



Ministério da Educação

Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo

Campus Serra

Verbum: Acervo Digital da Biblioteca do IFES Serra

5º período Curso Técnico em Informática para Internet | 2025-2

Erick Miguel Barbosa do Santos
Joice Bento Calixto da Costa
Katarina Veljovic
Vitória Nascimento Lima

Serra, ES

Novembro de 2025

Sumário

Sumário	2
1.1 Introdução	3
1.1.1 Pesquisa Exploratória	3
1.1.2 Técnicas de Empatia	3
1.1.3 Criação de Personas	3
1.2 Definição do Problema e Ideação	4
1.2.1 Declaração de problema (Point of View - POV)	4
1.2.2 Brainstorming	4
1.3 Prototipagem e Validação da Proposta de Valor	4
1.3.1 Criação de Protótipos de Baixa Fidelidade	5
1.3.2 Teste de Conceito com Usuários	5
1.3.3 Refinamento da Proposta de Valor	5
2.1 Project Model Canvas	5
3.1 Contextualização do Projeto	8
3.2 Escopo da Solução e Requisitos do Sistema	8
3.3 Planejamento e Metodologia de Gerenciamento	17
3.4 Arquitetura e Design Técnico do Sistema	22
3.5 Ecossistema de Tecnologias e Ferramentas	24
3.6 Estratégia de Qualidade e Plano de Testes	26
3.7 Conclusões e Perspectivas Futuras	29
Apêndices	30
Apêndice A: Glossário de Termos Chave	30
Referências citadas	31

Verbum: Acervo Digital da Biblioteca do IFES Serra

1.1 Introdução

O ponto de partida do nosso projeto integrador foi a identificação de um problema real enfrentado pela comunidade acadêmica do IFES Campus Serra: a dificuldade de acesso e gerenciamento das informações da biblioteca. Observou-se que muitos usuários encontravam obstáculos para realizar consultas, reservas, renovações e pagamentos de multas de forma prática e centralizada. Diante disso, surgiu a necessidade de desenvolver uma solução tecnológica que tornasse esses processos mais ágeis, intuitivos e acessíveis. Com base na metodologia de Aprendizagem Baseada em Problemas (ABP), o grupo buscou compreender as reais demandas dos estudantes e servidores. A partir dessa análise, foi escolhido como problemática a otimização do sistema de empréstimo e consulta da biblioteca, com o propósito de facilitar a interação entre os usuários e o acervo.

1.1.1 Pesquisa Exploratória

A etapa inicial do projeto consistiu em uma pesquisa exploratória, cujo objetivo foi compreender o contexto geral do problema enfrentado pelos usuários da biblioteca do IFES Campus Serra. Essa fase buscou mapear as principais dificuldades relacionadas ao acesso às informações sobre o acervo, às reservas, às renovações de empréstimos e ao pagamento de multas de forma presencial e remota.

1.1.2 Técnicas de Empatia

Após a compreensão inicial do problema, a equipe aplicou técnicas de empatia para conhecer diretamente as necessidades e experiências dos usuários. Foram realizadas entrevistas semiestruturadas com estudantes do campus, a fim de compreender suas rotinas de uso da biblioteca e as principais dificuldades enfrentadas no processo de empréstimo e devolução de livros.

1.1.3 Criação de Personas

Persona 1- Lucas: Estudante do curso técnico em Informática. Usa frequentemente o celular e busca praticidade em ferramentas digitais. Costuma esquecer prazos de devolução de livros e gostaria de receber notificações automáticas.

Persona 2- Cláudia: Servidora do IFES. Utiliza a biblioteca para recomendar livros técnicos para seus estudantes.

1.2 Definição do Problema e Ideação

Com um entendimento profundo do usuário e de seu contexto, a equipe refinou sua compreensão sobre o problema identificado. Na fase de Definição do Design Thinking, os aprendizados obtidos durante a imersão foram sintetizados em uma declaração de problema clara, prática e centrada no ser humano, permitindo direcionar o desenvolvimento da solução para as reais necessidades dos usuários da biblioteca.

1.2.1 Declaração de problema (Point of View - POV)

"Os estudantes e servidores do IFES Campus Serra precisam de uma forma prática, intuitiva e centralizada de consultar, reservar e renovar livros da biblioteca porque o processo atual é demorado, pouco acessível e carece de ferramentas digitais integradas que facilitem o gerenciamento de empréstimos e multas."

1.2.2 Brainstorming

Em seguida, foi realizada a sessão de Ideação (Brainstorming), momento em que a equipe se dedicou à geração livre de ideias para resolver o problema definido. Nessa fase, a ênfase foi colocada na quantidade e diversidade de propostas, sem julgamentos ou restrições iniciais, com o objetivo de estimular a criatividade e explorar diferentes possibilidades de solução. Entre as ideias levantadas, destacaram-se a criação de um aplicativo móvel e de um sistema web integrados, que permitissem não apenas a consulta e reserva de livros, mas também o pagamento de multas, o histórico de empréstimos, a avaliação de obras e até recomendações personalizadas com base no histórico de leitura do usuário.

1.3 Prototipagem e Validação da Proposta de Valor

A etapa de Ideação resultou em diversas possibilidades de solução para os desafios identificados, mas antes de avançar para o desenvolvimento efetivo, tornou-se essencial testar as ideias mais promissoras. No Design Thinking, essa fase corresponde à Prototipagem, que consiste em criar representações de baixo custo e

baixa fidelidade das ideias com o objetivo de testá-las com usuários reais. O foco dessa etapa não é avaliar a tecnologia empregada, mas sim verificar se a proposta de valor realmente atende às necessidades e expectativas dos usuários da biblioteca.

1.3.1 Criação de Protótipos de Baixa Fidelidade

Com base nas ideias geradas, a equipe selecionou as soluções mais relevantes: o aplicativo móvel e o sistema web integrados para a biblioteca. Em seguida, foram desenvolvidos protótipos de baixa fidelidade, representando telas e fluxos principais como login, consulta ao acervo, reserva, renovação, histórico de empréstimos, pagamento de multas e avaliação de livros.

1.3.2 Teste de Conceito com Usuários

Para validar as propostas, os protótipos foram apresentados a estudantes do IFES Campus Serra, que representam o público-alvo do sistema. Durante os testes de conceito, os participantes foram convidados a simular ações como buscar um livro e visualizar seu histórico de empréstimos. As reações e comentários dos usuários foram cuidadosamente observados, e suas opiniões coletadas por meio de questionários e entrevistas curtas.

1.3.3 Refinamento da Proposta de Valor

A partir do feedback recebido, foram realizados ajustes na organização das telas, na clareza das informações exibidas e na navegação entre as funções. O resultado dessa fase foi o refinamento da proposta de valor, validando a criação de uma plataforma integrada.

2.1 Project Model Canvas

A seguir, o template do *Project Model Canvas*, organizado em torno das 6 questões fundamentais do gerenciamento de projetos: Por quê?, O quê?, Quem?, Como?, Quando? e Quanto?

POR QUÊ?	O QUÊ?	QUEM?	COMO?	QUANDO e
-----------------	---------------	--------------	--------------	-----------------

				QUANTO?
Justificativa	Produto	Stakeholders	Premissas	Riscos
Objetivos SMART	Requisitos	Equipe	Grupo de Entregas	Linha do Tempo
<ul style="list-style-type: none"> Para substituir o Pergamum, sistema antigo e pouco prático. Para criar uma experiência mais intuitiva e rápida para estudantes e servidores. 	<p>Um aplicativo e sistema web com interface moderna que facilite o acesso ao acervo da biblioteca da escola, sendo possível reservar, consultar e renovar o empréstimo de livros, acessar o histórico de empréstimos, marcar os livros favoritos, pagar multas e avaliar obras.</p>	<p>Estudantes e servidores do campus, além da equipe da biblioteca e da coordenação de TI.</p>	<p>Os usuários devem ter acesso ao celular ou computador/not ebook.</p> <p>Os usuários terão acesso à internet dentro da escola ou em casa.</p> <p>O uso será gratuito e restrito à comunidade escolar.</p> <p>O app e o site estarão integrados a um mesmo banco de dados da biblioteca.</p>	<p>Falta de tempo.</p> <p>Defeito nos computadores da escola ou nos dos integrantes da equipe.</p> <p>Perda de dados.</p>
<p>O projeto Verbum tem como objetivo substituir o sistema Pergamum da biblioteca do Campus Serra, tornando a experiência de busca, reserva e renovação de livros mais intuitiva e rápida. Espera-se que os usuários consigam localizar livros em poucos cliques, reduzir em até 50% o tempo gasto em</p>	<p>O app deve possuir uma interface simples e responsiva, integração completa com a base de dados da biblioteca, funcionalidades de pesquisa, reserva e renovação de livros,</p>	<p>A equipe que está desenvolvendo o app é formada por um grupo de estudantes do ensino médio técnico do IFES.</p> <p>O grupo é constituído por: Joice Bento, Katarina Veljovic, Erick Miguel e Vitória Lima.</p>	<p>As entregas principais do projeto incluem a criação do protótipo da interface, a integração com o banco de dados da biblioteca, a implementação das funcionalidades de pesquisa, reserva e renovação, o</p>	<p>O projeto durará 1 ano</p>

<p><i>pesquisas e receber notificações de prazos de devolução, garantindo resultados específicos, mensuráveis, atingíveis, relevantes e com prazo definido.</i></p>	<p><i>além de enviar notificações automáticas sobre os prazos de devolução. O sistema deve ser compatível com dispositivos Android e iOS e ser fácil de usar por estudantes e professores.</i></p>		<p><i>sistema de notificações, a realização de testes de usabilidade e, finalmente, o lançamento oficial do app Verbum.</i></p>	
Benefícios			Restrições	Custos
<p><i>Facilidade de acesso ao acervo sem precisar ir fisicamente à biblioteca.</i></p> <p><i>Os alunos podem reservar um livro online e retirá-lo depois.</i></p> <p><i>Maior interesse pela leitura, já que o processo se torna mais prático e interativo.</i></p> <p><i>Autonomia no uso da biblioteca, aprendendo a buscar e gerenciar suas leituras.</i></p> <p><i>Notificações e lembretes sobre os prazos dos empréstimos.</i></p>			<p><i>Prazo.</i></p> <p><i>Devem ser usadas ferramentas gratuitas.</i></p> <p><i>Pouco tempo semanal disponível para o desenvolvimento.</i></p> <p><i>Compatibilidade com navegadores comuns.</i></p> <p><i>Sistema simples e de fácil entendimento.</i></p>	<p><i>Tempo.</i></p>

3.1 Contextualização do Projeto

- **Propósito:** Reformular o sistema remoto (web) atual e a criação de um aplicativo mobile para acessar os serviços – consultar, reservar, renovar, avaliar livros e pagar multas – de forma simples e intuitiva.
- **Instruções Detalhadas:**
 - **Contextualização do Problema:** Os usuários da biblioteca do IFES Campus Serra enfrentam dificuldades para acessar os serviços da biblioteca remotamente de forma prática e intuitiva. Atualmente, muitos processos são manuais ou pouco acessíveis, gerando demora, falta de informação em tempo real e baixa autonomia do usuário.
 - **Justificativa:** O projeto é importante porque moderniza o sistema remoto atual, facilita a comunicação com o usuário e amplia o acesso aos serviços da biblioteca. Há uma clara oportunidade de inovação ao centralizar funções essenciais em um aplicativo e sistema web intuitivos, tornando a experiência mais prática e alinhada às necessidades contemporâneas de digitalização no ambiente acadêmico.
 - **Objetivo Geral:** Desenvolver um sistema digital integrado que facilite e modernize o acesso dos usuários aos serviços da biblioteca do IFES Campus Serra.
 - **Objetivos Específicos (SMART):**
 - Desenvolver até o final do semestre um aplicativo e um sistema web que permitam consulta ao acervo em tempo real.
 - Disponibilizar as funções de reserva e renovação de livros de forma digital.
 - Integrar notificações automáticas de prazos e disponibilidade até o lançamento da versão MVP.
 - Criar uma área administrativa que permita ao bibliotecário gerenciar obras, empréstimos e relatórios ainda no semestre letivo atual.
 - Garantir que 90% dos usuários testados consigam realizar tarefas básicas (consultar, reservar, renovar) em até 3 minutos, validando a usabilidade do sistema.

- **Conexão Conceitual:** Esta seção formaliza o "POR QUÊ?" do Project Model Canvas, articulando justificativa, objetivos e benefícios. Ela sintetiza as reflexões feitas sobre o problema central e direciona o desenvolvimento do projeto para soluções práticas e alinhadas às necessidades dos usuários.

3.2 Escopo da Solução e Requisitos do Sistema

- **Propósito:** O escopo do produto inclui um sistema web responsivo para consulta ao acervo da biblioteca do IFES campus Serra, reservas, renovações e notificações pagamento de multas, relatórios e recomendações básicas, segurança (hashing, SSL, autenticação) e interface acessível e intuitiva.
- **Instruções Detalhadas:**
 - **Declaração de Escopo do Produto:** O produto tem como objetivo o desenvolvimento de um aplicativo e sistema web para consulta ao acervo digital da biblioteca do IFES campus Serra. Ele permitirá que os usuários realizem a reserva e renovação de empréstimos de livros, além de possibilitar a avaliação das obras e o pagamento de multas associadas a atrasos. O sistema será acessível via dispositivos móveis e navegadores e terá uma interface amigável e intuitiva, visando proporcionar uma experiência fluida e eficiente para os usuários. O resultado final será uma plataforma digital que conecta os leitores ao acervo da biblioteca, otimizando o processo de empréstimos, interações e pagamentos.
 - **Fora de Escopo:** A venda de livros ou qualquer funcionalidade relacionada ao comércio de obras literárias, integração com sistemas externos de bibliotecas (ex: outras bibliotecas de redes diferentes), suporte para idiomas além do português, sistema de leitura ou visualização digital dos livros (apenas informações sobre os livros serão fornecidas).
 - **Requisitos Funcionais (User Stories):** O sistema deve permitir o cadastro de usuários e, após esse processo, disponibilizar uma tela de login para acesso às funcionalidades. Uma vez autenticado, o usuário poderá consultar o acervo da biblioteca, visualizar a disponibilidade das obras — indicando se estão disponíveis, emprestadas ou reservadas —, além de verificar a quantidade de exemplares existentes. O sistema também oferecerá a possibilidade de edição do perfil e de realização de logoff a

qualquer momento. Caso um livro esteja emprestado, o usuário poderá efetuar uma reserva e será notificado quando a obra estiver novamente disponível. O sistema enviará lembretes de devolução antes da data limite e disponibilizará uma opção para pagamento de multas pendentes. O usuário ainda poderá acessar seu histórico de empréstimos e reservas, bem como receber recomendações de livros com base em seu histórico de leitura.

- **Requisitos Não Funcionais:**

- **Usabilidade:**

- A interface deve ser intuitiva, limpa e organizada, permitindo que o usuário aprenda a utilizá-la sem necessidade de treinamento.
 - O sistema deve seguir padrões de acessibilidade (WCAG) para garantir uso por pessoas com deficiência visual, auditiva ou motora.

- **Desempenho:**

- O sistema deve estar disponível 99% do tempo, exceto em períodos de manutenção programada.
 - O sistema deve lidar com entradas de dados incorretas sem travar ou gerar erros críticos.

- **Segurança:**

- Senhas devem ser armazenadas utilizando hashing seguro (ex: bcrypt ou Argon2).
 - Os usuários só devem ter acesso aos dados e funcionalidades para os quais foram explicitamente autorizados.

- **Manutenibilidade:**

- O sistema deve ser modularizado, de modo que a alteração de um dos módulos não cause erros nos outros e que seja de fácil manutenção.

- **Conexão Conceitual:** As User Stories apresentadas foram formuladas com base em insights obtidos na fase de Design Thinking, garantindo que as funcionalidades atendam diretamente às necessidades e dores dos usuários, como a reserva de livros, renovação de empréstimos, pagamento de multas, e a avaliação dos livros. As metodologias ágeis, como o Scrum, serão utilizadas para implementar e iterar rapidamente sobre essas funcionalidades, com o foco constante no valor entregue aos usuários. O escopo do projeto foi cuidadosamente definido para evitar o aumento descontrolado de funcionalidades (scope creep), assegurando que a entrega final seja fiel ao propósito original e às necessidades dos usuários.

- **Tabela de Requisitos (User Stories)**

- Esta tabela é o coração do escopo do produto e a base para o planejamento do desenvolvimento.

ID	Título da Story	Descrição (Como um..., eu quero..., para que...)	Critérios de Aceitação	Prioridade (MoSCoW)
US01	Login de Usuário	Como um novo usuário, eu quero entrar no sistema com matrícula e senha para que eu possa salvar minhas preferências e acessar minhas informações.	1. O sistema deve validar o formato da matrícula. 2. A senha deve ter no mínimo 8 caracteres.	M
US02	Pagamento de taxa de atraso	Como um usuário cadastrado, eu quero acessar o exemplar emprestado para que eu possa pagar a taxa de atraso.	1. O sistema deve mostrar os exemplares emprestados. 2. Deve haver uma opção "Taxa de atraso". 3. Ao clicar na opção "Taxa de atraso", o sistema deve disponibilizar formas de pagamento para a taxa.	M
US03	Renovação de empréstimo	Como um usuário cadastrado, eu quero acessar o exemplar emprestado para que eu	1. O sistema deve mostrar os exemplares emprestados. 2. Deve haver uma opção	M

		<i>possa renovar o empréstimo.</i>	<p>"Renovar Empréstimo".</p> <p>3. Ao clicar na opção "Renovar Empréstimo", o sistema deve automaticamente estender o prazo de devolução do exemplar.</p>	
<i>US04</i>	<i>Consulta ao acervo com filtros e busca avançada.</i>	<i>Como um usuário cadastrado, eu quero pesquisar títulos e filtrar a busca para que eu possa encontrar o resultado de maneira eficaz.</i>	<p>1. Ao digitar um termo no campo de pesquisa, o sistema deve retornar uma lista de títulos que contenham total ou parcialmente o termo buscado.</p> <p>2. Ao digitar um termo inexistente, o sistema deve exibir uma mensagem informando que "nenhum resultado foi encontrado".</p> <p>3. Ao selecionar um ou mais filtros disponíveis, o sistema deve atualizar os resultados exibindo apenas os títulos que correspondam aos filtros aplicados.</p> <p>4. Se ocorrer uma falha na</p>	<i>M</i>

			<p><i>busca (ex.: erro de rede ou servidor), o sistema deve exibir uma mensagem amigável ao usuário e permitir tentar novamente.</i></p>	
<i>US05</i>	<i>Avaliação e recomendação de obras com base no histórico de leitura.</i>	<i>Como um usuário cadastrado, eu quero poder avaliar os títulos para que minha página inicial possua recomendações baseadas nas minhas avaliações.</i>	<p><i>1. Ao acessar os detalhes de um título, o usuário deve poder atribuir uma avaliação.</i></p> <p><i>2. O usuário deve poder editar ou remover sua avaliação.</i></p> <p><i>3. Caso ocorra uma falha ao registrar a avaliação (erro de rede, servidor etc.), o sistema deve exibir uma mensagem de erro amigável e permitir tentar novamente.</i></p> <p><i>4. Caso o usuário tenha avaliado algum título, o sistema deve exibir uma seção de recomendações personalizadas baseadas em suas avaliações anteriores. Caso não, o sistema deve exibir</i></p>	<i>S</i>

			<p><i>recomendações genéricas.</i></p> <p><i>5. As avaliações do usuário devem ser visíveis apenas para ele.</i></p> <p><i>6. A média de avaliações de um título (de todos os usuários) deve ser exibida.</i></p> <p><i>7. O sistema deve permitir avaliar de forma simples e intuitiva.</i></p>	
<i>US06</i>	<i>Marcar obras como "Favoritas"</i>	<i>Como um usuário cadastrado, eu quero acessar o acervo online da Biblioteca para que eu possa favoritar obras.</i>	<p><i>1. O sistema deve disponibilizar o acervo na página inicial.</i></p> <p><i>2. Ao clicar em um exemplar, deve haver um botão para adicionar a obra à minha lista de "Favoritos"</i></p> <p><i>3. No perfil do usuário, o sistema deve automaticamente ao clicar no botão de favoritar, adicionar a obra à lista de "Favoritos" e exibi-lá.</i></p> <p><i>4. Na lista de Favoritos, o usuário poderá remover a obra</i></p>	<i>S</i>

			<i>da lista a qualquer momento.</i>	
<i>US07</i>	<i>Acesso ao histórico de empréstimos</i>	<i>Como um usuário cadastrado, eu quero acessar minha lista de empréstimos para que eu consiga ter controle do que já li.</i>	<p><i>1. O sistema deve ter um botão "Histórico de Empréstimo" na página de perfil do usuário.</i></p> <p><i>2. Ao clicar no botão, o sistema deve exibir a lista de todos os exemplares emprestados, com as datas de empréstimo e devolução.</i></p> <p><i>3. O sistema deve adicionar uma obra ao histórico de empréstimo quando o usuário retirar o livro na biblioteca.</i></p>	<i>M</i>
<i>US08</i>	<i>Perfil de Adm.</i>	<i>Como um administrador cadastrado, eu quero ter acesso ao acervo para modificá-lo e cadastrar novos usuários.</i>	<i>1. O sistema Web deve possuir perfil tipo administrador.</i>	<i>S</i>
<i>US09</i>	<i>Sistema de Gamificação</i>	<i>Como um usuário cadastrado, eu quero ter um sistema competitivo que incentive a leitura, havendo</i>	<i>1. Para o usuário fazer parte do ranking, é necessário fazer parte do Clube do Livro, que é realizado</i>	<i>C</i>

		<i>ranking de leitura, para comprovar, seja necessário realizar review.</i>	<i>presencialmente uma vez por mês</i>	
--	--	---	--	--

3.3 Planejamento e Metodologia de Gerenciamento

- O objetivo desta seção é descrever como o projeto Verbum será gerenciado e executado, detalhando a abordagem metodológica escolhida, a estrutura do time Scrum, os artefatos utilizados e o plano proativo de gerenciamento de riscos. Aqui são definidas as práticas que garantirão organização, alinhamento entre os membros e adaptação contínua durante o desenvolvimento, assegurando que o produto final atenda às necessidades dos usuários da biblioteca.
- **Instruções Detalhadas:**
 - **Metodologia de Gerenciamento:** O projeto Verbum será gerenciado utilizando o *framework Scrum*, um método ágil amplamente adotado no desenvolvimento de software. A escolha do *Scrum* se justifica por sua flexibilidade, permitindo adaptações naturais conforme surgem novos *feedbacks* dos usuários; por suas entregas incrementais, que garantem evolução contínua do produto; e pela colaboração intensa entre os membros da equipe, característica indispensável em projetos educativos e multidisciplinares como este. Além disso, o *Scrum* favorece inspeção frequente, transparência e melhoria contínua, princípios cruciais para manter o projeto alinhado às necessidades reais dos estudantes e servidores do IFES.
 - **O Time Scrum:** Com base na composição oficial da equipe do projeto Verbum, os papéis do *Scrum* foram definidos da seguinte forma:

■ **Product Owner (PO) - Erick Miguel:**

Responsável por:

- Gerenciar o *Product Backlog*;
- Priorizar as funcionalidades;
- Representar as necessidades dos usuários e *stakeholders*;

- Garantir que o time produza o maior valor possível a cada *Sprint*

■ Scrum Master (SM) - Joice Bento

Responsável por:

- Assegurar o uso correto do *framework Scrum*;
- Facilitar cerimônias;
- Incentivar a comunicação eficaz;
- Remover impedimentos que atrapalhem o progresso.

■ Time de Desenvolvimento - Vitória Lima e Katarina Veljovic:

Responsáveis por:

- Desenvolvimento do aplicativo mobile e sistema web;
- Implementação de funcionalidades, banco de dados e interfaces;
- Testes, revisão de código e entrega de incrementos funcionais.

- **Eventos Scrum:** Descreva o plano de execução dos eventos (cerimônias) do Scrum⁵:

- **Eventos Scrum:** Descreva o plano de execução dos eventos (cerimônias) do Scrum⁵:
 - **Duração da Sprint:** As Sprints terão duração de 2 semanas, garantindo um ritmo equilibrado de entregas, validação e adaptação contínua.
 - **Sprint Planning:** No início de cada *Sprint*, o PO apresenta as User Stories priorizadas no *Product Backlog*.
 - **Daily Scrum:** Reunião diária de 15 minutos para alinhamento do time.
 - **Sprint Review:** Ao final da *Sprint*, o Time apresenta o incremento funcional aos stakeholders, principalmente ao PO e à equipe da biblioteca. *Feedbacks* são coletados e adicionados ao *Product Backlog*.
 - **Sprint Retrospective:** Após a Review, a equipe se reúne para identificar: O que funcionou bem, o que pode ser melhorado, ações para melhorar.

- **Artefatos Scrum:** Apresente os artefatos iniciais¹⁶:

- **Artefatos Scrum:** Apresente os artefatos iniciais¹⁶:
 - **Product Backlog:** O **Product Backlog inicial** do projeto é composto pela **Tabela de Requisitos (User Stories)** apresentada na seção 3.2 do documento.
 - **Definition of Done (DoD):** Uma *User Story* será considerada “**Pronta**” quando cumprir todos os critérios abaixo:

- Código implementado conforme os requisitos.
- Código revisado por outro membro da equipe.

- Interface aderente ao protótipo do Figma.
 - Critérios de aceitação totalmente atendidos.
 - Testes funcionais realizados e aprovados.
 - Documentação atualizada (tela, fluxo, endpoints, regras).
 - Incremento integrado ao repositório sem conflitos.
 - Nenhum erro crítico pendente.
- **Conexão Conceitual:** Esta seção aplica diretamente os conceitos do Scrum Guide.¹⁶ A inclusão do gerenciamento de riscos, mesmo de forma simplificada, introduz uma prática essencial da área de conhecimento de Gerenciamento de Riscos do PMBOK®, ensinando a equipe a ser proativa em vez de reativa.¹⁵
- **Tabela de Riscos Simplificada**
 - Esta tabela introduz o pensamento proativo sobre possíveis problemas, uma habilidade crucial para qualquer gerente de projeto.

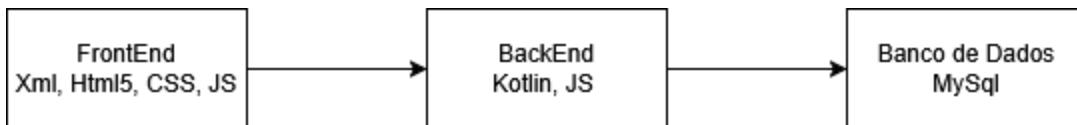
ID do Risco	Descrição do Risco	Probabilidade (Baixa/Média/Alta)	Impacto (Baixo/Médio/Alto)	Ação de Mitigação/Contingência
R01	<i>Sobrecarga acadêmica de um membro da equipe, reduzindo sua disponibilidade.</i>	Média	Alta	Mitigação: <i>Criar um cronograma de tarefas realista e distribuído.</i> Contingência: <i>Documentar o trabalho de forma clara para que outro membro possa assumir se necessário.</i>
R02	<i>A API de terceiros escolhida para o projeto se torna instável ou é descontinuada.</i>	Baixa	Alta	Mitigação: <i>Desenvolver uma camada de abstração no código para facilitar a troca da API.</i> Contingência: <i>Pesquisar e listar 1-2 APIs</i>

				<i>alternativas.</i>
<i>R03</i>	<i>Dificuldade em obter dados reais do acervo institucional.</i>	Média	Alta	<p><i>Mitigação:</i> solicitar antecipadamente à equipe da biblioteca acesso a dados simulados.</p> <p><i>Contingência:</i> usar base de dados fictícia para testes.</p>
<i>R04</i>	<i>Erros de comunicação ou falta de alinhamento entre os membros do time.</i>	Média	Média	<p><i>Mitigação:</i> realizar reuniões diárias curtas e registrar decisões no Trello ou Figma.</p> <p><i>Contingência:</i> Mediar reuniões extras quando houver divergências.</p>
<i>R05</i>	<i>Complexidade técnica maior que o esperado (autenticação, integração, etc.).</i>	Alta	Alta	<p><i>Mitigação:</i> dividir o sistema em módulos simples e priorizar o MVP.</p> <p><i>Contingência:</i> reduzir escopo e entregar apenas as funções essenciais.</p>
<i>R06</i>	<i>Falta de engajamento dos usuários nos testes ou feedback insuficiente.</i>	Média	Alta	<p><i>Mitigação:</i> convidar usuários desde o início.</p> <p><i>Contingência:</i> realizar testes com colegas e professores simulando o uso real.</p>

R07	Mudança de requisito durante o desenvolvimento	Alta	Média	<p>Mitigação: validar cada alteração com o Product Owner antes de implementar.</p> <p>Contingência: priorizar as mudanças mais críticas e reagendar o restante para versões futuras.</p>
-----	--	------	-------	--

3.4 Arquitetura e Design Técnico do Sistema

- Propósito:** Descrever a estrutura técnica fundamental da solução, incluindo a organização do software, a modelagem dos dados e o design da interface com o usuário. Esta seção é o elo entre os requisitos e a implementação.



- Instruções Detalhadas:**

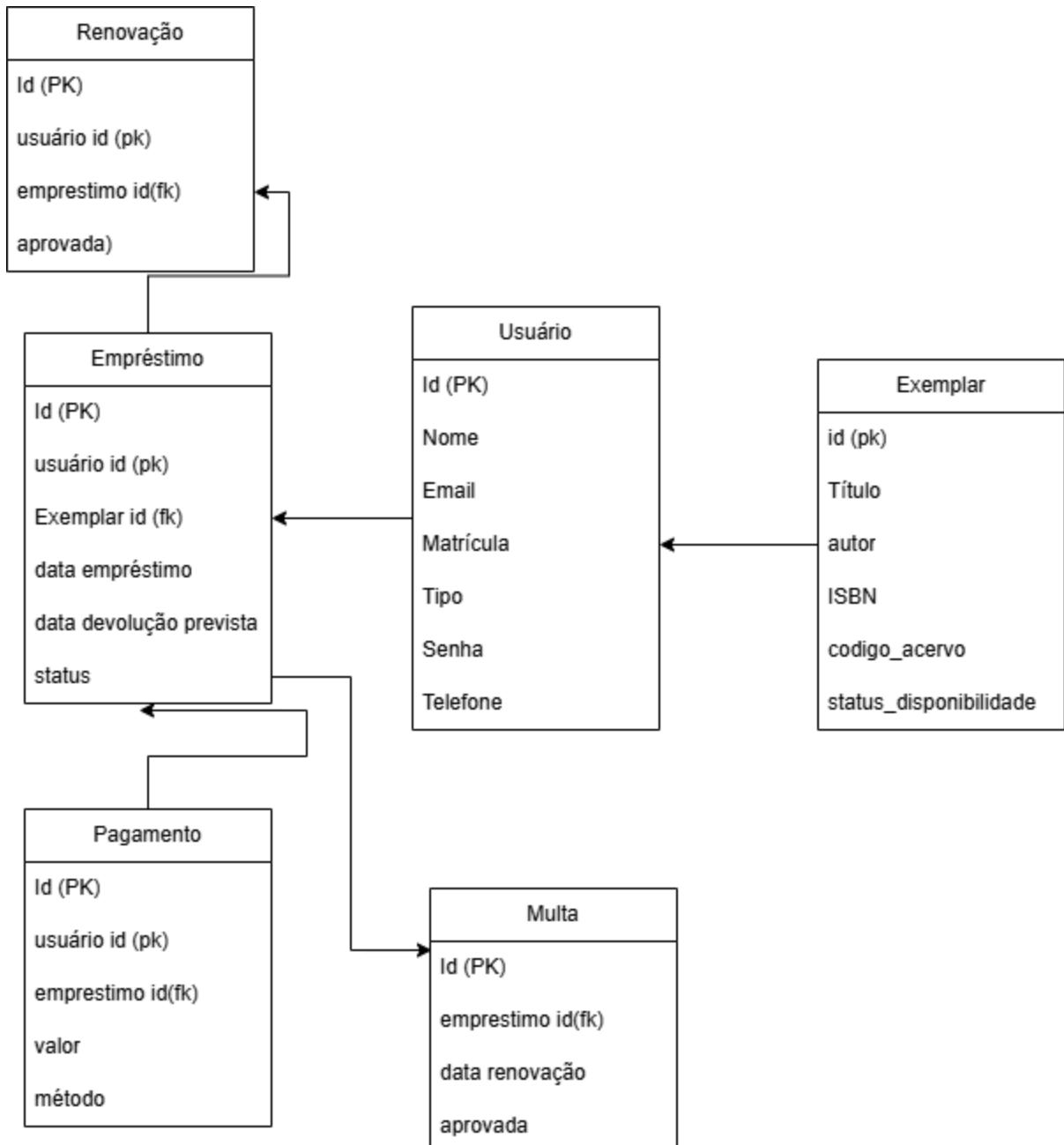
Padrão arquitetural escolhido: Arquitetura em Camadas (Client — Application/API — Data), seguindo princípios Cliente-Servidor e separação de responsabilidades. Em termos de estilo de organização interna do backend, adotamos o padrão MVC (*Model-View-Controller*) ou, para APIs, o equivalente *Model-Service-Controller*, o que facilita manutenção e testes.

Justificativa:

- Adequação ao escopo: a arquitetura em camadas é simples, didática e suficiente para um MVP acadêmico, permitindo evolução para microserviços se necessário.
- Separação de responsabilidades: Frontend, regras de negócio e persistência ficam desacoplados, permitindo desenvolvimento paralelo por diferentes integrantes do grupo.

- Facilidade de teste e deploy: cada camada pode ser testada isoladamente; backend pode expor APIs versionadas e ser hospedado independentemente do frontend.
- Escalabilidade e opções de hospedagem: Backend e DB podem ser migrados para serviços gerenciados (Heroku, Railway, AWS/GCP) sem alterar o frontend.

- **Modelo de Dados:**



- **Design da Interface de Usuário (UI) e Experiência do Usuário (UX):**
Apresente protótipos de média ou alta fidelidade (wireframes ou mockups) das telas principais do sistema. Descreva a jornada do usuário através dessas telas, explicando as decisões de design tomadas para garantir uma boa usabilidade e experiência.
- **Conexão Conceitual:**
 - Engenharia de Software:** tomada de decisões arquiteturais (camadas, padrões MVC, versionamento de API) e práticas de qualidade (testes unitários, CI/CD, documentação da API).
 - Banco de Dados:** modelagem ER para garantir integridade referencial, uso de índices para consultas por ISBN/código/usuário e estratégias de backup/restore.
 - IHC (Interação Humano-Computador):** prototipagem, avaliação heurística (visibilidade do status do sistema, consistência), e testes de usabilidade para priorizar funções essenciais como renovação e pagamento.

3.5 Ecossistema de Tecnologias e Ferramentas

- **Linguagens de Programação:**
 - Kotlin: Recomendamos Kotlin para o desenvolvimento mobile Android, pois é a linguagem oficial da plataforma e oferece uma sintaxe moderna, segura e totalmente integrada ao AndroidStudio.
 - Xml: Recomendamos XML para a definição das interfaces gráficas no Android, já que ele é o padrão utilizado pelo Android Studio para estruturar layouts e componentes visuais.
 - JavaScript: Recomendamos JavaScript para o desenvolvimento web, pois é a principal linguagem de programação para o front-end e permite a integração direta com o React.js.
 - HTML5: Recomendamos HTML5 para a estruturação das páginas web, garantindo compatibilidade com navegadores modernos e suporte a recursos multimídia.
 - CSS: Recomendamos CSS para o estilo das interfaces web, pois

ele permite personalizar cores, fontes e layouts, assegurando uma boa experiência visual ao usuário.

- **Frameworks/Bibliotecas (Front-end e Back-end):**
 - React: Recomendamos React.js para o front-end porque ele é baseado em componentes, o que facilita a criação e reutilização das telas que desenhamos no Figma.
 - Node.js: Recomendamos Node.js para o back-end, pois ele utiliza JavaScript no servidor e permite a criação de APIs rápidas e escaláveis.
 - Django: Recomendamos Django como opção de back-end por ser um framework robusto em Python, com estrutura organizada e foco em segurança e rapidez no desenvolvimento.
 - SpringBoot: Recomendamos SpringBoot para o desenvolvimento back-end em Java, pois ele simplifica a criação de aplicações complexas e oferece integração com diversos bancos de dados.
- **Sistema de Gerenciamento de Banco de Dados (SGBD):**
 - MySQL: Recomendamos MySQL como banco de dados relacional por sua estabilidade, ampla documentação e fácil integração com frameworks como Spring Boot e Django.
- **Ambiente de Desenvolvimento (IDEs):**
 - VSCode: Recomendamos o Visual Studio Code, por ser um editor leve e altamente personalizável, excelente para o desenvolvimento web (JavaScript, HTML, CSS). Possui uma vasta biblioteca de extensões e integração com o Git, o que facilita o trabalho em equipe.
 - Figma: Recomendamos o Figma para o design e prototipação das interfaces, pois permite colaboração em tempo real, criação de protótipos interativos e integração direta com desenvolvedores, garantindo consistência entre design e implementação.
 - AndroidStudio: Recomendamos o Android Studio para o desenvolvimento de aplicativos Android, pois ele é a IDE oficial do Google e oferece ferramentas completas para testar e

depurar o app.

- **Sistema de Controle de Versão:**

- GitHub: Recomendamos GitHub para o controle de versão e colaboração da equipe, pois ele permite armazenar, versionar e revisar o código de forma organizada.

- **Ferramentas de Gerenciamento de Projeto:**

- Trello: Recomendamos Trello para o gerenciamento do projeto, pois ele oferece uma interface intuitiva baseada em quadros e cartões, facilitando o acompanhamento das tarefas e prazos.

3.6 Estratégia de Qualidade e Plano de Testes

História de Usuário a ser testada:

- Como um novo usuário, eu quero entrar no sistema com matrícula e senha para que eu possa salvar minhas preferências e acessar minhas informações.

Cenário de Teste (O que dizer ao usuário):

- Imagine que você é um novo usuário do sistema da biblioteca e deseja acessar sua conta para visualizar suas informações e preferências de leitura. Por favor, use este protótipo e tente realizar o login utilizando sua matrícula e senha. Lembre-se de pensar em voz alta enquanto realiza as ações, explicando o que está fazendo e o que espera que aconteça a cada etapa.

Passos Esperados (Critérios de Sucesso):

1. O usuário acessa a tela inicial e visualiza os campos “Matrícula” e “Senha”.
2. O usuário digita corretamente sua matrícula e senha válidas.
3. Ao clicar no botão “Entrar”, o sistema valida as credenciais.
4. Caso as informações estejam corretas, o usuário é redirecionado para a tela inicial do sistema, onde pode ver seu perfil e preferências.
5. Caso a matrícula ou senha estejam incorretas, o sistema exibe uma mensagem clara: “Matrícula ou senha inválidas. Tente novamente.”
6. O botão “Esqueci minha senha” deve estar visível e funcional.
7. O tempo de resposta do login não deve exceder 5 segundos.

Critério de Falha (Quando o teste falha?):

- O teste será considerado uma falha se o usuário não conseguir identificar

facilmente os campos de matrícula e senha, se o botão “Entrar” não responder ou não redirecionar corretamente, se o sistema não exibir mensagens de erro claras ao inserir credenciais inválidas, ou se o login demorar mais de 5 segundos para carregar. Também será considerado falha se o usuário demonstrar confusão, hesitar por mais de 30 segundos para localizar o botão “Entrar” ou se o sistema permitir o acesso com campos em branco ou dados incorretos.

História de Usuário a ser testada:

- Como um usuário cadastrado, eu quero acessar o exemplar emprestado para que eu possa pagar a taxa de atraso.

Cenário de Teste (O que dizer ao usuário):

- Imagine que você é um usuário que possui um livro emprestado e percebeu que ele está atrasado. Você deseja acessar o sistema para visualizar o exemplar e realizar o pagamento da taxa de atraso. Por favor, use este protótipo e tente efetuar o pagamento. Lembre-se de pensar em voz alta enquanto realiza as ações, dizendo o que está fazendo e o que espera que aconteça.

Passos Esperados (Critérios de Sucesso):

1. O usuário acessa o sistema com suas credenciais válidas.
2. O usuário visualiza uma lista de exemplares emprestados em sua conta.
3. O exemplar atrasado aparece identificado com um aviso de atraso.
O usuário clica na opção “Pagar multa” abaixo do exemplar correspondente.
4. O sistema exibe o valor da taxa e as opções de pagamento disponíveis.
5. O usuário seleciona uma forma de pagamento (ex: cartão ou boleto) e confirma a operação.
6. O sistema exibe uma mensagem de confirmação: “Pagamento realizado com sucesso.”
7. O status do exemplar é atualizado, removendo o aviso de atraso.

Critério de Falha (Quando o teste falha?):

- O teste será considerado uma falha se o usuário não conseguir identificar facilmente o exemplar atrasado, se o botão “Pagar multa” não estiver visível ou não funcionar, ou se o sistema não exibir corretamente o valor e as opções de pagamento. Também será considerado falha se o usuário demonstrar confusão sobre como efetuar o pagamento, se o sistema permitir a conclusão com dados incompletos, ou se não apresentar a mensagem de confirmação após a

operação.

História de Usuário a ser testada:

- Como um usuário cadastrado, eu quero acessar o exemplar emprestado para que eu possa renovar o empréstimo.

Cenário de Teste (O que dizer ao usuário):

- Imagine que você possui um livro emprestado da biblioteca e deseja continuar com ele por mais alguns dias. Para isso, você vai acessar o sistema e tentar renovar o empréstimo do exemplar. Por favor, use este protótipo e realize a renovação. Lembre-se de pensar em voz alta, explicando o que está fazendo e o que espera que aconteça a cada passo.

Passos Esperados (Critérios de Sucesso):

1. O usuário faz login no sistema com matrícula e senha válidas.
2. O usuário visualiza a lista de exemplares atualmente emprestados em "Minha Biblioteca".
3. O exemplar disponível para renovação mostra a opção "Renovar Empréstimo" ao lado.
4. O usuário clica em "Renovar Empréstimo".
5. O sistema atualiza automaticamente a data de devolução do livro.
6. Uma mensagem de confirmação aparece: "Renovação concluída."
7. O novo prazo de devolução é exibido corretamente.

Critério de Falha (Quando o teste falha?):

- O teste será considerado uma falha se o usuário não conseguir localizar a opção "Renovar Empréstimo", se o botão não responder ao clique ou se o sistema não atualizar corretamente a data de devolução. Também será considerado falha se não aparecer uma mensagem de confirmação após a renovação ou se o participante demonstrar confusão e demorar mais de 30 segundos para entender como renovar o exemplar.

História de Usuário a ser testada:

- Como um usuário cadastrado, eu quero pesquisar títulos e filtrar a busca para que eu possa encontrar o resultado de maneira eficaz.

Cenário de Teste (O que dizer ao usuário):

Imagine que você é um usuário cadastrado e deseja encontrar um livro ou artigo específico, então acessa a página de busca. Na barra de pesquisa, você digita um termo e clica no botão de Pesquisar. O sistema processa sua solicitação e exibe uma lista de resultados relacionados ao termo digitado. Para tornar a pesquisa mais objetiva você decide refinar sua busca aplicando filtros: primeiro escolhe a categoria “Conto”, e depois define o filtro “Nacional”. Imediatamente, a lista é atualizada e exibe apenas os títulos que correspondem ao termo buscado e aos filtros aplicados. Você então analisa os resultados e encontra exatamente o que procurava.

Passos Esperados (Critérios de Sucesso):

1. O usuário acessa o sistema com matrícula e senha válidas.
2. O usuário acessa a página de busca através do ícone de lupa no topo da tela.
3. O usuário escreve sua busca e identifica o local onde aplicar os filtros.
4. O usuário consegue aplicar os filtros desejados e removê-los quando quiser.

Critério de Falha (Quando o teste falha?):

- Se o usuário não conseguir identificar o ícone de pesquisa com facilidade.
- Se o usuário não conseguir encontrar o local de aplicação do filtro.
- Se o usuário não conseguir encontrar o filtro desejado.
- Se o filtro falhar e não for devidamente aplicado no sistema.

3.7 Conclusões e Perspectivas Futuras

O presente projeto teve como ponto de partida a identificação de um problema real enfrentado pelos estudantes e servidores do IFES Campus Serra: a dificuldade de acessar, gerenciar e acompanhar as informações relacionadas ao acervo da biblioteca de forma prática e centralizada. A partir dessa necessidade, desenvolveu-se a proposta de criar um aplicativo móvel e um sistema web integrados, com o intuito de facilitar a consulta, reserva, renovação, avaliação de obras e pagamento de multas, oferecendo uma experiência mais ágil, intuitiva e moderna aos usuários.

O projeto surgiu da necessidade de melhorar o acesso às informações e serviços da biblioteca do IFES Campus Serra. A solução proposta é um aplicativo e um sistema web integrados que permitem consultar, reservar, renovar, avaliar livros e pagar

multas de forma simples e intuitiva. O objetivo principal é oferecer ao usuário uma experiência mais prática, ágil e centralizada no uso do acervo.

Os principais desafios envolveram compreender profundamente as necessidades dos usuários e transformar essas demandas em requisitos claros. A equipe aprendeu a aplicar técnicas de empatia, organizar dados coletados e alinhar ideias durante a ideação. Também desenvolveu maior habilidade de colaboração, comunicação e tomada de decisão centrada no usuário.

Entre as possibilidades de evolução do sistema estão: gamificação, chatbot de atendimento, incluir painel administrativo e expansão para outros campi do IFES. Essas melhorias tornariam o sistema mais completo e versátil.

Apêndices

Apêndice A: Glossário de Termos Chave

- **Design Thinking:** Uma abordagem para a inovação centrada no ser humano, que utiliza um conjunto de ferramentas para integrar as necessidades das pessoas, as possibilidades da tecnologia e os requisitos para o sucesso do negócio.³
- **MVP (Minimum Viable Product):** Produto Mínimo Viável. A versão de um novo produto que permite a uma equipe coletar a quantidade máxima de aprendizado validado sobre os clientes com o mínimo de esforço.
- **PMBOK® (Project Management Body of Knowledge):** Um guia de boas práticas, terminologias e diretrizes para o gerenciamento de projetos, publicado pelo Project Management Institute (PMI).⁶
- **Product Backlog:** Uma lista ordenada e emergente de tudo o que é conhecido ser necessário no produto. É a única fonte de trabalho a ser realizada pelo Time Scrum.¹⁶
- **Project Model Canvas (PMC):** Uma ferramenta visual e colaborativa para criar o plano de um projeto de forma simplificada, respondendo a perguntas fundamentais em 13 blocos.⁴
- **Scrum:** Um framework ágil dentro do qual as pessoas podem abordar

problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos de maior valor possível.⁵

- **Sprint:** Um período de tempo fixo (geralmente de 1 a 4 semanas) durante o qual um incremento de produto "Pronto", utilizável e potencialmente liberável é criado.⁵
- **User Story:** Uma descrição curta e simples de uma funcionalidade contada da perspectiva da pessoa que deseja o novo recurso, geralmente um usuário ou cliente do sistema.

Referências citadas

1. TII 5 - Projeto Integrador I - Ementa
2. APRENDIZAGEM BASEADA EM PROBLEMAS ... - SciELO Brasil, acessado em outubro 16, 2025, <https://www.scielo.br/j/edur/a/qWyNpvw94bycsjL9Qw6pZxC/>
3. Design Thinking - Hitachi Solutions, acessado em outubro 16, 2025, <https://www.hitachi-solutions.pt/design-thinking/>
4. Project Model Canvas, acessado em outubro 16, 2025, <https://assets-sam.mkt.dynamics.com/91224084-773a-496b-b0ba-c1eade9d34b0/digitalassets/docs/7dc7d3ba-7c64-ef11-a670-002248dec411?ts=63860364045000000>
5. Scrum para iniciantes: O guia completo para você começar ..., acessado em outubro 16, 2025, <https://massimus.com/scrum-para-iniciantes-o-guia-completo-para-voce-comecar/>
6. White paper: Guia PMBOK® - 4ª Edição - TRT7, acessado em outubro 16, 2025, https://www.trt7.jus.br/files/institucional/governanca_ti/processos/processo-trt7/TRT Compartilhado/guidances/whitepapers/PMBOK_Guia_623B7F29.html
7. Aprendizagem Baseada em Problemas aplicada à Programação de ..., acessado em outubro 16, 2025, <https://seer.ufrgs.br/index.php/renote/article/download/110298/60053/455510>
8. Como aplicar o design thinking na gestão de TI - Supero, acessado em outubro 16, 2025, <https://www.supero.com.br/como-aplicar-o-design-thinking-na-gestao-de-ti/>
9. Design Thinking na TI: benefícios e como implementar - Escola Superior de Redes, acessado em outubro 16, 2025, <https://esr.rnp.br/metodos-ageis-e-inovacao/design-thinking-na-ti/>
10. Detalhe do registro: Um estudo sobre design thinking ... - ReP USP, acessado em outubro 16, 2025, <https://repositorio.usp.br/item/003021879>

11. Elicitação de requisitos de Sistemas e Design Thinking no âmbito público - Um estudo de caso na <
12. Project Model Canvas: O que é e como aplicar em projetos. - Euax, acessado em outubro 16, 2025, <https://www.euax.com.br/2024/11/project-model-canvas/>
13. A Diferença entre o Plano de Projeto e o Project Model Canvas, acessado em outubro 16, 2025, <https://www.projectbuilder.com.br/blog/a-diferenca-entre-o-plano-de-projeto-e-o-project-model-canvas/>
14. Documento de Projeto de Software - Ceará Transparente, acessado em outubro 16, 2025, <https://cearatransparente.ce.gov.br/attachments/679ba76d0fbe6ea4331cf0477fe57f0bf5c582e8/store/bb0f164e41ea26f55847ec609d59d76e3483d558aeec7fc7876ada97b3d4/Documenta%C3%A7%C3%A3oCearaInterativo.pdf>
15. Quais os benefícios da Gestão de Projetos e da estruturação de um ..., acessado em outubro 16, 2025, <https://netproject.com.br/blog/quais-os-beneficios-da-gestao-de-projetos-e-da-estruturação-de-um-pmo/>
16. Guia SCRUM, acessado em outubro 16, 2025, <https://www.iffarroupilha.edu.br/component/k2/attachments/download/5702/87f24f4c8f7ed3238ef6f4a71b010a14>
17. Framework Agile Scrum: Guia Completo para Transformação Organizacional, acessado em outubro 16, 2025, <https://scrum.pt/growth-mindset-blog/fundamentos-do-scrum/framework-agile-scrum-guia-completo-para-transformacao-organizacional/>
18. Guia do Scrum™, acessado em outubro 16, 2025, <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>
19. Como aplicar os 12 Princípios PMBOK® 7 no gerenciamento de projetos na prática - PMP® 2022 - Aula 03 - YouTube, acessado em outubro 16, 2025, <https://www.youtube.com/watch?v=avqOTUgfrQQ>
20. O que é o Design Thinking? | IBM, acessado em outubro 16, 2025, <https://www.ibm.com/br-pt/think/topics/design-thinking>
21. PMBOK: Conceito, importância, vantagens e implementação - TOTVS, acessado em outubro 16, 2025, <https://www.totvs.com/blog/negocios/pmbok/>
22. Scrum: Introdução para iniciantes - DEV Community, acessado em outubro 16, 2025, https://dev.to/hivecommunity/_/scrum-uma-introducao-para-iniciantes-3bkh
23. Guia do Scrum: conheça as cerimônias e artefatos | monday.com Blog, acessado em outubro 16, 2025, <https://monday.com/blog/pt/desenvolvimento/guia-scrum-cerimonias-e-artefatos/>