disponível para eventuais dúvidas.
- IV. **Componente Curricular:** O PostgreSQL será usado na disciplina “Banco de Dados”, facilitando o uso da tecnologia e possibilitando um acompanhamento com a professora dessa cadeira.

4. Modelagem de Interface: Scenebuilder e JavaFX

O JavaFX é uma plataforma de desenvolvimento de software e uma biblioteca de interface gráfica de usuário (GUI) que faz parte do ecossistema Java. Ela é utilizada para criar aplicações desktop interativas e visualmente atraentes, com botões, caixas de texto, gráficos e vários outros componentes.

- I. **Integração e Modernidade:** JavaFX é uma plataforma de desenvolvimento de interface gráfica de usuário (GUI) moderna e rica em recursos, é a plataforma GUI oficial para Java desde a versão 8, o que significa que oferece uma integração perfeita com a linguagem de programação Java.
- II. **Estilo e Personalização:** JavaFX permite estilizar e personalizar a aparência da interface de usuário por meio do uso de CSS (Cascading Style Sheets). Isso significa que você pode criar designs visualmente atraentes e manter uma identidade visual consistente em todo o sistema.

O Scenebuilder é uma ferramenta de design visual que simplifica a criação de interfaces gráficas em JavaFX. Com essa ferramenta, você pode criar interfaces de usuário arrastando e soltando componentes na tela, o que facilita a prototipagem e o design da interface. O escolhemos por esses motivos:

- I. **Facilidade:** Sua curva de aprendizagem é bastante baixa, já que possui uma UI intuitiva, ideal para novos programadores.

- II. **Agilidade:** Sua abordagem *drag-and-drop* permite o desenvolvimento ágil do sistema.

5. Controle de Versão e Armazenamento do Código: Git e GitHub

O Git é um sistema de controle de versão. Ele é usado para rastrear e gerenciar as alterações feitas no código-fonte de um projeto de software. O escolhemos pelos seguintes motivos:

- I. **Maturidade:** É um dos sistemas de controle mais utilizados no mundo, maduro no mercado e com bastante conteúdo didático disponível.
- II. **Componente Curricular:** É o sistema que estamos usando no componente curricular “Gerência de Configuração”, então teremos suporte do professor dessa cadeia.

Já o GitHub é uma plataforma de hospedagem de código na nuvem que permite o compartilhamento e o manuseio do código por uma equipe. Abaixo, listamos as razões que motivaram essa escolha:

- I. **Integração:** O GitHub já possui uma integração padrão com o sistema de controle de versão Git.
- II. **Plano Universitário:** Essa plataforma oferece um plano universitário gratuito com diversos benefícios úteis para o projeto, como armazenamento estendido e controle de grupos.
- III. **Componente Curricular:** Estamos fazendo uso dessa ferramenta na disciplina de “Gerência de Configuração”.

6. Ambiente de Desenvolvimento: NetBeans

O NetBeans é uma IDE (Ambiente Integrado de Desenvolvimento) de código aberto que é usada para desenvolver, testar e depurar software. Escolhemos essa IDE pelos seguintes motivos:

- I. **Multiplataforma:** Está disponível para Windows, macOS e Linux. Isso é útil para nossa equipe, já que nem todos os membros fazem uso do mesmo sistema.
- II. **Integração:** O NetBeans oferece integração nativa com sistemas de controle de versão como Git, além da criação de GUIs com JavaFX e Scenebuilder.

7. Prototipação: InVision e Framer

O InVision é uma plataforma de prototipagem e colaboração de design que permite aos designers criar protótipos interativos. Usaremos ele para criar o protótipo de baixa fidelidade, já que é mais fácil de usar do que as ferramentas mais completas. Aqui estão alguns pontos positivos:

- I. **Colaboração:** Permite a edição do protótipo em conjunto com outras pessoas e em tempo real.
- II. **Simplicidade:** Suas funcionalidades são simples, ideal para protótipos de baixa fidelidade, que não precisam de um design detalhado.

O Framer, assim como o InVision, é uma plataforma de prototipagem em nuvem. Usaremos ele para criar o protótipo de alta fidelidade. Os motivos são:

- I. **Completo:** Possui mais funcionalidades do que o InVision, permitindo um protótipo mais detalhado e fiel.
- II. **Gratuito:** Não possui custos para projetos pequenos.

8. Gerenciamento de Equipe: Scrum

O Scrum é uma metodologia ágil amplamente utilizada para o gerenciamento de equipes e projetos de desenvolvimento de software. O Scrum se concentra na colaboração, adaptabilidade e entrega contínua ao cliente. Utilizaremos essa abordagem para colocar o cliente no centro do projeto, garantindo que as necessidades e prioridades dele sejam atendidas. Além disso, temos familiaridade com esse processo. Faremos uso de seus papéis, eventos e artefatos.

- I. **Papéis:** Scrum Master, Product Owner e Equipe de Desenvolvimento.
- II. **Eventos:** Sprint, Planejamento, Revisão e Retrospectiva da Sprint.
- III. **Artefatos:** Backlog do produto e da Sprint, Incremento.

9. Cronograma

1.1. Cronograma do Projeto

Como será usado o scrum como metodologia de desenvolvimento, teremos o desenvolvimento dividido em *sprints*, com cada *sprint* tendo uma duração de duas semanas. O projeto terá 6 *sprints* no total, dispostas em um período de 3 meses.

Segue um resumo de cada *sprint*:

Sprint 1 (Semana 1-2): Sprint Concluída

- RF001 (Prototipação)
- RF002 (Prototipação)

Sprint 2 (Semana 3-4):

- RF003 (Listagem de animais) (**Em desenvolvimento**)
- RF004 (Cadastro de animais)

Sprint 3 (Semana 5-6):

- RF005 (Gerenciamento de Adoção)
- RF006 (Registrar doações e finanças)

Sprint 4 (Semana 7-8):

- RF007 (Registrar despesas)
- RF008 (Documentos e comprovantes)
- RF009 (Registrar consultas)

Sprint 5 (Semana 9-10):

- RF010 (Agendamento de visitas)
- RF011 (Gerir voluntários)
- RF012 (Cronograma de afazeres)

Sprint 6 (Semana 11-12):

- RF013 (Notificações via e-mail)
- RF014 (Relatório e análises)