



**UNIVERSIDADE DE ITAÚNA
PRÓ-REITORIA DE ENSINO
COORDENAÇÃO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

DAVI VENTURA CARDOSO PERDIGÃO

**DESENVOLVIMENTO DE SOFTWARE PARA MOBILIZAÇÃO
EMPRESARIAL: ESTRATÉGIAS DE ENGENHARIA DE
SOFTWARE E PROCESSOS DE GERENCIAMENTO ÁGIL**

**ITAÚNA
2023**

DAVI VENTURA CARDOSO PERDIGÃO

**DESENVOLVIMENTO DE SOFTWARE PARA MOBILIZAÇÃO
EMPRESARIAL: ESTRATÉGIAS DE ENGENHARIA DE SOFTWARE E
PROCESSOS DE GERENCIAMENTO ÁGIL**

Projeto submetido à Coordenadoria do Curso de Bacharelado em Ciência da Computação da Universidade de Itaúna - Campus Verde, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Área de Pesquisa: Engenharia de Software

Orientador: Professor Me. Adriano Benigno Moreira

ITAÚNA
2023

Resumo

Este estudo trata acerca do processo de gerência do desenvolvimento de um *software* para mobilização empresarial, de forma a representar um impacto positivo em um curto período de tempo. A empresa *Alfa Engenharia e Montagens Industriais* é apresentada como um contexto em que a contratação de colaboradores em grande escala e a mobilização dos colaboradores existentes são desafios. Até então, o processo é realizado de forma manual, mesmo que por meio de planilhas aperfeiçoadas durante anos, ainda assim exige-se muito tempo e mão de obra dos setores de recursos humanos e administrativo.

Diante disso, surge a necessidade de desenvolver uma solução que otimize o tempo gasto nesse processo descrito. A hipótese levantada é que, a aplicação de práticas de Engenharia de *Software* e metodologias ágeis pode resultar em um produto de alta qualidade, que atenda às necessidades do cliente e aumente a produtividade dos setores envolvidos. Sendo assim, o objetivo é demonstrar a eficácia da aplicação dessas estratégias no desenvolvimento de uma solução.

Para atingir esses objetivos, serão investigadas diferentes metodologias ágeis e práticas de Engenharia de *Software*, além da análise dos requisitos e necessidades dos usuários. Será desenvolvida uma arquitetura de *software* adequada, com foco na segurança das informações e na escalabilidade do sistema. Também serão realizados testes de aceitação e validação para garantir a qualidade da solução.

Este estudo contribuirá para o conhecimento científico e prático no campo da Engenharia de *Software* e gestão de projetos, fornecendo recomendações e boas práticas para um desenvolvimento ágil, baseado nessa solução aplicada em um contexto real.

Palavras-chave: mobilização, metodologias ágeis, Engenharia de *Software*, desenvolvimento de *software*, gestão de projetos, produtividade.

Lista de Figuras

1	Pesquisa de Mercado - Valores <i>Softwares</i> de Mobilização	30
2	Principais Menus do <i>Bitrix24</i>	31
3	Kanban Aplicado no Módulo de Mobilização	31
4	Cronograma das Atividades MVP (Mínimo Produto Viável)	32
5	Gráfico de <i>Gantt</i> apenas com as tarefas master do Módulo de Mobilização	33
6	Trecho de <i>Gantt</i> em Plena Aplicação no Módulo de Mobilização	34
7	Requisitos Funcionais do Módulo de Mobilização	35
8	Requisitos Não Funcionais do Módulo de Mobilização	35
9	Regras de Negócio do Módulo de Mobilização	36
10	BPMN <i>Software</i> de Mobilização	37
11	EAP do Módulo de Mobilização	38
12	Esboço Primeiro Diagrama de Classes	39
13	Primeiro Diagrama de Classes	41
14	Diagrama de Recursos	44
15	Menu e submenus de Mobilização	46
16	Formulário Externo	47
17	Formulário no Alfald	48

18	Tela de Controle de Mobilização	48
19	Tela de Requisição de Pessoal	49
20	Tela de ASO (Atestado de Saúde Ocupacional)	50

Sumário

1	Introdução	8
1.1	Problema de Pesquisa	9
1.2	Hipótese	10
1.3	Motivação	10
1.4	Objetivos	13
1.4.1	Objetivos Específicos	13
1.5	Metodologia	15
2	Trabalhos relacionados	17
2.0.1	<i>Agile Development for Enterprise Mobility: A Systematic Literature Review</i>	17
2.0.2	<i>Agile Software Development Practices in Mobile Application Development: A Systematic Literature Review</i>	18
2.0.3	Metodologias ágeis de gerenciamento de projetos para startups	18
2.0.4	Implantação da metodologia ágil <i>SCRUM</i> em um ambiente de desenvolvimento	19
3	Referencial Teórico	20

3.1	Transformação digital	20
3.2	Metodologias ágeis	21
3.3	Práticas de Engenharia de <i>Software</i>	22
4	Plano de Trabalho	24
4.1	Cronograma	24
5	Resultados Preliminares	28
5.1	Pesquisa de mercado	28
5.2	Definição de metodologias ágeis a serem utilizadas	30
5.3	Planejamento e cronograma	32
5.4	Definição do escopo e requisitos	34
5.4.1	Levantamento de requisitos funcionais e não funcionais .	34
5.4.2	Definição das regras de negócio	36
5.4.3	BPMN (<i>Business Process Model and Notation</i>)	37
5.4.4	EAP (Estrutura Analítica do Projeto)	37
5.4.5	Diagrama de classes	38
5.4.6	Definição de recursos	42
5.5	Elaboração e validação dos protótipos	45
	Referências Bibliográficas	51

1 *Introdução*

A transformação digital é capaz de otimizar diversos processos internos dentro de um negócio, adaptando a sua estrutura para se alinhar com as principais atualizações do mercado. Basicamente, trata-se de aproveitar bem as ferramentas tecnológicas para melhorar a produtividade e a eficácia das atividades organizacionais. A aplicação de diversas ferramentas digitais implica em uma dinâmica que proporciona mais praticidade no cotidiano. Como consequência, o rendimento do trabalho é maior, os processos são mais eficazes e a empresa consegue ser mais bem-sucedida.

Segundo estudo apresentado pela Parkes (2018) à CA Technologies, **81%** dos executivos acreditam que as metodologias ágeis são necessárias para uma transformação digital bem-sucedida e melhor desempenho do negócio. Com base nisso, para auxiliar na gerência do projeto, utilizou-se metodologias ágeis como forma de otimizar as entregas. Trata-se de métodos focados no fracionamento de entregas para o cliente final em ciclos menores. Com isso, comprovadamente, eventuais problemas podem ser corrigidos rapidamente e o planejamento pode ser revisto.

A gerência de mão de obra é uma tarefa essencial no contexto de uma empresa, já que são os funcionários que ditam o rumo do empreendimento, e fatalmente, se ele irá ser bem ou mal sucedido. Sendo assim, os gastos com contratação, benefícios, encargos sociais e trabalhistas, demissão, bem como a própria remuneração de uma mão de obra, seja ela direta ou indireta, são

parte dos principais custos de uma organização.

Frente aos assuntos abordados até então, encontra-se o contexto da empresa *Alfa Engenharia e Montagens Industriais*, na qual o número de colaboradores até dezembro de **2022** era **1.450**, e que em **2023** estima-se esse número a atingir **4.000**, com pico de admissão de 2.500 colaboradores nos primeiros dois meses do ano, faz-se necessário uma alternativa de forma a otimizar essa contratação em grande escala, além da mobilização dos colaboradores já atuantes. Diante disso, a proposta deste trabalho é discorrer à respeito da metodologia por trás da gerência do desenvolvimento do projeto de mobilização implementado pela equipe de desenvolvimento de *software* dessa empresa, em conjunto com o entendimento de todo o processo por parte do setor responsável pelo mesmo.

1.1 Problema de Pesquisa

O método utilizado até então pela empresa para o processo de mobilização trata-se de uma planilha no *Google Sheets*. Essa planilha passou por diversas melhorias ao longo dos anos, com diversas fórmulas e demonstrativos implementados por diferentes pessoas, chegando a ter mais de 100 páginas, porém, esse método exige mais tempo de mão de obra por parte dos colaboradores dos setores de recursos humanos, departamento pessoal e administrativo. É de se indagar também que, com o pico de admissões citado anteriormente, diversos colaboradores desses setores precisariam inserir uma quantidade demasiada de informações simultâneamente, podendo ocasionar uma lentidão na planilha, ou até mesmo seu colapso.

Frente a esse cenário, como desenvolver uma solução em tempo hábil, que cause um impacto positivo no tempo de mão de obra gasto nesse processo de mobilização? Quais processos podem ser automatizados ou delegados? E até mesmo, o quão eficaz e completa essa solução deve ser, dado que

estará substituindo uma planilha que fora construída e melhorada durante tanto tempo?

1.2 Hipótese

Ao utilizar práticas de Engenharia de *Software*, tais como análise de requisitos, *design* de *software*, programação e testes, em conjunto com metodologias ágeis, é possível que a equipe de desenvolvimento entregue um produto de alta qualidade que atenda às necessidades do cliente e aumente a produtividade.

Com um *software* de mobilização, a organização alcançará um público mais amplo, independentemente da localização geográfica, atingindo pessoas e ampliando sua base apoiadores. Com essa solução, será possível fazer a gestão das tarefas de forma mais eficiente, atribuindo tarefas aos colaboradores de forma rápida e simples, e monitorando o progresso das atividades em tempo real. Além disso, é possível manter uma comunicação mais clara e direta com os colaboradores, garantindo que todos estejam alinhados em relação às metas e objetivos da empresa.

1.3 Motivação

Comprovar à respeito da importância de processos de gerenciamento ágil e de etapas da Engenharia de *Software* em um projeto de desenvolvimento, para garantir o sucesso do mesmo. Nesse contexto, as motivações apresentam-se da seguinte forma:

1. Eficiência e produtividade: As estratégias ágeis, como o Scrum e o Kanban, aliadas às etapas de Engenharia de *Software*, permitem um

fluxo contínuo de trabalho, maximizando a eficiência e a produtividade da equipe de desenvolvimento.

2. Entrega incremental de valor: Entregar incrementos funcionais do *software* em curtos períodos de tempo. Isso possibilita que os usuários tenham acesso antecipado aos benefícios do *software* e proporciona um *feedback* valioso para direcionar o desenvolvimento;
3. Colaboração e comunicação efetiva: Incentivar a colaboração e a comunicação contínua entre os membros da equipe de desenvolvimento, bem como com os *stakeholders*. Isso promove a transparência, a troca de conhecimento e a identificação precoce de possíveis problemas;
4. Garantia de qualidade: As etapas de Engenharia de *Software*, como a definição do escopo e requisitos, EAP (Estrutura Analítica do Projeto), gestão de mudanças e riscos e os testes, asseguram a qualidade da solução desenvolvida. Isso reduz erros e retrabalho, resultando em um produto final mais confiável e robusto;
5. Satisfação do cliente: Ao adotar o gerenciamento ágil e seguir etapas de Engenharia de *Software*, o foco no cliente é ampliado, resultando em uma solução que atenda às suas necessidades e expectativas. Isso fortalece a relação de confiança entre a equipe de desenvolvimento e o cliente.

Agora, tratando das motivações quanto ao produto final, ou seja, do *software* de mobilização:

1. Aumento da produtividade dos setores envolvidos: Com um *software* de mobilização, é possível otimizar o processo de alocação e atribuição de tarefas aos colaboradores, garantindo que cada um seja designado para as atividades mais adequadas às suas habilidades e experiência. Isso resulta em uma maior eficiência operacional e no aumento da produtividade geral da equipe;

2. Tomada de decisões estratégicas: Com acesso a dados e informações atualizadas sobre a gestão de mão de obra, a gerência pode tomar decisões mais embasadas e estratégicas. O *software* de mobilização fornece *insights* sobre a disponibilidade de recursos humanos, habilidades específicas dos colaboradores, histórico de funções, entre outros. Isso permite uma alocação mais eficiente de recursos, identificação de necessidades de treinamento e planejamento de longo prazo;
3. Segurança e proteção de dados: A aplicação de tecnologias e ferramentas avançadas na solução garante a segurança das informações da empresa. Com medidas de autenticação e controle de acesso, criando um ambiente confiável e confidencial para o tratamento das informações dos colaboradores;
4. Eficiência no ciclo de contratação: Ao abranger desde o planejamento da mão de obra até a contratação e gerenciamento contínuo dos colaboradores, o *software* de mobilização simplifica e agiliza o ciclo de contratação. Processos manuais demorados são substituídos por fluxos automatizados, reduzindo erros e aumentando a eficiência na seleção, contratação, realocação e demissão de colaboradores;
5. Redução de custos: A eficiência trazida pelo *software* de mobilização resulta em uma redução de custos significativa para a organização. Menos tempo é gasto em processos burocráticos, menos erros ocorrem devido a fluxos automatizados e menos recursos são desperdiçados devido a uma alocação inadequada de colaboradores, além de, claro, todo processo de desenvolvimento ser efetuado por colaboradores desse setor da empresa.

1.4 Objetivos

Demonstrar a eficácia da aplicação de etapas de Engenharia de *software* aliadas à processos de gerenciamento ágil no processo de desenvolvimento de um *software* complexo em um curto espaço de tempo, além de destacar os benefícios e impactos positivos que a aplicação dessas metodologias e práticas pode trazer para a eficiência e qualidade do *software* desenvolvido. Em suma, o *software* de mobilização proporciona uma solução inteligente para diminuir custos, aumentar a eficiência e a produtividade dos funcionários envolvidos nesse processo, resultando em benefícios para a organização como um todo. Além disso, busca-se garantir a segurança das informações de candidatos e colaboradores. O estudo irá investigar os requisitos e necessidades dos usuários, aplicar metodologias ágeis e práticas de Engenharia de *Software*, e avaliar o impacto do *software* de mobilização no desempenho, na qualidade dos dados e na tomada de decisões estratégicas.

1.4.1 Objetivos Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos devem ser alcançados:

- Investigar e comparar diferentes metodologias ágeis, como *Scrum*, *Kanban* ou XP, para identificar qual é a mais adequada para o desenvolvimento do *software*;
- Realizar uma análise detalhada das etapas e práticas de Engenharia de *Software*, como análise de requisitos, *design*, programação e testes, e avaliar como essas etapas podem ser aplicadas de forma ágil e eficaz em um contexto real de desenvolvimento;
- Avaliar o impacto da aplicação dessas metodologias e práticas no

processo de desenvolvimento, considerando fatores como velocidade de entrega, adaptabilidade a mudanças e qualidade do produto final;

- Identificar boas práticas e recomendações específicas para o desenvolvimento ágil de *software*, levando em consideração aspectos como colaboração, comunicação, priorização de requisitos e gestão de mudanças;
- Identificar os principais requisitos funcionais e não funcionais, levando em consideração aspectos como escalabilidade, segurança e usabilidade;
- Desenvolver uma arquitetura de *software* adequada, levando em conta a integração com outros sistemas e a capacidade de processamento de grandes volumes de dados;
- Garantir a segurança das informações e dados manipulados pelo *software*, implementando mecanismos de autenticação, criptografia e controle de acesso, de forma a bloquear possíveis fraudes existentes nesses processos;
- Avaliar a escalabilidade do *software*, considerando seu potencial de expansão para atender a um maior número de usuários e demandas crescentes;
- Realizar testes de aceitação e validação, garantindo que a solução atenda aos requisitos e expectativas dos usuários e *stakeholders*;
- Contribuir para o conhecimento científico e prático no campo da Engenharia de *Software* e gestão de projetos, por meio de uma pesquisa rigorosa e embasada sobre o desenvolvimento e a implementação do *software*.

1.5 Metodologia

Para gerir o desenvolvimento de um *software* de mobilização, é extremamente recomendável utilizar processos de gerenciamento ágil, que permitem uma entrega rápida e de valor ao usuário final. Aliada a isso, a aplicação de algumas etapas da Engenharia de *Software* será realizada no projeto em questão. A seguir está uma proposta de metodologia passo a passo para garantir os objetivos estabelecidos:

1. Realização de revisão bibliográfica: realização de uma revisão bibliográfica abrangente sobre metodologias ágeis, práticas de Engenharia de *Software* e desenvolvimento de *software* de mobilização, afim de obter um conhecimento sólido sobre as abordagens e técnicas relevantes;
2. Definição de metodologias ágeis mais adequadas: Identificar dentre as opções, as metodologias ágeis mais adequadas para o desenvolvimento do *software*. Fatores como natureza do projeto, tamanho da equipe, requisitos de negócio e características do ambiente de desenvolvimento, devem ser considerados;
3. Planejamento e Cronograma:
 - Estimativa de prazo e MVP (Mínimo Produto Viável) de produto com as funcionalidades prioritárias;
 - Elaboração do cronograma com gráfico de *Gantt*;
4. Definição do escopo e requisitos:
 - Levantamento de requisitos funcionais e não funcionais;
 - Definição da EAP (Estrutura Analítica do Projeto) e notação BPMN (*Business Process Model and Notation*) através de uma representação gráfica dos processos;

- Definição das regras de negócio;
- Priorização das funcionalidades mais importantes para o usuário;
- Elaboração do primeiro Diagrama de Classes;
- Definição de recursos, isto é, equipe envolvida no projeto, ferramentas, tecnologias e linguagens de programação a serem utilizadas;
- Definição do Earned Value;

5. Gestão de mudanças e riscos:

- Adaptação do planejamento às mudanças de requisitos ou prioridades;
- Monitoramento e mitigação dos riscos identificados;

6. Aplicação de técnicas de *UI/UX Design*:

- Matriz de certezas, suposições e dúvidas;
- Entrevistas com *stakeholders* e demais partes interessadas;
- Elaboração e validação de protótipos;

7. Realizar testes de validação: Realização de testes de aceitação e validação para garantir que o *software* atenda aos requisitos e expectativas dos usuários e *stakeholders*. Isso inclui testes funcionais, de desempenho e de segurança;

8. Entrega e implantação:

- Preparação para a implantação do *software* com testes de comunicação com *web-service*;
- Lançamento do *software* para uso do usuário final em conjunto com um treinamento;
- Identificar potenciais melhorias através de *feedbacks* e aprimorar as funcionalidades.

2 *Trabalhos relacionados*

Com o objetivo de adquirir maior conhecimento na área de Engenharia de *Software* aliadas à metodologias de gestão ágil no processo de produção de um *software*, o capítulo corrente é destinado a listar uma sequência de artigos científicos:

2.0.1 *Agile Development for Enterprise Mobility: A Systematic Literature Review*

O objetivo do estudo em questão foi investigar a aplicação de métodos ágeis no desenvolvimento de soluções de mobilidade empresarial. Os autores ??) conduzem uma revisão sistemática da literatura para identificar os principais desafios, benefícios e práticas relacionadas ao desenvolvimento ágil nesse contexto. O artigo oferece uma visão abrangente do estado atual da aplicação de métodos ágeis no contexto de mobilidade empresarial, fornecendo *insights* importantes para profissionais e pesquisadores interessados nesse campo. Ele serve como uma referência útil para compreender as práticas ágeis mais eficazes e as considerações necessárias ao desenvolver soluções de mobilidade empresarial de forma ágil.

2.0.2 *Agile Software Development Practices in Mobile Application Development: A Systematic Literature Review*

O objetivo do estudo em questão foi realizar uma revisão sistemática da literatura para identificar e analisar as práticas ágeis utilizadas no desenvolvimento de aplicativos móveis. Os autores Raj & Ghosh (2020) examinaram um conjunto de artigos selecionados e destacaram as principais práticas ágeis adotadas nesse contexto, bem como seus impactos na qualidade do *software* desenvolvido. Em suma, o artigo fornece uma visão abrangente e atualizada das práticas ágeis no desenvolvimento de aplicativos móveis, destacando suas vantagens, desafios e lições aprendidas. Ele serve como um recurso valioso para profissionais e pesquisadores que desejam adotar abordagens ágeis em seus projetos, promovendo a entrega eficiente, a qualidade do *software* e a satisfação do usuário.

2.0.3 Metodologias ágeis de gerenciamento de projetos para startups

Discute a aplicação de metodologias ágeis no gerenciamento de projetos em startups. O autor Jesus (2016) reconhece que as metodologias tradicionais de gerenciamento de projetos podem ser inadequadas para as necessidades e características das startups, que operam em um ambiente altamente dinâmico e incerto. O artigo explora diferentes metodologias ágeis, como Scrum, Kanban e Lean Startup, e analisa como essas abordagens podem ser adaptadas e utilizadas de forma eficaz nas startups. O autor destaca os benefícios das metodologias ágeis, como flexibilidade, feedback contínuo, foco no valor entregue e capacidade de adaptação às mudanças. Além disso, o artigo discute desafios e melhores práticas na implementação das metodologias ágeis nas startups, visando melhorar a eficiência, a produtividade e o sucesso dos

projetos.

2.0.4 Implantação da metodologia ágil *SCRUM* em um ambiente de desenvolvimento

O autor Souza (2014) destaca a crescente demanda por abordagens ágeis na indústria de desenvolvimento de software devido à necessidade de maior flexibilidade, adaptabilidade e eficiência. O artigo descreve o *Scrum* como uma metodologia ágil amplamente adotada e discute os princípios e práticas fundamentais do *Scrum*, como as reuniões diárias (*daily scrum*), *sprints* e a definição de papéis, como o *Scrum Master* e o *Product Owner*. O autor compartilha a experiência de implantação do *Scrum* em um ambiente de desenvolvimento e discute os desafios enfrentados e as lições aprendidas durante o processo. O artigo ressalta os benefícios alcançados com a adoção do *Scrum*, como a melhoria da comunicação, a entrega de valor mais rápida e a maior satisfação dos clientes. No geral, o artigo destaca a importância da metodologia *Scrum* como uma abordagem ágil eficaz para aprimorar o processo de desenvolvimento de software em um ambiente de trabalho.

3 *Referencial Teórico*

Este capítulo abordará alguns conceitos que serão necessários durante o desenvolvimento deste trabalho para que o projeto tenha fundamentos concretos.

3.1 Transformação digital

A transformação digital exige uma mudança cultural na empresa. Segundo Westerman & Bonnet (2019), é necessário criar uma cultura de inovação que valorize a experimentação e o aprendizado contínuo. Além disso, é importante incentivar a colaboração entre os funcionários da empresa, para que ideias e soluções inovadoras possam surgir de forma mais natural. Porém, os autores alertam que é necessário avaliar quais tecnologias são mais relevantes para a empresa e como elas podem ser integradas para criar uma experiência digital mais completa para os clientes. Além disso, é importante ter uma estratégia clara para a implementação da tecnologia, para evitar desperdícios e retrabalho. Por fim, é importante destacar que a transformação digital não é uma jornada de curto prazo, mas sim um processo contínuo de adaptação e aprendizado. Segundo ROSS *et al.* (2019), as empresas precisam ter uma visão de longo prazo para a transformação digital, considerando não apenas as tecnologias atuais, mas também as emergentes e as futuras. A empresa deve estar sempre avaliando o impacto das tecnologias nas suas operações e no seu modelo de negócios, e realizando ajustes e inovações para se manter

competitiva e relevante no mercado digital.

3.2 Metodologias ágeis

O artigo "Manifesto for Agile Software Development", escrito por Beck *et al.* (2001), é um documento fundamental para a compreensão das metodologias ágeis, que vêm ganhando cada vez mais espaço no mundo do desenvolvimento de *software*. É importante entender que as metodologias ágeis surgiram de um contexto no qual os profissionais de Tecnologia da Informação buscavam uma alternativa às metodologias tradicionais de gestão de projetos. Os autores do artigo citado anteriormente identificaram que as abordagens tradicionais muitas vezes resultavam em atrasos, custos excessivos e falta de adaptação às mudanças. O manifesto ágil, por outro lado, baseado em valores e princípios, propõe uma abordagem mais flexível e colaborativa para o desenvolvimento de *software*, com foco na entrega contínua de valor ao cliente. Além do Manifesto Ágil, existem diversas metodologias ágeis que são utilizadas atualmente no desenvolvimento de *software*, como o *Scrum* e o *Kanban*, cada uma dessas metodologias possui suas particularidades, mas todas compartilham da mesma filosofia de valorização da entrega contínua de valor ao cliente e da colaboração entre as equipes envolvidas no projeto. Outro ponto importante a ser destacado é a importância da comunicação efetiva entre as equipes, principalmente em um contexto ágil, no qual as mudanças são frequentes e o feedback é constante. Como afirma Cohn (2004) em seu livro "User Stories Applied", a comunicação é fundamental para o sucesso do desenvolvimento ágil de *software*, e a utilização de histórias de usuário pode ajudar a garantir uma comunicação mais clara e objetiva entre os membros da equipe. Por fim, é importante mencionar que o uso de metodologias ágeis no desenvolvimento de *software* tem sido cada vez mais adotado por empresas de diversos setores, uma vez que essa abordagem permite uma maior flexibilidade e adaptabilidade

a mudanças no mercado e nas necessidades dos clientes. Como destaca Serrador & Pinto (2015) em seu artigo "Does Agile work? A quantitative analysis of agile project success", os projetos ágeis tendem a apresentar maior satisfação do cliente e maior probabilidade de entrega no prazo e dentro do orçamento previsto em comparação com as metodologias tradicionais.

3.3 Práticas de Engenharia de Software

A aplicação de práticas de engenharia de *software* em um projeto é essencial para garantir a qualidade, confiabilidade e segurança do produto desenvolvido. A engenharia de *software* é uma disciplina que se preocupa com a aplicação de métodos e técnicas para a criação, manutenção e evolução de sistemas de *software* complexos e confiáveis. Essa disciplina busca garantir que o *software* desenvolvido atenda às necessidades dos usuários, cumpra com os requisitos de qualidade e segurança, e seja desenvolvido de forma eficiente e eficaz. Dessa forma, além de reduzir os custos e o tempo de desenvolvimento, a engenharia de *software* oferece uma série de técnicas e métodos que podem ser aplicados em projetos de diferentes tamanhos e complexidades, garantindo que o mesmo atenda às necessidades dos usuários e da organização (PRESSMAN, 2010). Aliada a isso, a experiência do usuário (*UX - User Experience*) e a interface do usuário (*UI - User Interface*) são duas disciplinas importantes na criação de produtos digitais de sucesso. A aplicação de práticas de *UX/UI* em um projeto é fundamental para garantir a satisfação do usuário final, melhorar a usabilidade do produto e aumentar sua eficiência (BROWN, 2016). Além disso, é importante destacar que a engenharia de *software* não se resume apenas ao desenvolvimento do produto em si, mas também envolve a gestão do processo de desenvolvimento, incluindo o planejamento, a execução, o monitoramento e o controle do projeto. A gestão de projetos de *software* é fundamental para garantir que o produto

seja entregue dentro do prazo, do orçamento e com a qualidade esperada (PMBOK, 2017). Por fim, é importante destacar que a engenharia de *software* e as disciplinas de *UX/UI* e gestão de projetos devem trabalhar em conjunto para garantir o sucesso do produto final (SOMMERVILLE, 2015).

4 *Plano de Trabalho*

As seguintes disciplinas, cursadas ao decorrer da graduação de Bacharelado em Ciência da Computação, terão impacto direto para os fundamentos do projeto em questão:

- Introdução a Pesquisa em Informática;
- Engenharia de *Software* I;
- Engenharia de *Software* II;
- UX e UI Design.

4.1 Cronograma

Definiu-se o cronograma a partir das atividades listadas abaixo:

1. Leitura de Artigos Relacionados: ler artigos e pesquisas relacionados ao tema do projeto. Identificar pontos relevantes que possam ser aplicados no projeto;
2. Análise de Requisitos do Módulo: entender o processo e alinhar as expectativas com os principais *Stakeholders*, além da definição do escopo e requisitos e do planejamento e cronograma;

3. Pesquisa de Mercado: elaborar relatório de análise de mercado com base nas ferramentas já existentes para um processo de mobilização de candidatos e funcionários;
4. Definição dos Prazos: definir o cronograma do projeto em função dos prazos estabelecidos pela empresa e pelos objetivos do projeto. Além de definir prazos para cada etapa específica do projeto para que seja possível ajustar o cronograma conforme necessidade, caso houver;
5. Design Inicial do Protótipo: definir as principais funcionalidades que serão implementadas no módulo, afim de elaborar o design inicial do protótipo, utilizando a ferramenta *Figma*;
6. Formular o Diagrama de Classes Inicial: definir a estrutura de classes e objetos do sistema e elaborar o diagrama de classes inicial;
7. Desenvolvimento do Código: desenvolver o código do módulo utilizando a tecnologia *Angular* para o *Front-End* (linguagens *HTML*, *CSS* e *TypeScript*), a linguagem *PHP* para o *Back-End* e a tecnologia *SpringBoot* (linguagens *HTML* e *JavaScript*) para o formulário externo. Utilizar também a ferramenta Ágil Bitrix24 para gerenciamento do projeto e acompanhar o progresso e efetuar ajustes conforme necessidade;
8. Testes de *Web-Service*: realizar testes de integração com outras aplicações e testes de desempenho e segurança do sistema;
9. Entrega do Módulo: realizar testes finais antes da entrega. Concluir a entrega do módulo para a empresa e seus *Stakeholders* dentro do prazo estabelecido. Elaborar a documentação técnica e de usuário;
10. Análise dos Resultados: analisar os resultados do módulo e identificar problemas e oportunidades de melhoria;

11. Melhorias e Manutenibilidade: realizar ajustes e melhorias no sistema de acordo com os resultados da análise e *feedbacks* de usuários. Garantir a manutenibilidade do sistema para futuras atualizações;
12. Formular Escopo da Documentação: Definir o escopo da documentação a ser entregue. O mesmo servirá como base para o pré-projeto e para a monografia;
13. Escrita do Pré-Projeto: escrever o pré-projeto de acordo com as normas estabelecidas pela instituição de ensino. Incluir informações sobre a justificativa, objetivo, metodologia e referencial teórico do projeto;
14. Preparar Roteiro de Apresentação: Preparar um roteiro de apresentação para a defesa do pré-projeto. Treinar a apresentação;
15. Entrega do Pré-Projeto: entregar o pré-projeto dentro do prazo estabelecido pela instituição de ensino;
16. Escrita da Monografia: escrever a monografia de acordo com as normas estabelecidas pela instituição de ensino. Incluir informações sobre a introdução, referencial teórico, metodologia, resultados e conclusão do projeto;
17. Preparar Roteiro de Apresentação: preparar um roteiro de apresentação para a defesa da monografia. Treinar a apresentação;
18. Entrega da Monografia: entregar a monografia dentro do prazo estabelecido pela instituição de ensino.

[illegible]

5 *Resultados Preliminares*

Este capítulo é dedicado a apresentar os resultados preliminares obtidos até o momento da pesquisa. As pesquisas, análises e protótipos foram conduzidos com o objetivo de investigar e validar as hipóteses levantadas no estudo.

É importante ressaltar que os resultados preliminares apresentados aqui servem como base para a continuidade da pesquisa e aprimoramento dos métodos utilizados.

5.1 Pesquisa de mercado

Com o objetivo de obter uma compreensão mais aprofundada do mercado de *softwares* de mobilização existentes, foi realizada uma pesquisa de mercado. Para isso, foram selecionados dois *softwares* de referência, cujos nomes não serão mencionados neste contexto. Além das funcionalidades, a pesquisa de mercado também buscou informações sobre os preços dos *softwares* selecionados. Foram analisados os diferentes modelos de licenciamento disponíveis, como licenças individuais, por usuário, por assinatura mensal ou anual, bem como possíveis descontos ou planos especiais para empresas.

Essa pesquisa de mercado proporcionou uma visão abrangente do cenário dos *softwares* de mobilização disponíveis no mercado, permitindo uma melhor compreensão das opções existentes e fornecendo *insights* valiosos para o desenvolvimento do *software* de mobilização empresarial no projeto em questão.

Um aspecto relevante identificado foi o custo elevado por candidato admitido, conforme demonstrado na figura 1. No caso da pesquisa em questão, considerou-se valores em euro e em dólar, pois os *softwares* com os melhores custos são encontrados com essas moedas, porém realizou-se uma conversão para real afim de exibir o resultado. Além disso, obteve-se os valores por funcionário, pois é o modelo de cobrança que esses *softwares* adotam, porém também os custos mensais (1.250 colaboradores admitidos nos primeiros dois meses do ano, considerando o pior caso em relação a custo).

Esses *softwares* geralmente são projetados para atender a diferentes setores e necessidades, o que pode resultar em uma ampla gama de funcionalidades que nem sempre são aplicáveis ao processo de contratação e mobilização de uma empresa específica. Essa observação é importante, pois a adoção de um *software* de mobilização com funcionalidades excessivas pode resultar em um investimento desnecessário. Portanto, ao desenvolver o *software* de mobilização empresarial do projeto em questão, é fundamental considerar as necessidades e processos específicos da *Alfa Engenharia e Montagens Industriais*. Buscar um enfoque personalizado, que atenda às demandas de contratação e mobilização de forma eficiente e direcionada, pode resultar em um sistema mais eficaz e com custos mais adequados, evitando o pagamento por funcionalidades que não serão utilizadas e proporcionando um melhor aproveitamento dos recursos financeiros. Isso contribuirá para a otimização do processo de contratação e mobilização, resultando em um maior retorno sobre o investimento e um aumento na eficiência operacional da empresa.

Figura 1: Pesquisa de Mercado - Valores *Softwares* de Mobilização

Fonte: Compilação do Autor.

5.2 Definição de metodologias ágeis a serem utilizadas

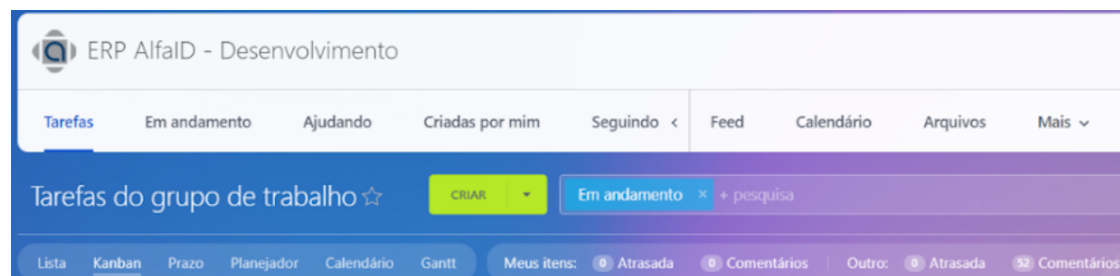
Com base na revisão bibliográfica realizada, foram identificadas as metodologias ágeis mais adequadas para o desenvolvimento do *software*. Levaram-se em consideração fatores como a natureza do projeto, o tamanho da equipe, os requisitos de negócio e as características do ambiente de desenvolvimento. As duas metodologias selecionadas foram:

- *Scrum*: Trata-se de uma metodologia ágil que auxilia a equipe no gerenciamento do desenvolvimento por meio de *sprints*, que são iterações de trabalho. O *Scrum* envolve reuniões diárias (*dailys*), planejamento de *sprint*, revisão e retrospectiva de *sprint*.
- *Kanban*: É uma técnica que permite visualizar o fluxo de trabalho, limitar o número de tarefas em progresso e aumentar a eficiência da equipe. O *Kanban* ajuda a manter o foco, a organização e a identificação de

gargalos no processo de desenvolvimento.

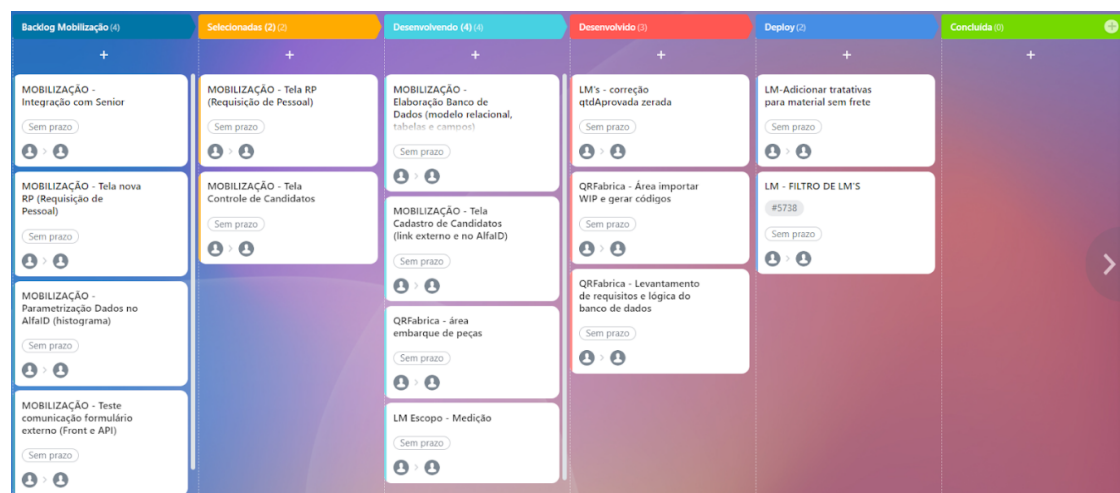
Para o gerenciamento do projeto, será utilizada a ferramenta *Bitrix24*, que oferece recursos abrangentes, conforme ilustrado na figura 2. Com essa ferramenta, será possível acompanhar o progresso do projeto e realizar ajustes quando necessário. Entre os recursos disponíveis estão os quadros *Kanban* (observado na Figura 3), a atribuição de tarefas, a comunicação em equipe e o monitoramento do tempo. Esses recursos contribuem para manter a equipe alinhada e garantir a eficiência do desenvolvimento do *software*.

Figura 2: Principais Menus do *Bitrix24*



Fonte: Compilação do Autor.

Figura 3: Kanban Aplicado no Módulo de Mobilização



Fonte: Compilação do Autor.

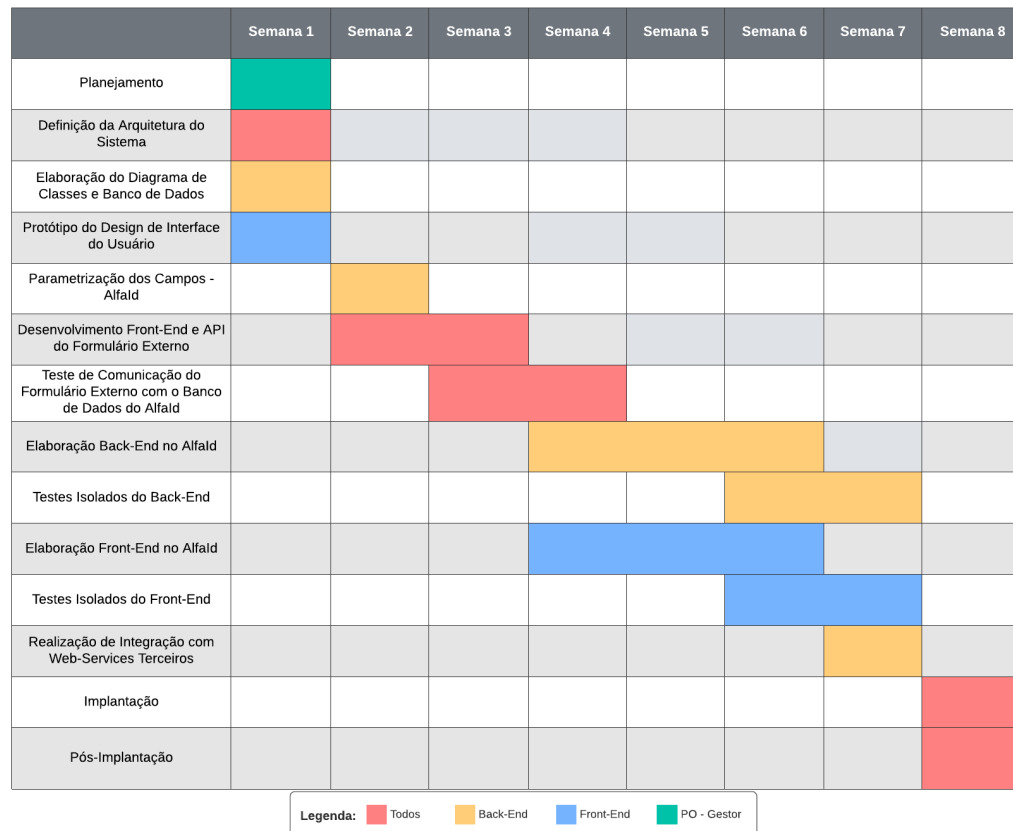
5.3 Planejamento e cronograma

Nas figuras 4 e 5, serão apresentados o cronograma em forma de tabela e o gráfico *Gantt*, respectivamente, do trabalho em questão. O cronograma é uma ferramenta fundamental para o planejamento e acompanhamento das atividades ao longo do projeto, permitindo visualizar as etapas, prazos e responsáveis envolvidos em cada tarefa. O gráfico *Gantt*, por sua vez, fornece uma representação visual das atividades do projeto ao longo do tempo, mostrando a duração de cada tarefa e suas dependências. Ele proporciona uma visão clara do fluxo de trabalho e ajuda a identificar possíveis sobreposições ou atrasos.

Figura 4: Cronograma das Atividades MVP (Mínimo Produto Viável)

Cronograma Mobilização - MVP				
ID	Atividades Planejadas	ID Dependente	Equipe/ Responsável	Duração (Dias)
1	Planejamento	N/A	PO - Gerente	2
2	Definição da Arquitetura do Sistema	1	Todos	1
3	Elaboração do Diagrama de Classes e Banco de Dados	1	Back-End	2
4	Protótipo do Design de Interface do Usuário	1	Front-End	2
5	Parametrização dos Campos - Alfald	3	Back-End	1
6	Desenvolvimento Front-End e API do Formulário Externo	5	Front-End / Back-End	10
7	Teste de Comunicação do Formulário Externo com o Banco de Dados do Alfald	6	Todos	4
8	Elaboração Back-End no Alfald	5	Back-End	28
9	Testes Isolados do Back-End	8	Back-End	5
10	Elaboração Front-End no Alfald	5	Front-End	28
11	Testes Isolados do Front-End	10	Front-End	5
12	Realização de Integração com Web-Services Terceiros	9	Back-End	3
13	Implantação	12	Todos	7
14	Pós-Implantação	13	Todos	N/A

Fonte: Compilação do Autor.

Figura 5: Gráfico de *Gantt* apenas com as tarefas master do Módulo de Mobilização

Fonte: Compilação do Autor.

Com base no cronograma e no gráfico *Gantt*, é possível acompanhar o progresso do projeto, garantir o cumprimento dos prazos e realizar ajustes conforme necessário.

Na figura 6 é possível visualizar um gráfico *Gantt* em execução, que ilustra algumas das subetapas do desenvolvimento do projeto. No gráfico *Gantt* em questão, é possível visualizar as subetapas que estão sendo executadas em um determinado período de tempo. Porém, vale destacar que a figura exibida representa apenas um trecho do gráfico *Gantt* completo.

Figura 6: Trecho de *Gantt* em Plena Aplicação no Módulo de Mobilização

Fonte: Compilação do Autor.

5.4 Definição do escopo e requisitos

5.4.1 Levantamento de requisitos funcionais e não funcionais

O *software* de mobilização possui uma série de requisitos funcionais e não funcionais que são essenciais para o seu correto funcionamento e atendimento às necessidades da empresa. Os requisitos funcionais definem as funcionalidades e recursos específicos do sistema, como o cadastro de candidatos, controle de vagas, seleção de candidatos, entre outros. Já os requisitos não funcionais estabelecem as características e qualidades desejadas para o sistema, tais como desempenho, segurança, usabilidade, integração com outros sistemas, escalabilidade, entre outros.

A relação completa dos requisitos funcionais e não funcionais pode ser visualizada nas figuras 7 e 8, respectivamente. Nessas figuras, são apresentadas tabelas detalhadas com os códigos, títulos e descrições de cada requisito, proporcionando uma visão clara e organizada das necessidades do software de mobilização.

Figura 7: Requisitos Funcionais do Módulo de Mobilização

Código	Requisito Funcional	Descrição
RF 01	Cadastro de candidatos	Disponibilizar um formulário, externo e dentro do Alfald, para que seja possível preencher os dados cadastrais e informações do candidato, necessários para iniciar o processo de recrutamento.
RF 02	Preenchimento de Histograma e PCMSO	Liberação do Alfald com alimentação da base de dados de treinamentos PCMSO e histograma, necessários para o início do planejamento do projeto.
RF 03	Solicitação de Recrutamento	O Alfald deverá permitir a criação de RP's (Requisição de Pessoal). Esse documento é o que inicia o processo de contratação, sendo necessário para a liberação de vagas a serem preenchidas e/ou indicação de candidatos.
RF 04	Controle de Candidatos e Vagas	O Alfald deve permitir o gerenciamento eficiente dos candidatos e das vagas disponíveis. O objetivo é facilitar o processo seletivo, garantindo a eficiência na busca por candidatos qualificados e o preenchimento das vagas de forma adequada.
RF 05	Seleção de Candidatos	Selecionar candidatos para prosseguir no processo seletivo. Novos campos deverão ser adicionados ao formulário cadastral para que o status do candidato seja alterado.
RF 06	Registro e Consulta de ASO (Atestado de Saúde Ocupacional)	O Alfald será disponibilizado para as clínicas cadastradas como fornecedores para registrar e atestar que o candidato está clinicamente apto para exercer as atividades/funções profissionais de seu cargo.
RF 07	Aprovação Multinível para Contratação de Colaboradores	Implementação de um fluxo de aprovação, no qual a contratação de novos colaboradores passa por diferentes níveis de aprovação dentro da empresa.
RF 08	Comunicação com Web-Service de Terceiros	Envio da documentação do candidato a ser admitido para Web-Service de uma empresa terceira, onde o setor responsável realizará a conferência de todos os dados cadastrais e documento. Se estiver tudo em conformidade, o setor realiza a contratação pelo Web-Service e o mesmo retorna essa informação para o Alfald.
RF 09	Acompanhamento de Todo Processo de Mobilização	O Alfald deve permitir o acompanhamento do processo de mobilização, fornecendo informações atualizadas sobre o status das candidaturas, vagas em aberto, vagas preenchidas, entre outros.

Fonte: Compilação do Autor.

Figura 8: Requisitos Não Funcionais do Módulo de Mobilização

Código	Requisito Não Funcional	Descrição
RNF 01	Interface Intuitiva	O módulo deve possuir uma interface intuitiva, com disposição clara dos elementos e fluxos de navegação simples, visando facilitar o uso pelos usuários.
RNF 02	Tempo de Resposta Hábil	O módulo deve apresentar um tempo de resposta rápido para a realização das principais operações, a fim de otimizar a produtividade dos usuários.
RNF 03	Proteção de Dados	A proteção de dados sensíveis dos candidatos deve ser garantida, adotando medidas de segurança adequadas, como criptografia e controle de acesso.
RNF 04	Disponibilidade 24/7	O Alfald e o formulário externo devem estar disponíveis para acesso 24 horas por dia, 7 dias por semana, a fim de atender às necessidades dos usuários em diferentes momentos.
RNF 05	Confiabilidade	O módulo deve ser confiável, evitando falhas críticas, erros de processamento e perda de dados, para garantir a integridade e a consistência das informações.
RNF 06	Escalabilidade	O Alfald deve ser capaz de lidar com um aumento significativo no número de candidatos e vagas, mantendo o desempenho e a funcionalidade em situações de alta demanda.
RNF 07	Documentação Detalhada	Deve ser fornecida uma documentação completa, contendo manuais de usuário, guias de administração e especificações técnicas para facilitar o entendimento e uso do sistema.
RNF 08	Suporte Técnico	Deve ser oferecido suporte técnico eficiente aos usuários, por meio de canais de comunicação adequados, como telefone e e-mail, para solução de problemas e esclarecimento de dúvidas.
RNF 09	Atender os Processos de Mobilização da empresa	O módulo deve ser capaz de atender aos processos específicos de mobilização da empresa, como requisitos de contratação, etapas do processo seletivo, aprovações necessárias, e demais fluxos e procedimentos estabelecidos pela empresa.

Fonte: Compilação do Autor.

5.4.2 Definição das regras de negócio

O *software* de mobilização também possui um conjunto de regras de negócio que são fundamentais para orientar o funcionamento adequado do sistema e garantir a conformidade com os processos e políticas da empresa. Essas regras de negócio definem as diretrizes, restrições e procedimentos a serem seguidos no contexto da mobilização de colaboradores.

A relação completa das regras de negócio pode ser visualizada na figura 9, onde é apresentada uma tabela com os códigos e descrições de cada regra, fornecendo uma visão consolidada das diretrizes estabelecidas para o módulo de mobilização.

Figura 9: Regras de Negócio do Módulo de Mobilização

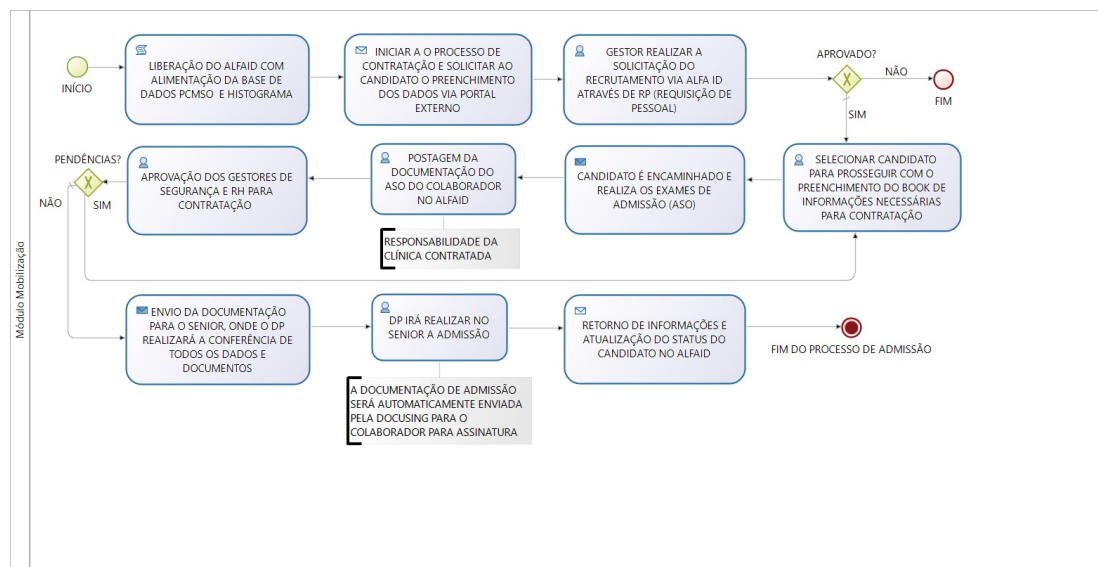
Código	Regra de Negócio
RN 01	Os formulários (externo e no Alfald) não devem permitir o cadastro de candidatos sem o preenchimento dos campos cadastrais obrigatórios.
RN 02	Deve ocorrer a validação da consistência dos dados cadastrais dos candidatos, como CPF, RG, e-mail, telefone, evitando duplicidade de registros e inconsistências nos mesmos.
RN 03	O Alfald deve permitir que apenas usuários autorizados criem, editem ou excluam requisições de pessoal (RP), garantindo a segurança das informações e a integridade do processo de contratação.
RN 04	O módulo deve facilitar a seleção de candidatos com base em critérios predefinidos, como qualificações, experiência profissional, disponibilidade, entre outros, visando a identificação dos candidatos mais adequados para as vagas disponíveis.
RN 05	O Alfald deve gerar automaticamente um número de ASO (Atestado de Saúde Ocupacional) único para cada candidato registrado, garantindo a rastreabilidade e a integridade dos registros médicos.
RN 06	O módulo deve seguir as normas e regulamentações de proteção de dados pessoais, como a Lei Geral de Proteção de Dados (LGPD), garantindo a privacidade e o tratamento adequado das informações dos candidatos.
RN 07	O Alfald deve permitir a visualização e o acompanhamento do status de cada etapa do processo de mobilização, oferecendo informações atualizadas sobre a progressão dos candidatos, vagas em aberto, vagas preenchidas e demais indicadores relevantes.
RN 08	O Alfald deve registrar todas as interações e histórico de atividades relacionadas aos candidatos e ao processo de mobilização, proporcionando um registro completo e rastreável das ações realizadas.
RN 09	O Alfald deve permitir a geração de relatórios e análises sobre o processo de mobilização, fornecendo informações gerenciais e indicadores para auxiliar na tomada de decisões e no monitoramento do desempenho do processo.
RN 10	O Alfald deve fornecer mecanismos de segurança adequados para proteger as informações do software, como autenticação, controle de acesso e criptografia, garantindo a confidencialidade e a integridade dos dados.
RN 11	O módulo deve ser escalável e capaz de lidar com um grande volume de candidatos e vagas, garantindo o desempenho e a disponibilidade do sistema.
RN 12	O Alfald deve ser capaz de integrar-se com outros sistemas utilizados pela empresa, como sistemas de ERP, para garantir a sincronização dos dados e evitar inconsistências entre os sistemas.

Fonte: Compilação do Autor.

5.4.3 BPMN (*Business Process Model and Notation*)

BPMN (*Business Process Model and Notation*) é uma linguagem gráfica utilizada para modelar processos de negócio. No contexto do projeto, o BPMN desempenha um papel importante na representação visual dos processos de negócio envolvidos, oferecendo uma notação padronizada e compreensível, permitindo que diferentes *stakeholders* compreendam e colaborem na análise, documentação e melhoria dos processos. O resultado observa-se na figura 10.

Figura 10: BPMN *Software de Mobilização*



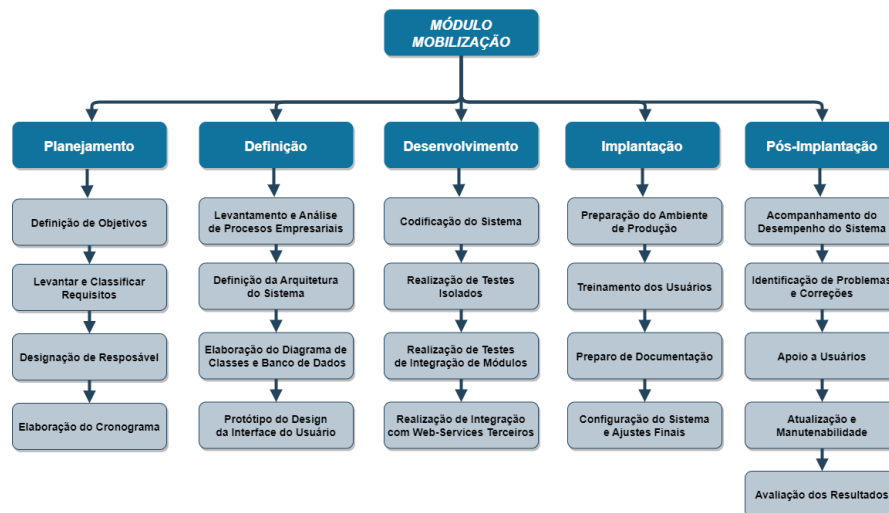
Fonte: Compilação do Autor.

5.4.4 EAP (*Estrutura Analítica do Projeto*)

Um EAP (*Estrutura Analítica do Projeto*), também conhecido como WBS (*Work Breakdown Structure*), é uma representação hierárquica e visual das entregas e atividades que compõem um projeto. No contexto do projeto, se trata de uma estrutura organizada das tarefas relacionadas ao desenvolvimento e implementação do *software* de mobilização.

O objetivo principal do EAP é decompor o escopo do trabalho em atividades específicas e gerenciáveis, relacionadas ao desenvolvimento do projeto. Ele ajuda a identificar e visualizar todas as etapas necessárias para a criação, configuração e otimização do mesmo, conforme observa-se na figura 11.

Figura 11: EAP do Módulo de Mobilização



Fonte: Compilação do Autor.

5.4.5 Diagrama de classes

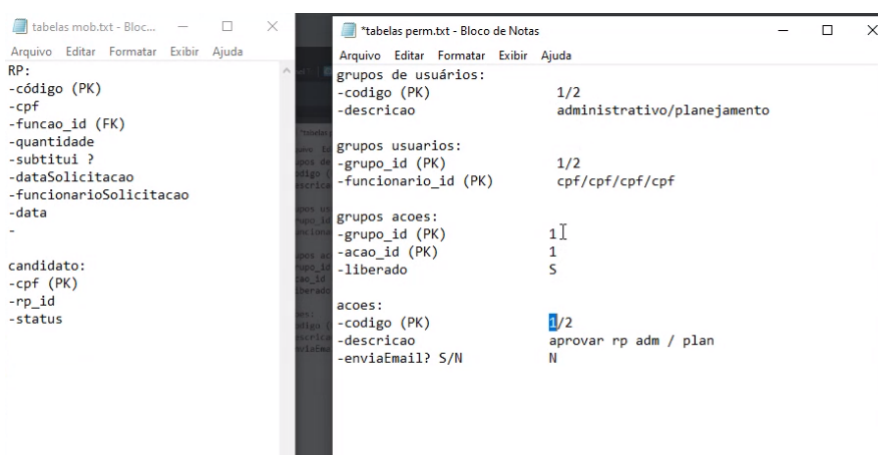
Um diagrama de classes em banco de dados é uma representação visual das tabelas, relacionamentos e estrutura de um banco de dados. Ele é usado para modelar as entidades do sistema, seus atributos e os relacionamentos entre elas. No contexto do projeto, ele é essencial para definir e organizar a estrutura de dados do *software*, identificando as entidades, seus atributos e os relacionamentos entre elas.

Com o diagrama de classes, é possível ter uma visão clara e organizada da estrutura do sistema, facilitando o desenvolvimento e a manutenção do *software*. Além disso, ele serve como base para a geração automática

de *scripts* de criação do banco de dados, agilizando a implementação e colaboração entre os desenvolvedores.

A figura 12 apresenta o primeiro esboço, ainda em bloco de notas, da estrutura do banco de dados do projeto. Nessa representação, constam algumas tabelas, os relacionamentos e outras informações relevantes para a organização dos dados do sistema.

Figura 12: Esboço Primeiro Diagrama de Classes



Fonte: Compilação do Autor.

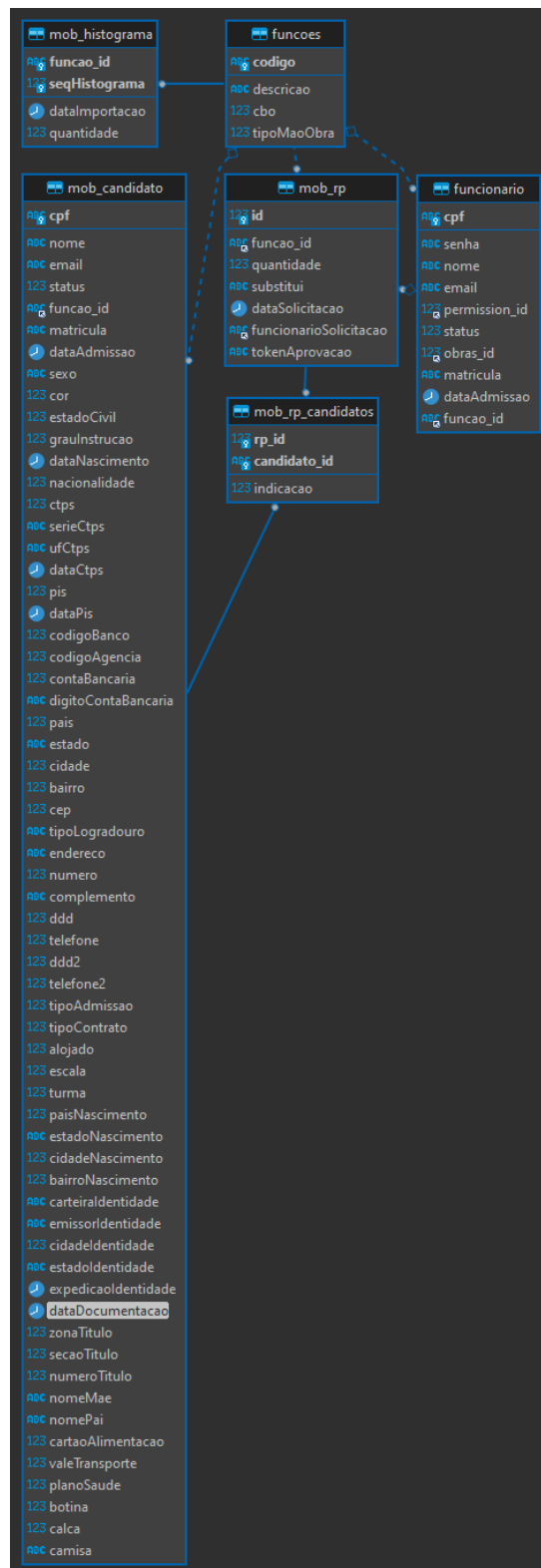
Inicialmente, a estrutura do diagrama de classes do projeto incluiu um total de seis tabelas, sendo duas delas já existentes na estrutura geral de banco de dados do sistema *Alfald*, enquanto as outras quatro foram criadas especificamente para o módulo de mobilização.

As duas tabelas existentes são denominadas '*funcoes*' e '*funcionario*'. Essas tabelas são parte integrante do ecossistema de mobilização, uma vez que contêm dados essenciais que precisaram ser consultados e manipulados durante o processo. Esses dados incluem informações de permissões, detalhes dos funcionários, histórico de funções desempenhadas, entre outros.

As quatro tabelas criadas são chamadas de '*mob-histograma*',

'mob-candidato', 'mob-rp' e 'mob-rp-candidatos'. Essas tabelas foram desenvolvidas para armazenar os principais dados relacionados ao módulo de mobilização, especialmente informações sensíveis dos candidatos. Através dessas tabelas, é possível registrar e gerenciar detalhes como histórico de mobilização, candidatos envolvidos em cada mobilização, responsáveis pelo processo de mobilização e os respectivos candidatos relacionados.

Essa estrutura do diagrama de classes, conforme ilustrado na figura 13, permite uma organização eficiente dos dados do projeto, fornecendo uma visão clara das entidades envolvidas no módulo de mobilização e suas respectivas relações.

Figura 13: Primeiro Diagrama de Classes

Fonte: Compilação do Autor.

5.4.6 Definição de recursos

A etapa de definição de recursos é de extrema importância no desenvolvimento de um projeto de *software*, pois está diretamente relacionada ao sucesso e eficiência da equipe envolvida. Nessa etapa, são identificados e definidos os recursos necessários para o projeto, incluindo a equipe, ferramentas, tecnologias e linguagens de programação a serem utilizadas.

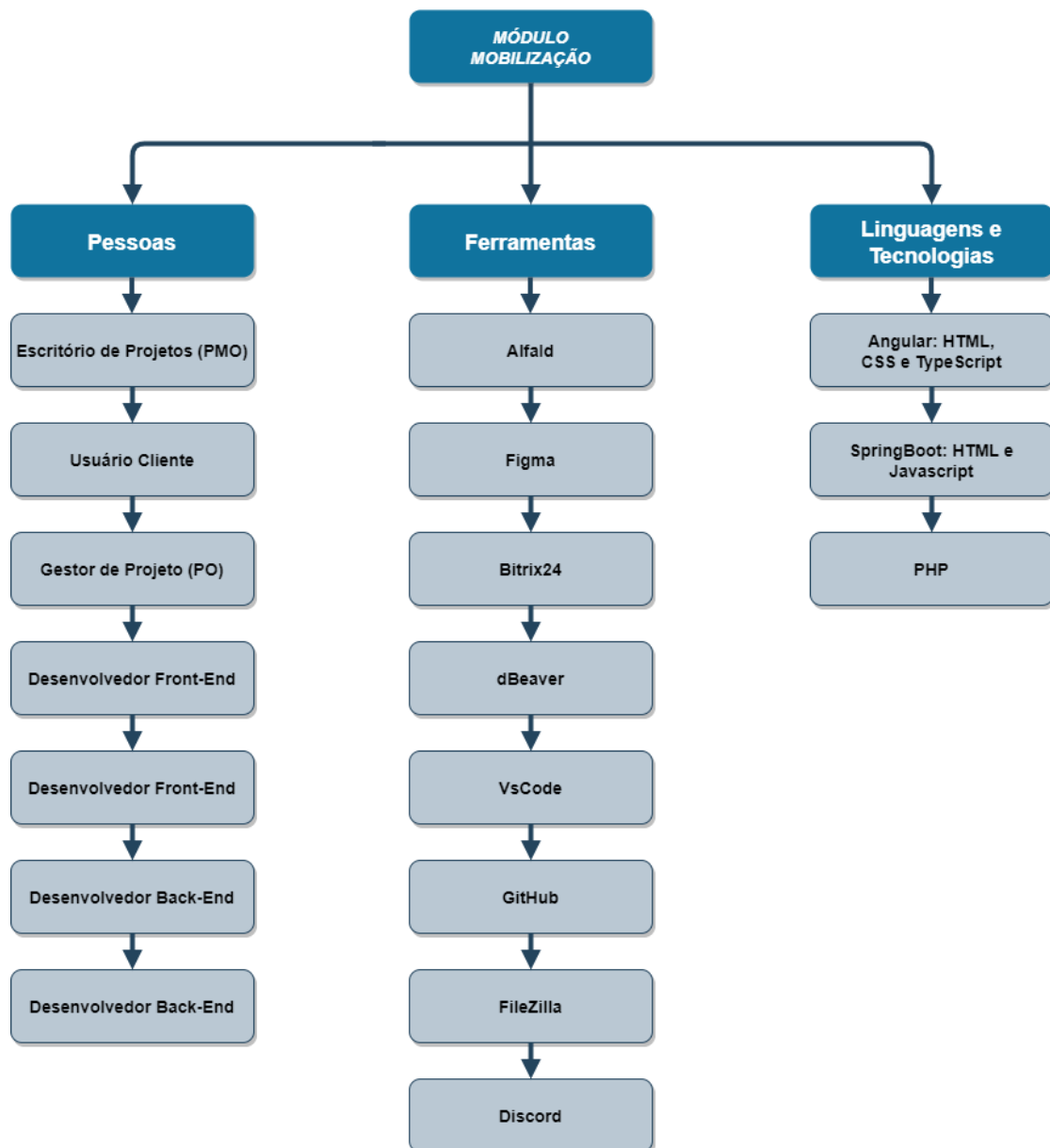
A equipe envolvida no projeto desempenha um papel crucial, pois cada membro traz habilidades e experiências específicas para contribuir com o desenvolvimento do *software*. O 'Escritório de Projetos (PMO)' é responsável por fornecer diretrizes e apoio ao projeto, garantindo que ele esteja alinhado com as estratégias e objetivos da organização. O 'Usuário Cliente' desempenha um papel fundamental ao fornecer informações e requisitos para o desenvolvimento do *software*, garantindo que ele atenda às necessidades do usuário final. O 'Gestor de Projeto (PO)' é responsável por coordenar e gerenciar a equipe, garantindo o cumprimento dos prazos, a qualidade do trabalho e a comunicação efetiva entre os membros. Os 'desenvolvedores *Front-End*' são responsáveis por criar a interface do usuário, ou seja, a parte visual e interativa do *software*, essencial para proporcionar uma experiência de usuário agradável, intuitiva e responsiva. Já os 'desenvolvedores *Back-End*' são responsáveis pela lógica e estrutura que sustentam o *software*, garantindo que o mesmo seja seguro, eficiente e capaz de lidar com as demandas de processamento e armazenamento de dados.

As ferramentas utilizadas no projeto desempenham um papel importante na produtividade e colaboração da equipe. O 'Discord', por exemplo, é utilizado para comunicação em tempo real entre os membros da equipe. O 'FileZilla' permite transferir arquivos para um servidor. O 'GitHub' é uma plataforma para compartilhamento de código e colaboração em equipe. O 'VsCode', 'dBeaver' e 'Figma' são ferramentas de desenvolvimento amplamente utilizadas para codificação, gerenciamento de bancos de dados e design de interface de

usuário. O 'Alfald' trata-se da ferramenta ERP (*Enterprise Resource Planning*) da *Alfa Engenharia e Montagens Industriais* e o 'Bitrix24' uma ferramenta que reúne um compilado de processos de gerenciamento ágil, também serão utilizadas no contexto do projeto.

As tecnologias e linguagens de programação escolhidas também desempenham um papel fundamental no desenvolvimento do software. No caso, o 'Angular' é usado para o desenvolvimento *Front-End*, utilizando *HTML*, *CSS* e *TypeScript*. O *PHP* é usado para o desenvolvimento *Back-End*. Já o *SpringBoot* é utilizado para o desenvolvimento de ambos, com *HTML* e *JavaScript*.

Figura 14: Diagrama de Recursos



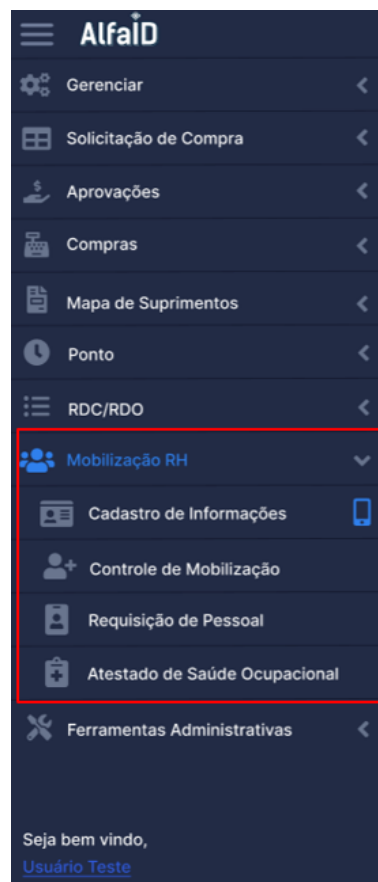
Fonte: Compilação do Autor.

5.5 Elaboração e validação dos protótipos

Um dos passos cruciais no processo de desenvolvimento de *software* é a validação dos protótipos com os principais *stakeholders*. Nesse contexto, a utilização da plataforma *Figma* desempenha um papel fundamental, proporcionando uma experiência de prototipagem interativa e colaborativa.

Através do *Figma*, é possível desenvolver protótipos de alta fidelidade, simular interações e fluxos de navegação, e compartilhá-los com os *stakeholders* para obter *feedback* e validar a usabilidade e a eficácia do design proposto. Isso permite que todas as partes interessadas tenham uma compreensão clara das funcionalidades do *software*, visualizem sua aparência e interajam com os elementos de interface de forma realista. Isso economiza tempo, reduz retrabalhos e maximiza as chances de sucesso do produto final. Nas próximas figuras, serão apresentados os protótipos desenvolvidos e validados junto aos principais *stakeholders*:

- Figura 15: Novo Menu e seus respectivos submenus criados para o novo módulo de mobilização. A explicação de cada tela (submenus) se dará em sequência.

Figura 15: Menu e submenus de Mobilização

Fonte: Compilação do Autor.

- Figura 16: Formulário externo ao AlfaID, disponível em *link da web*. Caso uma pessoa tenha interesse em fazer parte do corpo de funcionários da empresa, esse é o formulário que o mesmo deverá preencher para se candidatar. No formulário o potencial candidato preencherá seus dados pessoais, além de anexar documentos como currículo, comprovante de endereço e de identidade. É importante ressaltar que, atualmente, os funcionários do setor de recursos humanos fazem essa etapa manualmente, entrando em contato com os potenciais candidatos e inserindo essas informações na planilha já mencionada anteriormente.

Figura 16: Formulário Externo

12:30 50%

alfaengenharia.ind.br

AlfaID

VEM SER ALFA

ATENÇÃO: o preenchimento dessas informações é de suma importância para o seu prosseguimento no processo seletivo. Todos os campos são OBRIGATORIOS, então se atente às informações preenchidas.

Nome Completo:

Função:

Contato: Data Nascimento:

(37) 99999-9999 EX: 00/00/0000

Naturalidade: Estado Civil:

Escolaridade: CEP:

Ex: xxxxx-xxx

Endereço:

Fonte: Compilação do Autor.

- Figura 17: O mesmo formulário que está disponível em *link da web*, também está disponível dentro do AlfaID. Isso porque, caso algum potencial candidato tenha qualquer problema para preencher o formulário externo, seja por qual motivo for, os funcionários do setor de recursos humanos também podem fazê-lo.

Figura 17: Formulário no Alfald

Fonte: Compilação do Autor.

- Figura 18: De fato, a principal tela do módulo, pois na mesma encontram-se diversas funcionalidades, além de um grande fluxo de informações e demonstrativos. Cadastro, consulta, aprovação e admissão de candidatos. Um verdadeiro controle de mobilização.

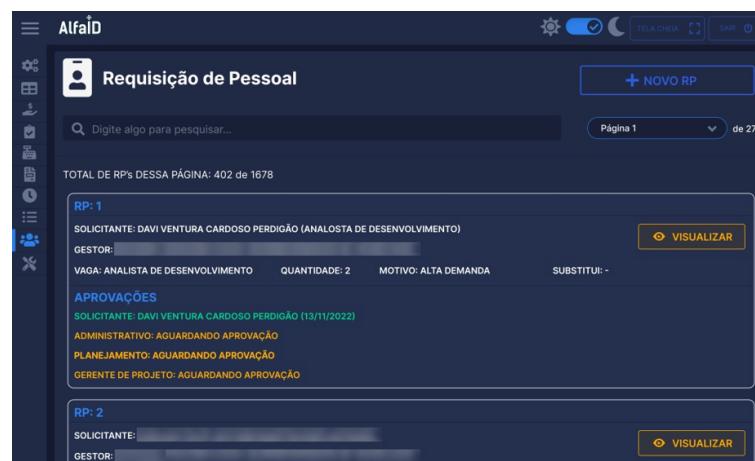
Figura 18: Tela de Controle de Mobilização

	NOME	FUNÇÃO	CONTATO	CONTATO DE EMERGÊNCIA (NOME / NÚMERO)	CEP
<input type="checkbox"/>	DAVI VENTURA CARDOSO PERDIGÃO	ANALISTA DE DESENVOLVIMENTO			
<input checked="" type="checkbox"/>		ANALISTA DE DESENVOLVIMENTO			
<input checked="" type="checkbox"/>		ANALISTA DE DESENVOLVIMENTO			
<input type="checkbox"/>		IMPORTAÇÃO			
<input type="checkbox"/>		SUPERINTENDENTE TECNOLOGIA			
<input type="checkbox"/>		ANALISTA DE DESENVOLVIMENTO			
<input type="checkbox"/>		ANALISTA DE DESENVOLVIMENTO			
<input type="checkbox"/>		ANALISTA DE DESENVOLVIMENTO			
<input type="checkbox"/>		ANALISTA DE DESENVOLVIMENTO			

Fonte: Compilação do Autor.

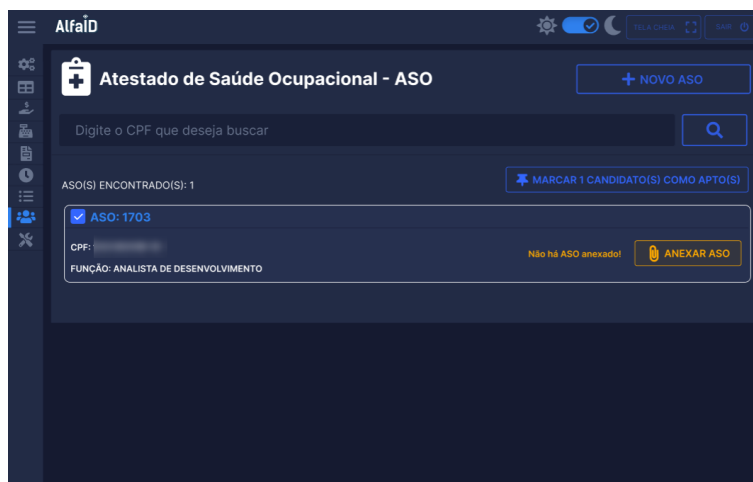
- Figura 19: Essa tela trata-se de um dos procedimentos internos da empresa com relação ao fluxo de mobilização. Nela é possível registrar e demonstrar o início do processo de contratação. Inclui ferramenta de aprovação multinível, necessária para a liberação de contratação de funcionários e/ou liberação de vagas.

Figura 19: Tela de Requisição de Pessoal



Fonte: Compilação do Autor.

- Figura 20: Essa tela faz parte de uma particularidade que o módulo de mobilização possui com relação a outros módulos desenvolvidos anteriormente para o AlfaiD. Pela primeira vez, usuários que não são colaboradores da empresa terão acesso a esse *software*. Esses usuários se tratam dos funcionários das clínicas que realizam os exames de saúde ocupacional. Nessa tela, eles registrarão e atestarão que o candidato está clinicamente apto para exercer as atividades/funções profissionais de seu cargo.

Figura 20: Tela de ASO (Atestado de Saúde Ocupacional)

Fonte: Compilação do Autor.

Referências Bibliográficas

BECK, Beedle; M., van Bennekum; A., Cockburn; A., Cunningham; W., Fowler; M., Grenning; J., Highsmith; J., Hunt; A., Jeffries; R., Kern; J., Marick; B., Martin; R., Mellor; S., Schwaber; K., Sutherland; ; THOMAS, D. Manifesto for Agile Software Development. **Agile Alliance**, Agile Alliance, 2001.

BROWN, D. C. **Communicating Design: Developing Web Site Documentation for Design and Planning**. [S.l.]: New Riders, 2016.

COHN, Mike. **User stories applied: for agile software development**. [S.l.]: Pearson Education, 2004.

JESUS, Italo. METODOLOGIAS ÁGEIS DE GERENCIAMENTO DE PROJETOS PARA STARTUPS. **Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação**, Universidade Federal de Minas Gerais, p. 1–56, 2016.

PARKES, Coleman. Metodologias são necessárias para uma transformação digital bem-sucedida e melhor desempenho do negócio? **CA Technologies**, CA Technologies, 2018.

PMBOK. **Guia PMBOK: um guia para o conjunto de conhecimentos em gerenciamento de projetos**. [S.l.]: Project Management Institute, 2017.

PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. [S.l.]: Bookman Editora, 2010.

RAJ, Rakesh; GHOSH, Sabyasachi. **Agile Software Development Practices in Mobile Application Development: A Systematic Literature Review**. Tese (Doutorado), 2020.

ROSS; W., BEATH J.; M., GOULD C.; M. Designed for digital: How to architect your business for sustained success. **MIT Press**, MIT, 2019.

SERRADOR, Pedro; PINTO, Jeffrey K. Does Agile work? A quantitative analysis of agile project success. **International Journal of Project**

Management, International Journal of Project Management, v. 33, n. 5, p. 1040–1051, 2015.

SOMMERVILLE, Ian. Engenharia de Software. **Pearson Education do Brasil**, Pearson Education, 2015.

SOUZA, Diogo. IMPLANTAÇÃO DA METODOLOGIA ÁGIL SCRUM EM UM AMBIENTE DE DESENVOLVIMENTO. **UNIVERSIDADE FEDERAL DE SANTA CATARINA, CAMPUS ARARANGUÁ**, UNIVERSIDADE FEDERAL DE SANTA CATARINAS, p. 1–63, 2014.

WESTERMAN, George; BONNET, Didier. How to Start a Digital Transformation. **Harvard Business Review**, Harvard Business Review, v. 3, n. 97, p. 134–141, 2019.