**UNIVERSIDADE PAULISTA**

**FERNANDA MAYUMI SATO**

**JULIA CAROLINA DE BIAZI CARDOSO**

**PAMELA BRANDÃO OLIVAL**

**VELORA**

Aplicação de Gerenciamento Ágil

**SÃO PAULO**

**2024**

**FERNANDA MAYUMI SATO**

**JULIA CAROLINA DE BIAZI CARDOSO**

**PAMELA BRANDÃO OLIVAL**

**VELORA**

Aplicação de Gerenciamento Ágil

Trabalho de Conclusão de Curso para obtenção do título de graduação em Ciência da Computação apresentado à Universidade Paulista – UNIP.

Orientador: Prof. Me. Lauro Henrique de Castro Tomiatti

**SÃO PAULO**

**2024**

CIP - Catalogação na Publicação

VELORA - Aplicação de Gerenciamento Ágil / FERNANDA SATO, JULIA CARDOSO, PAMELA OLIVAL. - 2024.

95 f. : il. color

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto de Ciência Exatas e Tecnologia da Universidade Paulista, SÃO PAULO, 2024.

Área de Concentração: ..

Orientador: Prof. Me. Lauro Tomiatti.

1. Metodologias Ágeis. 2. Desenvolvimento de Software. I. FERNANDA SATO. II. CARDOSO, JULIA . III. OLIVAL, PAMELA . IV. Tomiatti , Lauro (orientador). V. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da Universidade Paulista com os dados fornecidos pelo(a) autor(a).

**FERNANDA MAYUMI SATO**

**JULIA CAROLINA DE BIAZI CARDOSO**

**PAMELA BRANDÃO OLIVAL**

**VELORA**

Aplicação de Gerenciamento Ágil

Trabalho de Conclusão de Curso para obtenção do título de graduação em Ciência da Computação apresentado à Universidade Paulista – UNIP.

Orientador: Prof. Me. Lauro Henrique de Castro Tomiatti

Aprovado em:

BANCA EXAMINADORA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_/\_\_\_\_\_\_\_\_

Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Universidade Paulista - UNIP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_/\_\_\_\_\_\_\_\_

Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Universidade Paulista - UNIP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_/\_\_\_\_\_\_\_\_

Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Universidade Paulista - UNIP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_/\_\_\_\_\_\_\_\_

**AGRADECIMENTOS**

Este trabalho é o resultado do esforço conjunto e, como grupo, gostaríamos de expressar nossa gratidão a todos que nos apoiaram ao longo desta jornada.

Primeiramente, agradecemos aos nossos familiares, que sempre nos ofereceram apoio, compreensão e incentivo, mesmo nos momentos mais desafiadores. Sem o suporte deles, esta conquista não teria sido possível. Também queremos expressar nossa gratidão ao nosso orientador, professor Lauro, pela orientação constante, paciência e valiosas contribuições durante o desenvolvimento deste projeto. Suas sugestões foram fundamentais para a evolução do nosso trabalho.

Agradecemos, ainda, a todos os amigos e colegas que, de diferentes maneiras, estiveram ao nosso lado, oferecendo apoio e motivação. A troca de experiências e conselhos ao longo do curso foi essencial para o nosso aprendizado e crescimento.

Por fim, nosso sincero agradecimento aos professores da Universidade, que compartilharam seu conhecimento e nos guiaram em nossa formação acadêmica e profissional. Cada um de vocês fez parte dessa conquista.

**EPÍGRAFE**

*“Nunca estou realmente satisfeita quanto a entender alguma coisa; porque, até onde entendo, a minha compreensão só pode ser uma fração infinitesimal de tudo o que eu quero compreender”.*

*(Ada Lovelace)*

**RESUMO**

O seguinte projeto tem como objetivo a criação de uma ferramenta que auxilia o uso de metodologias ágeis no ambiente coorporativo. A solução vem da necessidade de uma necessidade de poder gerenciar essas metodologias de uma forma facilitada e organizada.

O ágil é uma forma de acelerar entregas de projetos, tendo em vista o foco com o cliente final. Ele ajuda a coordenar melhor equipes, trazendo um método eficaz de práticas e aumentando a qualidade do produto.

Embora amplamente adotadas na área de tecnologia para a criação de projetos eficazes, rápidos e bem-organizados, as metodologias ágeis não são exclusivas para equipes técnicas, podendo ser aplicadas em diversos tipos de projetos.

A ferramenta resultante deste projeto visa possibilitar que empresas de qualquer porte incorporem a metodologia ágil em suas práticas organizacionais, promovendo uma cultura de adaptação e inovação contínua.

Palavras-Chave: Metodologias Ágeis; Tecnologia; Organização.

**ABSTRACT**

The following project aims to create a tool that helps the use of agile methodologies in the corporate environment. The solution comes from the need to be able to manage these methodologies in an easy and organized way.

Agile is a way of accelerating project deliveries, focusing on the end customer. It helps to better coordinate teams, bringing an effective method of practice and increasing the quality of the final product.

Although widely adopted in the technology area for creating effective, fast and well-organized projects, agile methodologies are not exclusive to technical teams and can be applied to different types of projects.

The tool resulting from this project aims to enable companies of any size to incorporate the agile methodology into their organizational practices, promoting a culture of adaptation and continuous innovation.

Keywords: Agile Methodologies; Technology; Organization.

**LISTA DE FIGURAS**

[Figura 1 - Ciclo do Scrum 23](#_Toc183421921)

[Figura 2 - 5 Princípios do Lean 26](#_Toc183421922)

[Figura 3 - Kanban 31](#_Toc183421923)

[Figura 4 - PDCA 34](#_Toc183421924)

[Figura 5 - Tela Fazer do Kanban durante o Desenvolvimento do Projeto 47](#_Toc183421925)

[Figura 6 - Tela Em Progresso do Kanban durante o desenvolvimento do projeto 48](#_Toc183421926)

[Figura 7 - Tela Pronto do Kanban durante o desenvolvimento do projeto 48](#_Toc183421927)

[Figura 8 - Tela Pronto do Kanban após terminar o desenvolvimento do projeto 49](#_Toc183421928)

[Figura 9 - Caso de Uso 54](#_Toc183421929)

[Figura 10 - Diagrama de Classe 61](#_Toc183421930)

[Figura 11 - Diagrama de Entidade e Relacionamento 62](#_Toc183421931)

[Figura 12 - Modelo Lógico 62](#_Toc183421932)

[Figura 13 - Protótipo: Página de Contratação 63](#_Toc183421933)

[Figura 14 - Protótipo: Login 64](#_Toc183421934)

[Figura 15 - Protótipo: Solicitação de Cadastro 64](#_Toc183421935)

[Figura 16 - Protótipo: Iniciando na Aplicação 65](#_Toc183421936)

[Figura 17 - Protótipo: Meus Projetos 65](#_Toc183421937)

[Figura 18 - Protótipo: Meus PDCAs 66](#_Toc183421938)

[Figura 19 - Protótipo: Criando PDCA 66](#_Toc183421939)

[Figura 20 - Protótipo: Kanban Pessoal 67](#_Toc183421940)

[Figura 21 - Protótipo: Criando Tarefa do Kanban Pessoal 67](#_Toc183421941)

[Figura 22 - Protótipo: Tela Projetos 68](#_Toc183421942)

[Figura 23 - Protótipo: Kanban de Equipe 68](#_Toc183421943)

[Figura 24 - Protótipo: Criando Tarefa no Kanban de Equipe 69](#_Toc183421944)

[Figura 25 - Protótipo: Kanban de Equipe - Documentos 69](#_Toc183421945)

[Figura 26 - Protótipo: Kanban de Equipe - Métricas 70](#_Toc183421946)

[Figura 27 - Protótipo: Kanban de Equipe - Backlog 70](#_Toc183421947)

[Figura 28 - Protótipo: Perfil 71](#_Toc183421948)

[Figura 29 - Protótipo: Dicas Ágeis 71](#_Toc183421949)

[Figura 30 - Logo da Aplicação: Velora 72](#_Toc183421950)

**LISTA DE TABELAS**

[Tabela 1 - Requisitos Funcionais 47](#_Toc180593512)

[Tabela 2 - Requisitos Não Funcionais 49](#_Toc180593513)

[Tabela 3 - Regras de Negócio 50](#_Toc180593514)

[Tabela 4 - Caso de Uso: 1. Solicitar Cadastro 52](#_Toc180593515)

[Tabela 5 - Caso de Uso: 2. Aprovar Cadastro 52](#_Toc180593516)

[Tabela 6 - Caso de Uso: 3. Fazer Login 53](#_Toc180593517)

[Tabela 7 - Caso de Uso: 4. Solicitar Acesso ao Kanban de Projeto 54](#_Toc180593518)

[Tabela 8 - Caso de Uso: 5. Aprovar Acesso ao Kanban de Projeto 54](#_Toc180593519)

[Tabela 9 - Caso de Uso: 6. Manter Kanban de Projeto 55](#_Toc180593520)

[Tabela 10 - Caso de Uso: 7. Consultar Kanban de Projeto 55](#_Toc180593521)

[Tabela 11 - Caso de Uso: 8. Solicitar Acesso ao PDCA 56](#_Toc180593522)

[Tabela 12 - Caso de Uso: 9. Manter Quadro PDCA 56](#_Toc180593523)

[Tabela 13 - Caso de Uso: 10. Manter Kanban Pessoal 57](#_Toc180593524)

[Tabela 14 - Caso de Uso: 11. Manter Usuários 57](#_Toc180593525)

[Tabela 15 - Caso de Teste: Solicitar Cadastro 71](#_Toc180593526)

[Tabela 16 - Caso de Teste: Manutenção de Usuários 71](#_Toc180593527)

[Tabela 17 - Caso de Teste: Alteração de Senha 73](#_Toc180593528)

[Tabela 18 - Caso de Teste: Login 73](#_Toc180593529)

[Tabela 19 - Caso de Teste: Criação de Kanban de Projeto 74](#_Toc180593530)

[Tabela 20 - Caso de Teste: Manutenção de Quadro PDCA 75](#_Toc180593531)

[Tabela 21 - Caso de Teste: Kanban Pessoal 75](#_Toc180593532)

[Tabela 22 - Alteração de Status do Kanban de Projeto 76](#_Toc180593533)

[Tabela 23 - Caso de Teste: Documentos de Projeto 76](#_Toc180593534)

[Tabela 24 - Caso de Teste: Visualização de Kanban e PDCA 77](#_Toc180593535)

[Tabela 25 - Caso de Teste: Adicionar Documentos 77](#_Toc180593536)

[Tabela 26 - Caso de Teste: Solicitar Entrada no Projeto 78](#_Toc180593537)

[Tabela 27 - Caso de Teste: Pointing Poker 78](#_Toc180593538)

[Tabela 28 - Caso de Teste: Encerramento do Pointing Poker 79](#_Toc180593539)

[Tabela 29 - Caso de Teste: Métricas 79](#_Toc180593540)

[Tabela 30 - Caso de Teste: Manutenção Kanban de Projetos 80](#_Toc180593541)

[Tabela 31 - Caso de Teste: Dicas de Ágil 80](#_Toc180593542)

[Tabela 32 - Caso de Teste: Tela de Projetos 81](#_Toc180593543)

[Tabela 33 - Caso de Teste: Pesquisa de Projetos 81](#_Toc180593544)

[Tabela 34 - Caso de Teste: Solicitação para um Projeto 82](#_Toc180593545)

**SUMÁRIO**

[1. INTRODUÇÃO 19](#_Toc183419521)

[1.1. Objetivos 21](#_Toc183419522)

[1.1.1. Objetivo Geral 21](#_Toc183419523)

[1.1.2. Objetivos Específicos 21](#_Toc183419524)

[1.2. Motivação 22](#_Toc183419525)

[1.2.1. Questão a Ser Resolvida 23](#_Toc183419526)

[2. REFERENCIAL TEÓRICO 24](#_Toc183419527)

[2.1. Metodologia Ágil 24](#_Toc183419528)

[2.2. Scrum 24](#_Toc183419529)

[2.2.1. Papéis no Scrum 25](#_Toc183419530)

[2.2.2. Artefatos do Scrum 25](#_Toc183419531)

[2.2.3. Eventos do Scrum 26](#_Toc183419532)

[2.2.4. Benefícios do Scrum 26](#_Toc183419533)

[2.2.5. Desvantagens do Scrum 26](#_Toc183419534)

[2.3. Lean 27](#_Toc183419535)

[2.3.1. Princípios do Lean 29](#_Toc183419536)

[2.4. Kanban 32](#_Toc183419537)

[2.5. PDCA 33](#_Toc183419538)

[3. MATERIAIS E FERRAMENTAS 36](#_Toc183419539)

[3.1. .NET 36](#_Toc183419540)

[3.2. Angular 36](#_Toc183419541)

[3.3. JSON 37](#_Toc183419542)

[3.4. SQL Server 37](#_Toc183419543)

[3.5. SQL Server Management Studio (SSMS) 38](#_Toc183419544)

[3.6. Visual Studio Code (VSCode) 38](#_Toc183419545)

[3.7. Visual Studio 2022 38](#_Toc183419546)

[3.8. Figma 39](#_Toc183419547)

[3.9. Lucidchart 39](#_Toc183419548)

[3.10. BrModelo 39](#_Toc183419549)

[4. METODOLOGIA 40](#_Toc183419550)

[4.1. Metodologia de Pesquisa 40](#_Toc183419551)

[4.1.1. Instrumento de Coleta de Dados 40](#_Toc183419552)

[4.1.2. Amostra e Perfil dos Participantes 41](#_Toc183419553)

[4.1.3. Resultados da Pesquisa 41](#_Toc183419554)

[4.1.4. Funcionalidades Valorizadas 42](#_Toc183419555)

[4.1.5. Barreiras na Adoção de Novas Ferramentas 42](#_Toc183419556)

[4.1.6. Software semelhantes 43](#_Toc183419557)

[4.1.6.1. Propósito e Funcionalidades Principais 43](#_Toc183419558)

[4.1.6.2. Metodologias Suportadas 44](#_Toc183419559)

[4.1.6.3. Público-Alvo 44](#_Toc183419560)

[4.1.6.4. Usabilidade e Interface 45](#_Toc183419561)

[4.1.6.5. Colaboração 45](#_Toc183419562)

[4.1.6.6. Relatórios e Análise 46](#_Toc183419563)

[4.1.6.7. Diferenciais 47](#_Toc183419564)

[4.2. Metodologia de Desenvolvimento 47](#_Toc183419565)

[5. MODELAGEM 51](#_Toc183419566)

[5.1. Requisitos Funcionais 51](#_Toc183419567)

[5.2. Requisitos Não Funcionais 53](#_Toc183419568)

[5.3. Regras de Negócio 54](#_Toc183419569)

[5.4. UML 54](#_Toc183419570)

[5.4.1. Caso De Uso 55](#_Toc183419571)

[5.4.2. Narrativa De Caso De Uso 56](#_Toc183419572)

[5.4.3. Diagrama de Classe 62](#_Toc183419573)

[5.5. Diagrama De Banco De Dados 63](#_Toc183419574)

[5.6. Protótipos 64](#_Toc183419575)

[6. DESENVOLVIMENTO 73](#_Toc183419576)

[6.1. Criação de Identidade Visual 73](#_Toc183419577)

[6.2. Repositórios 74](#_Toc183419578)

[7. TESTES E RESULTADOS 75](#_Toc183419579)

[8. CONCLUSÃO 87](#_Toc183419580)

[REFERÊNCIAS 88](#_Toc183419581)

[APÊNDICE A – GRÁFICO DA QUESTÃO 1 DO QUESTIONÁRIO DE PESQUISA 92](#_Toc183419582)

[APÊNDICE B – GRÁFICO DA QUESTÃO 2 DO QUESTIONÁRIO DE PESQUISA 92](#_Toc183419583)

[APÊNDICE C – GRÁFICO DA QUESTÃO 3 DO QUESTIONÁRIO DE PESQUISA 93](#_Toc183419584)

[APÊNDICE D – GRÁFICO DA QUESTÃO 4 DO QUESTIONÁRIO DE PESQUISA 93](#_Toc183419585)

[APÊNDICE E – GRÁFICO DA QUESTÃO 5 DO QUESTIONÁRIO DE PESQUISA 94](#_Toc183419586)

[APÊNDICE F – GRÁFICO DA QUESTÃO 6 DO QUESTIONÁRIO DE PESQUISA 94](#_Toc183419587)

[APÊNDICE G – GRÁFICO DA QUESTÃO 7 DO QUESTIONÁRIO DE PESQUISA 95](#_Toc183419588)

[APÊNDICE H – GRÁFICO DA QUESTÃO 8 DO QUESTIONÁRIO DE PESQUISA 95](#_Toc183419589)

[APÊNDICE I – DIAGRAMA DE ENTIDADE E RELACIONAMENTO 96](#_Toc183419590)

[APÊNDICE J – MODELO LÓGICO – BANCO DE DADOS 97](#_Toc183419591)

# INTRODUÇÃO

Em um cenário empresarial marcado por constantes mudanças e crescente complexidade, as metodologias ágeis vêm ganhando destaque como soluções inovadoras e eficazes para aumentar a eficiência e promover a adaptabilidade. Diferente dos modelos tradicionais de gestão, as abordagens ágeis, como Kanban, Scrum, PDCA e Lean, são baseadas em ciclos curtos e iterativos de desenvolvimento, o que facilita entregas rápidas e contínuas. Essas metodologias têm sido amplamente adotadas em diversos setores, principalmente por sua capacidade de responder de forma ágil às demandas do mercado, garantindo maior flexibilidade e eficiência na execução dos projetos.

No entanto, a adoção dessas práticas não está isenta de desafios. Muitas organizações e profissionais enfrentam dificuldades ao tentar implementar múltiplas metodologias ágeis de maneira integrada. A combinação dessas abordagens pode resultar em falhas de comunicação entre as equipes, perda de foco nas prioridades e dificuldade em manter uma estrutura de trabalho coesa e eficiente. Segundo a TOTVS (2021), as metodologias ágeis são essenciais para promover maior transparência e otimizar a colaboração nas organizações. Contudo, a ausência de ferramentas centralizadas que integrem essas práticas pode limitar o potencial das metodologias ágeis, comprometendo seus benefícios.

Com o objetivo de preencher essa lacuna, o presente projeto propõe o desenvolvimento de uma plataforma digital que centralize as principais metodologias ágeis, como Kanban, Scrum, PDCA e Lean. A ideia é oferecer uma ferramenta acessível e intuitiva, que possibilite tanto às empresas quanto aos profissionais a aplicação dessas metodologias de maneira personalizada, adaptando-as às suas necessidades específicas. De acordo com Miro, as metodologias ágeis também desempenham um papel fundamental na promoção de uma cultura de inovação, permitindo que as organizações reajam de forma mais rápida às mudanças e otimizem seus processos internos de forma contínua e eficiente.

Além disso, conforme os estudos de Nagai e Sbragia (2023), os princípios que fundamentam as metodologias ágeis estão fortemente ligados ao Lean e ao Desenvolvimento Incremental e Iterativo, práticas originalmente adotadas pela indústria, mas que foram amplamente adaptadas para diversos setores ao longo do tempo. Esses conceitos indicam que a aplicação das metodologias ágeis não só melhora a eficiência operacional, mas também contribui para o fortalecimento de uma cultura de melhoria contínua, alinhada aos objetivos estratégicos das organizações.

Portanto, a plataforma proposta visa não só facilitar a implementação dessas metodologias em ambientes corporativos, mas também as tornar mais acessíveis, permitindo que profissionais as apliquem em suas atividades diárias. Ao reunir em um único local ferramentas ágeis, versáteis e práticas, a expectativa é que a plataforma contribua para a simplificação e ampliação da gestão ágil, permitindo que diferentes tipos de usuários aproveitem os benefícios dessas metodologias de maneira flexível e personalizada.

## Objetivos

### Objetivo Geral

O objetivo geral deste projeto é desenvolver uma plataforma web integrada para aplicar metodologias ágeis no ambiente empresarial, oferecendo ferramentas que facilitem a gestão de processos, a organização de tarefas e a melhoria contínua por meio de Scrum, PDCA, Lean e Kanban. A proposta é centralizar as soluções, ajudando as equipes a entenderem e gerenciar suas responsabilidades, alinhando-as aos objetivos estratégicos da empresa, aumentando a eficiência operacional e promovendo a colaboração entre os colaboradores.

### Objetivos Específicos

* Implementar um módulo de Scrum que permita a criação e gerenciamento de quadros Kanban, tanto individuais quanto de equipe, facilitando a visualização do progresso das tarefas e promovendo a transparência no processo de desenvolvimento.
* Desenvolver funcionalidades para a gestão de Sprints, incluindo quadros de início e final de Sprint, para auxiliar na organização e no monitoramento das atividades planejadas e concluídas ao longo dos ciclos de desenvolvimento.
* Incorporar uma ferramenta de Pointing Poker para permitir a estimativa colaborativa do esforço necessário para a realização das tarefas, promovendo o consenso e a precisão nas estimativas de tempo e recursos.
* Criar um módulo de PDCA, facilitando a aplicação do ciclo de melhoria contínua nas equipes, com funcionalidades que auxiliem no planejamento, execução, verificação e ajustes dos processos.
* Incorporar um check-list automatizado baseado em Lean, que auxilie na identificação e mitigação de desperdícios nas atividades e processos do projeto, promovendo a eficiência e a redução de custos.
* Criar um módulo de gerenciamento de documentos, onde cada equipe possa armazenar e organizar seus documentos de projeto de forma segura e acessível, garantindo que todas as informações essenciais estejam centralizadas.
* Desenvolver uma ferramenta de sugestão de modelos de documentos\*\* que auxilie na criação de documentação padronizada, assegurando a consistência e a conformidade com as melhores práticas de gerenciamento de projetos.
* Garantir a usabilidade e a integração das funcionalidades\*\* da plataforma, oferecendo uma interface intuitiva que permita aos usuários navegarem e utilizar as ferramentas de forma eficiente e eficaz, contribuindo para a melhoria do desempenho das equipes.
* Validar a eficácia da plataforma através de testes com usuários reais\*\*, ajustando e refinando as funcionalidades com base no feedback coletado, para garantir que a solução atenda às necessidades práticas das equipes e gestores.

## Motivação

Considerando que o tema “Metodologias Ágeis” tem ganhado crescente relevância e presença em diversos setores da área de tecnologia, nossa motivação para este trabalho surgiu a partir da reflexão sobre o uso dessas metodologias no mercado. Foi observado como elas são aplicadas e analisada a sua efetividade em projetos. Dessa forma, reconhecendo o potencial de melhoria na eficiência e no fluxo de produção, foi proposto o desenvolvimento de uma plataforma web para otimizar a aplicação, o gerenciamento, o desenvolvimento e a organização das metodologias ágeis, especificamente Scrum, Lean, PDCA e Kanban. A plataforma será projetada para ser utilizada pela equipe responsável, com o objetivo de proporcionar um melhor desenvolvimento dos projetos e assegurar uma conclusão satisfatória.

### Questão a Ser Resolvida

Atualmente, a maioria das organizações enfrenta dificuldades ao tentar implementar e gerenciar diversas metodologias de forma integrada e eficiente dentro de seus processos ágeis. A ausência de ferramentas centralizadas que possam unificar práticas como Scrum, Kanban, PDCA e Lean resulta em falhas de comunicação entre as equipes, perda de foco nas prioridades e dificuldades em manter uma estrutura de trabalho coesa. A falta de padronização no uso dessas abordagens acaba por agravar os problemas de colaboração e acompanhamento de tarefas.

As empresas ainda se deparam com a dificuldade de adaptar rapidamente suas práticas às mudanças do mercado, o que gera processos menos eficientes e uma menor capacidade de inovação. Embora as metodologias ágeis estejam cada vez mais difundidas, muitas organizações ainda não conseguem aplicá-las plenamente, devido à complexidade de implementação e à falta de ferramentas acessíveis que ofereçam a flexibilidade necessária para diferentes tipos de projetos e equipes.

Esses problemas culminam na dificuldade de visualizar o progresso dos projetos, na falta de transparência nos processos e na ineficiência da comunicação entre os membros das equipes. Isso pode comprometer tanto a qualidade quanto a velocidade das entregas, além de prejudicar a colaboração, essencial para o sucesso dos projetos.

# REFERENCIAL TEÓRICO

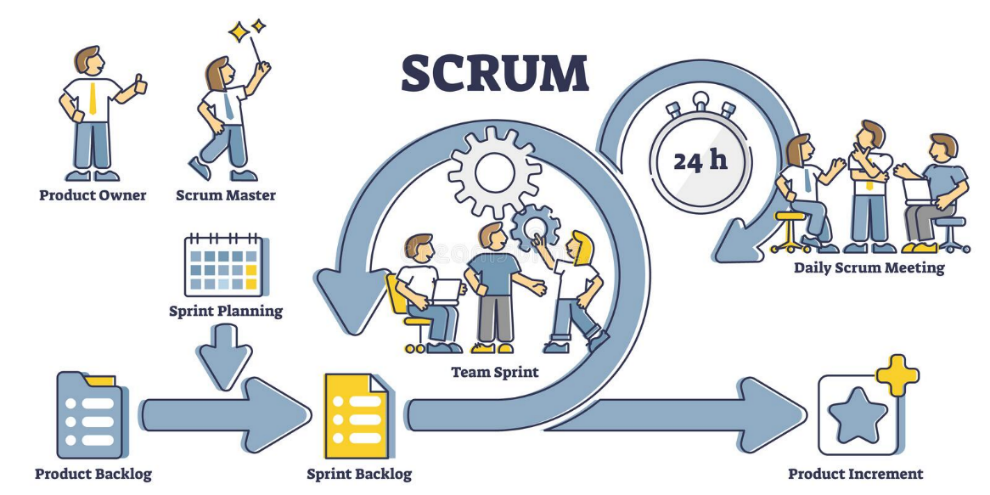
## Metodologia Ágil

As metodologias ágeis são abordagens de gestão de projetos que focam na entrega rápida e contínua de valor, permitindo maior flexibilidade e adaptação às mudanças. Entre elas, destacam-se o Scrum, que organiza o trabalho em ciclos curtos; o Lean, que busca eliminar desperdícios e maximizar valor; o Kanban, que visualiza o fluxo de trabalho para otimizar processos; e o PDCA, uma ferramenta para melhorar continuamente as práticas e processos dentro de um projeto. Cada uma dessas metodologias possui características específicas que ajudam equipes a trabalhar de maneira mais eficiente e colaborativa

## Scrum

O Scrum é uma metodologia ágil que facilita o gerenciamento de projetos complexos, permitindo que equipes se adaptem rapidamente a mudanças e entreguem valor continuamente. De acordo com Schwaber (2004), o Scrum organiza o trabalho em ciclos curtos, conhecidos como Sprints, que permitem ajustes rápidos e entregas incrementais. Essa abordagem é especialmente útil em projetos onde os requisitos podem mudar com frequência.

Figura 1 - Ciclo do Scrum



Fonte: <https://www.benner.com.br/tecnologia/wp-content/uploads/2023/12/scrum.png>, 2023

### Papéis no Scrum

O Scrum é composto por três papéis principais: o Product Owner, o Scrum Master e o Development Team.

Product Owner: É responsável por gerenciar o Product Backlog e priorizar as tarefas que trazem mais valor ao produto. Schwaber (2004) destaca que o Product Owner deve representar os interesses dos stakeholders e garantir que a equipe de desenvolvimento esteja focada nas tarefas mais importantes.

Scrum Master: Sua função é garantir que a equipe siga os princípios e práticas do Scrum, além de remover obstáculos e promover a melhoria contínua. Sutherland (2019) afirma que o Scrum Master não atua como um gerente tradicional, mas como um facilitador que ajuda a equipe a ser autônoma e produtiva.

Development Team: Composto por profissionais multifuncionais, o Development Team trabalha em conjunto para entregar incrementos do produto. A equipe organiza seu próprio trabalho e decide como implementar as tarefas para alcançar os objetivos da Sprint (Schwaber, 2004).

### Artefatos do Scrum

Product Backlog: É uma lista priorizada de todas as tarefas necessárias para o desenvolvimento do produto. Pichler (2010) afirma que o Product Backlog deve ser continuamente refinado e priorizado para garantir que a equipe esteja trabalhando nas tarefas mais valiosas para o produto.

Sprint Backlog: Consiste nos itens selecionados do Product Backlog que serão trabalhados durante a Sprint. A equipe é responsável por planejar como essas tarefas serão realizadas e dividir as responsabilidades entre os membros.

Incremento: Refere-se ao resultado do trabalho realizado ao final de cada Sprint, que deve ser uma versão utilizável do produto, agregando valor ao projeto (Schwaber, 2004).

### Eventos do Scrum

Sprint Planning: Reunião realizada no início de cada Sprint, onde a equipe define o que será feito e como será feito. Schwaber (2004) enfatiza que essa reunião é fundamental para garantir que todos os membros da equipe compreendam claramente o que será entregue e como alcançar os objetivos da Sprint.

Daily Scrum: Uma reunião diária de até 15 minutos, em que a equipe discute o progresso e ajusta o planejamento conforme necessário. Sutherland (2014) explica que o Daily Scrum facilita uma comunicação rápida e contínua, permitindo que a equipe se mantenha alinhada e identifique impedimentos.

Sprint Review: Ao final de cada Sprint, a equipe apresenta o que foi desenvolvido e avalia o progresso em relação aos objetivos estabelecidos. Pichler (2010) ressalta que essa reunião é essencial para garantir a transparência e alinhar as expectativas dos stakeholders.

Sprint Retrospective: Após o encerramento da Sprint, a equipe realiza uma reunião focada na melhoria contínua, discutindo o que funcionou bem e o que pode ser aprimorado no próximo ciclo.

### Benefícios do Scrum

O Scrum oferece diversos benefícios, como a capacidade de adaptação rápida às mudanças, a entrega contínua de valor e a melhoria constante. Schwaber (2004) ressalta que essa metodologia permite que as equipes entreguem resultados de alta qualidade e se ajustem rapidamente às mudanças nos requisitos, o que é fundamental em ambientes de projeto dinâmicos.

### Desvantagens do Scrum

Apesar de seus benefícios, o Scrum também apresenta desvantagens. Sua implementação pode ser desafiadora para equipes que não estão familiarizadas com práticas ágeis. A dependência de um Scrum Master experiente é crucial para o sucesso do processo. Além disso, o Scrum pode enfrentar dificuldades em projetos de grande escala, o que pode exigir adaptações ou o uso de frameworks adicionais, como o Scrum of Scrums (Sutherland, 2019).

## Lean

A metodologia Lean é uma filosofia de gestão que busca elevar a produtividade e maximizar o valor entregue ao cliente por meio da eliminação de desperdício de tempo e recursos, redução de custos e otimização dos processos. O Lean orienta a empresa que que está utilizando essa metodologia na adoção do método “desenvolvimento ágil”, essas práticas são desenvolvidas em conjunto com o cliente. (Barbosa, et al., 2022; Lean Institute Brasil, [s.d.])

Figura 2 - 5 Princípios do Lean

Diagrama

Descrição gerada automaticamente

Fonte: <https://www.kanbanchi.com/wp-content/uploads/2023/04/lean-principles.png>, 2023

O termo filosofia Lean originou-se do Japão, em grande parte, no desenvolvimento do Sistema Toyota de Produção (STP) após a Segunda Guerra Mundial. Nessa época, a empresa automobilística enfrentava desafios econômicos difíceis e sofreu com uma séria escassez de recursos. Portanto, a Toyota procurou maneiras de otimizar seu processo de produção. Apesar da inspiração do método de produção em massa de Henry Ford, a Toyota adaptou o sistema às suas condições e necessidades locais e criou o STP. A principal característica distinta do STP era a eliminação de desperdícios e melhoria contínua e foco na adição de valor. (Briales, 2022)

Posteriormente, o livro A Máquina que Mudou o Mundo introduziu a palavra Lean para descrever esse novo paradigma de produção. Isso foi validado por uma pesquisa extensa do MIT sobre a indústria automotiva. A pesquisa mostrou que a Toyota poderia produzir muitos produtos diferentes com volumes baixos, baixo investimento e altos padrões de qualidade. Tudo isso foi feito com metade do esforço humano, espaço, ativos fixos, tempo para desenvolver novos produtos e estoques. É preciso ressaltar que a tecnologia e a qualidade não foram comprometidas. O paradigma Lean é uma ruptura com a produção em larga escala, pois é muito flexível. Porém, isso não diminui o seu compromisso com a qualidade e a consistência, isto é, a eficiência. O método Just-in-Time é essencial para evitar desperdícios de estoque e maximizar o uso dos recursos. (Briales, 2022; FM2S, 2023).

De acordo com o Lean Intitute Brasil, para aplicar o Lean:

“Identifique o que é valor para o cliente. Então, concentre energia nas atividades que criam valor, eliminando todo o esforço desnecessário (desperdícios) que permeia a organização do trabalho e as etapas do processo. Então, busque estabelecer um fluxo contínuo de produção e entregas, respeitando a cadência e as necessidades do cliente. Perpetue a busca pelo valor perfeito para o cliente, envolvendo todos os níveis da organização em inovação, melhoria e solução de problemas.” (Lean Institute Brasil, [s.d.])

### Princípios do Lean

Valor

No gerenciamento Lean, o primeiro passo na fabricação de um produto de um cliente é identificar o valor, ou seja, identificar o motivo pelo qual um cliente pagará prontamente pelo produto ou serviço fornecido. Assim, qualquer processo ou atividade que não agrega qualquer utilidade ou é irrelevante ao produto é considerado desperdício e é imediatamente eliminado. (Fm2s, 2023; Totvs, 2023)

Fluxo de Valor

Depois de estabelecer o valor que o produto deve oferecer, a próxima etapa é o mapeamento do “fluxo de valor”, o que inclui o caminho e os processos de todas as etapas que é preciso fazer para transformar matérias-primas em um produto e entregá-lo ao cliente. Esse mapeamento é necessário para entender todas as ações que estão sendo realizadas e com que finalidade para a eliminação de desperdícios, que fornece uma visualização mais lógica e detalhada do fluxo das atividades. (Fm2s, 2023; Totvs, 2023)

Fluxo Continuo

Depois da remoção do desperdício do fluxo de valores, é necessário garantir que as etapas restantes sigam uns aos outros sem dificuldades, atraso ou gargalos, criando uma cadeia de produção ágil e acelerada. Isso também incluiria ultrapassar barreiras entre departamentos e promover a abordagem multifuncional, o que, embora seja um desafio, pode render ganhos de produtividade e eficiência de mais de 50% do tempo. Dividir o processo de trabalho em passos menores e usar instrumentos específicos, como Kanban, que permite uma melhor visualização e comunicação entre as equipes, é crucial para garantir a ausência de interrupções e bloqueios. Desta maneira, acabará por tornar-se um processo mais acelerado e produtivo, com menos erros, volume de produção e entrega mais alta qualidade em menos tempo. (Fm2s, 2023; Totvs, 2023)

Produção Puxada

Por fim, a produção puxada, como técnica lean, visa reduzir o desperdício nos processos de produção ao permitir que os trabalhos sejam iniciados apenas com base em sua demanda. Isso permite reduzir a carga física e otimizar os custos de armazenamento. Além disso, possibilita manter um estoque virtual mais enxuto ao invés do justo o que, dada a natureza sem desperdícios de todo o fluxo, não compromete os produtos de entrega. Ao calibrar o fluxo de prioridades e aprimorar processos, a produção puxada, que é o alicerce do gerenciamento enxuto, valoriza a melhoria contínua como a prioridade máxima dentre todos os princípios. (Fm2s, 2023; Totvs, 2023)

Melhoria Contínua

O quinto passo da Metodologia Lean diz respeito à incorporação de melhoria contínua à cultura corporativa, e é este o fator responsável pelo sucesso do sistema. A quinta filosofia Lean afirma que a perfeição é inalcançável e que a mentalidade otimista de tentar melhorar é importante. Isso significa que tudo, desde a qualidade dos processos internos até a dependência do relacionamento entre as diversas equipes e a utilização de tecnologia para facilitar a execução, está em questão. Resolver desafios futuros, monitorar atentamente e garantir um produto final que forneça valor ao cliente por meio do envolvimento de todos os funcionários em todos os estágios é fundamental. (Fm2s, 2023; Totvs, 2023)

Desperdícios do Lean

Neste sentido, o texto apresenta sete principais tipos de desperdícios que podem ocorrer num processo produtivo e argumenta sobre a necessidade de eliminá-los para otimizar o trabalho e a produtividade de uma organização. A produção em excesso é considerada a “mãe” de todos os desperdícios, uma vez que leva à utilização de recursos de forma desnecessária e à manutenção de estoques que não serão consumidos. A espera também é um dos tipos mais comuns da ineficiência, uma vez que envolve o tempo durante o qual uma pessoa ou uma máquina fica parada por falta de sincronização entre as etapas. (Lean Institute Brasil, [s.d.]; Totvs, 2023)

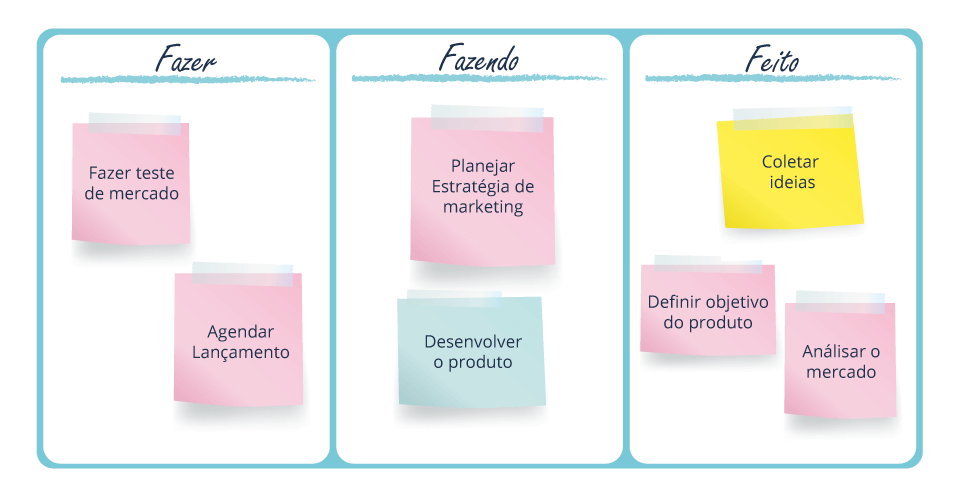
Outro problema é o processamento desnecessário, que ocorre quando certas atividades que não agregam valor são continuadas, como verificações repetitivas ou tratamento contínuo que não é mais necessário. O estoque excessivo é mencionado como um dos principais indicadores de problemas em um sistema produtivo, uma vez que exige capital e pode encobrir defeitos, atrasando a detecção e correção. Os próximos desperdícios a serem considerados são associados ao transporte e movimentação interna, com deslocação desnecessária de materiais e pessoas que consome tempo e recursos. (Lean Institute Brasil, [s.d.]; Totvs, 2023)

Finalmente, o desperdício relacionado à correção de erros é mencionado por ser muito comum: é desperdiçado tempo e recursos para refazer itens que foram produzidos de forma incorreta. O autor acredita que o meta de criar consciência de todos esses problemas e promover um sentimento de inadequação é essencial para resolvê-los em breve. Finalmente, o desperdício sob a forma de correção de erros é identificado entre os mais comuns em que é preciso prever tempo e esforço para reprojetar o que foi feito erradamente. O autor conclui que alcunhar tudo o que foi mencionado acima requer vai questionar o que se passa e criar um senso de insatisfação com ele, de forma a eliminar esses problemas e contribuir para a melhoria contínua (Lean Institute Brasil, [s.d.]; Totvs, 2023).

## Kanban

Originado do termo de origem japonesa “kanban” que significa “cartão” ou “sinalização”, a ferramenta indica o andamento de um fluxo de produção.

Figura 3 - Kanban



Fonte: <https://artia.com/wp-content/uploads/2019/01/kanban-fazer-fazendo-feito.png>, 2019

Essa metodologia ágil foi introduzida por Taiichi Ohno em 1953 dentro do Sistema Toyota, era um cartão para sinalizar a conclusão de uma produção e o início de uma nova, facilitando a gestão de recursos. (VILLANOVA, 2021)

O quadro Kanban pode ser tanto usado de forma física com post-its quanto de forma digital em quadros organizados online. Esses são organizados em divisões sendo normalmente as três etapas “Pendente”, “Em Andamento” e “Concluído”, onde serão posicionados os cartões onde estão localizadas as tarefas.

Os cartões possuem as descrições básicas das tarefas, sendo algumas: o responsável, descrição e tempo estimado para sua conclusão.

Os benefícios que essa metodologia pode trazer são:

* Flexibilidade e Versatibilidade: Concentrando a equipe nas tarefas atuais e planejando as próximas adequando o fluxo de trabalho.
* Visualização: A possibilidade de verificar visualmente onde os esforços da equipe estão sendo concentrados e como estão a continuidade das tarefas.
* Produtividade: Uma maneira de estimular a evolução da produção de projetos de forma organizada.

## PDCA

O Ciclo PDCA, também conhecido como "Ciclo de Deming", foi desenvolvido nas décadas de 1920 e 1930 pelo estatístico norte-americano Walter A. Shewhart. Esse método consiste em um loop contínuo de planejamento, execução, verificação e ação, sendo um ciclo de controle de projetos que pode ser repetido em qualquer processo ou problema, com o objetivo de identificar e corrigir falhas, mantendo os processos que funcionam e melhorando aqueles que necessitam de ajustes. O PDCA foi amplamente popularizado na década de 1950 pelo especialista em qualidade W. Edwards Deming, após sua contribuição ao progresso das empresas japonesas, auxiliando na melhoria contínua e no monitoramento da qualidade em seus processos. Por essa razão, o ciclo também ficou conhecido como "Ciclo de Deming".

O ciclo PDCA fundamenta-se na melhoria contínua dos processos de desenvolvimento, sendo caracterizado como um método iterativo. Caso um dos elementos do ciclo se mostre ineficaz ou sem melhorias, o ciclo recomeça e é repetido até que o resultado desejado seja alcançado.

Campos (1996, p. 262) define o ciclo PDCA da seguinte forma: “O PDCA é um método de gerenciamento de processos ou de sistemas. É o caminho para se atingirem as metas atribuídas aos produtos dos sistemas empresariais.”

O ciclo PDCA é dividido em quatro etapas: "Plan (Planejar)", "Do (Fazer)", "Check (Checar ou Verificar)" e "Act (Agir)".

Plan (Planejar): Este é o primeiro e mais importante estágio do ciclo, pois requer uma compreensão detalhada do problema a ser resolvido e da importância de sua resolução. Badiru (1993) afirma que, nesta fase, devem ser discutidas questões como: Qual é o objetivo a ser alcançado pela organização? Quais pessoas devem estar envolvidas nesse processo? Qual é o prazo para a conclusão do plano de ação? Quais recursos serão disponibilizados para a execução do plano? Quais dados esperamos obter no processo?

O módulo "Plan" é subdividido em cinco etapas essenciais, conforme descrito por Campos (1996) e Melo (2001):

Entender o problema: Identificar o resultado indesejado de um processo.

Estabelecer as metas: Definir o que se deseja obter ao final do projeto.

Análise de fenômenos: Compreender todas as características do problema.

Análise de processos (causas): Identificar as possíveis causas do problema.

Plano de Ação: Determinar as ações que devem ser tomadas para alcançar as metas estabelecidas.

Do (Fazer): Esta etapa envolve a execução das soluções para os problemas identificados. Só é viável realizá-la se houver um plano de ação bem estruturado. O ideal é implementar as mudanças em pequena escala para evitar impactos significativos na produção e nos ciclos subsequentes. Caso alguma ação cause um impacto negativo, os danos ao processo do projeto serão minimizados, assim como o uso de recursos.

Este módulo é dividido em duas partes importantes:

Fase de Treinamento: Esta fase envolve a conscientização da equipe responsável pelo projeto. Os planos de ação devem ser comunicados aos envolvidos, destacando a importância de cada ação e do papel de cada membro.

Fase de Execução: Após a divulgação aos membros, é hora de colocar o plano em prática.

Check (Checar): Esta etapa consiste na verificação e análise. Com base nas etapas anteriores, verifica-se a eficácia das soluções implementadas nos processos. O objetivo é garantir que os resultados sejam satisfatórios e que as soluções estejam funcionando conforme os objetivos estabelecidos ou, caso contrário, identificar a necessidade de melhorias adicionais para atingir as metas. Se ajustes forem necessários, deve-se retornar à etapa de planejamento para reavaliar as alternativas e resolver o problema.

Act (Agir): Esta etapa consiste em padronizar as ações bem-sucedidas implementadas nas etapas anteriores que melhoraram os processos. Isso inclui documentar o passo a passo das ações tomadas e orientar todos os envolvidos a seguir o plano de ação definido.

Figura 4 - PDCA

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: https://voitto.com.br/blog/artigo/o-que-e-o-ciclo-pdca, 2017

# MATERIAIS E FERRAMENTAS

## .NET

Durante a fase de planejamento do projeto, foram analisadas uma série de plataformas e ferramentas para decidir qual atenderia melhor às necessidades específicas do desenvolvimento desse projeto. O .NET se destacou, não apenas pela sua versatilidade em suportar diferentes tipos de aplicações, seja web, desktop ou mobile, mas também pela sua capacidade de integrar todos esses ambientes de forma fluida. O fato de o .NET estar sempre se atualizando, com melhorias constantes, também pesou em nossa escolha, pois isso garante o uso de tecnologias de ponta, essenciais para assegurar o sucesso e longevidade do projeto.

Outro fator que chama atenção é a flexibilidade no uso de várias linguagens de programação dentro da mesma plataforma. Isso facilitou bastante a integração de diferentes componentes. Frameworks como o ASP.NET Core, voltado para aplicações web, e o Entity Framework Core, que auxilia no gerenciamento de bancos de dados, formando a base estrutural do sistema.

Ainda vale ressaltar que o ecossistema do .NET, com sua vasta gama de bibliotecas e ferramentas adicionais, nos deu uma base sólida para enfrentar desafios mais complexos. Sabe-se que, qualquer que fosse o problema técnico, haveria soluções já estabelecidas ou facilmente adaptáveis para nossa realidade.

## Angular

Para a construção da interface do usuário, a escolha foi o Angular, que, além de ser amplamente utilizado no mercado, oferece uma excelente estrutura para desenvolver Single Page Applications (SPA). Essas SPAs são bastante eficazes para criar uma experiência de navegação fluida, o que impacta diretamente a experiência do usuário. Um ponto adicional foi o uso do TypeScript, que ajudou a melhorar a qualidade do código com sua tipagem estática, reduzindo a margem de erro durante o desenvolvimento.

O Angular-CLI foi outro grande diferencial. Através dessa ferramenta de linha de comando é possível automatizar a criação de componentes, o que não só trouxe agilidade ao processo de desenvolvimento, mas também garantiu uma maior organização do código.

## JSON

No que diz respeito à comunicação entre os componentes do sistema, a escolha pelo JSON (JavaScript Object Notation) foi praticamente automática. Ele se destacou por ser um formato de dados leve e de fácil leitura, tanto por humanos quanto por máquinas. Essa característica simples, porém, poderosa, tornou o JSON ideal para ser o padrão de troca de informações entre o front-end e o back-end.

Além disso, a compatibilidade do JSON com várias linguagens de programação, como JavaScript, Python e .NET, facilitou muito a integração de diferentes partes do sistema. Isso permite uma interoperabilidade eficaz entre as camadas de aplicação, sem comprometer a performance.

Sua estrutura simples e legível também facilita bastante a manutenção e futura expansão do código.

## SQL Server

Para o gerenciamento de banco de dados, o SQL Server é uma escolha sólida para uma aplicação web baseada em Angular e .NET devido à sua integração robusta com o ecossistema Microsoft. Ele oferece alta performance, confiabilidade e segurança, além de suporte nativo para APIs e frameworks da plataforma .NET. O SQL Server também é otimizado para trabalhar em ambientes híbridos, permitindo fácil integração com o Azure para escalabilidade e serviços adicionais, como backups automáticos e análise de dados. Sua interface amigável e recursos avançados, como consultas em tempo real e suporte a dados relacionais e não relacionais, tornam-no ideal para aplicações modernas que demandam flexibilidade e eficiência.

## SQL Server Management Studio (SSMS)

O SQL Server Management Studio (SSMS) é uma ferramenta essencial para gerenciar e configurar o SQL Server em aplicações web, como aquelas baseadas em Angular e .NET. Ele oferece uma boa interface para criar e administrar bancos de dados, executar consultas em T-SQL, configurar permissões e monitorar o desempenho do servidor. Além disso, o SSMS facilita tarefas como criação de backups e otimização de consultas, integrando-se perfeitamente ao SQL Server para simplificar o gerenciamento e melhorar a produtividade do desenvolvedor.

## Visual Studio Code (VSCode)

O Visual Studio Code (VSCode) foi escolhido como um dos editores de código por ser uma ferramenta leve e altamente personalizável, que atende tanto projetos pequenos quanto de grande escala. Um dos principais benefícios é a integração nativa com sistemas de controle de versão, como o Git, que facilita o acompanhamento das alterações no código e permite uma colaboração mais eficiente entre os membros da equipe.

O recurso IntelliSense, que sugere automaticamente trechos de código é fundamental para evitar erros e melhorar a produtividade. Permitindo assim focar mais nas soluções criativas e menos nas correções de pequenos problemas de sintaxe.

## Visual Studio 2022

O Visual Studio foi escolhido para a edição do código .NET, o suporte da IDE para a criação de códigos dessa linguagem é extremamente robusto. Assim permitindo a compilação e execução do código de forma ágil e completa. Também ajudando na identificação de erros e correções.

## Figma

No desenvolvimento da interface do usuário, foi utilizado o Figma, uma ferramenta que se destacou pela sua capacidade de colaboração em tempo real. Isso foi especialmente útil, pois permitiu que diferentes membros da equipe, de áreas como design e desenvolvimento, trabalhassem juntos no mesmo projeto sem a necessidade de trocas contínuas de arquivos. Essa integração facilitou muito o fluxo de trabalho, assegurando que todos estivessem alinhados durante o processo de criação.

Além disso, o Figma oferece uma interface intuitiva, o que torna a criação de protótipos e layouts bem mais rápida e precisa.

## Lucidchart

Para a organização de ideias e criação de fluxogramas e diagramas, foi optado pelo Lucidchart. A ferramenta provou ser bastante intuitiva, permitindo que todos os membros da equipe, independentemente de suas especializações, conseguissem utilizá-la sem dificuldade. A colaboração em tempo real foi outro ponto positivo, facilitando o processo de brainstorming e o planejamento em equipe.

Além disso, o Lucidchart possui integração com ferramentas como o Google Drive e o Microsoft Office, o que facilitou o compartilhamento dos diagramas em apresentações e documentos.

## BrModelo

Foi utilizado o BrModelo para a modelagem de dados, especialmente na criação de diagramas de Modelo Entidade-Relacionamento (ER). A interface simples da ferramenta foi essencial para a visualização e planejamento da estrutura do banco de dados de forma clara. Isso garantiu que todos os membros da equipe tivessem uma compreensão aprofundada do funcionamento dos dados, o que facilitou as tomadas de decisão ao longo do desenvolvimento.

# METODOLOGIA

## Metodologia de Pesquisa

Para validar a plataforma Velora e compreender as necessidades e desafios dos usuários, foi realizada uma pesquisa exploratória com o intuito de identificar as percepções sobre ferramentas de gestão de projetos e metodologias ágeis. O objetivo principal dessa pesquisa foi obter insights sobre como os profissionais utilizam essas metodologias, quais são os maiores desafios enfrentados, e quais funcionalidades são consideradas essenciais em uma nova plataforma de gestão.

### Instrumento de Coleta de Dados

O instrumento de coleta de dados utilizado foi um questionário online, desenvolvido de maneira a capturar tanto dados quantitativos quanto qualitativos. O questionário foi dividido em seções que cobriam tópicos variados, desde a familiaridade dos participantes com metodologias ágeis até os desafios enfrentados no uso de ferramentas de gestão de projetos.

As perguntas foram elaboradas para proporcionar uma visão ampla e detalhada das experiências dos participantes. A pesquisa continha perguntas fechadas, para gerar dados estruturados e facilitar análises quantitativas, e perguntas abertas, que permitiram uma análise mais profunda das sugestões e percepções dos respondentes. Isso possibilitou uma análise tanto dos padrões gerais quanto das nuances individuais, trazendo insights valiosos sobre os pontos de dor e as melhorias esperadas em uma plataforma de gestão.

A pesquisa foi disponibilizada por um período de 29 dias, e o tempo médio para preenchimento do questionário foi de 4 minutos e 19 segundos.

### Amostra e Perfil dos Participantes

A amostra foi composta por 59 profissionais de diversas áreas de atuação e níveis de experiência, o que garantiu uma visão diversificada sobre o uso de metodologias ágeis e ferramentas de gestão de projetos. Os participantes foram selecionados com base em suas experiências prévias no uso de metodologias ágeis, permitindo uma amostra que refletisse tanto os usuários experientes quanto aqueles que estavam iniciando o uso dessas práticas.

A maior parte dos respondentes (64%) estava na faixa etária de 20 a 29 anos, enquanto os demais se distribuíram entre as faixas de 30 a 39 anos (7%), 40 a 49 anos (14%), e outras faixas etárias. Em relação ao uso de metodologias ágeis, 51% dos participantes relataram utilizar essas metodologias no dia a dia, enquanto 29% afirmaram já ter ouvido falar, mas não utilizam regularmente, e 20% desconheciam completamente essas práticas.

### Resultados da Pesquisa

A pesquisa revelou informações importantes sobre os desafios enfrentados pelas equipes ao utilizar ferramentas de gestão de projetos. Entre os problemas mais citados, estavam:

Falta de uniformidade e terminologia comum entre equipes, o que dificulta a comunicação e a integração no uso de metodologias ágeis.

Dificuldade em manter as ferramentas atualizadas, refletindo a realidade dos projetos, especialmente em ambientes onde o fluxo de trabalho é intenso.

Interface confusa e complexa das ferramentas atuais, o que torna a usabilidade difícil para os usuários, especialmente para aqueles que não possuem conhecimento técnico aprofundado.

Resistência à mudança dentro das equipes, o que dificulta a adoção de novas ferramentas e metodologias.

Falta de conhecimento técnico, tanto no uso das ferramentas quanto na aplicação das metodologias ágeis.

Esses pontos destacam as dificuldades de implementação de novas soluções no ambiente corporativo e reforçam a importância de criar uma plataforma intuitiva e de fácil uso, como a Velora.

### Funcionalidades Valorizadas

Quando questionados sobre quais funcionalidades consideravam essenciais em uma ferramenta de gestão de projetos, os participantes destacaram:

Organização de tarefas, que foi considerada a funcionalidade mais importante.

Visualização do progresso das atividades, considerada fundamental para acompanhar o andamento dos projetos.

Acompanhamento de prazos, essencial para garantir a entrega no tempo planejado.

Colaboração entre equipes, apontada como uma característica essencial para a boa execução de projetos, permitindo que todos os membros da equipe tenham uma visão clara do que está sendo feito.

Essas informações ajudaram a definir as principais funcionalidades da plataforma Velora, garantindo que ela atenda às necessidades e expectativas dos usuários.

### Barreiras na Adoção de Novas Ferramentas

A pesquisa também revelou que a principal barreira encontrada pelas equipes ao adotar novas ferramentas de gestão é a resistência à mudança. Muitas equipes ainda hesitam em mudar suas práticas tradicionais e adotar soluções ágeis, mesmo sabendo dos benefícios que essas metodologias podem trazer. Além disso, a falta de conhecimento técnico foi citada como uma dificuldade recorrente, especialmente para equipes que não têm experiência com metodologias ágeis.

O custo elevado foi mencionado por uma parte menor dos respondentes, indicando que, embora seja uma preocupação, não é o principal fator que impede a adoção de novas ferramentas.

A plataforma Velora visa preencher essas lacunas, oferecendo uma solução prática, eficiente e adaptada às necessidades dos usuários, facilitando a adoção de metodologias ágeis no ambiente de trabalho.

### Software semelhantes

#### Propósito e Funcionalidades Principais

Velora: A Velora é uma plataforma focada em fornecer uma solução integrada para a aplicação de metodologias ágeis no ambiente corporativo. Suas funcionalidades incluem gerenciamento de Scrum, Kanban, PDCA e Lean. A ferramenta é voltada para facilitar a organização de tarefas, promover a colaboração entre equipes, otimizar a aplicação de metodologias ágeis e fornecer modelos de documentos para as equipes.

Miro: O Miro é uma plataforma de colaboração visual usada principalmente para brainstorming, mapeamento de processos, workshops e colaboração em tempo real. Suas funcionalidades são centradas em quadros brancos interativos, permitindo que os usuários criem diagramas, fluxogramas e outros tipos de visualização. É uma ferramenta altamente flexível que pode ser adaptada a diversos cenários de trabalho, como design thinking, mapeamento de jornadas de clientes e prototipação de processos.

Jira: O Jira é uma plataforma desenvolvida pela Atlassian, altamente focada em gerenciamento de projetos ágeis. Suas funcionalidades principais incluem gerenciamento de backlogs, sprints e issues (tarefas), com forte ênfase no acompanhamento de progresso e relatórios detalhados. É uma ferramenta amplamente usada por equipes de desenvolvimento de software para o acompanhamento de bugs, gerenciamento de sprints e planejamento de releases.

#### Metodologias Suportadas

Velora: A Velora se destaca por integrar várias metodologias ágeis, como Scrum, Kanban, PDCA e Lean. Ela oferece suporte tanto para o gerenciamento pessoal (com Kanban pessoal) quanto para o gerenciamento de equipe (Scrum e Kanban em nível de equipe), além de funcionalidades que facilitam o aprendizado contínuo e a melhoria de processos através do ciclo PDCA.

Miro: O Miro não é diretamente uma ferramenta de metodologias ágeis, mas pode ser usado em conjunto com elas. Ele não oferece uma estrutura específica para metodologias como Scrum ou Kanban, mas é extremamente útil para atividades de planejamento, brainstorming e visualização de processos. Equipes ágeis muitas vezes utilizam o Miro para mapear fluxos de trabalho e processos visuais.

Jira: O Jira é uma ferramenta amplamente focada em Scrum e Kanban. Ele suporta o planejamento de sprints, gestão de backlog, relatórios de burndown, boards Kanban e o acompanhamento de tarefas. Jira é uma das plataformas mais robustas para equipes que trabalham especificamente com metodologias ágeis, particularmente no contexto de desenvolvimento de software.

#### Público-Alvo

Velora: A Velora tem como público-alvo empresas que desejam aplicar metodologias ágeis de forma integrada, oferecendo uma ferramenta que pode ser utilizada tanto por equipes de TI quanto por outros setores. Além disso, a plataforma busca atender pequenas e médias empresas que ainda não têm soluções robustas de gestão ágil, mas que desejam adotar práticas eficientes para melhorar a produtividade e a colaboração.

Miro: O público-alvo do Miro é diverso, abrangendo designers, facilitadores de workshops, equipes de produto e inovação, além de qualquer equipe que precise colaborar em tempo real. O Miro é usado em sessões de design thinking, planejamento estratégico, mapeamento de processos e outras atividades que exigem colaboração visual.

Jira: O Jira é amplamente adotado por equipes de desenvolvimento de software e organizações que seguem metodologias ágeis estritamente estruturadas. Ele é especialmente útil para empresas de tecnologia e setores que demandam um alto controle sobre tarefas, bugs, issues e acompanhamento detalhado de sprints e releases.

#### Usabilidade e Interface

Velora: A Velora foi projetada com uma interface intuitiva, visando atender tanto usuários experientes em metodologias ágeis quanto iniciantes. Um dos focos é simplificar o acesso a ferramentas como Scrum e PDCA, além de fornecer uma visualização clara das tarefas em andamento e dos documentos do projeto. A proposta de Velora é criar uma ferramenta de fácil uso, mesmo para aqueles que não têm familiaridade com plataformas complexas.

Miro: O Miro possui uma interface visual altamente intuitiva, com foco em interações simples e rápidas. A curva de aprendizado é baixa, especialmente para quem já está acostumado com ferramentas de quadro branco digital. O ponto forte do Miro é sua flexibilidade, permitindo que os usuários criem seus próprios fluxos e formas de trabalho, com liberdade para personalizar a interface.

Jira: O Jira é uma ferramenta muito poderosa, mas sua interface pode parecer complexa para iniciantes. A curva de aprendizado é mais acentuada, especialmente devido à vasta quantidade de funcionalidades, customizações e integrações que oferece. No entanto, para usuários experientes, ele é uma das plataformas mais completas para o gerenciamento de projetos ágeis, especialmente no contexto de desenvolvimento de software.

#### Colaboração

Velora: A Velora promove a colaboração entre equipes, permitindo que os membros vejam o progresso das tarefas em tempo real, além de possibilitar uma comunicação contínua sobre o andamento do projeto. O quadro Kanban, tanto pessoal quanto de equipe, facilita a transparência das atividades. A integração com ferramentas de gestão de documentos também favorece o compartilhamento de informações essenciais entre os membros da equipe.

Miro: O Miro é uma das plataformas mais colaborativas disponíveis no mercado. Ele permite que múltiplos usuários editem, desenhem e façam anotações simultaneamente em um mesmo quadro, tudo em tempo real. O foco do Miro está na colaboração criativa e visual, tornando-o ideal para workshops, sessões de brainstorming e reuniões colaborativas.

Jira: O Jira também oferece um alto nível de colaboração, principalmente em ambientes de desenvolvimento de software. A equipe pode atribuir tarefas, comentar issues e acompanhar o progresso de sprints em tempo real. No entanto, o foco é mais voltado para o acompanhamento do progresso e a gestão de tarefas do que para a colaboração criativa ou visual, como no Miro.

#### Relatórios e Análise

Velora: A Velora tem como foco fornecer relatórios simples e claros sobre o progresso das tarefas, a conclusão de Sprints e a melhoria contínua dos processos. Relatórios personalizados podem ser criados para analisar a performance da equipe e identificar gargalos no fluxo de trabalho, ajudando na tomada de decisões estratégicas.

Miro: O Miro não possui ferramentas robustas de relatórios e análise, pois seu foco está na criação visual e na colaboração em tempo real. No entanto, ele pode ser integrado com outras ferramentas que fornecem essas funcionalidades.

Jira: O Jira é conhecido por seus relatórios detalhados e capacidade de análise. Ele oferece gráficos de burndown, relatórios de progresso de sprints, rastreamento de desempenho e muito mais. Essas funcionalidades tornam o Jira uma escolha robusta para equipes que precisam de uma análise precisa do desempenho das suas equipes de desenvolvimento.

#### Diferenciais

Velora: A principal vantagem da Velora é a integração de diversas metodologias ágeis (Scrum, Kanban, Lean, PDCA) em uma única plataforma. Ela também é pensada para facilitar o acesso a equipes que estão começando a adotar essas metodologias, com uma interface amigável e funcionalidades que não são excessivamente complexas. Outro diferencial é a inclusão de uma área específica para o gerenciamento de documentos, algo que nem Jira nem Miro oferecem de maneira integrada.

Miro: O grande diferencial do Miro é sua flexibilidade visual e a capacidade de colaborar em tempo real de forma extremamente fluida e criativa. Ele é uma das melhores opções para equipes que necessitam de uma plataforma de brainstorming e visualização de ideias.

Jira: O Jira se destaca por ser uma das plataformas mais robustas para o gerenciamento de projetos ágeis, com funcionalidades altamente customizáveis, integração com diversas ferramentas e capacidade de gerar relatórios detalhados. É amplamente utilizado por grandes equipes de desenvolvimento de software que precisam de controle rigoroso sobre o progresso dos projetos.

## Metodologia de Desenvolvimento

A metodologia de desenvolvimento adotada no projeto foi o Kanban, uma abordagem ágil que permite a organização e visualização clara das tarefas em andamento. O Kanban foi escolhido devido à sua flexibilidade e simplicidade, facilitando o controle do fluxo de trabalho e promovendo uma gestão eficiente das atividades. Por meio de um quadro Kanban, foi possível dividir as tarefas em categorias como "A fazer", "Em progresso" e "Concluído", o que proporcionou uma visualização mais clara do status de cada atividade ao longo do desenvolvimento.

O uso do Kanban também ajudou a equipe a priorizar tarefas e identificar possíveis gargalos no processo, permitindo ajustes rápidos e uma distribuição mais equilibrada das responsabilidades. Além disso, a abordagem ágil garantiu que, ao final de cada ciclo de desenvolvimento, fosse possível fazer revisões e adaptações com base no progresso e nas necessidades do projeto. Isso facilitou a melhoria contínua do sistema, alinhando o desenvolvimento com os objetivos e requisitos definidos inicialmente.

Essa metodologia não só melhorou a organização interna da equipe como também aumentou a transparência do processo, o que é fundamental em projetos colaborativos. Com o Kanban, a equipe conseguiu manter um ritmo constante e previsível de entregas, garantindo que o projeto evoluísse conforme esperado.

Figura 5 - Tela Fazer do Kanban durante o Desenvolvimento do Projeto

Uma imagem contendo Linha do tempo

Descrição gerada automaticamente

Fonte: Autores, 2024.

Figura 6 - Tela Em Progresso do Kanban durante o desenvolvimento do projeto

Linha do tempo

Descrição gerada automaticamente com confiança média

Fonte: Autores, 2024

Figura 7 - Tela Pronto do Kanban durante o desenvolvimento do projeto

Uma imagem contendo Retângulo

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 8 - Tela Pronto do Kanban após terminar o desenvolvimento do projeto

Linha do tempo

Descrição gerada automaticamente

Fonte: Autores, 2024

# MODELAGEM

A modelagem de software foi feita utilizando conceitos de engenharia de software. Primeiro utilizando a engenharia de requisitos assim sendo possível identificar os requisitos funcionais e não funcionais do sistema.

E após a compreensão dos requisitos foram feitos os diagramas usando o UML (Unified Modeling Language). Ainda, a partir da visão de desenvolvimento proporcionada pelos diagramas, foram construídos os protótipos para identificar ainda mais detalhadamente como seria feito o desenvolvimento do sistema.

## Requisitos Funcionais

Tabela 1 - Requisitos Funcionais

|  |  |
| --- | --- |
| Número | Detalhamento |
| REQF\_001 | O sistema deverá permitir que o usuário solicite o cadastro. |
| REQF\_002 | A aplicação deve permitir que o usuário com perfil “Gestão” efetue a manutenção (incluir, alterar, excluir) dos demais usuários. |
| REQF\_003 | A aplicação deve pedir a alteração da senha a partir do primeiro login do usuário cadastrado pela Gestão. |
| REQF\_004 | A aplicação deve conter uma tela de login, o usuário deve entrar com e-mail e senha (cadastrados ou aprovados pela Gestão). |
| REQF\_005 | Os usuários com perfil “Superior” deverão ser os únicos a poder criar um Kanban de projeto. |
| REQF\_006 | Os usuários com perfil “Superior” deverão ser os únicos a poder criar, alterar e excluir um quadro PDCA. |
| REQF\_007 | A aplicação deve permitir que os usuários possam visualizar e alterar seus próprios Kanban Pessoais. |
| REQF\_008 | A aplicação deve permitir que os usuários relacionados a um Kanban de Projeto possam alterar o status das tarefas que estão ligadas ao mesmo. |
| REQF\_009 | Os documentos de um projeto só podem ser acessados pelos usuários relacionados àquele projeto. |
| REQF\_010 | A aplicação deve permitir que os usuários possam visualizar os quadros Kanban e PDCA aos quais tenham a permissão concedida pelo “Superior”. |
| REQF\_011 | Os usuários devem poder adicionar Documentos ao Projeto com a aprovação de um usuário com perfil “Superior” ou “Lead”. |
| REQF\_012 | Um usuário poderá solicitar a entrada a um projeto, sua entrada só poderá ser aprovada pelo usuário dono do projeto. |
| REQF\_013 | Ao iniciar uma tarefa, deverá se iniciar uma votação “Pointing Poker” onde os usuários poderão dar a sugestão de horas para conclusão da tarefa. |
| REQF\_014 | O “Pointing Poker” se encerra ao momento que o usuário “Superior” ou “Lead” define a quantidade de horas para concluir a tarefa. |
| REQF\_015 | A aplicação deve gerar a partir de um quadro Kanban visualizações de métricas de dados. |
| REQF\_016 | Os usuários com perfis “Superior” e “Lead” poderão fazer a manutenção (incluir, alterar, excluir) as tarefas do Kanban de Projeto do qual fazem parte. |
| REQF\_017 | A aplicação deve ter uma área de dicas para “ensinar” aos usuários o uso das metodologias Lean e Scrum. |
| REQF\_018 | A aplicação deve ter uma tela que mostre ao usuário os projetos ao qual ele está ligado. |
| REQF\_019 | A aplicação deve ter uma tela de pesquisa para que o usuário seja capaz de ver os projetos ativos (de forma simplificada). |
| REQF\_020 | O usuário ao pesquisar e achar um projeto ao qual deva participar, deverá ter a capacidade “pedir para ingressar no projeto”. |

Fonte: Autores, 2024

## Requisitos Não Funcionais

Tabela 2 - Requisitos Não Funcionais

|  |  |
| --- | --- |
| Número | Detalhamento |
| REQNF\_001 | O sistema deverá ser de fácil aprendizagem e operação para seus usuários. |
| REQNF\_002 | O sistema deve ser ágil e fornecer quadros e documentos em tempo real e ser atualizado imediatamente quando sofrer alterações. |
| REQNF\_003 | Deverá estar disponível 24 horas por dia. |
| REQNF\_004 | Deverá ser acessível por Navegador Web. |
| REQNF\_005 | Trabalhará com banco de dados SQL Server. |
| REQNF\_006 | A senha dos usuários deve ser protegida com criptografia. |
| REQNF\_007 | O sistema deverá ter um bom nível de segurança para manter informações e documentos confidenciais. |
| REQNF\_008 | Alterações e Cadastro do usuário devem ser informados no e-mail. |

Fonte: Autores, 2024

## Regras de Negócio

Tabela 3 - Regras de Negócio

|  |  |
| --- | --- |
| Número | Detalhamento |
| RN\_001 | O e-mail para cadastro deve ser aquele provido pela empresa. |
| RN\_002 | Somente o Perfil Gestor pode fazer a manutenção do sistema. |

Fonte: Autores, 2024

## UML

UML (Unified Modeling Language ou Linguagem de Modelagem Unificada) é uma linguagem de modelagem para softwares. Contendo diversos tipos de diagramas é possível usá-los para descrever a estrutura, comportamento, objetos e outras coisas que compõem o software.

“A base para a razão do sucesso da UML é que ela nos proporciona todos os ícones de desenho necessários para capturar a maioria dos conceitos ou mecanismos que considerarmos valiosos para a resolução dos problemas reais de negócios.” (LEE, pag. 16, 2001)

Foram usados para a modelagem do “Velora” os principais diagramas de UML para descrever os comportamentos e mostrar visualmente o desenvolvimento do software.

### Caso De Uso

O Caso de uso representa como os usuários irão interagir com o software, representando de forma que o usuário e o desenvolvedor possam entender. No caso do Velora os usuários são representados pelos atores: Gestão e Funcionário, além dos atores Superior e Lead.

Figura 9 - Caso de Uso

Diagrama

Descrição gerada automaticamente

Fonte: Autores, 2024

### Narrativa De Caso De Uso

A Narrativa de Caso de Uso traz alguns detalhes sobre as interações dos atores e descreve o fluxo para a realização do caso. Para cada caso foi criada uma narrativa e atrelada aos seus respectivos atores.

Tabela 4 - Caso de Uso: 1. Solicitar Cadastro

|  |  |
| --- | --- |
| Caso de Uso | 1. Solicitar Cadastro |
| Ator Principal: | Funcionário |
| Ator Secundário: |  |
| Descrição: | O funcionário solicita a criação de uma conta no sistema. |
| Fluxo Principal: | O funcionário preenche os dados necessários para o cadastro.  O sistema valida os dados e encaminha a solicitação para o gestor.  O funcionário é informado de que a solicitação foi enviada com sucesso e está aguardando aprovação. |
| Pós-condição: | A solicitação de cadastro está registrada no sistema e aguarda aprovação do gestor. |

Fonte: Autores, 2024

Tabela 5 - Caso de Uso: 2. Aprovar Cadastro

|  |  |
| --- | --- |
| Caso de Uso | 2. Aprovar Cadastro |
| Ator Principal: | Gestão |
| Ator Secundário: |  |
| Descrição: | O gestor recebe as solicitações de cadastro e decide se aprova ou rejeita a solicitação. |
| Fluxo Principal: | O gestor visualiza a solicitação de cadastro do funcionário.  O gestor aprova ou rejeita o cadastro.  Caso aprovado, o funcionário recebe um e-mail de confirmação e pode acessar o sistema. Se rejeitado, o funcionário é informado sobre a rejeição. |
| Pós-condição: | O funcionário aprovado está habilitado para acessar o sistema, e o status do cadastro é atualizado conforme a decisão do gestor. |

Fonte: Autores, 2024

Tabela 6 - Caso de Uso: 3. Fazer Login

|  |  |
| --- | --- |
| Caso de Uso | 3. Fazer Login |
| Ator Principal: | Funcionário, Gestão |
| Ator Secundário: |  |
| Descrição: | O usuário realiza login no sistema após ter se cadastrado. |
| Fluxo Principal: | O usuário insere suas credenciais (usuário e senha).  O sistema valida as credenciais e autentica o usuário.  O usuário é redirecionado para sua área de trabalho no sistema. |
| Pós-condição: | O usuário está autenticado no sistema e tem acesso à sua área de trabalho. |

Fonte: Autores, 2024

Tabela 7 - Caso de Uso: 4. Solicitar Acesso ao Kanban de Projeto

|  |  |
| --- | --- |
| Caso de Uso | 4. Solicitar Acesso ao Kanban de Projeto |
| Ator Principal: | Funcionário |
| Ator Secundário: | Lead |
| Descrição: | O funcionário solicita acesso ao Kanban de um projeto específico. |
| Fluxo Principal: | O funcionário seleciona o projeto ao qual deseja ter acesso.  A solicitação é encaminhada ao lead do projeto.  O funcionário é informado de que a solicitação foi enviada e aguarda aprovação. |
| Pós-condição: | A solicitação de acesso ao Kanban do projeto está registrada e aguarda aprovação do lead. |

Fonte: Autores, 2024

Tabela 8 - Caso de Uso: 5. Aprovar Acesso ao Kanban de Projeto

|  |  |
| --- | --- |
| Caso de Uso | 5. Aprovar Acesso ao Kanban de Projeto |
| Ator Principal: | Lead |
| Ator Secundário: | Funcionário |
| Descrição: | O lead do projeto recebe a solicitação de acesso e decide se aprova ou rejeita. |
| Fluxo Principal: | O lead visualiza as solicitações de acesso ao projeto.  O lead aprova ou rejeita o acesso do funcionário.  O funcionário é notificado sobre a decisão. |
| Pós-condição: | O funcionário tem ou não acesso ao Kanban do projeto, dependendo da aprovação do lead. |

Fonte: Autores, 2024

Tabela 9 - Caso de Uso: 6. Manter Kanban de Projeto

|  |  |
| --- | --- |
| Caso de Uso | 6. Manter Kanban de Projeto |
| Ator Principal: | Funcionário, Lead, Superior |
| Ator Secundário: |  |
| Descrição: | O funcionário, lead ou superior pode adicionar, remover ou atualizar tarefas no Kanban de um projeto. |
| Fluxo Principal: | O ator seleciona o projeto desejado.  O ator altera o Kanban, atualizando tarefas, movendo-as entre colunas, ou adicionando novas tarefas.  O sistema atualiza as informações do Kanban e as disponibiliza para os demais usuários. |
| Pós-condição: | O Kanban do projeto está atualizado com as novas tarefas ou alterações feitas. |

Fonte: Autores, 2024

Tabela 10 - Caso de Uso: 7. Consultar Kanban de Projeto

|  |  |
| --- | --- |
| Caso de Uso | 7. Consultar Kanban de Projeto |
| Ator Principal: | Funcionário, Lead, Superior |
| Ator Secundário: |  |
| Descrição: | O ator consulta o estado atual do Kanban de um projeto específico. |
| Fluxo Principal: | O ator seleciona o projeto desejado.  O sistema exibe o Kanban com as tarefas atuais e seus respectivos estados (To Do, Doing, Done). |
| Pós-condição: | O ator visualizou o estado atual do Kanban de um projeto específico, mas não houve mudanças no sistema. |

Fonte: Autores, 2024

Tabela 11 - Caso de Uso: 8. Solicitar Acesso ao PDCA

|  |  |
| --- | --- |
| Caso de Uso | 8. Solicitar Acesso ao PDCA |
| Ator Principal: | Funcionário |
| Ator Secundário: | Superior |
| Descrição: | O funcionário solicita acesso ao quadro PDCA (Planejar, Fazer, Verificar, Agir). |
| Fluxo Principal: | O funcionário faz a solicitação de acesso ao PDCA.  O superior avalia a solicitação.  O funcionário é informado se o acesso foi concedido ou não. |
| Pós-condição: | A solicitação de acesso ao PDCA está registrada e aguarda avaliação do superior. |

Fonte: Autores, 2024

Tabela 12 - Caso de Uso: 9. Manter Quadro PDCA

|  |  |
| --- | --- |
| Caso de Uso | 9. Manter Quadro PDCA |
| Ator Principal: | Superior |
| Ator Secundário: |  |
| Descrição: | O superior gerencia o quadro PDCA, atualizando as ações de acordo com as etapas do ciclo PDCA. |
| Fluxo Principal: | O superior acessa o quadro PDCA.  O superior altera o estado das atividades no ciclo (Planejar, Fazer, Verificar, Agir).  O sistema atualiza o quadro PDCA conforme as mudanças. |
| Pós-condição: | O quadro PDCA foi atualizado de acordo com as alterações feitas pelo superior. |

Fonte: Autores, 2024

Tabela 13 - Caso de Uso: 10. Manter Kanban Pessoal

|  |  |
| --- | --- |
| Caso de Uso | 10. Manter Kanban Pessoal |
| Ator Principal: | Funcionário |
| Ator Secundário: |  |
| Descrição: | O funcionário gerencia seu próprio Kanban pessoal. |
| Fluxo Principal: | O funcionário adiciona ou altera suas tarefas pessoais no Kanban.  O sistema atualiza o Kanban pessoal conforme as alterações feitas. |
| Pós-condição: | O Kanban pessoal do funcionário foi atualizado com as novas tarefas ou mudanças. |

Fonte: Autores, 2024

Tabela 14 - Caso de Uso: 11. Manter Usuários

|  |  |
| --- | --- |
| Caso de Uso | 11. Manter Usuários |
| Ator Principal: | Gestão |
| Ator Secundário: |  |
| Descrição: | O gestor gerencia os usuários do sistema, podendo editar informações, desativar ou ativar contas. |
| Fluxo Principal: | O gestor acessa a lista de usuários do sistema.  O gestor edita as informações ou desativa/ativa usuários conforme necessário.  O sistema salva as alterações feitas pelo gestor. |
| Pós-condição: | As informações dos usuários foram atualizadas e as contas ativas/inativas foram devidamente alteradas no sistema. |

Fonte: Autores, 2024

### Diagrama de Classe

Figura 10 - Diagrama de Classe

Uma imagem contendo Linha do tempo

Descrição gerada automaticamente

Fonte: Autores, 2024

## Diagrama De Banco De Dados

A modelagem do banco de dados foi feita a partir do programa brModelo para que assim pudesse ser criado o script. É possível ver em tamanho maior os diagramas nos Apêndices I e J.

Figura 11 - Diagrama de Entidade e Relacionamento

Diagrama, Esquemático

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 12 - Modelo Lógico

Diagrama

Descrição gerada automaticamente

Fonte: Autores, 2024

## Protótipos

Para complemento da modelagem foram feitos diversos protótipos das telas principais a serem desenvolvidas. Como explica Sommerville (2018) utilizando a prototipação é possível testar opções, conceitos e assim entender melhor o problema e a solução proposta.

Figura 13 - Protótipo: Página de Contratação

Diagrama

Descrição gerada automaticamente com confiança baixa

Fonte: Autores, 2024

Figura 14 - Protótipo: Login

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 15 - Protótipo: Solicitação de Cadastro

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 16 - Protótipo: Iniciando na Aplicação

Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 17 - Protótipo: Meus Projetos

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 18 - Protótipo: Meus PDCAs

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 19 - Protótipo: Criando PDCA

Diagrama

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 20 - Protótipo: Kanban Pessoal

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 21 - Protótipo: Criando Tarefa do Kanban Pessoal

Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 22 - Protótipo: Tela Projetos

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 23 - Protótipo: Kanban de Equipe

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Autores, 2024

Figura 24 - Protótipo: Criando Tarefa no Kanban de Equipe

**Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente**

Fonte: Autores, 2024

Figura 25 - Protótipo: Kanban de Equipe - Documentos

**Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente**

Fonte: Autores, 2024

Figura 26 - Protótipo: Kanban de Equipe - Métricas

**Gráfico, Gráfico de cascata

Descrição gerada automaticamente**

Fonte: Autores, 2024

Figura 27 - Protótipo: Kanban de Equipe - Backlog

**Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente**

Fonte: Autores, 2024

Figura 28 - Protótipo: Perfil

**Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente**

Fonte: Autores, 2024

Figura 29 - Protótipo: Dicas Ágeis

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Autores, 2024

# DESENVOLVIMENTO

O desenvolvimento do projeto foi feito separadamente o Front-End, Back-End e Banco de Dados. O Front-End feito em Angular usando a IDE Visual Studio Code, o Back-End em .NET usando a IDE Visual Studio e o Banco de Dados no SQL Server pelo SQL Server Management Studio.

## Criação de Identidade Visual

A escolha do nome Velora combina os conceitos de "velo", que remete a velocidade e agilidade, com uma sonoridade que evoca força, organização e modernidade, com "Aurora", trazendo a ideia de agilidade e novos começos. Criando uma identidade forte e dinâmica para o software.

Com base nos conceitos de agilidade, velocidade e mitologia. A paleta de cores utiliza tons de roxo e azul, simbolizando sabedoria e tecnologia, com toques de dourado para representar excelência e eficiência.

Figura 30 - Logo da Aplicação: Velora

Ícone

Descrição gerada automaticamente com confiança baixa

Fonte: Autores, 2024

O Logo reúne o nome e a paleta de cores, sua criação traz a ideia do ciclo de agilidade em seu desenho.

## Repositórios

Todas as informações gerais do projeto podem ser acessadas no repositório do GitHub “https://github.com/DevPam/TCC\_CC\_2024”

Para visualizar os códigos Front-End e Back-End eles podem ser acessados na parte do repositório GitHub: “https://github.com/DevPam/TCC\_CC\_2024/tree/main/Code”

E Para o Banco de Dados: “https://github.com/DevPam/TCC\_CC\_2024/tree/main/BD”

# TESTES E RESULTADOS

Tabela 15 - Caso de Teste: Solicitar Cadastro

|  |  |
| --- | --- |
| Requisito | REQF\_001: Solicitar Cadastro |
| Caso de Teste 001.01: | Verificar se o sistema permite ao usuário solicitar o cadastro preenchendo as informações obrigatórias. |
| Pré-condição: | O usuário não possui cadastro no sistema. |
| Passos: | Acessar a página de cadastro.  Preencher os campos obrigatórios.  Submeter a solicitação de cadastro. |
| Resultado Esperado: | O sistema exibe uma mensagem de sucesso indicando que a solicitação foi enviada para aprovação. |

Fonte: Autores, 2024

Tabela 16 - Caso de Teste: Manutenção de Usuários

|  |  |
| --- | --- |
| Requisito | REQF\_002: Manutenção de Usuários (Perfil Gestão) |
| Caso de Teste 002.01: | Verificar se o usuário com perfil "Gestão" pode incluir novos usuários. |
| Pré-condição: | O usuário logado possui perfil "Gestão". |
| Passos: | Acessar a tela de manutenção de usuários.  Preencher as informações do novo usuário.  Submeter o formulário. |
| Resultado Esperado: | O novo usuário é incluído no sistema com sucesso. |
| Caso de Teste 002.02: | Verificar se o usuário com perfil "Gestão" pode alterar dados de um usuário existente. |
| Pré-condição: | O usuário logado possui perfil "Gestão". |
| Passos: | Acessar a lista de usuários.  Selecionar um usuário para editar.  Alterar as informações do usuário.  Salvar as alterações. |
| Resultado Esperado: | As alterações são salvas com sucesso e refletidas no sistema. |
| Caso de Teste 002.03: | Verificar se o usuário com perfil "Gestão" pode excluir um usuário. |
| Pré-condição: | O usuário logado possui perfil "Gestão". |
| Passos: | Acessar a lista de usuários.  Selecionar um usuário para excluir.  Confirmar a exclusão. |
| Resultado Esperado: | O usuário é excluído do sistema. |

Fonte: Autores, 2024

Tabela 17 - Caso de Teste: Alteração de Senha

|  |  |
| --- | --- |
| Requisito | REQF\_003: Alteração de Senha no Primeiro Login |
| Caso de Teste 003.01: | Verificar se o sistema solicita a alteração de senha no primeiro login do usuário. |
| Pré-condição: | O usuário foi cadastrado pela Gestão. |
| Passos: | Fazer login no sistema com a senha provisória.  Verificar se o sistema solicita a alteração de senha.  Alterar a senha. |
| Resultado Esperado: | O sistema solicita a alteração de senha e a senha é alterada com sucesso. |

Fonte: Autores, 2024

Tabela 18 - Caso de Teste: Login

|  |  |
| --- | --- |
| Requisito | REQF\_004: Tela de Login |
| Caso de Teste 004.01: | Verificar se a tela de login permite a autenticação com e-mail e senha aprovados. |
| Pré-condição: | O usuário tem um e-mail e senha cadastrados ou aprovados pela Gestão. |
| Passos: | Acessar a tela de login.  Inserir o e-mail e senha válidos.  Tentar realizar o login. |
| Resultado Esperado: | O login é realizado com sucesso e o usuário é redirecionado para sua área de trabalho. |

Fonte: Autores, 2024

Tabela 19 - Caso de Teste: Criação de Kanban de Projeto

|  |  |
| --- | --- |
| Requisito | REQF\_005: Criação de Kanban de Projeto (Perfil Superior) |
| Caso de Teste 005.01: | Verificar se apenas usuários com perfil “Superior” podem criar um Kanban de Projeto. |
| Pré-condição: | O usuário logado possui perfil "Superior". |
| Passos: | Acessar a funcionalidade de criação de Kanban de Projeto.  Criar um novo Kanban de Projeto. |
| Resultado Esperado: | O Kanban é criado com sucesso. |
| Caso de Teste 005.02: | Verificar se usuários sem perfil “Superior” não podem criar um Kanban de Projeto. |
| Pré-condição: | O usuário logado não possui perfil "Superior". |
| Passos: | Tentar acessar a funcionalidade de criação de Kanban de Projeto. |
| Resultado Esperado: | O sistema não mostra a funcionalidade de criação de Kanban de projeto. |

Fonte: Autores, 2024

Tabela 20 - Caso de Teste: Manutenção de Quadro PDCA

|  |  |
| --- | --- |
| Requisito | REQF\_006: Manutenção de Quadro PDCA (Perfil Superior) |
| Caso de Teste 006.01: | Verificar se apenas usuários com perfil “Superior” podem criar, alterar e excluir um quadro PDCA. |
| Pré-condição: | O usuário logado possui perfil "Superior". |
| Passos: | Acessar a funcionalidade de manutenção de quadro PDCA.  Criar, alterar e excluir um quadro PDCA. |
| Resultado Esperado: | O usuário consegue realizar as ações com sucesso. |

Fonte: Autores, 2024

Tabela 21 - Caso de Teste: Kanban Pessoal

|  |  |
| --- | --- |
| Requisito | REQF\_007: Kanban Pessoal |
| Caso de Teste 007.01: | Verificar se o usuário pode visualizar e alterar seu próprio Kanban Pessoal. |
| Pré-condição: | O usuário está logado. |
| Passos: | Acessar o Kanban Pessoal.  Alterar o status de uma tarefa. |
| Resultado Esperado: | O usuário consegue visualizar e alterar seu Kanban Pessoal com sucesso. |

Fonte: Autores, 2024

Tabela 22 - Alteração de Status do Kanban de Projeto

|  |  |
| --- | --- |
| Requisito | REQF\_008: Alteração de Status no Kanban de Projeto |
| Caso de Teste 008.01: | Verificar se os usuários relacionados a um Kanban de Projeto podem alterar o status das tarefas. |
| Pré-condição: | O usuário está relacionado ao Kanban de Projeto. |
| Passos: | Acessar o Kanban de Projeto.  Alterar o status de uma tarefa. |
| Resultado Esperado: | O status da tarefa é alterado com sucesso. |

Fonte: Autores, 2024

Tabela 23 - Caso de Teste: Documentos de Projeto

|  |  |
| --- | --- |
| Requisito | REQF\_009: Acesso a Documentos de Projeto |
| Caso de Teste 009.01: | Verificar se apenas usuários relacionados a um projeto podem acessar os documentos do projeto. |
| Pré-condição: | O usuário está relacionado ao projeto. |
| Passos: | Tentar acessar os documentos do projeto. |
| Resultado Esperado: | O acesso aos documentos é permitido. |

Fonte: Autores, 2024

Tabela 24 - Caso de Teste: Visualização de Kanban e PDCA

|  |  |
| --- | --- |
| Requisito | REQF\_010: Visualização de Quadros Kanban e PDCA |
| Caso de Teste 010.01: | Verificar se os usuários podem visualizar os quadros Kanban e PDCA aos quais tenham permissão. |
| Pré-condição: | O usuário tem permissão para visualizar os quadros. |
| Passos: | Acessar os quadros Kanban e PDCA. |
| Resultado Esperado: | O usuário consegue visualizar os quadros com sucesso. |

Fonte: Autores, 2024

Tabela 25 - Caso de Teste: Adicionar Documentos

|  |  |
| --- | --- |
| Requisito | REQF\_011: Adicionar Documentos ao Projeto |
| Caso de Teste 011.01: | Verificar se os usuários podem adicionar documentos a um projeto, com a aprovação de um "Superior" ou "Lead". |
| Pré-condição: | O usuário está relacionado ao projeto. |
| Passos: | Tentar adicionar um documento ao projeto.  Solicitar aprovação ao "Superior” ou "Lead". |
| Resultado Esperado: | O documento é adicionado após a aprovação. |

Fonte: Autores, 2024

Tabela 26 - Caso de Teste: Solicitar Entrada no Projeto

|  |  |
| --- | --- |
| Requisito | REQF\_012: Solicitar Entrada em um Projeto |
| Caso de Teste 012.01: | Verificar se um usuário pode solicitar a entrada em um projeto. |
| Pré-condição: | O usuário não está relacionado ao projeto. |
| Passos: | Acessar a lista de projetos.  Solicitar entrada em um projeto. |
| Resultado Esperado: | A solicitação é enviada com sucesso e aguarda aprovação. |

Fonte: Autores, 2024

Tabela 27 - Caso de Teste: Pointing Poker

|  |  |
| --- | --- |
| Requisito | REQF\_013: Iniciar Votação Pointing Poker |
| Caso de Teste 013.01: | Verificar se a votação "Pointing Poker" é iniciada ao iniciar uma tarefa. |
| Pré-condição: | O usuário está relacionado ao projeto. |
| Passos: | Iniciar uma tarefa no Kanban.  Verificar se a votação "Pointing Poker" é iniciada. |
| Resultado Esperado: | A votação "Pointing Poker" é iniciada automaticamente. |

Fonte: Autores, 2024

Tabela 28 - Caso de Teste: Encerramento do Pointing Poker

|  |  |
| --- | --- |
| Requisito | REQF\_014: Encerramento do Pointing Poker |
| Caso de Teste 014.01: | Verificar se a votação "Pointing Poker" é encerrada quando o "Superior" ou “Lead” define as horas da tarefa. |
| Pré-condição: | Definir a quantidade de horas para uma tarefa. |
| Passos: | Iniciar uma tarefa no Kanban.  Verificar se a votação "Pointing Poker" é iniciada. |
| Resultado Esperado: | A votação é encerrada e o tempo é definido. |

Fonte: Autores, 2024

Tabela 29 - Caso de Teste: Métricas

|  |  |
| --- | --- |
| Requisito | REQF\_015: Visualizações de Métricas de Dados |
| Caso de Teste 015.01: | Verificar se a aplicação gera visualizações de métricas de dados a partir de um quadro Kanban. |
| Pré-condição: | O quadro Kanban está preenchido. |
| Passos: | Acessar o Kanban de Projeto.  Gerar visualizações de métricas. |
| Resultado Esperado: | O sistema gera visualizações das métricas de dados. |

Fonte: Autores, 2024

Tabela 30 - Caso de Teste: Manutenção Kanban de Projetos

|  |  |
| --- | --- |
| Requisito | REQF\_016: Manutenção de Tarefas no Kanban de Projeto |
| Caso de Teste 016.01: | Verificar se usuários com perfil "Superior" ou "Lead" podem fazer a manutenção das tarefas do Kanban de Projeto. |
| Pré-condição: | O usuário tem perfil "Superior" ou "Lead". |
| Passos: | Acessar o Kanban de Projeto.  Criar, alterar ou excluir tarefas. |
| Resultado Esperado: | As tarefas são mantidas conforme as alterações feitas. |

Fonte: Autores, 2024

Tabela 31 - Caso de Teste: Dicas de Ágil

|  |  |
| --- | --- |
| Requisito | REQF\_017: Área de Dicas de Lean e Scrum |
| Caso de Teste 017.01: | Verificar se a aplicação possui uma área de dicas para ensinar o uso de metodologias Lean e Scrum. |
| Pré-condição: | O usuário está logado. |
| Passos: | Acessar a área de dicas. |
| Resultado Esperado: | O usuário visualiza dicas sobre Lean e Scrum. |

Fonte: Autores, 2024

Tabela 32 - Caso de Teste: Tela de Projetos

|  |  |
| --- | --- |
| Requisito | REQF\_018: Tela de Projetos do Usuário |
| Caso de Teste 018.01: | Verificar se a aplicação exibe ao usuário os projetos aos quais ele está ligado. |
| Pré-condição: | O usuário está logado e relacionado a projetos. |
| Passos: | Acessar a tela de projetos. |
| Resultado Esperado: | O usuário visualiza os projetos aos quais está ligado. |

Fonte: Autores, 2024

Tabela 33 - Caso de Teste: Pesquisa de Projetos

|  |  |
| --- | --- |
| Requisito | REQF\_019: Tela de Pesquisa de Projetos Ativos |
| Caso de Teste 019.01: | Verificar se a aplicação possui uma tela de pesquisa para ver projetos ativos. |
| Pré-condição: | O usuário está logado. |
| Passos: | Acessar a tela de pesquisa de projetos.  Pesquisar por projetos ativos. |
| Resultado Esperado: | O sistema exibe uma lista de projetos ativos de forma simplificada. |

Fonte: Autores, 2024

Tabela 34 - Caso de Teste: Solicitação para um Projeto

|  |  |
| --- | --- |
| Requisito | REQF\_020: Solicitação para Ingressar em Projeto |
| Caso de Teste 020.01: | Verificar se o usuário pode solicitar ingresso em um projeto após encontrá-lo na pesquisa. |
| Pré-condição: | O usuário não está relacionado ao projeto. |
| Passos: | Pesquisar um projeto ativo.  Solicitar ingresso no projeto. |
| Resultado Esperado: | A solicitação de ingresso é enviada com sucesso. |

Fonte: Autores, 2024

# CONCLUSÃO

Tendo em mente os objetivos definidos no início do projeto e o desenvolvimento da aplicação Velora, pode-se afirmar que os resultados obtidos foram satisfatórios e atenderam às expectativas. Todo o processo de desenvolvimento baseou-se na aplicação de metodologias ágeis e na escolha prudente de ferramentas tecnológicas, garantindo o bom funcionamento da plataforma, além de facilidade de uso.

Embora haja muitos desafios ao tentar incorporar várias metodologias ágeis em uma única plataforma, foi possível superar esses obstáculos por meio de uma abordagem iterativa e colaborativa. A plataforma Velora, conforme planejada, centraliza metodologias como Scrum, Kanban, PDCA e Lean em uma solução de gestão de projetos fácil de usar. As ferramentas foram escolhidas com base em uma análise detalhada das necessidades da equipe de desenvolvimento e dos usuários finais, com o intuito de garantir que a aplicação não apenas apoiasse a organização e o acompanhamento das atividades, mas também promovesse a melhoria contínua e o trabalho em equipe.

Em termos de desempenho, foram realizados testes que demonstraram a robustez da aplicação na adaptação às diferentes necessidades dos usuários, integrando diversas metodologias de forma coesa. No entanto, como em todo projeto, há espaço para melhorias e futuras expansões. Poderá ser considerada a implementação de novas funcionalidades, como a inclusão de relatórios mais detalhados sobre a evolução dos projetos e a integração de novos módulos que permitam uma personalização ainda maior da plataforma, de acordo com as necessidades específicas de cada organização.

Em resumo, a aplicação Velora foi uma boa realização dos objetivos estipulados, oferecendo uma solução prática e acessível para a gestão de projetos ágeis. Além disso, a plataforma pode sempre evoluir, tornando-se um software indispensável no ambiente corporativo.

###### REFERÊNCIAS

AMAZON. O que o .net? –Explicação sobre o .net – AWS. Disponível em: <https://aws.amazon.com/pt/what-is/net/>. Acesso em: 24 set. 2024.

ANGULAR. Angular. Disponível em: <https://angular.dev/tools/cli>. Acesso em: 24 set. 2024.

ARTIA. Kanban: guia completo sobre a metodologia. Disponível em: https://homolog.artia.com/blog/kanban-guia/. Acesso em: 18 out. 2024.

Atlassian. O que é Kanban? Disponível em: <https://www.atlassian.com/br/agile/kanban>. Acesso em: 01 ago. 2024.

BARBOSA, Amanda Teixeira; SILVA, César Carriço da; CAETANO, Rebecca Leal; SILVA, Deborah Paredes Soares da; BARBOSA, Júlio Vianna; PINTO, Zeneida Teixeira. METODOLOGIAS ÁGEIS: E SUA APLICABILIDADE NA ÁREA DE MARKETING. Revista Ibero-Americana de Humanidades, Ciências e Educação, [S. l.], v. 8, n. 3, p. 1659–1669, 2022. DOI: 10.51891/rease.v8i3.4751. Disponível em: <https://periodicorease.pro.br/rease/article/view/4751>. Acesso em: 20 jul. 2024.

BENNER. Scrum: o que é e como funciona. Disponível em: https://www.benner.com.br/tecnologia/2023/12/scrum/. Acesso em: 18 out. 2024.

BOOCH, G. et al. UML guia do usuário. [s.l.] Rio De Janeiro Elsevier Campus, 2006.

BRIALES, Julio Aragon. Lean business: melhoria contínua e transformação cultural nas organizações. 1. ed. Curitiba: Intersaberes, 2022. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 23 jul. 2024.

BRMODELO. BR Modelo Web - Documentação de uso. Disponível em: <https://docs.brmodeloweb.com/#/>. Acesso em: 24 set. 2024.

Como Aplicar a Metodologia Lean Na Sua Empresa. Disponível em: <https://www.totvs.com/blog/negocios/metodologia-lean/>. Acesso em: 23 jul. 2024.

Definição e Aplicações - Lean Intitute Brasil. Disponível em: <https://www.lean.org.br/o-que-e-lean.aspx>. Acesso em: 23 jul. 2024.

FIGMA. Figma: a ferramenta de design de interface colaborativa e online. Disponível em: <https://www.figma.com/pt-br/>. Acesso em: 23 set. 2024.

IDE do Visual Studio, Editor de Código, Azure DevOps e App Center - Visual Studio. Disponível em: <https://visualstudio.microsoft.com/pt-br/>. Acesso em: 25 ago. 2024.

JSON.ORG. JSON. Disponível em: <https://www.json.org/json-en.html>. Acesso em: 23 set. 2024.

KANBANCHI. Metodologia de gestão Lean: tudo o que você precisa saber. Disponível em: https://www.kanbanchi.com/pt/blog/metodologia-de-gestao-lean. Acesso em: 18 out. 2024.

KJO-MSFT. Introdução ao Visual Studio Code - Training. Disponível em: <https://learn.microsoft.com/pt-br/training/modules/introduction-to-visual-studio-code/>. Acesso em: 23 out. 2024.

LEE, R. C.; TEPFENHART, W. M. UML e C++: guia prático de desenvolvimento orientado a objeto. São Paulo: Pearson, 2001. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 26 abril 2024.

LUCID SOFTWARE INC. Lucidchart overview | Lucidchart. Disponível em: <https://www.lucidchart.com/pages/product>. Acesso em: 21 out. 2024.

LUCIDCHART. O que é um diagrama UML? Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-uml>.‌ Acesso em: 26 abril 2024.

MARKINGMYNAME. SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS). Disponível em: <https://learn.microsoft.com/pt-br/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>. Acesso em: 23 set. 2024.

MICROSOFT. IDE do Visual Studio, Editor de Código, Azure DevOps e App Center - Visual Studio. Disponível em: <https://visualstudio.microsoft.com/pt-br/>. Acesso em: 23 set. 2024.

MICROSOFT. What is .NET? An open-source developer platform. Disponível em: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>. Acesso em: 23 set. 2024.

MUNIZ, Antonio; IRIGOYEN, Analia; MAFRA, Cleiton; TRIERVEILER, Fábio; VILLANOVA, Guilherme. Jornada Kanban na prática: unindo teoria e prática com o objetivo de acelerar o aprendizado do Kanban para quem está iniciando. 1. ed. Rio de Janeiro: Brasport, 2021. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 01 ago. 2024.

PAGE-JONES, Meilir. Fundamentos do desenho orientado a objeto com UML. São Paulo: Pearson, 2001. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 20 set. 2024.

PICHLER, Roman. Agile Product Management with Scrum: Creating Products that Customers Love. Upper Saddle River: Addison-Wesley, 2010.

Princípios Lean: Quais São Os Que Todos Deveriam saber? - FM2S. Disponível em: <https://www.fm2s.com.br/blog/principios-lean>. Acesso em: 24 jul. 2024.

RWESTMSFT. O que é o SQL Server? - SQL Server. Disponível em: <https://learn.microsoft.com/pt-br/sql/sql-server/what-is-sql-server?view=sql-server-ver16>. Acesso em: 23 set. 2024.

SCHWABER, Ken. Scrum: A Arte de Fazer o Dobro do Trabalho na Metade do Tempo. Rio de Janeiro: Sextante, 2004.

SOMMERVILLE, Ian. Engenharia de software. 10. ed. São Paulo, SP: Pearson, 2018. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 20 set. 2024.

SUTHERLAND, Jeff. Scrum: A Revolução Ágil para o Seu Negócio. São Paulo: Leya, 2019.

TERRYGLEE. Visão geral do Visual Studio. Disponível em: <https://learn.microsoft.com/pt-br/visualstudio/get-started/visual-studio-ide?view=vs-2022>. Acesso em: 25 ago. 2024.

Totvs. O que é Kanban? Disponível em: <https://www.totvs.com/blog/negocios/kanban/#:~:text=O%20termo%20%E2%80%9CKanban%E2%80%9D%20%C3%A9%20de,ele%20se%20move%20pelo%20processo>. Acesso em: 01 ago. 2024.

VOITTO. O que é o ciclo PDCA? Disponível em: https://voitto.com.br/blog/artigo/o-que-e-o-ciclo-pdca. Acesso em: 18 out. 2024.

###### APÊNDICE A – GRÁFICO DA QUESTÃO 1 DO QUESTIONÁRIO DE PESQUISA

Gráfico, Gráfico de barras

Descrição gerada automaticamente

###### APÊNDICE B – GRÁFICO DA QUESTÃO 2 DO QUESTIONÁRIO DE PESQUISA

Gráfico

Descrição gerada automaticamente

###### APÊNDICE C – GRÁFICO DA QUESTÃO 3 DO QUESTIONÁRIO DE PESQUISA

Gráfico, Gráfico de pizza

Descrição gerada automaticamente

###### APÊNDICE D – GRÁFICO DA QUESTÃO 4 DO QUESTIONÁRIO DE PESQUISA

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

###### APÊNDICE E – GRÁFICO DA QUESTÃO 5 DO QUESTIONÁRIO DE PESQUISA

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

###### APÊNDICE F – GRÁFICO DA QUESTÃO 6 DO QUESTIONÁRIO DE PESQUISA

Gráfico, Gráfico de pizza

Descrição gerada automaticamente

###### APÊNDICE G – GRÁFICO DA QUESTÃO 7 DO QUESTIONÁRIO DE PESQUISA

Tabela

Descrição gerada automaticamente

###### APÊNDICE H – GRÁFICO DA QUESTÃO 8 DO QUESTIONÁRIO DE PESQUISA

Tabela

Descrição gerada automaticamente

###### APÊNDICE I – DIAGRAMA DE ENTIDADE E RELACIONAMENTO

Diagrama, Esquemático

Descrição gerada automaticamente

###### APÊNDICE J – MODELO LÓGICO – BANCO DE DADOS

Diagrama

Descrição gerada automaticamente