

Corrigge

# VISÃO DO PRODUTO E PROJETO

Versão 1.0

## Histórico de Revisão

Data	Versão	Descrição	Autor
05/04/2025	0.1	Completando seções 1 e 2	Otávio
08/04/2025	0.2	Corrigindo erros apontados na versão 0.1	Otávio
21/04/2025	1.0	Adicionando seções 4,5 e 6	Todos

# 1 Cenário Atual do Cliente e do Negócio

## 1.1 Introdução ao Negócio e Contexto

A Guia do PAS é uma empresa especializada na preparação de estudantes para o Programa de Avaliação Seriada (PAS) da Universidade de Brasília (UnB), um dos processos seletivos mais concorridos do país. Atuando no setor educacional, a empresa oferece conteúdos direcionados, simulados, correções comentadas, aulas ao vivo e materiais exclusivos, com foco na aprovação dos alunos. A empresa, nos últimos anos, também tem expandido seu mercado para o abranger o atendimento à outras instituições de ensino, oferecendo a aplicação de simulados, listas de exercícios personalizadas e controle de desempenho dos alunos. Além disso, a empresa tem buscado expandir sua atuação para outros processos seletivos além do PAS, visando aumentar sua base de clientes e diversificar seus serviços.

A empresa já tentou desenvolver uma solução anteriormente, também chamada Corrigge, mas esta primeira versão apresentou diversos problemas técnicos que comprometeram sua utilização efetiva. O sistema anterior, que era uma aplicação web, necessitava de manutenção constante e apresentava várias limitações operacionais: o servidor precisava estar sempre ativo em um computador específico com boa capacidade de processamento, onde um script Python precisava ser executado manualmente para permitir que outros usuários acessassem o sistema. Além disso, o sistema apresentava instabilidades frequentes, com múltiplas ocorrências de crashes tanto no servidor quanto na aplicação, sem mecanismos de auto-recuperação. A necessidade de ajustes manuais de parâmetros pelos usuários finais também se mostrou um obstáculo significativo para a adoção do sistema. A funcionalidade do sistema anterior era limitada apenas à identificação das respostas dos alunos, sem oferecer recursos adicionais como análise de desempenho, geração de relatórios ou personalização de gabaritos.

## 1.2 Identificação da Oportunidade ou Problema

Com o crescimento da base de usuários e o aumento no número de clientes B2B buscando a realização de simulados em suas escolas, percebeu-se algumas dificuldades na obtenção das respostas dos alunos, na inserção desses dados no banco de dados da empresa e a visualização desses dados de forma efetiva pelos coordenadores das escolas parceiras. Após a aplicação do simulado é necessário registrar as respostas de cada aluno e associar esses dados com a matrícula do aluno, utilizando desses dados em uma planilha .csv para ser cadastrada no sistema da empresa. Mesmo fazendo uso de serviços de outras empresas para realizar essa identificação dos gabaritos, essas soluções eram demoradas e muito custosas (demorando mais de um mês para receber os resultados e custando em média R\$1500.00 por simulado), não tendo sido possível encontrar uma solução rápida, de fácil uso e barata para a realização de grandes simulados ou de simulados em várias escolas em um curto período de tempo.

Diante dessas experiências anteriores e da necessidade de uma solução mais robusta, a empresa busca o desenvolvimento de uma nova versão do sistema como uma aplicação desktop, que permita a correção automática dos gabaritos de forma eficiente e rápida, desejando também oferecer esse serviço para outras instituições de ensino não associadas à empresa.



Figura 1: Diagrama de Ishikawa (Fishbone) com os principais desafios do projeto

### 1.3 Desafios do Projeto

Os principais desafios do projeto são: A utilização e parametrização de ferramentas de CV (Visão computacional) para a identificação das respostas dos alunos, a não padronização dos gabaritos preenchidos pelos alunos, a não familiaridade de alguns membros da equipe com as ferramentas a serem utilizadas, a criação de um design amigável, eficiente e intuitivo para a utilização pelos membros da empresa, e o empacotamento das ferramentas necessárias para integrar o CV em uma aplicação multiplataforma em flutter.

### 1.4 Segmentação de Clientes

O público alvo atual da empresa são alunos do ensino médio e alunos de cursinhos que buscam uma preparação estratégica e eficiente para o PAS, além de instituições de ensino que desejam ter uma preparação mais específica para o PAS e simulados personalizados para esta preparação.

O perfil dos alunos é de jovens entre 15 e 20 anos, que buscam uma preparação diferenciada para o PAS, com foco em resultados e aprovação. Eles valorizam a qualidade do conteúdo, a eficiência dos simulados e a possibilidade de acompanhar seu desempenho ao longo do tempo.

O perfil dos coordenadores das instituições de ensino são profissionais com formação superior, que atuam na área de educação e buscam soluções tecnológicas para otimizar a gestão acadêmica e melhorar a experiência dos alunos. Eles valorizam a eficiência e a facilidade de uso das ferramentas que utilizam.

## 2 Solução Proposta

### 2.1 Objetivos do Produto

O objetivo principal do produto é otimizar o processo de correção de gabaritos da Guia do PAS, reduzindo o tempo de processamento de 1 mês para menos de 24 horas, diminuindo o custo por correção de R\$5,00 para menos de R\$1,00, e aumentando a precisão da identificação das respostas para mais de 99%. Além disso, o sistema deve permitir a análise automática do desempenho individual e coletivo dos alunos, gerando relatórios detalhados que incluem estatísticas por questão, comparações entre turmas e identificação de padrões de erro. O produto também visa expandir a base de clientes da empresa, permitindo a oferta do serviço de correção para outras instituições de ensino, com meta de atingir pelo menos 10 novas escolas parceiras no primeiro ano de operação. Por fim, o sistema deve resolver os problemas técnicos encontrados na versão anterior, como instabilidades, necessidade de manutenção constante e limitações operacionais, oferecendo uma solução robusta e multiplataforma que não dependa de servidores dedicados.

### 2.2 Características da Solução

O produto será um aplicativo desktop multiplataforma (Windows, Linux e MacOS) que deve ser capaz de processar imagens de gabaritos preenchidos pelos alunos, identificar as respostas dadas pelos alunos em cada questão, identificar a matrícula do aluno, comparar as respostas com o gabarito correto, gerar relatórios detalhados (com o desempenho do aluno — incluindo notas, acertos e erros —, comparações entre alunos, comparações entre grupos de alunos e dados estatísticos de cada questão). O produto também deve ser capaz de gerar templates de gabaritos e permitir a personalização dos gabaritos de acordo com as necessidades de cada instituição de ensino. Além disso, o produto também deve permitir a exportação dos dados e relatórios para arquivos .csv e .pdf, para facilitar a integração com outros sistemas e a análise dos dados. A escolha por uma aplicação desktop também visa facilitar o processo de desenvolvimento, permitindo que a equipe de desenvolvimento da empresa adicione mais funcionalidades ao longo do tempo de forma ágil e eficiente.

### 2.3 Tecnologias a Serem Utilizadas

As tecnologias a serem utilizadas para o desenvolvimento do produto são:

- Frontend: Dart + Flutter
- Identificação de imagem local: Python + OpenCV
- Backend: Typescript + Express.js
- Banco de dados: Supabase (PostgreSQL)
- Hospedagem: Azure + VM Linux

### 2.4 Pesquisa de Mercado e Análise Competitiva

- **RemarkOffice:** Possui um preço muito alto (\$1,195.00), não oferece a possibilidade de personalização dos gabaritos e não possui precificação por gabarito analisado (deixando o serviço mais barato para pequenas empresas).

- **Graden:** É gratuito mas pouco conhecido. Também não oferece estatísticas de desempenho do aluno, apenas a correção do gabarito. Também não oferece a possibilidade de personalização dos gabaritos.
- **Prova Fácil:** Custa \$0.15 por correção, não oferece estatísticas de desempenho do aluno, apenas a correção do gabarito. Também não oferece a possibilidade de personalização dos gabaritos.

## 2.5 Análise de Viabilidade

A viabilidade técnica do projeto é alta, uma vez que a equipe possui experiência em desenvolvimento de plataformas similares e familiaridade com as tecnologias propostas como **Flutter**, **PostgreSQL**, **Typescript** e a com os **serviços de cloud** a serem utilizados. O prazo é de 3 meses para o desenvolvimento da solução, o que é viável considerando a complexidade do projeto e a experiência da equipe. O custo estimado para o desenvolvimento é de R\$1,000.00 para eventuais pagamentos de serviços como o banco de dados e hospedagem dos servidores backend durante o desenvolvimento, o que é viável considerando a demanda do mercado e o preço cobrado pelos concorrentes. O retorno financeiro inicial esperado é de R\$1,000.00 por mês (uma estimativa baixa considerando apenas 2000 correções por mês à R\$0.50 por correção, o que é facilmente atingível dado o crescimento atual da empresa no mercado de Brasília e a possibilidade de oferecimento desse serviço para diversas empresas de educação do país), considerando a venda do serviço para outras instituições de ensino e o crescimento da confiabilidade na empresa, o que permite o oferecimento do serviço para várias instituições de Brasília.

## 2.6 Impacto da Solução

A solução proposta terá um impacto significativo na eficiência operacional da empresa, reduzindo o tempo (de um mês para algumas horas) e os custos associados (de R\$5,00 para menos de R\$1,00 por correção) à correção de gabaritos. Além disso, a solução permitirá a venda desse serviço para outras instituições de ensino, aumentando a receita da empresa em pelo menos R\$12.000,00 por ano e expandindo sua atuação no mercado educacional, expandindo a influência da empresa no mercado de preparação para o PAS. A solução também melhorará a experiência do usuário, proporcionando resultados mais rápidos (análise de 100 gabaritos por minuto) e precisos, além de relatórios detalhados sobre o desempenho dos alunos.

# 3 Estratégias de Engenharia de Software

## 3.1 Estratégia Priorizada

- **Abordagem de Desenvolvimento de Software:** Ágil com elementos dirigidos por plano.
- **Ciclo de vida:** Híbrido (iterativo e incremental).
- **Processo de Engenharia de Software:** OpenUP.

O OpenUP será utilizado, pois é um processo ágil e leve, adequado para projetos de médio porte com necessidades de validação rápida e evoluções constantes.

As quatro fases do OpenUP serão seguidas:

- **Concepção:** Definição dos objetivos do projeto, levantamento dos requisitos iniciais, entendimento do problema de correção de gabaritos, escolha das tecnologias (Flutter, OpenCV, Typescript, Supabase) e planejamento inicial de prazos e custos.
- **Elaboração:** Refinamento dos requisitos (como detalhamento das funcionalidades de correção, exportação e geração de relatórios), desenvolvimento de protótipos iniciais de reconhecimento de gabaritos, e avaliação dos principais riscos (como o reconhecimento de imagens e integração entre as tecnologias).
- **Construção:** Desenvolvimento iterativo da aplicação desktop, integração contínua entre frontend, backend e sistemas de visão computacional, testes de funcionalidades como reconhecimento automático e geração de relatórios, além de ajustes baseados no feedback interno.
- **Transição:** Testes finais de desempenho e robustez (ex: correção de 100 gabaritos por minuto), implantação para o ambiente de produção, produção de documentação de uso para a equipe comercial e suporte inicial aos clientes externos.

O cronograma de execução do projeto será alinhado às entregas previstas em cada uma dessas fases, respeitando o prazo de 3 meses para conclusão do produto mínimo viável (MVP).

### 3.2 Quadro Comparativo

Característica	Unified Process (UP)	Open Unified Process (OpenUP)
<b>Objetivo</b>	Metodologia robusta com forte foco em documentação e controle	Processo leve e ágil com foco em colaboração e entregas contínuas
<b>Foco no Processo</b>	Estruturado e formal, com controle rigoroso	Flexível e centrado na produtividade da equipe
<b>Complexidade</b>	Elevada, ideal para projetos críticos e de grande porte	Reduzida, ideal para projetos de médio porte que exigem agilidade
<b>Fases</b>	Concepção, Elaboração, Construção, Transição (com ampla documentação)	Concepção, Elaboração, Construção, Transição (com entregas enxutas)
<b>Disciplinas</b>	Abrange várias disciplinas como Modelagem, Design, Testes, Gerência de Projeto	Enfoque em Requisitos, Implementação, Testes e Gerência de Projeto
<b>Documentação</b>	Detalhada e exigida em cada fase	Essencial apenas para manter a produtividade da equipe
<b>Iteratividade</b>	Iterativo e incremental, com processos mais pesados	Iterativo e incremental, com ciclos curtos e adaptáveis
<b>Gestão de Riscos</b>	Formal e concentrada na fase de Elaboração	Contínua e integrada de forma leve
<b>Flexibilidade</b>	Menor adaptabilidade a mudanças rápidas	Alta flexibilidade, facilitando ajustes frequentes
<b>Colaboração</b>	Papéis bem definidos e estrutura hierárquica	Equipes multifuncionais, com papéis flexíveis e foco na colaboração
<b>Recomendado para</b>	Projetos grandes, críticos, com alta demanda por rastreabilidade e validações formais	Projetos médios ou pequenos, com necessidade de agilidade – como no projeto Corrigge

### 3.3 Justificativa

O projeto da nova solução **Corrigge** para a Guia do PAS demanda um processo ágil, adaptável e eficiente. O **OpenUP** se mostra a escolha ideal por:

- Permitir a adaptação rápida a mudanças no escopo, como ajustes em funcionalidades (ex: novos tipos de gabaritos ou estatísticas).
- Exigir menos documentação formal, favorecendo o foco em entregas práticas e ágeis, especialmente importante para uma equipe enxuta.

- Ser mais leve que o UP tradicional, o que reduz a carga de burocracia, otimiza o tempo e diminui o custo do desenvolvimento — aspectos cruciais para um projeto com orçamento de aproximadamente R\$ 1.000,00.
- Permitir iterações rápidas e entregas frequentes, essenciais para testar rapidamente o reconhecimento de imagens, relatórios e exportações, validando a qualidade da solução antes do lançamento comercial.
- Promover a colaboração ativa da equipe de desenvolvimento com os stakeholders (Guia do PAS e instituições parceiras), para garantir que o produto atenda às expectativas tanto para uso interno quanto para venda externa.

Em resumo, o **OpenUP** encaixa-se perfeitamente nas necessidades do projeto Corrigge: é leve, iterativo, adaptável, reduz riscos e acelera o retorno sobre o investimento.

## 4 Cronograma e Entregas

Fase	Sprint	Início	Fim	Objetivos e Entregas Esperadas
<b>Concepção</b>	Sprint 0	21/04/2025	08/05/2025	<ul style="list-style-type: none"> <li>- Planejamento geral do projeto</li> <li>- Definição do escopo</li> <li>- Organização da documentação inicial</li> <li>- Definição das tecnologias principais</li> <li>- Definição do MVP do Corrigge</li> <li>- Definição do DoR e DoD</li> <li>- Definição de User Stories</li> <li>- Declaração de requisitos funcionais e não funcionais</li> <li>- Entrega da Unidade 1 em 22/04/2025 (terça-feira)</li> </ul>
<b>Elaboração</b>	Sprint 1	09/05/2025	22/05/2025	<ul style="list-style-type: none"> <li>- Capacitação técnica (Flutter, OpenCV, Supabase)</li> <li>- Protótipo no Figma</li> <li>- Definição da Arquitetura</li> <li>- Análise de riscos técnicos</li> <li>- Configuração do banco de dados (Supabase)</li> <li>- Ambiente de desenvolvimento inicial</li> </ul>



Fase	Sprint	Início	Fim	Objetivos e Entregas Esperadas
<b>Construção</b>	Sprint 2	23/05/2025	05/06/2025	<ul style="list-style-type: none"> <li>- Implementação inicial do frontend desktop, login e criação de conta</li> <li>- Implementação inicial do reconhecimento de questões</li> <li>- Implementação inicial da identificação das questões com OpenCV</li> <li>- Integração de frontend com a biblioteca python local</li> <li>- Criação dos endpoints backend para criação de conta</li> <li>- Personalização de gabaritos</li> <li>- Entrega da Unidade 2 em 27/05/2025 (terça-feira)</li> </ul>
	Sprint 3	06/06/2025	19/06/2025	<ul style="list-style-type: none"> <li>- Continuação da implementação do design no frontend</li> <li>- Testes de reconhecimento e comparação</li> <li>- Testes de criação de gabaritos</li> <li>- Melhorias de UI/UX</li> <li>- Ajustes baseados em feedback do cliente</li> <li>- Implementação inicial da geração de dashboards</li> </ul>
	Sprint 4	20/06/2025	03/07/2025	<ul style="list-style-type: none"> <li>- Geração de relatórios simples (acertos e erros) para cada aluno</li> <li>- Implementação inicial de pagamentos com Stripe</li> <li>- Exportação de dados em .csv</li> <li>- Exportação de relatórios em .pdf</li> <li>- Validação das exportações</li> <li>- Entrega da Unidade 3 em 24/06/2025 (terça-feira)</li> </ul>
<b>Transição</b>	Sprint 5	04/07/2025	17/07/2025	<ul style="list-style-type: none"> <li>- Testes finais de performance (100 gabaritos/min)</li> <li>- Testes de estabilidade e recuperação</li> <li>- Melhorias de UI/UX</li> <li>- Ajustes baseados em feedback interno</li> <li>- Hospedagem do backend (Azure)</li> <li>- Walkthrough com o cliente (Guia do PAS)</li> <li>- Encerramento do projeto</li> <li>- Entrega da Unidade 4 em 15/07/2025 (terça-feira)</li> </ul>

#### Datas específicas de entrega:

- Unidade 1 → 22/04/2025 (terça-feira)

- Unidade 2 → 27/05/2025 (terça-feira)
- Unidade 3 → 24/06/2025 (terça-feira)
- Unidade 4 → 15/07/2025 (terça-feira)

## 5 Interação entre Equipe e Cliente

### 5.1 Composição da Equipe

Papel	Descrição	Responsável	Participantes
Gerente de Projeto	Responsável por coordenar o projeto, manter a comunicação entre a equipe e o cliente, e garantir o cumprimento dos prazos e entregas.	Otavio Maya	-
Desenvolvedor FrontEnd	Cuida do desenvolvimento da interface do usuário, garantindo o design e a implementação das funcionalidades no lado do cliente.	Nathan Benigno	Otavio Maya
Desenvolvedor Backend	Desenvolve a lógica de negócio, realiza a integração com banco de dados e serviços externos por meio de APIs.	Esdras de Sousa	Atyrson Souto, Pedro Victor
Analista de Requisitos	Responsável por levantar, documentar e gerenciar requisitos funcionais e não funcionais, garantindo o alinhamento com os objetivos do projeto.	Eduardo Ferreira	Nathan Benigno, Esdras de Sousa, Otavio
Analista de QA	Cria pipelines de integração e entrega contínua, desenvolve testes automatizados e realiza testes funcionais para assegurar a qualidade do produto.	Marcelo Adrian	-

Tabela 3: Composição da Equipe e Responsabilidades

### 5.2 Comunicação

#### 5.2.1 Ferramentas Utilizadas

- **WhatsApp:** será utilizado para comunicações rápidas e diárias entre os membros da equipe e também com o cliente, sempre que necessário.
- **Google Meet:** usado para reuniões mais formais de sprint e também para os encontros de validação com o cliente.

- **GitHub Projects (Kanban):** ferramenta usada para o acompanhamento visual das tarefas da equipe durante o desenvolvimento.

### 5.2.2 Tipos e Frequência das Reuniões

- **Revisão de Sprint** (a cada duas semanas):
  - Toda segunda-feira, a equipe fará uma reunião para revisar o que foi entregue na sprint anterior, identificando o que funcionou bem e o que pode ser melhorado.
- **Planejamento de Sprint** (a cada duas semanas):
  - Logo após a revisão, será feito o planejamento da nova sprint, com base no cronograma e nas prioridades do projeto.
- **Validação com o Cliente:**
  - As reuniões com o cliente ocorrerão **exclusivamente ao final das sprints**, alinhando-se à preferência do stakeholder por acompanhar apenas os fechamentos.
  - O objetivo dessas validações é apresentar o progresso alcançado, receber feedback e realizar ajustes, se necessário.
  - As reuniões acontecerão **preferencialmente às segundas ou sextas-feiras**, e não será exigida a presença de toda a equipe — apenas os membros responsáveis pelas entregas em validação participarão para conduzir a apresentação e esclarecer eventuais dúvidas.

## 5.3 Frequência de Contato com o Cliente

- **Validações Principais:**
  - Previstas para ocorrer apenas ao final das sprints, momento destinado à validação do progresso do projeto.
  - Como o stakeholder possui conhecimento em tecnologia, optou por não realizar reuniões semanais, considerando mais eficiente acompanhar o desenvolvimento apenas nos fechamentos de sprint.
- **Comunicação pelo WhatsApp:**
  - Em situações de dúvidas rápidas ou discussões informais ao longo do projeto, será utilizada essa ferramenta pela sua praticidade e acessibilidade.

## 5.4 Processo de Validação

O processo de validação das funcionalidades do sistema Corigge será dividido em três etapas principais:

### 1. Definition of Ready (DoR)

- Antes de iniciar o desenvolvimento de qualquer funcionalidade, será verificado se:

- Os requisitos estão bem definidos e alinhados com os objetivos do projeto
- Existe documentação adequada, incluindo critérios técnicos e funcionais
- Os critérios de aceitação foram estabelecidos e validados pela equipe

## 2. Definition of Done (DoD)

- A funcionalidade só será considerada concluída se:
  - Passar por testes unitários, de integração e validações internas
  - Atender aos critérios de aceitação previamente definidos
  - Estar aprovada visual e funcionalmente pela equipe, garantindo qualidade, estabilidade e alinhamento com os objetivos do projeto

## 3. Teste de Aceitação pelo Cliente

- Após a validação interna de funcionalidades como reconhecimento automático de respostas, identificação da matrícula e exportação de relatórios, o sistema será entregue ao cliente para testes finais.
- A validação será feita com base nos critérios definidos durante o planejamento (DoR) e ocorrerá nos fechamentos de sprint, conforme acordado com o stakeholder.

# 6 Lições Aprendidas

## 6.1 Unidade 1

### 6.1.1 Organização dos Requisitos

**Desafio:** Desenvolver uma estrutura clara e hierárquica dos requisitos do sistema, garantindo uma priorização eficiente que atenda às necessidades do cliente e aos objetivos do projeto.

**Ação de melhoria:** Implementar uma ferramenta especializada em gerenciamento de requisitos para facilitar o rastreamento, versionamento e priorização das funcionalidades.

### 6.1.2 Escolha do Processo de Engenharia de Software

**Desafio:** Selecionar uma metodologia de desenvolvimento que equilibre a necessidade de flexibilidade para adaptações com a estrutura necessária para manter o projeto organizado, considerando as restrições de disponibilidade do cliente.

**Ação de melhoria:** Realizar uma análise prévia das necessidades do projeto, do perfil do cliente e da maturidade da equipe para escolher a metodologia mais adequada, aproveitando as experiências anteriores dos membros.

### 6.1.3 Organização do Tempo da Equipe

**Desafio:** Gerenciar eficientemente a disponibilidade dos membros da equipe para garantir a realização das reuniões de alinhamento e o cumprimento dos prazos de entrega, considerando as diferentes agendas e responsabilidades individuais.

**Ação de melhoria:** Implementar um sistema de gestão de tempo baseado em calendário compartilhado com horários fixos para reuniões, estabelecer marcos claros para

entregas e adotar ferramentas de gestão de tarefas para monitoramento assíncrono do progresso.

## 6.2 Unidade 2

## 6.3 Unidade 2

## 6.4 Unidade 3

## 6.5 Unidade 4

# 7 Referências Bibliográficas

1. SILVA, Luciane. OpenUp: um processo integrado e ágil. Disponível em: <https://medium.com/@LucianeS/openup-um-processo-integrado-e-agil-a4400c17ce62>. Acesso em: 11 nov. 2024.
2. CATOLICA. METODOLOGIAS ÁGEIS DE DESENVOLVIMENTO: OPENUP, FDD, DSDM E LEAN. Disponível em: [https://conteudo.catolica.edu.br/conteudos/unileste\\_cursos/disciplinas/nucleo\\_formacao\\_geral/Gestao\\_de\\_projetos\\_e\\_metodos\\_ageis/tema\\_03/index.html](https://conteudo.catolica.edu.br/conteudos/unileste_cursos/disciplinas/nucleo_formacao_geral/Gestao_de_projetos_e_metodos_ageis/tema_03/index.html). Acesso em: 11 nov. 2024.
3. EDUCATIVE.IO. What is a unified process model. Disponível em: <https://www.educative.io/answers/what-is-a-unified-process-model>. Acesso em: 11 nov. 2024.
4. EDEKI, Charles. Agile Unified Process. Disponível em: <https://interhad.nied.unicamp.br/courses/roberto-pereira/ci163-projeto-de-software-ufpr-1/agenda/auppaper.pdf>. Acesso em: 11 nov. 2024.
5. TREINAWEB. O que é RUP (Rational Unified Process). Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-rup-rational-unified-process>. Acesso em: 11 nov. 2024.
6. LEFFINGWELL, D.; WIDRIG, D. Managing Software Requirements: A Use Case Approach. 2. ed. Addison-Wesley, 2003.
7. AMBLER, S. Agile Modeling. Wiley, 2002.
8. LEFFINGWELL, Dean. Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise. 1. ed. Addison-Wesley Professional, 2011.
9. PATTON, Jeff; ECONOMY, Peter. User story mapping: discover the whole story, build the right product. O'Reilly Media, Inc., 2014.
10. SCHWABER, Ken; SUTHERLAND, Jeff. O Guia do Scrum: O Guia Definitivo para o Scrum: As Regras do Jogo. Novembro de 2020.