

Prof. esp. Thalles Canela

- **Graduado:** Sistemas de Informação - Wyden Facimp
- **Pós-graduado:** Segurança em redes de computadores - Wyden Facimp
- **Professor:** Todo núcleo de T.I. (Graduação e Pós) - Wyden Facimp
- **Diretor:** SCS
- **Gerente de Projetos:** Motoca Systems

Redes sociais:

- **Linkedin:** <https://www.linkedin.com/in/thalles-canela/>
- **YouTube:** <https://www.youtube.com/aXR6CyberSecurity>
- **Facebook:** <https://www.facebook.com/axr6PenTest>
- **Instagram:** https://www.instagram.com/thalles_canela
- **Github:** <https://github.com/ThallesCanela>
- **Github:** <https://github.com/aXR6>
- **Twitter:** <https://twitter.com/Axr6S>



LTD

LABORATÓRIO DE
TRANSFORMAÇÃO DIGITAL

Introdução aos Princípios de Engenharia de Software e Gerenciamento de Projetos:

- Breve explicação sobre metodologias ágeis, desenvolvimento iterativo e incremental, e a importância do planejamento e execução em sprints.

Entendendo o Escopo do Projeto:

- Definição clara do problema a ser resolvido.
- Importância da comunicação eficaz com os stakeholders para capturar os requisitos.

Descoberta de Requisitos:

- Técnicas para a descoberta de requisitos, como entrevistas, questionários, observação, e workshops com stakeholders.
- Diferença entre requisitos funcionais e não funcionais.

Brainstorming e Prototipagem:

- Estratégias para brainstorming eficaz, incluindo sessões guiadas, uso de whiteboards, e ferramentas de colaboração online.
- A importância da prototipagem rápida para validar ideias com usuários finais e stakeholders.

Design da Tela de Cadastro:

- Princípios de design de interface do usuário (UI) e experiência do usuário (UX) para criar telas intuitivas e acessíveis.
- Elementos chave da tela de cadastro, como campos de formulário, validações, feedback ao usuário, e questões de segurança como criptografia de dados.

Planejamento de Funcionalidades Adicionais:

- Como identificar e priorizar funcionalidades adicionais, como relatórios, notificações, integrações com outros sistemas, etc.
- Uso de histórias de usuários e mapas de jornada do usuário para visualizar como essas funcionalidades se encaixam na experiência geral do usuário.

Testes e Feedback:

- A importância de testes contínuos, incluindo testes unitários, de integração, e de usabilidade, para garantir a qualidade do software.
- Estratégias para coletar e incorporar feedback do usuário no ciclo de desenvolvimento.

Implementação e Lançamento:

- Melhores práticas para a fase de implementação, incluindo revisão de código, integração contínua e entrega contínua (CI/CD).
- Considerações para o lançamento bem-sucedido e monitoramento pós-lançamento.

Para visualizações, pode-se incluir:

- Fluxogramas: Para mapear o processo de descoberta de requisitos e desenvolvimento.
- Wireframes ou mockups: Para visualizar a tela de cadastro e outras interfaces do usuário.
- Gráficos de Gantt: Para planejar e monitorar o progresso do projeto.
- Mapas de jornada do usuário e personas: Para ilustrar como diferentes tipos de usuários interagirão com o sistema.



Formação das equipes

1. Gerente de Projeto/Product Owner (1 pessoa)

- **Responsabilidades:**
- Comunicar-se com os stakeholders para entender as necessidades do projeto.
- Definir o escopo e os objetivos do projeto.
- Priorizar as funcionalidades e gerenciar o backlog do produto.
- Garantir que a equipe esteja alinhada com os objetivos do projeto e prazos.
- Gerenciar riscos e resolver problemas durante o projeto.

2. Analista de Sistemas/Engenheiro de Requisitos (1 pessoa)

- **Responsabilidades:**
- Liderar a descoberta de requisitos através de entrevistas, pesquisas, e sessões de brainstorming.
- Documentar os requisitos funcionais e não funcionais.
- Comunicar requisitos de forma clara para a equipe de desenvolvimento e design.
- Validar soluções propostas com stakeholders e ajustar conforme necessário.

3. Designer de UI/UX (1 pessoa)

- **Responsabilidades:**
- Projetar wireframes e mockups da tela de cadastro e outras interfaces do usuário.
- Garantir uma experiência do usuário intuitiva e acessível.
- Trabalhar em estreita colaboração com o desenvolvedor frontend para implementar o design.
- Realizar testes de usabilidade para coletar feedback e refinar a interface.

4. Desenvolvedor Frontend (1 pessoa)

- **Responsabilidades:**
- Implementar a interface do usuário conforme os designs fornecidos.
- Garantir a responsividade e acessibilidade da aplicação.
- Trabalhar com o desenvolvedor backend para integrar APIs e outros sistemas.
- Realizar testes unitários e de integração no frontend.

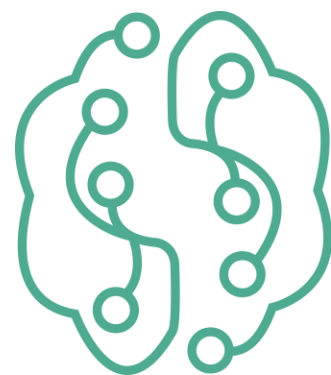
5. Desenvolvedor Backend (1 pessoa)

- **Responsabilidades:**
- Projetar e implementar a lógica de backend, incluindo bancos de dados, autenticação, e APIs.
- Garantir a segurança e a eficiência dos dados.
- Trabalhar com o desenvolvedor frontend para garantir a integração eficaz.
- Realizar testes unitários e de integração no backend.

6. Especialista em Qualidade/Testador (1 pessoa)

- **Responsabilidades:**

- Desenvolver planos de teste para garantir a cobertura dos requisitos.
- Realizar testes manuais e automatizados para identificar bugs e problemas de usabilidade.
- Trabalhar em estreita colaboração com a equipe de desenvolvimento para corrigir os problemas identificados.
- Monitorar a qualidade do produto e garantir que ele atenda aos padrões estabelecidos.



LTD

LABORATÓRIO DE
TRANSFORMAÇÃO DIGITAL

Ferramentas para as funções

Gerente de Projeto/Product Owner

- **Ferramentas:**

- Jira: Para gerenciar o backlog do produto, priorizar tarefas, e acompanhar o progresso do projeto através de sprints. Jira é eficaz para a gestão ágil de projetos e pode ser usado para relatar bugs, atribuir tarefas e monitorar o desenvolvimento.
- Trello: Pode ser usado como uma ferramenta alternativa ou complementar ao Jira para a gestão visual de tarefas, especialmente útil para equipes que preferem um layout de quadro kanban para organizar o trabalho.

Analista de Sistemas/Engenheiro de Requisitos

- **Ferramentas:**

- Jira: Para documentar e gerenciar requisitos como histórias de usuário ou tarefas. O Jira permite criar uma documentação detalhada dos requisitos que pode ser facilmente acessada pela equipe de desenvolvimento e design.

Designer de UI/UX

- **Ferramentas:**

- Figma: Ferramenta primária para o design de interfaces de usuário (UI) e experiência do usuário (UX). Figma é utilizado para criar wireframes, mockups, e protótipos interativos da tela de cadastro e outras interfaces, facilitando a colaboração e o feedback em tempo real com a equipe.

Desenvolvedor Frontend

- **Ferramentas:**

- HTML, CSS, JS: Tecnologias fundamentais para o desenvolvimento do frontend, responsáveis pela estrutura, estilo e funcionalidades interativas da interface do usuário.
- VSCode: Editor de código-fonte usado para o desenvolvimento de software. VSCode suporta HTML, CSS, JavaScript, e muitas outras linguagens, oferecendo funcionalidades como destaque de sintaxe, snippets de código, e extensões para melhorar a produtividade.

Desenvolvedor Backend

- **Ferramentas:**

- PHP: Linguagem de script do lado do servidor utilizada para desenvolver a lógica de backend e manipular a comunicação com o banco de dados.
- Laravel: Framework PHP para o desenvolvimento web que oferece ferramentas e recursos para a construção de aplicações web robustas, incluindo autenticação, roteamento, gestão de sessões e mais.
- VSCode: Para escrever e editar o código-fonte PHP e Laravel, com suporte a extensões específicas que melhoram o desenvolvimento PHP.

Especialista em Qualidade/Testador

- **Ferramentas:**

- Jira: Para registrar, atribuir e rastrear bugs e problemas de usabilidade identificados durante os testes.
- Ferramentas de teste (não especificadas): Embora não mencionadas na lista, ferramentas como Selenium para testes automatizados de interfaces de usuário, ou PHPUnit para testes unitários em PHP, seriam essenciais para o especialista em qualidade.