



Processo, tecnologias e ferramentas

1. Processo:

1.1 Scrum

Foi decidido utilizar o Scrum, devido ao fato deste framework ser orientado fortemente ao gerenciamento, além de todo o feedback positivo de todos os membros do time para com este framework. O scrum será levemente adaptado para este projeto. Manteremos um product backlog, onde em cada sprint planning, serão selecionadas quais histórias de usuário irão compor o backlog da sprint atual. Cada sprint terá duração de duas semanas, onde foi decidido não incorporar o daily scrum, como parte do processo. Ao final dessas duas semanas, teremos um sprint review, onde um de nossos membros, atuará até o final do projeto, como Product Owner devido ao alto conhecimento do domínio ao que destina-se esta aplicação. O incremento será validado por meio de testes de aceitação. Serão utilizados story points, para auxiliar o time de desenvolvedores a estimar um sprint backlog mais acurado. Por fim, ao final de cada ciclo, será feita uma retrospectiva da sprint, afim de identificar pontos a serem melhorados e/ou adicionar novas abordagens, visando uma melhor qualidade do processo de software.

2. Tecnologias:

2.1 Linguagem de programação: Java - Oracle OpenJDK 20.0.2.

Para o desenvolvimento do nosso projeto, escolhemos a linguagem de programação Java, mais especificamente a versão Oracle OpenJDK 20.0.2. Essa escolha se deve ao fato de que o Java, compilado e executado na JVM, permite que o mesmo código seja executado em diferentes sistemas operacionais. Além disso, o Java é conhecido por seguir o paradigma de orientação a objetos, permitindo a aplicação dos princípios do S.O.L.I.D, uma melhor modularização e reutilização de código. O ecossistema Java é rico, com uma comunidade bastante ativa que mantém bibliotecas e frameworks gratuitos, tornando-o uma escolha sólida para o nosso projeto.

2.2 Framework para criação de interfaces desktop: JavaFX.

Para a criação de interfaces desktop modernas e estilizadas, optamos pelo JavaFX. Este framework permite a utilização de código CSS para estilização, integra-se ao Scene Builder para a edição de interfaces gráficas e oferece suporte embutido para animações e efeitos. Além disso, possui uma documentação robusta, o que facilita o desenvolvimento de interfaces de alta qualidade.

2.3 Persistência de dados: PostgreSQL - 16.RC1.

Para a persistência de dados, escolhemos o PostgreSQL devido à sua total compatibilidade com a linguagem SQL. Isso o torna compatível com muitas aplicações e ferramentas que dependem do SQL para consultas e manipulação de dados. Além disso, o PostgreSQL permite a criação de funções, operadores e tipos de dados personalizados. Sua comunidade ativa de desenvolvedores e usuários fornece suporte e atualizações regulares, além de uma documentação robusta.

3. Ferramentas

3.1 Software de versionamento local: Git.

Para o controle de versão local, utilizamos o Git. Esta ferramenta mantém um histórico completo de todas as alterações feitas no código, permitindo a fácil reversão de mudanças e resolução de conflitos. Além disso, o Git oferece recursos para a criação de branches, o que nos permite trabalhar em novos recursos ou correções de bugs

sem afetar a versão principal do código. Também possui ferramentas de mesclagem para incorporar alterações de um ramo para outro.

3.2 Plataforma de hospedagem do código-fonte: GitHub.

Para hospedar nosso código-fonte, escolhemos o GitHub, construído em cima do Git. Essa plataforma permite o rastreamento de todas as alterações feitas no sistema, incluindo ferramentas de rastreamento de problemas (issue tracking). Além disso, integra-se com uma variedade de ferramentas populares de desenvolvimento, como o Visual Studio Code, e permite o armazenamento de documentação. Facilita a colaboração distribuída em projetos e a criação e mesclagem/reestruturação de branches.

3.3 Gerenciamento das sprints: Notion.

Para o gerenciamento de nossas sprints, utilizamos o Notion. Essa ferramenta versátil permite a criação de documentos, listas de tarefas e quadros Kanban para o acompanhamento do processo. Além disso, pode ser integrada com outras ferramentas, como o GitHub, tornando nosso fluxo de trabalho mais eficiente e organizado.

3.4 Ferramenta de prototipação: Figma.

Para o design de interfaces de usuário, optamos pelo Figma. Esta ferramenta baseada na web permite que nossas equipes trabalhem juntas em projetos de design em tempo real. Ele oferece a criação de protótipos interativos de alta fidelidade, design responsivo e versionamento de design. Além disso, facilita a colaboração ao permitir comentários diretamente no design, simplificando o processo de revisão.

3.5 Comunicação entre membros da equipe, gestor de processo e indexação de artefatos: Discord.

Para a comunicação em tempo real entre os membros da equipe, gestores de processo e indexação de artefatos, utilizamos o Discord. Esta plataforma oferece comunicação via chat em tempo real, permitindo a criação de canais específicos para diferentes aspectos do desenvolvimento de software. Além disso, oferece recursos de chamadas de voz e vídeo de alta qualidade, compartilhamento de arquivos e a capacidade de criar bots para automação de tarefas comuns, como notificações.

3.6 Meio alternativo de comunicação: Telegram

Oferece a capacidade de criar grupos e canais; suporta bots personalizados que podem automatizar tarefas além de permitir o compartilhamento de arquivos, documentos e mídia.

3.7 IDEs: IntelliJ e Visual Studio Code + Java package + Code runner.

Para o desenvolvimento de código, utilizamos duas IDEs complementares:

- **IntelliJ:** Essa IDE é projetada especificamente para Java e oferece uma integração forte com a linguagem. Ela fornece sugestões de código inteligentes, refatoração de código e verificação de erros em tempo real, tornando-a ideal para o desenvolvimento Java. Além disso, oferece suporte integrado ao framework JavaFX.
- **Visual Studio Code:** Essa ferramenta é multiplataforma, disponível para Windows, macOS e Linux, e oferece um ambiente de desenvolvimento leve e rápido. Ela possui um terminal integrado que permite a execução de comandos diretamente do ambiente de desenvolvimento e integração nativa com o Git. O Visual Studio Code suporta uma ampla gama de extensões que podem ser instaladas para personalizar a experiência de desenvolvimento, incluindo extensões para desenvolvimento em Java e para melhorar o uso do Git.