PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS

ESCOLA POLITÉCNICA

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

APLICATIVO FAST PROJECT PARA APLICAÇÃO DE METODOLOGIA ÁGIL SCRUM

MOISÉS ALVES DA COSTA

GOIÂNIA
2022

MOISÉS ALVES DA COSTA

APLICATIVO FAST PROJECT PARA APLICAÇÃO DE METODOLOGIA ÁGIL SCRUM

Artigo apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás. Como uma atividade avaliativa correspondente TCC2.

Orientador(a): Fernando Gonçalves Abadia.

GOIÂNIA
2022

MOISÉS ALVES DA COSTA

APLICATIVO FAST PROJECT PARA APLICAÇÃO DE METODOLOGIA ÁGIL SCRUM

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Ciência da Computação, e aprovado em sua forma final pela Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, em 10 / 12 / 2022.

_____
Profa. Ma. Ludmilla Reis Pinheiro dos Santos
Coordenadora de Trabalho de Conclusão de Curso

Banca examinadora:

_____
Orientador: Prof. Me. Fernando Gonçalves Abadia

_____
Prof. Me. Rafael Leal Martins

_____
Prof. Me. Gustavo Vinhal Siqueira

GOIÂNIA
2022

Agradecimentos

Agradeço a Deus, que me abençoou com força e resiliência para chegar até o final desse trabalho.
A minha irmã, por sempre me apoiar e estar comigo nos momentos mais difíceis.
Ao meu pai, que sempre esteve comigo e me apoiou em todos os momentos.
A minha mãe, que sempre me consolou e me inspirou a seguir em frente.
Ao meu Orientador Fernando que me guiou e passou todo conhecimento necessário para conclusão.
A todos os professores de curso, que me passaram todo conhecimento necessário para conclusão dessa caminhada.

Dedico esse trabalho a Deus
E a minha Família que sempre esteve comigo.

RESUMO

Esta pesquisa tem como objetivo levantar dados sobre o que é uma metodologia ágil e como facilitar o uso para seus usuários, tendo eles conhecimento sobre os processos ou não. Contando da sua origem até como é conhecida nos dias de hoje, passando por cada uma de suas etapas, elucidando suas funções e como elas são executadas, como elas são utilizadas no dia a dia, quais seus benefícios, qual seus objetivos e suas utilidades para o usuário. A partir disso levantar informações de como os usuários fazem a interação com cada uma das etapas, fazendo com que seja possível o entendimento das principais dificuldades, e a partir disso criar um esboço do que seria um aplicativo que trata essas deficiências do usuário e ainda deixa claro a forma de uso da metodologia, para que qualquer usuário com conhecimento ou não possa fazer usufruir dos seus benefícios.

Palavras-chave: Metodologia ágil, *Scrum*, Gerência de Projeto, *Sprint*

ABSTRACT

This research aims to collect data on what an agile methodology is and how to facilitate its use for its users, whether they have knowledge about the processes or not. Counting from its origin to how it is known today, going through each of its stages, elucidating its functions and how they are performed, how they are used daily, what are their benefits, what are their objectives and their uses for the user. From this, gather information on how users interact with each of the steps, making it possible to understand the main difficulties, and from that create an outline of what would be an application that deals with these user difficulties and still leaves the way of using the methodology is clear, so that any user with or without knowledge can make use of its benefits.

Keywords: Agile methodology, Scrum, Project Management, Sprint.

LISTA DE FIGURAS

Figura 1 - Tela de login 24
Figura 2 - Tela de cadastro 25
Figura 3 - Tela de recuperação de senha 25
Figura 4 - Tela de confirmação de envio 26
Figura 5 – Tela de inicial 27
Figura 6 – Tela de Linha do tempo 28
Figura 7 – Tela de tarefas 29
Figura 8 – Tela de configurações 30
Figura 9 – Tela do chat 31
Figura 10 – Tela de envio de documentos 32

SUMÁRIO

1 INTRODUÇÃO 11

1.1 Objetivo geral 12

1.2 Objetivos específicos 12

1.3 Métodos 12

2 CONCEITOS TEÓRICOS RELEVANTES. 14

2.1 Métodos ágeis 14

2.2 Práticas e etapas do *Scrum* 15

2.3 O Time *Scrum* 17

2.4 Eventos *Scrum* 18

2.4.1 *Sprint planning meeting 19*

2.4.2 *Daily Meeting 19*

2.4.3 *Sprint Review 20*

2.4.4 *Sprint Retrospective 20*

2.4.5 *Burndown Chart 20*

3 METODOLOGIA 21

3.1 Tipo de pesquisa 21

3.2 Ferramentas utilizadas 23

3.2.1 *Figma 23*

4 Implementação e resultados 24

4.1 Tela de login 24

4.2 Tela de cadastro 24

4.3 Tela de recuperação de senha 25

4.4 Tela inicial 26

4.5 Linha do tempo das tarefas 27

4.6 Tela das tarefas 28

4.7 Tela de configurações 29

4.8 Tela do chat 30

4.9 Tela de envio de documentos 31

5 Considerações finais 33

1 INTRODUÇÃO

As metodologias de desenvolvimento ágil, no Brasil, têm gerado grande entusiasmo entre seus usuários, assim como na comunidade acadêmica. Destacam-se, principalmente, os aspectos relacionados as melhorias nos resultados que as empresas desejam obter, como o aprimoramento de seus processos internos e de sua estrutura organizacional (VARASCHIM, 2009).

Vários métodos ágeis são usados atualmente no desenvolvimento de projetos. Um deles é o *SCRUM*, um *framework de gerenciamento de projetos*. Que consiste em separar o projeto em ciclos, cada ciclo de atividade é planejado previamente e é chamado *Sprint*, composto por um período predefinido em que as tarefas devem ser realizadas pela equipe.

Segundo Schwaber (2011)

> Scrum é um framework para desenvolver e manter produtos complexos, esta definição consiste em papéis, eventos, artefatos e as regras do Scrum que unem os demais e os mantém integrados. Ken Schwaber e Jeff Sutherland desenvolveram o Scrum, o Guia do Scrum é escrito e fornecido por eles.

No *Scrum*, os projetos são divididos em chamados de *Sprints*. O *Sprint* representa um *Time Box* dentro do qual um conjunto de atividades deve ser executado. Metodologias ágeis de desenvolvimento de *software* são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de *Sprints* no caso do Scrum (DESENVOLVIMENTO AGIL, 2014).

As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como *Product Backlog*. No início de cada *Sprint*, faz-se um *Sprint Planning Meeting*, ou seja, uma reunião de planejamento na qual o *Product Owner* prioriza os itens do *Product Backlog* e a equipe seleciona as atividades que ela será capaz de implementar durante o *Sprint* que se inicia. As tarefas alocadas em um *Sprint* são transferidas do *Product Backlog* para o *Sprint Backlog* (DESENVOLVIMENTO AGIL, 2014).

A cada dia de uma Sprint, a equipe faz uma breve reunião (normalmente de manhã), chamada *Daily Scrum*. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.

Ao final de um *Sprint*, a equipe apresenta as funcionalidades implementadas em uma *Sprint Review Meeting*. Finalmente, faz-se uma *Sprint Retrospective* e a

equipe parte para o planejamento do próximo *Sprint*. Assim reinicia-se o ciclo. (DESENVOLVIMENTO AGIL, 2014)

Grandes empresas utilizam métodos ágeis para o gerenciamento de suas equipes e projetos, com eles o trabalho em equipe e eficiência são potencializados, deixando mais fácil e organizado o andamento deles, trazendo *feedbacks* e atualizações dia a dia do andamento dos projetos.

Mesmo presente em bastante corporações atualmente, várias empresas têm problemas ou dificuldades na implantação ou uso da metodologia para o seu contexto e filosofia organizacional. Segundo Varaschim (2009) um dos grandes desafios do *Scrum* é manter um *backlog* de projeto organizado, seja com relação às prioridades ou com relação à complexidade necessária para o cumprimento das histórias.

Compreender quais são os principais problemas e desafios presenciados na implantação da metodologia *Scrum* pode ajudar as empresas a se preparar de forma adequada, fazendo com que a probabilidade de sucesso do projeto seja maior.

Portanto, o objetivo deste trabalho é identificar as principais dificuldades enfrentadas na implementação da metodologia ágil *Scrum*, e tratá-las de forme com que o usuário possa usufruir de um ambiente simples e de fácil manuseio.

1.1 Objetivo geral

Este projeto consiste na criação de telas para um aplicativo que ajude na utilização método ágil *SCRUM*, e facilite a iteração dos envolvidos no projeto por meio de um aplicativo acessível e simples.

1.2 Objetivos específicos

- Mostrar a dificuldade dos métodos atuais.
- Desenhar as possíveis combinações de software.
- Validação da ferramenta.

1.3 Métodos

Este trabalho quanto à natureza caracteriza-se como um trabalho original. Já que busca apresentar um conhecimento novo para o mundo e possui uma implicação

prática em sua realização (WAZLAWICK, 2014). Será construído um protótipo de um sistema de gerência de equipe e projeto com base na metodologia *SCRUM*.

Quanto aos objetivos é uma pesquisa exploratória. Será reunido conhecimento técnico e específico depois será desenvolvido um sistema e será observado suas funcionalidades. (WAZLAWICK, 2014).

Em relação aos procedimentos técnicos, este trabalho, caracteriza-se como pesquisa bibliográfica e experimental. Inicialmente, será reunido conhecimento técnico e específico, sendo realizada uma revisão bibliográfica com utilização de livros, sites, revistas, artigos e similares o que é característico de uma pesquisa bibliográfica. (WAZLAWICK, 2014).

AMOLIM (2016) mostra os seguintes passos para uma pesquisa experimental:

O primeiro passo será a formulação do problema, na qual será definido o objeto de estudo que será como tornar mais fácil a interação do usuário com o aplicativo.

Em seguida a construção das hipóteses, no qual será definido qual melhor forma de resolver o problema em questão, a opção escolhida é com base em testes com usuários e pesquisas de ferramentas já existentes perceber qual a maior dificuldade quando se trata de metodologia *SCRUM* para usuários leigos.

Em seguida será a Operacionalização das variáveis. Nesse passo o *front-end* do aplicativo será feito na ferramenta *Figma*, dando uma premissa de como ele ficará para o usuário.

Em seguida Definição do plano experimental. Será estabelecido um procedimento para avaliar os resultados, que será avaliado por meio experiências dos usuários que receberão o primeiro esboço.

Em seguida Determinação do ambiente. Será definido o ambiente alvo. No caso deste projeto será qualquer lugar onde seja utilizado metodologia ágil.

Em seguida a Coleta de dados na qual serão coletados os resultados da exibição das telas. Essa coleta de dados levará em conta experiências dos usuários que visualizarão o protótipo do aplicativo.

Depois da coleta de dados serão avaliados e analisados os dados adquiridos e as medidas possíveis para resolver qualquer problema

A seguir será feita a coleta dos dados dos usuários que visualizaram o esboço. Com base nos dados coletados será feita uma versão final do protótipo.

2 CONCEITOS TEÓRICOS.

Neste capítulo são apresentados os conceitos mais importantes utilizados neste projeto. Estes são necessários ao entendimento do processo de criação utilizando métodos ágeis, tanto quanto a apresentação do método *Scrum*.

Os conceitos apresentados são: Métodos ágeis, *Scrum*, seus eventos e *Sprint*, proporcionando soluções e problemas que podem ser encontrados durante o decorrer deste projeto.

2.1 Métodos ágeis

Ao longo dos anos, vários métodos de desenvolvimento de produtos foram apresentados. Entre eles, existem os chamados métodos ágeis (AMBLER, 2002), também chamados de métodos leves (FOWLER, 2000). Estes métodos são adaptativos e mais flexíveis em relação aos tradicionais.

Os métodos tradicionais de desenvolvimento se concentram na geração de documentação do projeto e na aplicação de processos rigorosos. Os métodos ágeis, por outro lado, focam na entrega contínua de produtos (MUNDIM et al., 2002) e nas interações entre os indivíduos.

Além disso, eles são recomendados para cenários em que possuem constante mudança de requisitos e os resultados devem ser entregues em curtos espaços de tempo. Geralmente, esses métodos dividem o desenvolvimento em várias iterações de ciclos menores e fazem entregas ao final de cada uma delas, de forma que o cliente (interno ou externo) receba uma versão que some valor ao seu negócio.” (DANTAS, 2003, p.37).

Sendo assim, as mudanças de requisitos podem ser assistidas pelos desenvolvedores no começo de cada ciclo. E há *feedback* do cliente para a equipe de desenvolvimento, o que reduz o risco do projeto.

Essas abordagens são fortemente influenciadas pelas melhores práticas industriais japonesas, em particular os princípios de manufatura enxuta implementados pela Honda e Toyota e as estratégias de gestão do conhecimento de Takeuchi e Nonaka (2004) e Senge (1990).

Nessas abordagens, a fase de planejamento inicial é minimizada, de modo que os desenvolvedores se concentram em entregar o produto no final de cada

iteração, em vez de desenvolver diretrizes e planos para todo o projeto.

Metodologias ágeis de desenvolvimento de *software* têm crescido em popularidade nos últimos anos. No entanto, há pouca pesquisa empírica sobre este tema. Uma recente revisão sistemática da literatura sobre esse tema (DYBÅ; DINGSOYR (2008) encontrou 1.996 artigos na área. Destes, apenas 36 são estudos empíricos com rigor metodológico aceitável, credibilidade e relevância, representando apenas 1,8% dos trabalhos. Além do *Scrum*, existem várias metodologias ágeis: *Extreme Programming, Feature Driven Development, Dynamic Systems Development Methods, Adaptive Software Development, Crystal, Practical Programming e Test Driven Development*. Para o desenvolvimento deste trabalho, o método de escolha foi o Scrum, que é detalhado nos tópicos abaixo.

2.2 Práticas e etapas do Scrum

Este método não requer ou fornece qualquer técnica específica para a fase de desenvolvimento, apenas estabelece conjuntos de regras e práticas gerenciais que devem ser adotadas para o sucesso de um projeto.

O ponto inicial do *Scrum* é o *Backlog* do Produto, sendo considerada a prática responsável pelo armazenamento e gerenciamento dos requisitos coletados, conforme aponta Schwaber e Beedle (2002).

Nesta prática, por meio de reuniões com todos os envolvidos, investidores, clientes e parceiros no projeto, são apontadas todas as necessidades do negócio e as funcionalidades a serem desenvolvidas. Assim, o *Backlog* do Produto é uma lista de funcionalidades, ordenadas por prioridade, que provavelmente serão desenvolvidas durante o projeto.

A reunião diária de *Scrum* (*Daily Scrum*) é um rápido encontro que ocorre entre os membros do time para definir quais serão as tarefas do dia e saber os resultados das tarefas do dia anterior.

Esta reunião é também chamada de *stand up Meeting* (reunião em pé), já que é de praxe que todos os membros a realizem de pé, de forma a conseguir maior agilidade. Três perguntas são respondidas por cada membro sobre suas responsabilidades (RISING; JANOFF, 2000): O que foi feito ontem? O que será feito hoje? Há algum obstáculo à realização das atividades? Na reunião diária de Scrum, os membros do time não respondem a essas perguntas como forma de

prestar contas à gerência, mas sim como formalização do comprometimento com o resto da equipe. Assim, todos os membros do time conhecem as metas individuais de cada integrante, conhecem seus impedimentos (riscos) e podem cobrar compromissos assumidos.

O *Sprint* é considerado a principal prática do Scrum. É o período no qual são implementados os itens de trabalho definidos no *Backlog* do Produto pela equipe *Scrum*. Conforme Abrahamsson et al. (2002), ele normalmente dura de uma a quatro semanas, mas não há uma regra para isto; as equipes que decidem a duração a ser adotada para o projeto.

No caso do desenvolvimento de *software*, o *Sprint* inclui as fases tradicionais do desenvolvimento de *software*: requisitos, análise, projeto e entrega.

O *Backlog* do *Sprint* é um subconjunto do *Backlog* do Produto. Ele é uma lista de atividades a serem desenvolvidas durante o *Sprint*. Sua definição acontece durante a Reunião de Planejamento do *Sprint*. Já a Reunião de Revisão do *Sprint* (*Sprint Review Meeting*) é a reunião que acontece após cada *Sprint*. Nela, a equipe discute sobre seus erros, acertos e lições aprendidas.

No início do projeto, cliente e desenvolvedores definem o *Backlog* do Produto (sua lista de requisitos). Também são estimados os custos do projeto e definidas as datas para entrega de resultados a partir da priorização mais favorável ao cliente. Uma análise inicial de riscos é preparada.

As ferramentas de trabalho e os integrantes das equipes são escolhidos. Um dos desenvolvedores é eleito "*Scrum Master*", cujo papel se assemelha a um gerente de projetos.

O *Scrum Master* trabalha para que o processo *Scrum* aconteça e para que não existam impedimentos para os membros da equipe realizarem seu trabalho. Remover os obstáculos apontados na reunião de *Scrum* diária é seu dever, de modo que os desenvolvedores se concentrem apenas nas questões técnicas.

Esses obstáculos são colocados em uma lista chamada de *Backlog* de Impedimentos, que fica à vista de todos. Outro papel importante no método é o do Dono do Produto (*Product Owner*).

Este membro do time geralmente representa o cliente (interno ou externo). Ele define quais são os requisitos e qual é o grau de importância e prioridade de cada um deles. Este membro necessita conhecer muito bem as regras de negócios do cliente, de forma que ele possa tirar qualquer dúvida que o time possa ter em

relação às funcionalidades do produto.

No início de cada *Sprint*, quando as equipes fazem a lista das atividades que precisam ser realizadas naquele *Sprint* (*Backlog do Sprint*), as responsabilidades são distribuídas. Os desenvolvedores discutem os padrões que serão adotados e as atividades de análise, codificação e testes se iniciam.

Ao final de cada *Sprint*, uma versão do produto (no caso do produto de *software*, um executável do *software*) é apresentada ao cliente para obter a retroalimentação. Os defeitos encontrados são adicionados ao *Backlog* do produto.

Ao longo de todo o projeto, são aplicados mecanismos de gerência *Scrum*, como o acompanhamento de alguns controles. A quantidade de funcionalidades não entregues, a necessidade de mudanças para corrigir defeitos ou para atualização tecnológica, os problemas técnicos encontrados e os riscos e as estratégias para evitá-los são exemplos de controles observados durante o desenvolvimento. O último artefato do *Scrum* é o gráfico *burndown*.

Trata-se de uma representação gráfica do trabalho restante em comparação com o trabalho já realizado. Geralmente, coloca-se a quantidade de trabalho no eixo vertical e o tempo no eixo horizontal. Ele é muito útil para predizer quando todo o trabalho será completado e para alarmar o time em caso de atraso (que ficará bastante aparente). Geralmente é traçada uma linha com a representação da execução do trabalho. Esta linha representa o esforço já realizado na execução das tarefas. Espera-se que a execução das atividades (tarefas) leve a linha de início em Y ao encontro de X. Este encontro representa o término das execuções das tarefas.

2.3 O Time *Scrum*

A primeira tarefa de um Time *Scrum* é que ele é seja auto-organizado, sempre escolhendo as melhores decisões no desenvolvimento de suas tarefas. A equipe garante que a entrega incremental está "pronta" conforme exigido pela história, e o projeto desenvolvido até agora está de forma funcional. Para ser eficaz, deve conter de 5 (cinco) a 9 (nove) membros, entre eles: *Product Owner*, Time de Desenvolvimento e *Scrum Master*. Os papéis de cada membro são os seguintes (Machado, 2009):

• *Product Owner*: É o dono do projeto, ou quem pagou por ele ou ainda mesmo é o responsável, e tem informações de como o projeto deve chegar, prioridades e necessidades. Além de controlar e gerenciar a entrega, é sua responsabilidade tornar seus objetivos totalmente claros e visíveis para a equipe originadora no *Product Backlog*.

• *Scrum Master*: Responsável por gerenciar tudo o que acontece durante o desenvolvimento do projeto e manter a transparência entre os membros. É responsável por agendar reuniões focadas no *Scrum* e garantir que a comunicação da equipe seja perfeita. O *Scrum Master* também garante que todos os rituais do Scrum sejam executados e seus artefatos sejam usados corretamente. Também é seu trabalho remover quaisquer obstáculos que impeçam o progresso de sua equipe.

• Equipe: é o grupo que realiza a construção do projeto, ou seja: desenvolvimento e testes, e tem autoridade para tomar as decisões necessárias e organizá-las para executar cada tarefa dentro do tempo estimado de cada *Sprint Backlog*. Não há hierarquia entre eles, o objetivo é se autogerenciar nas atribuições de tarefas.

2.4 Eventos Scrum

As atividades são necessárias para minimizar as reuniões fora do plano *Scrum*. Geralmente tem uma duração máxima para que não haja discussão além do que está definido, mas pode terminar quando tudo estiver realmente esclarecido e definido.

Cada evento é uma oportunidade para examinar e ajustar algo, por isso é necessário tirar todas as dúvidas para esclarecê-lo, garantindo assim a transparência de acordo com a definição do encontro (Sutherland e Schwaber, 2013). Os eventos mencionados acima são: *Sprint Planning Meeting, Daily Meeting e Sprint Retrospective*. Também é muito comum ajudar a melhorar a qualidade do desenvolvimento por meio de prazos de entrega, conhecidos como gráficos de *burndown*. Cada um desses itens será analisado a seguir (Brod, 2015).

2.4.1 *Sprint planning meeting*

Começa com a Reunião de Planejamento do *Sprint*, a reunião onde o *Sprint* é planejado. Após o *Product Owner* definir o projeto (metas e requisitos), é feita uma lista de histórias que contém o que é necessário para o desenvolvimento do projeto. Os pontos geralmente são definidos e destacados por usuários, sistemas e empresas ao usar um aplicativo.

Uma vez que esta lista é elaborada, o *Product Backlog* é concluído, dividido em versões, e um novo *Sprint Backlog* é gerado em cada versão. A equipe define os componentes de cada *Sprint* em função do *Product Backlog*, sempre analisando as tarefas que podem ser entregues dentro do prazo estimado, que pode variar de 1 a 4 semanas, e deixando o restante para o próximo *Sprint* do *Scrum One*, apenas se comprometa a fazer tarefas que você pode entregar no prazo.

Após a definição do *Sprint*, a responsabilidade pela execução de cada tarefa é dividida, não se limitando a quantas tarefas cada membro pode assumir. Na primeira reunião, também é definido o tempo para que todos executem, e nesta fase todas as equipes votam para participar com base em suas opiniões.

Muitas equipes optaram por um jogo chamado *Planning Poker*, no qual são encontradas cartas com pontuações e cada pessoa é mostrada uma ao mesmo tempo (as pontuações são definidas pela equipe) para simbolizar a dificuldade e o tempo de cada *Sprint*, essas pontuações podem ser nas cartas, ou ainda é um aplicativo que simboliza as cartas. Se as pontuações de cada membro estiverem muito distantes, discuta as dificuldades até chegar a um consenso.

2.4.2 *Daily Meeting*

Durante um *sprint*, a equipe se auto-organiza para que todos executem suas tarefas dentro dos prazos. O *Scrum Master* deve garantir que não haja distrações externas durante a execução, pois nada pode distrair os objetivos estimados.

As reuniões são realizadas diariamente para garantir a qualidade da entrega. Esses encontros podem ser presenciais ou por meio de conferências, idealmente sempre no mesmo local e no mesmo horário para que todos caibam em sua agenda, com duração de 15 a 20 minutos, permitindo que todos falem e desenvolvam uma conversa sobre o que fizeram, o que eles vão fazer, obstáculos e dúvidas. Isso é feito

melhor na frente de um quadro mostrando as tarefas que aparecem no Sprint, conforme um quadro tradicional ou na frente de um computador mostrando ferramentas com tarefas.

Geralmente começa no dia da manhã, já que não há hierarquia na equipe, este é o momento de esclarecer as dúvidas que surgiram no dia anterior e informar a equipe como está indo o *Sprint*, mas não necessariamente durante este período. Quando os membros da equipe mencionam obstáculos, é trabalho do *Scrum Master* resolver rapidamente esses obstáculos.

2.4.3 *Sprint Review*

A cada entrega, ou seja, de 1 a 4 semanas, será realizado um Sprint Review, e os produtos desenvolvidos no *Sprint* serão apresentados gradativamente sob a condição de funcionamento normal do sistema. Além da equipe *Scrum*, essa reunião pode incluir clientes, gerentes e até engenheiros de outros projetos.

2.4.4 Sprint Retrospective

Após essa reunião de Revisão, haverá uma reunião de Retrospectiva da *Sprint*, que é uma reunião para revisar tudo o que aconteceu na *Sprint*, apontar os pontos positivos e negativos que precisam ser melhorados, e as lições aprendidas ou uma dificuldade em alguma parte, seja no desenvolvimento de produtos, comunicação ou transparência. Essa reunião geralmente é feita pela equipe e pelo *Scrum Master*. É natural que a equipe cometa vários erros no primeiro *Sprint*, mas é esse tipo de reunião que garante o sucesso do próximo *Sprint*.

2.4.5 *Burndown Chart*

Ao final de cada *Sprint*, pode-se contar também com o *Burndown Chart*, que é artefato importante do Scrum, atuando no controle de qualidade.

De forma simples, é estimado quantos dias tem uma *Sprint*, com um total de 100 pontos que serão consumidos ao longo dos dias, sendo 20 pontos por dia a média de consumo correto, caso um dia se tenha consumido mais que 20, a ideia é recuperar no próximo dia.

3 METODOLOGIA

Este estudo baseou-se em uma estratégia qualitativa de pesquisa, de caráter exploratório, por meio de uma pesquisa de campo.

Neste capítulo, pretende-se demonstrar os procedimentos metodológicos do tipo de pesquisa utilizado. Será abordado também os critérios para a construção do universo de estudo, o método de coleta de dados, a forma de tratamento desses dados e, por fim, as limitações do método escolhido.

3.1 Tipo de pesquisa

Tomando como ponto de partida o objetivo desta pesquisa – que é investigar a melhor forma de criação da de um protótipo que satisfaça o usuário que utiliza o método SCRUM –, foi decidido adotar o método de pesquisa qualitativa, de caráter exploratório, sendo este considerado o mais apropriado para o tipo de análise que foi proposto. Antes, porém, se faz necessário contextualizar o tipo de pesquisa escolhido para um melhor entendimento a respeito.

Quanto ao objetivo, foi escolhido o tipo de pesquisa a ser realizada que é a pesquisa qualitativa. É classificada como pesquisa exploratória, E em há pouco conhecimento acumulado e sistemático de acordo com sua natureza de investigação, excluindo suposições que possam surgir durante a investigação do fim do estudo" (VERGARA, 2009, p. 42).

Sobre os meios pesquisa, foi optado por estudos de campo, que também se baseavam em Vergara, IS: "Uma investigação empírica em um só lugar ou ter elementos para explicá-lo. pode incluir entrevistas, seja para aplicar questionários, testes e observação" (2009, p. 43). Quanto aos procedimentos qualitativos, segundo Creswell (2007, p.184 e 188), "eles se baseiam em dados de texto e imagem, têm passos únicos na análise de dados e usam estratégias diversas de investigação".

O pesquisador irá para o local onde o entrevistado estava realizando a pesquisa, permitindo o pesquisador estar envolvido na experiência do participante ou entrevistado. A pesquisa qualitativa é interpretativa e o pesquisador se envolve de forma intensa com os entrevistados.

Tesch (1990, p. 55), por sua vez, lembra que, na investigação qualitativa, os pesquisadores coletam informações que não podem ser representadas

numericamente. No entanto, segundo os autores, a pesquisa qualitativa pode incluir outras Informações além das palavras, como desenhos e fotografias, para exemplo.

Vários aspectos emergem na pesquisa qualitativa: A pesquisa pode mudar e melhorar com a participação de pesquisadores com os participantes, descubra o que perguntar. Este processo permite Interpretação extensiva por pesquisadores à medida que aprendem padrões gerais apresentados na entrevista.

Esse fenômeno diz Fatos sobre pesquisadores que filtram dados através de lentes um indivíduo em um determinado momento, vendo um fenômeno em um determinado momento. Nessa perspectiva, "a pesquisa qualitativa se manifesta como Visão ampla, não microanálise (…). Os pesquisadores usam o raciocínio complexo, multifacetado, interativo e simultâneo" (CRESWELL, 2007, p. 186- 187).

Em uma abordagem mais ampla, Tesch (1990, p.57-58) argumenta que nas ciências sociais podemos considerar 46 estudos qualitativos. No entanto, os autores ressaltam que os tipos se sobrepõem e alguns são até sinônimos.

Dos tipos destacados por Tesch (1990), estamos particularmente interessados na fenomenologia – o método escolhido para este estudo. A fenomenologia é um método de pesquisa relativamente novo. Originou-se de um grupo de pesquisadores do Departamento de Educação da Universidade de Gotemburgo, na Suécia, e suas primeiras publicações datam do início da década de 1980. No entanto, o debate e as críticas a essa abordagem só se tornaram mais comuns na última década.

A fenomenografia é uma metodologia de pesquisa qualitativa, fundamentada em um paradigma interpretativo, baseado na compreensão do sentido humano, nas maneiras de se enxergar um fenômeno particular e nas ideias das pessoas a respeito do mundo a sua volta (AKERLIND, 2005; MARTON, 1981,1986).

Marton (1986, p.31) assevera que "quando investigamos o entendimento das pessoas a respeito de vários fenômenos, conceitos e princípios, nós repetidamente achamos que cada fenômeno, conceito ou princípio pode ser entendido qualitativamente de diferentes maneiras".

Marton (1986) prossegue dizendo que a fenomenografia é um método de pesquisa por meio do qual o pesquisador pretende mapear qualitativamente as maneiras de experimentação das pessoas, a forma como elas percebem, conceituam e entendem determinado fenômeno ou o mundo a sua volta.

Segundo Tesch (1990), a ênfase está no “como”. As abordagens fenomenológicas estudam como as pessoas explicam a si mesmas e aos outros, o que está por trás delas e como modificar essas explicações. Marton (1986, p.33) explicou que o objetivo não é tentar descrever as coisas "como elas são", mas tentar descrever como elas aparecem para as pessoas.

Os autores ensinam que o ponto de partida da fenomenologia é o "relacionamento", a relação entre um indivíduo e algum aspecto do mundo ao seu redor. Ainda segundo Marton (1999), o objetivo da pesquisa é descrever qualitativamente as diferentes formas como vários fenômenos (ou um mesmo fenômeno) são vivenciados e como as pessoas percebem determinada realidade. Essencialmente, é o estudo da mudança, uma mudança qualitativa entre as maneiras pelas quais o mesmo fenômeno é observado, experimentado e compreendido.

3.2 Ferramentas utilizadas

Para conclusão desse aplicativo foram utilizadas ferramentas de desenvolvimento *front-end*. Para construção e experiencia de usuário.

Interações entre as telas foram feitas na própria ferramenta proporcionando um vislumbre da aplicação.

3.2.1 *Figma*

Muitos projetos derivam de novas ideias, melhorias ou inovações em projetos existentes. No entanto, é preciso concordar que criar algo do zero leva tempo, dedicação e dinheiro, e antes de fazer grandes investimentos e apostas, é altamente recomendável testá-lo. É isso que o *Figma* torna possível: testar e testar novos conceitos, produtos ou soluções com baixo investimento. Imagine que seja criado algo que alguém quer ou precise, com o *Figma*, você pode: lançar protótipos digitais; criar um fluxo de navegação, criar e implementar sistemas de design. Com as funções adicionais disponíveis, a padronização do projeto também pode ser simplificada.

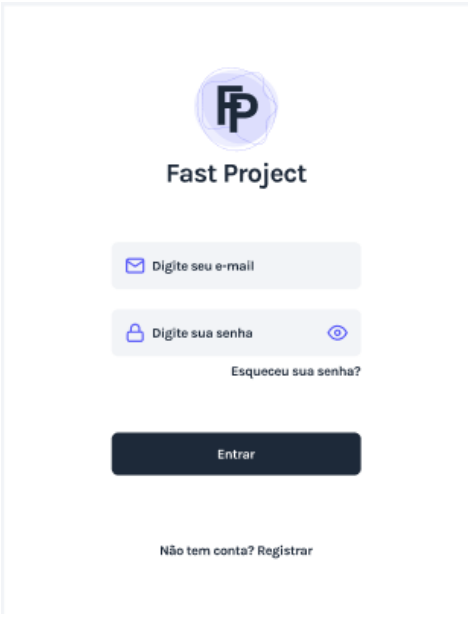
4 Implementação e resultados

Neste capítulo são apresentadas as telas do aplicativo *Fast Project*, mostrando as suas funcionalidades e detalhando as ações disponíveis para o usuário.

4.1 Tela de login

Na figura 1 se encontra a tela de login. A tela de login segue o padrão, sendo possível fazer o login com e-mail e senha.

Figura 1 - Tela de login



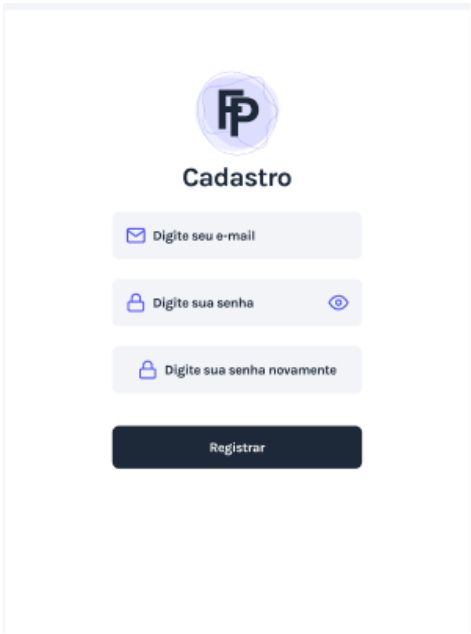
Fonte: Autoria própria.

Caso o usuário esqueça sua senha é possível fazer a recuperação dela, e também é possível fazer o registro caso a pessoa não tenha.

4.2 Tela de cadastro

Na figura 2, é apresentado a tela de cadastro, onde o usuário faz o registro no aplicativo. Sendo possível ser feito utilizando e-mail e senha válidos.

Figura 2 - Tela de cadastro



Fonte: Autoria própria.

4.3 Tela de recuperação de senha

Na figura 3 é apresentado a tela responsável pela recuperação da senha e, para que a senha seja recuperada, é necessário digitar um e-mail registrado.

Figura 3 - Tela de recuperação de senha

Fonte: Autoria própria.

Quando o email é enviado é mostrado uma imagem de confirmação como é apresentado na figura 4.

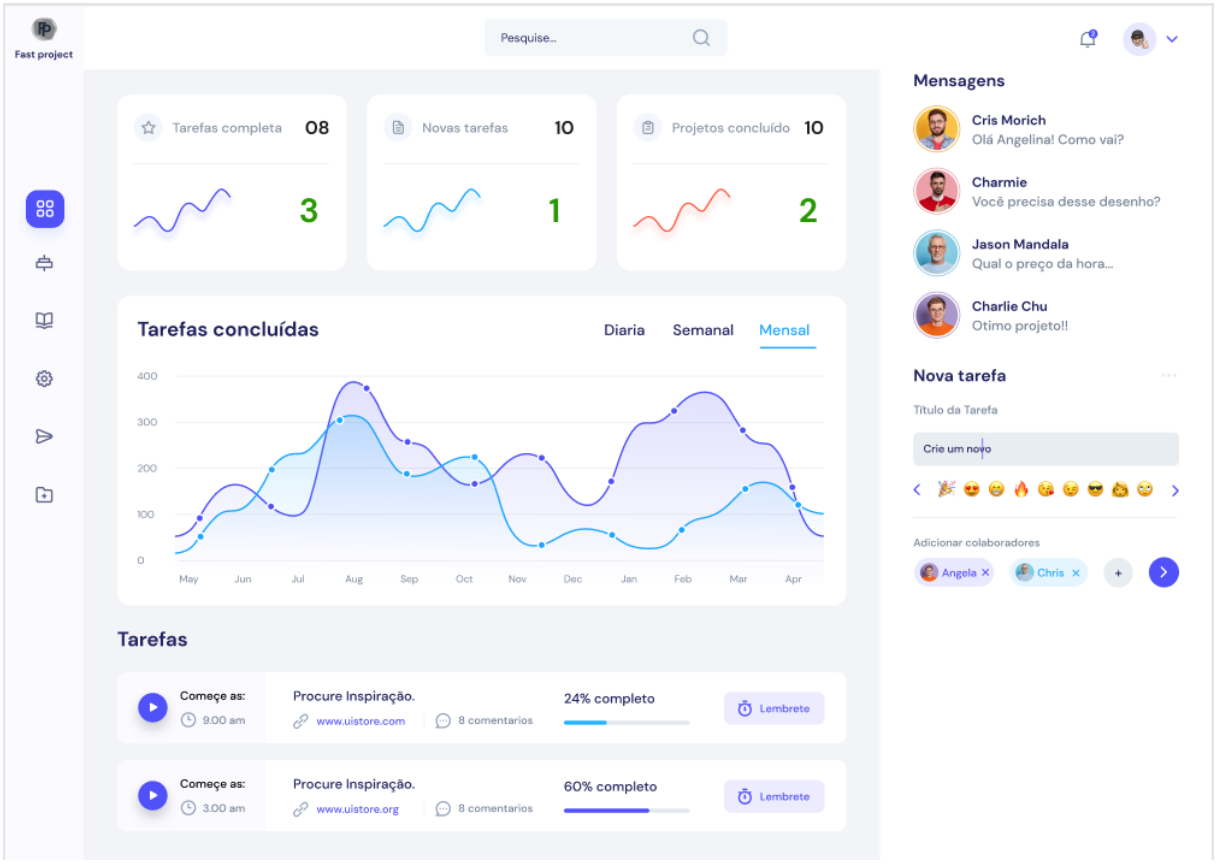
Figura 4 - Tela de confirmação de envio

Fonte: Autoria própria.

4.4 Tela inicial

A figura 5 representa a tela inicial do aplicativo, dentro dela é possível ir para qualquer parte do aplicativo.

Figura 5 – Tela de inicial



Fonte: Autoria própria.

Nesta tela pode ser encontrado o registro de tarefas completas, em progresso e concluídas.

Mostra-se também um gráfico informativo, sendo neste, possível filtrar os dados levantados por dia, semana e mês.

O temporizador de tarefas, também exibido nesta figura, contém os comentários, porcentagem concluída e acionador de lembrete.

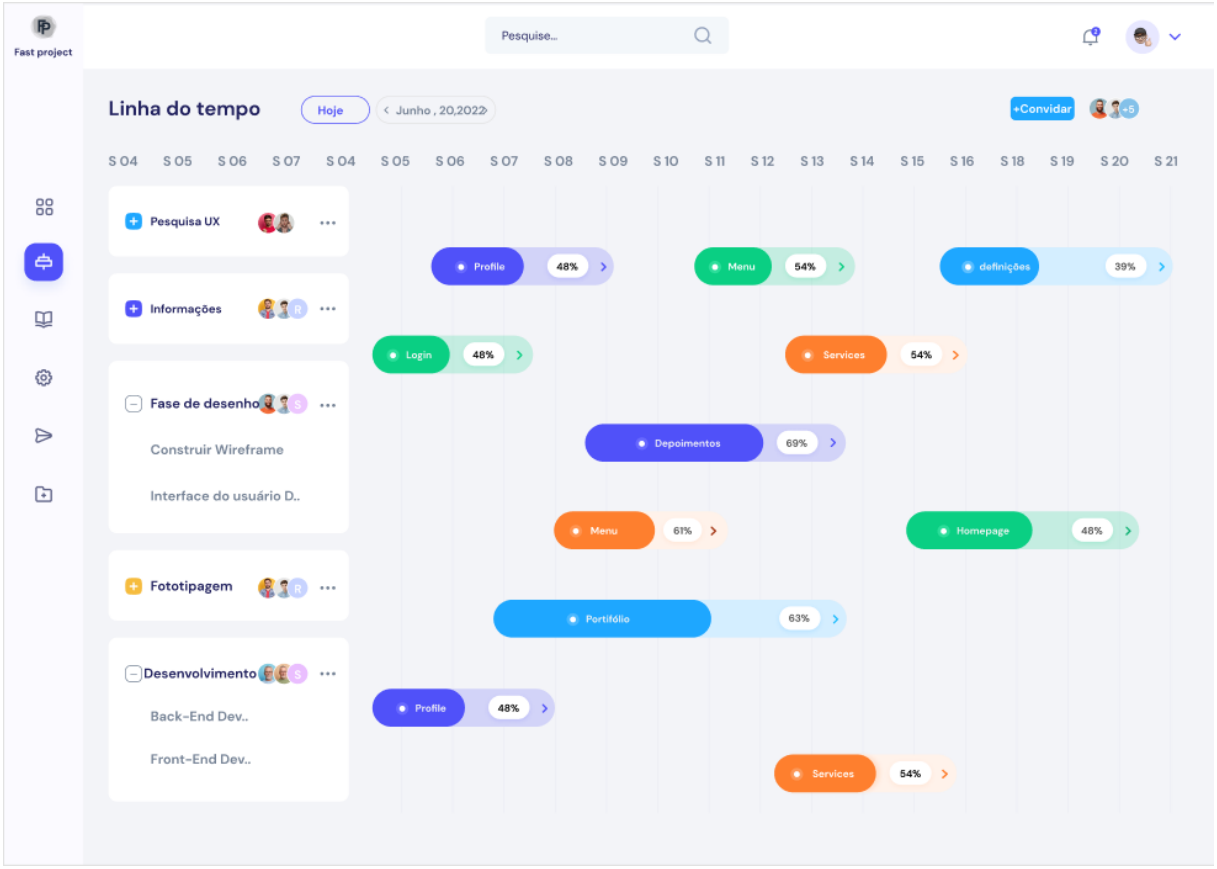
É mostrado também o chat contando suas últimas mensagens recebidas, sendo possível também realizar o envio de mensagens.

E por fim, é mostrado os botões ao lado direito para troca de telas.

4.5 Linha do tempo das tarefas

A figura 6 mostra a linha do tempo contendo todas as tarefas.

Figura 6 – Tela de Linha do tempo



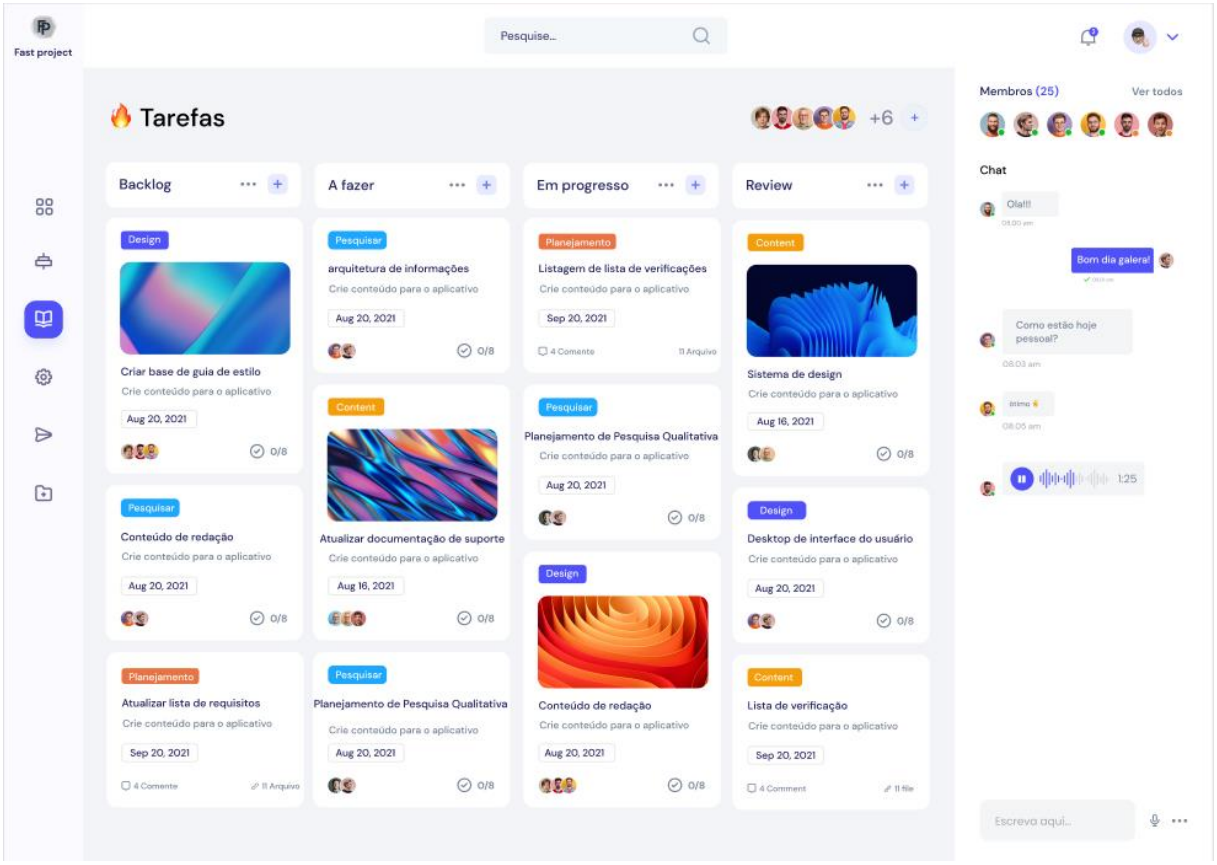
Fonte: Autoria própria.

Nesta tela é mostrada o informativo de todas as tarefas, porcentagem e técnicos envolvidos, assim como a linha do tempo com tarefas e tipo de tarefas.

4.6 Tela das tarefas

A figura 7 mostra a tela de informação sobre todas as tarefas.

Figura 7 – Tela de tarefas



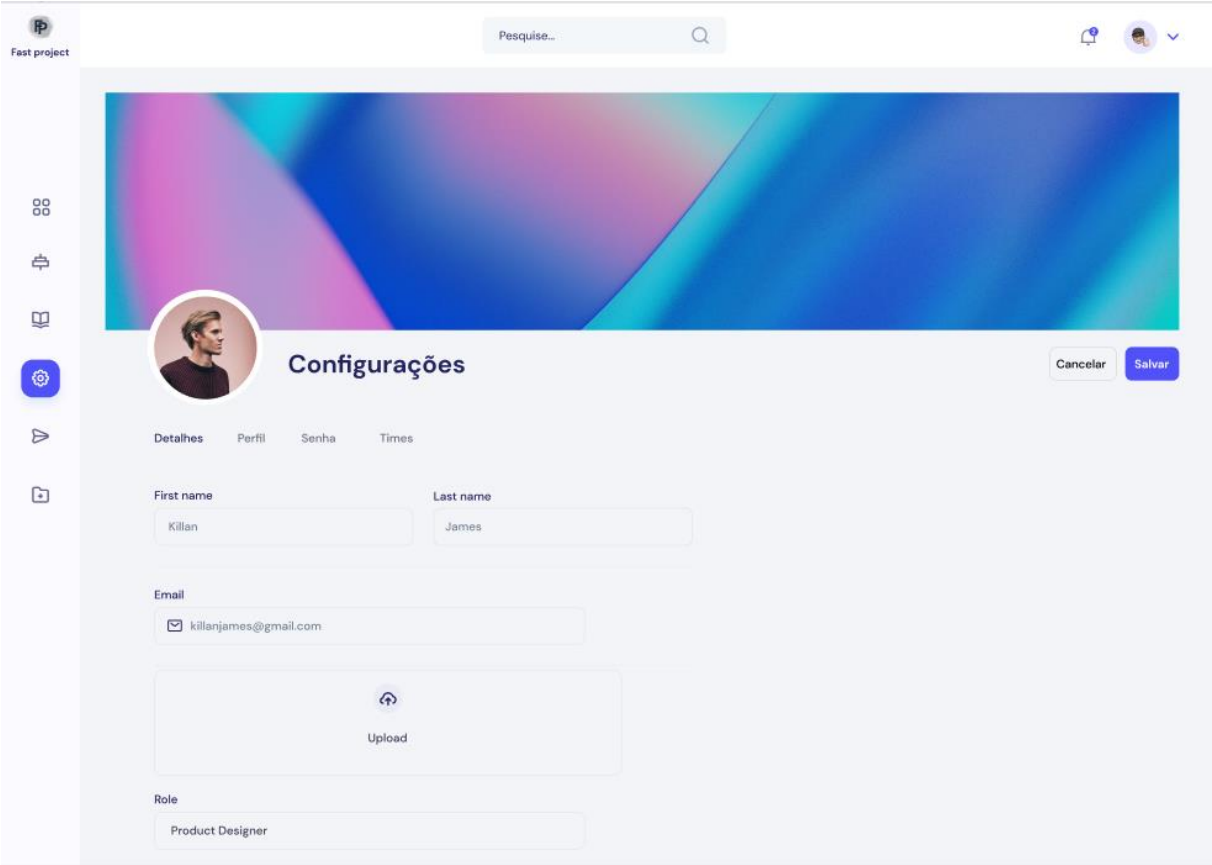
Fonte: Autoria própria.

Nesta tela, é mostrado todas as tarefas da empresa, informativo de pessoas, comentários e tarefas. Também possível a interação com o chat ,assim como a criação e edição de tarefas.

4.7 Tela de configurações

A figura 8 mostra a tela de configurações do usuário.

Figura 8 – Tela de configurações



Fonte: Autoria própria.

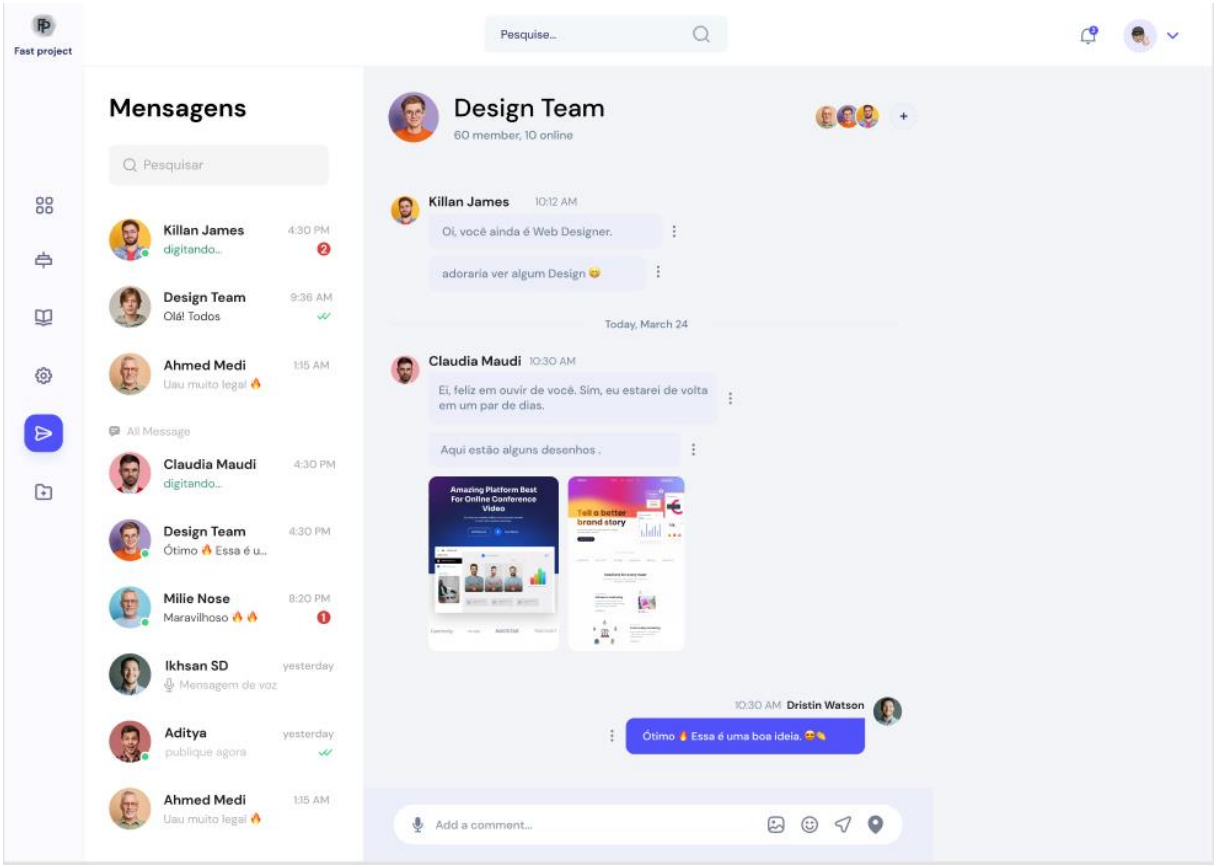
Já nesta tela de configuração, é possível fazer modificações no usuário como perfil, nome, e-mail e retrato. Esta tela também apresenta a possibilidade de troca de senha, sendo que, para este caso, necessário então colocar a última senha digitada como confirmação, para segurança da aplicação.

A aba times, desta figura, é responsável pelo cadastro e por adentrar a novos times cadastrados.

4.8 Tela do chat

A figura 9 mostra a tela do chat para interações entre os usuários do grupo.

Figura 9 – Tela do chat



Fonte: Autoria própria.

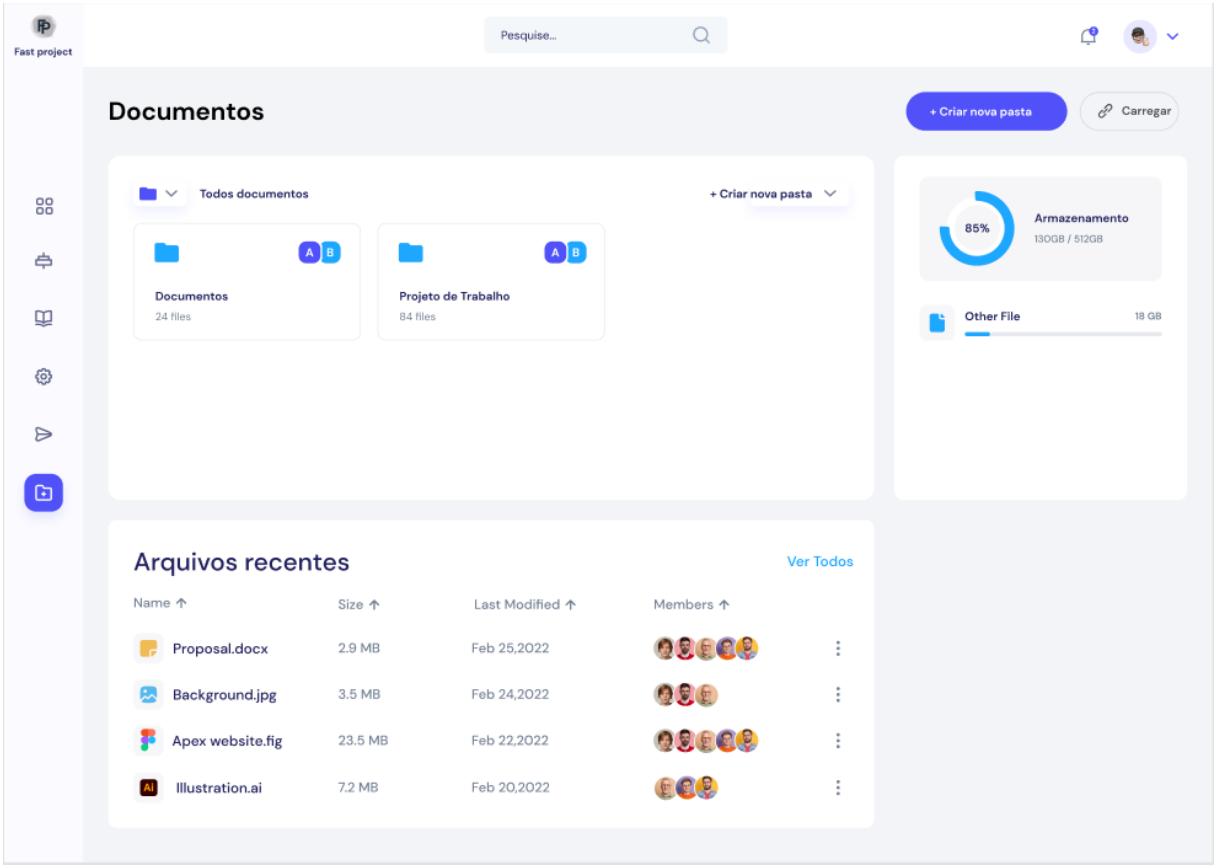
Esta figura mostra a tela do chat para interações entre os usuários do grupo. É responsável pela comunicação geral do time ou das pessoas envolvidas nos processos, assim como o envio de mensagens, sendo possível desta forma, enviar fotos, emojis e arquivos.

Também é possível realizar pesquisa por pessoas ou mensagens recebidas.

4.9 Tela de envio de documentos

A figura 9 mostra a tela de armazenamento de arquivos.

Figura 10 – Tela de envio de documentos



Fonte: Autoria própria.

Nesta tela, é trabalhado o envio e recebimento de documentos, sendo também mostrado uma possível pesquisa e filtro de documentos pelo nome.

É possível verificar que existe a opção de visualização dos últimos arquivos enviados, contendo a pessoa ou equipe que o enviou.

Mostra-se também a quantidade de arquivos armazenados.

5 Considerações finais

Através dos conceitos de UX, foi possível atender às necessidades do aplicativo, respeitando a experiência de usuário, como telas de fácil entendimento e objetivas.

Os resultados obtidos foram satisfatórios visto que foi possível criar telas objetivas, simples e de fácil interação. Também foram atendidos vários pontos da metodologia ágil, trazendo para telas seus processos de forma descomplicada.

Este trabalho apresentou a teoria e o desenvolvimento do aplicativo *FastProject* voltado ao *front-end* para aplicação da metodologia ágil, apresentando a sua justificativa, descrição geral, tecnologias utilizadas para o desenvolvimento e seus resultados.

Para justificativa do uso do método foi feito uma pesquisa profunda, trazendo todas as informações, opiniões e estatísticas apontando a importância da metodologia ágil Scrum. Foi desenvolvido uma pesquisa de como um novo aplicativo para aplicação da metodologia *scrum* poderia ser desenvolvido. Funções, funcionalidade e interação com usuário.

REFERENCIAS.

AMBLER, S. Agile modeling. New York: Wiley Computer Publishing, 2002.

CRESWELL, J. W. (2007). Qualitative inquiry & research design: Choosing among five approaches (2nd ed.). Thousand Oaks, CA: Sage.

DANTAS, V. F. Uma metodologia para o desenvolvimento de aplicações Web num cenário global. 2003. Dissertação (Mestrado)-Centro de Ciências e Tecnologia, Universidade Federal de Campina Grande, Campina Grande, 2003.

DYBÅ, T.; DINGSØYR, T. *Empirical studies of agile software development: a systematic review*. Information and Software Technology, v. 50, n. 9-10, p. 833-859, 2008. http://dx.doi.org/10.1016/jinfsof.2008.01.006.

FOWLER, M. Put your process on a diet. Software Development, 2000. Disponível em: <http://www.sdmagazine.com/articles/2000/0012/>. Acesso em: jul. 2003.

MACHADO, M. (2009). SCRUM – Método Ágil. Fonte: Faculdade do Guarujá: http://www.faculdadedoguaruja.edu.br/revista/downloads/edicao12009/Artigo_5_Prof_Marcos.pdf

MARTON, F. (1986). Phenomenography: A research approach to investigating different understandings f reality. Journal of Thought, 21 (3), 28-49

MARTON, F. (1994) Phenomenography. In Torsten Husén & Neville Postlethwaite (Eds.), The Inter- national Encyclopedia of Education (pp. 4424- A4429). Oxford: Pergamon Press.

MUNDIM, A. P. F. et al. Aplicando o cenário de desenvolvimento de produtos em um caso prático de capacitação profissional. Gestão & Produção, v. 9, n. 1, p. 1-16, 2002.

RISING, L.; JANOFF, N. S. *The Scrum software development process for small teams*. IEEE Software, v. 17, n. 4, p. 26-32, 2000. http://dx.doi.org/10.1109/52.854065.

SCHWABER, K.; BEEDLE, M. Agile software development with SCRUM. Prentice Hall, 2002.
SEBRAE; INSTITUTO DE PESQUISAS TECNOLÓGICAS.

SCHWABER, K., SUTHERLAND, J. (2013). O guia do Scrum. Fonte: Scrum Guides: http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf

SCHWABER, K. SCRUM Development process. 1995. Disponível em: <http://jeffsutherland.com/oopsla/schwapub.pdf>. Acesso: jul. 2003.

SCRUM. [2013]. Disponível em: <https://www.desenvolvimentoagil.com.br/scrum/>. Acesso em: 10 mai. 2021.

SUTHERLAND, J., & SCHWABER, K. (2013). The definitive guide to scrum: The rules of the game. Recuperado de www.scrum.org

TESCH, R. *Qualitative Research: Analysis Types and Software Tools. New York, Philadelphia; London: The Falmer Press*, 1990.

VARASCHIM, J. D. Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet, Rio de Janeiro: PUC, 2009.

VERGARA, S. C. Projetos e relatórios de pesquisa em administração. São Paulo: Atlas, 2009.

WAZLAWICK, R. S. Metodologia de Pesquisa para Ciência da Computação. 2. ed. Rio de Janeiro: Elsevier, 2014.