

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LEONARDO MENDES DA SILVA
TATIANE SILVA CACIQUE

FOCUS TIME

Gestão de tempo

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Carlos Alberto Lucas

FRANCA/SP

2025

Focus Time

Leonardo Mendes da Silva¹
Tatiane Silva Cacique²

Resumo

O presente trabalho desenvolve o FocusTime, um sistema integrado de gerenciamento de tempo baseado na Técnica Pomodoro, destinado a melhorar a produtividade e organização de tarefas de usuários. A problemática central aborda a dificuldade contemporânea em manter o foco e a disciplina no gerenciamento do tempo, especialmente em ambientes com múltiplas distrações digitais. O sistema foi concebido como uma aplicação móvel com backend em Node.js e frontend em React Native, incorporando funcionalidades de timer Pomodoro personalizável, gestão de tarefas com priorização e geração de relatórios de produtividade. A metodologia de desenvolvimento seguiu princípios ágeis, com modelagem de processos através de BPMN, especificação de requisitos funcionais e não funcionais, e diagramas UML para documentação arquitetural. Os resultados alcançados incluem a implementação completa do sistema com interface intuitiva, sincronização em tempo real, persistência de dados local e remota, e mecanismos de notificação inteligentes. A validação preliminar indica que a solução proposta efetivamente auxilia usuários na manutenção do foco e na melhoria da gestão do tempo, demonstrando potencial para impactar positivamente hábitos de produtividade.

Palavras-chave: Gerenciamento de tempo, Pomodoro, Produtividade, React Native, Sistema móvel, Timer.

Abstract

This paper develops FocusTime, an integrated time management system based on the Pomodoro Technique, designed to improve users' productivity and task organization. The central problem talks about the contemporary difficulty in maintaining focus and discipline in time management, especially in environments with multiple digital distractions. The system was conceived as a mobile application with Node.js backend and React Native frontend, incorporating features such as customizable Pomodoro timer, task management with prioritization, and productivity reports generation. The development methodology followed agile principles, with process modeling through BPMN, specification of functional and non-functional requirements, and UML diagrams for architectural documentation. The achieved results include the complete system implementation with intuitive interface, real-time synchronization, local and remote data persistence, and intelligent notification mechanisms. Preliminary validation indicates that the proposed solution effectively assists users in maintaining focus and improving time management, demonstrating potential to positively impact productivity habits.

¹ Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: [leonardo.mendes@fatec.sp.gov.br].

² Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: [tatiane.cacique@fatec.sp.gov.br].

Keywords: *Mobile system, Pomodoro, Productivity, React Native, Time management, Timer.*

1 Introdução

O presente trabalho tem como tema o desenvolvimento de um sistema integrado de gerenciamento de tempo baseado na Técnica Pomodoro, denominado FocusTime, que combina princípios de produtividade com tecnologia móvel para auxiliar usuários na organização de tarefas e otimização do tempo. Em um contexto marcado pela crescente demanda por eficiência e pela proliferação de distrações digitais, a gestão eficaz do tempo tornou-se uma competência essencial tanto no âmbito acadêmico quanto profissional.

A problemática central que este trabalho busca abordar refere-se à dificuldade contemporânea em manter o foco e a disciplina no gerenciamento do tempo de estudo e trabalho. Conforme evidências empíricas e estudos na área de produtividade, indivíduos enfrentam desafios significativos para:

- Manter a concentração em atividades que exigem atenção sustentada
- Organizar tarefas de forma prioritária e sistemática
- Resistir às constantes interrupções de dispositivos digitais e redes sociais
- Mensurar e analisar seu próprio desempenho produtivo

Esta problemática se manifesta especialmente entre estudantes universitários e profissionais que necessitam gerenciar múltiplas tarefas simultaneamente em ambientes com altos níveis de estímulos competitivos.

Parte-se das seguintes hipóteses para o desenvolvimento do trabalho:

1. A implementação da Técnica Pomodoro através de uma aplicação móvel pode aumentar significativamente a produtividade dos usuários
2. A combinação de timer Pomodoro com sistema de gestão de tarefas proporciona benefícios superiores ao uso isolado de cada ferramenta
3. Interface minimalista e intuitiva favorece a adoção e manutenção do uso do sistema

4. Feedback visual imediato e relatórios de progresso motivam a continuidade do uso

Em sequência, desenvolver um sistema integrado de gerenciamento de tempo, baseado na Técnica Pomodoro, que auxilie usuários na organização de tarefas, manutenção do foco e análise de produtividade através de uma aplicação móvel multi-plataforma.

Objetivos Específicos:

- Projetar e implementar um timer Pomodoro com ciclos personalizáveis de foco e pausa
- Desenvolver um sistema de gestão de tarefas com priorização e categorização
- Implementar mecanismos de registro e análise de estatísticas de produtividade
- Criar uma interface intuitiva e responsiva para dispositivos móveis
- Validar a usabilidade e eficácia do sistema através de testes com usuários
- Documentar todo o processo de desenvolvimento seguindo padrões de engenharia de software

A relevância deste trabalho fundamenta-se em múltiplas dimensões:

- Justificativa Técnica: A integração da Técnica Pomodoro com tecnologias móveis modernas representa uma evolução natural dos métodos tradicionais de gestão de tempo, permitindo automação, personalização e análise de dados em tempo real.
- Justificativa Social: Em uma sociedade cada mais conectada e sujeita a distrações, ferramentas que auxiliam na manutenção do foco contribuem para a saúde mental e o bem-estar dos indivíduos, reduzindo o estresse associado à sobrecarga de informações.
- Justificativa Acadêmica: O trabalho contribui para a área de Engenharia de Software ao demonstrar a aplicação prática de metodologias de desenvolvimento, padrões de projeto e ferramentas modernas no contexto de aplicações de produtividade.

- Justificativa Prática: A solução desenvolvida atende a uma necessidade concreta do mercado por ferramentas de produtividade que sejam ao mesmo tempo eficazes e de fácil adoção.

Por fim, o desenvolvimento do FocusTime seguiu uma abordagem metodológica mista, combinando:

- Pesquisa Aplicada: Com foco na solução de problemas concretos através do desenvolvimento de um produto tecnológico funcional.
- Desenvolvimento Ágil: Utilizando princípios de metodologias ágeis com ciclos iterativos e incrementais, permitindo adaptação a mudanças e entrega contínua de valor.
- Modelagem UML: Aplicação de diagramas padronizados (Casos de Uso, Classes, Sequência, Atividades, Estados) para documentação e planejamento arquitetural.
- Prototipagem Evolucionária: Desenvolvimento progressivo de protótipos funcionais que evoluíram até se tornarem o produto final.
- Testes de Usabilidade: Validação da interface e experiência do usuário através de testes com usuários reais.

1.1 Termo da Abertura do Projeto (TAP)

De acordo com PMI (2021), o Termo de Abertura do Projeto (TAP) é "o documento que formalmente autoriza a existência de um projeto e fornece ao gerente do projeto a autoridade para aplicar os recursos organizacionais às atividades do projeto". É o documento fundacional que estabelece as bases para o planejamento e execução do projeto.

Para KERZNER (2017), o TAP é crucial porque "define o escopo, objetivos, participantes e estrutura geral do projeto, servindo como referência para todas as decisões futuras". Sua importância reside em:

- Autorização formal: Estabelece a existência oficial do projeto
- Alinhamento estratégico: Garante que o projeto esteja alinhado com os objetivos organizacionais
- Definição de autoridade: Designa formalmente o gerente do projeto e sua autoridade
- Base para planejamento: Serve como fundamento para todos os planos subsequentes

Conforme SOMMERVILLE (2011), em projetos de software, o TAP é especialmente importante para "estabelecer limites claros, evitar expansão descontrolada de escopo e garantir que todos os stakeholders tenham expectativas alinhadas desde o início".

1.1.1 Situação atual

De acordo com pesquisa realizada com 35 stakeholders, identificou-se que 65,7% dos entrevistados relatam dificuldade frequente ou constante em manter o foco, enquanto 71,4% apontam redes sociais como principal fator de distração. Atualmente, 74,3% não utilizam nenhuma ferramenta específica de gestão de tempo, dependendo de métodos manuais como listas em papel, notas no celular ou organização mental.

A procrastinação foi identificada como problema significativo, com 68,6% dos respondentes admitindo adiar tarefas regularmente. A ausência de um sistema estruturado dificulta o acompanhamento do progresso e a manutenção da disciplina no gerenciamento do tempo.

1.1.2 Justificativa

A implementação do FocusTime justifica-se pela necessidade urgente de fornecer uma solução tecnológica que auxilie usuários na superação dos desafios de produtividade identificados. O sistema baseia-se na Técnica Pomodoro, metodologia cientificamente validada para melhoria do foco e gestão do tempo.

A viabilidade técnica é sustentada pela expertise da equipe em React Native, Node.js e tecnologias modernas de desenvolvimento. A relevância social é

evidenciada pelo crescente interesse em aplicativos de produtividade e bem-estar digital.

1.1.3 Propósito e Metas

O projeto visa desenvolver um sistema integrado de gerenciamento de tempo que combine timer Pomodoro, gestão de tarefas e analytics de produtividade em uma aplicação móvel multiplataforma.

- Implementar timer Pomodoro com personalização de ciclos (77,1% dos usuários desejam)
- Desenvolver sistema de tarefas com categorização (77,1% consideram importante)
- Criar dashboard de estatísticas de desempenho (60% valorizam)
- Garantir interface minimalista e intuitiva (74,3% preferem)
- Entregar versão funcional até Novembro 2025

1.1.4 Descrição

1.1.4.1 Missão, Visão e Valores

- Descrição: Define a razão de ser da empresa, seus objetivos de longo prazo e os princípios que orientam suas ações.
- Como é criado: Com base em entrevistas com o administrador da empresa para garantir alinhamento com a organização.
- Ferramenta: Microsoft Word.
- Importância: Guia a definição estratégica do projeto e garante que ele esteja alinhado com a cultura e objetivos da empresa.

1.1.4.2 Elicitação de Requisitos

- Descrição: Processo de levantamento das necessidades do cliente e funcionalidades desejadas para o sistema.
- Como é criado: Por meio de entrevistas, questionários (Google Forms).
- Ferramentas: Microsoft Word para consolidar a informação e Google Forms para coletar respostas.

- Importância: Garante que todas as expectativas e necessidades sejam compreendidas e documentadas.

1.1.4.3 Análise SWOT

- Descrição: Identificação de forças, fraquezas, oportunidades e ameaças que influenciam o projeto.
- Como é criada: Baseada em discussões e análise da situação da empresa e do mercado.
- Ferramenta: Canva, para criar uma apresentação visual clara.
- Importância: Auxilia na definição de estratégias para potencializar pontos fortes e mitigar riscos.

1.1.4.4 5W2H

- Descrição: Ferramenta para planejar ações com base nas perguntas: O quê, Por quê, Onde, Quando, Quem, Como e Quanto.
- Como é criado: Com base nas definições de atividades, responsáveis e recursos.
- Ferramenta: Microsoft Word.
- Importância: Garante clareza e organização no plano de ação.

1.1.4.5 Business Process Modeling Notation (BPMN)

- Descrição: Diagrama que modela os processos de negócios, mapeando o fluxo das atividades e interações entre diferentes setores ou sistemas.
- Como é criado: A partir da análise do fluxo atual de trabalho e identificação de melhorias nos processos.
- Ferramenta: Bpmn.io.
- Importância: Facilita a compreensão e otimização dos processos internos, garantindo eficiência e clareza no fluxo de trabalho.

1.1.4.6 Documentação de Requisitos

- Descrição: Documento que formaliza todos os requisitos do sistema, organizando as funcionalidades e restrições.
- Como é criado: A partir das informações coletadas na etapa de elicitação.
- Ferramenta: Microsoft Word.

- Importância: Serve como referência para desenvolvimento e validação do sistema.

1.1.4.7 Estrutura Analítica do Projeto (EAP)

- Descrição: Estrutura hierárquica que divide o projeto em tarefas menores e mais gerenciáveis.
- Como é criada: Com base no planejamento das atividades do projeto.
- Ferramenta: Miro.
- Importância: Facilita o gerenciamento e acompanhamento do progresso do projeto.

1.1.4.8 Termo de Abertura do Projeto (TAP)

- Descrição: Documento formal que autoriza o início do projeto e define seus objetivos e escopo.
- Como é criado: Com a colaboração do cliente e equipe de projeto para garantir alinhamento.
- Ferramenta: Microsoft Word.
- Importância: Estabelece as bases e responsabilidades para o sucesso do projeto.

1.1.4.9 Diagrama de Caso de Uso

- Descrição: Representação gráfica das interações entre os atores (usuários) e o sistema.
- Como é criado: Baseado nos requisitos e funcionalidades levantadas e no BPMN.
- Ferramenta: Astah.
- Importância: Auxilia na compreensão de como o sistema será utilizado e suas principais funcionalidades.

1.1.4.10 Documentação de Caso de Uso

- Descrição: Documento detalhado que descreve cada caso de uso, incluindo fluxos e exceções.
- Como é criado: A partir do diagrama de caso de uso e discussões com a equipe.
- Ferramenta: Microsoft Word.

- Importância: Garante clareza no desenvolvimento e testes do sistema, alinhando expectativas entre as partes envolvidas.

1.1.4.11 Diagrama de Classes

- Descrição: Representa a estrutura do sistema, com as classes, atributos, métodos e relações entre elas.
- Como é criado: A partir da análise dos requisitos para definir os componentes essenciais do sistema.
- Ferramenta: Draw.io.
- Importância: Garante uma visão clara da arquitetura do sistema, facilitando o desenvolvimento orientado a objetos.

1.1.4.12 Diagrama de Atividade

- Descrição: Mapeia o fluxo de atividades realizadas no sistema, demonstrando o caminho de execução de processos.
- Como é criado: A partir do mapeamento dos processos no BPMN.
- Ferramenta: Astah.
- Importância: Auxilia na identificação de pontos críticos e otimização dos processos.

1.1.4.13 Diagrama de Máquina de Estado

- Descrição: Representa os estados possíveis de um objeto ou sistema e as transições entre eles.
- Como é criado: A partir do mapeamento dos processos no BPMN.
- Ferramenta: Astah.
- Importância: Fundamental para entender o comportamento dinâmico do sistema.

1.1.4.14 Diagrama de Sequência

- Descrição: Mostra a interação entre diferentes objetos ou componentes ao longo do tempo.
- Como é criado: A partir do mapeamento dos processos no BPMN.
- Ferramenta: Astah.

- Importância: Garante uma visão clara do fluxo de mensagens e interações no sistema.

1.1.4.15 Documentação de Portabilidade

- Descrição: Documento que descreve como o sistema pode ser transferido e implementado em diferentes ambientes.
- Como é criado: A partir de testes em múltiplos dispositivos e plataformas.
- Ferramenta: Microsoft Word.
- Importância: Garante que o sistema funcione adequadamente em diferentes contextos tecnológicos.

1.1.4.16 Métricas

- Descrição: Indicadores quantitativos usados para avaliar o desempenho do sistema e do projeto.
- Como é criado: Calculando com base em um diagrama de classe de acordo com o projeto.
- Ferramenta: Excel.
- Importância: Permite indicar valores quantitativos para realizar o projeto.

1.1.4.17 Proposta Comercial

- Descrição: Documento formal que detalha a oferta de produtos ou serviços, incluindo escopo, prazos e valores.
- Como é criado: Com base na análise dos requisitos e expectativas do cliente.
- Ferramenta: Microsoft Word.
- Importância: Serve como base para alinhamento com o cliente e formalização do contrato.

1.1.5 Premissas

Sistema intuitivo e de fácil uso: O software deve ser desenvolvido com uma interface simples e intuitiva, visando facilitar a adoção por todos os usuários.

- Sistema intuitivo e de fácil uso: O software deve ser desenvolvido com uma interface simples e intuitiva, visando facilitar a adoção por todos os usuários, tanto clientes quanto o administrador.
- Organização das tarefas: O desenvolvimento e a implementação do sistema devem garantir as demandas do dia a dia, garantindo que todas sejam finalizadas conforme planejado.
- Foco na fácil implantação e manutenção: O sistema deve ser projetado para facilitar a sua implantação e futuras manutenções, minimizando a necessidade de suporte técnico contínuo e agilizando correções ou atualizações.
- Validação contínua com as partes interessadas: Todos os artefatos e projeto desenvolvidos devem ser validados regularmente com os integrantes do projeto, para garantir que estejam alinhados com as expectativas.
- Revisão periódica da documentação: A documentação do projeto deve ser revisada constantemente para garantir seu aprimoramento, clareza e utilidade, facilitando a compreensão e futura referência.
- Segurança e integridade dos dados: Deve-se garantir que o sistema atenda aos requisitos de segurança necessários para proteger os dados e informações dos usuários.

1.1.6 Restrições

- Capacitação técnica da equipe: Todos os membros da equipe do projeto são alunos e estão em fase de aprendizado, o que pode exigir novos conhecimentos adicionais para garantir a realização do projeto.

- Tempo disponível dos membros da equipe: Os membros da equipe, possuem limitações de disponibilidade, o que impacta o cronograma de desenvolvimento e revisão.

1.1.7 Stakeholders

Os *stakeholders* são as partes interessadas que desempenham papéis essenciais na implementação e no sucesso do sistema. No contexto deste projeto, destacam-se dois grupos principais de usuários finais:

- Usuários :

Os usuários finais que acessam a plataforma digital para administração e organização de demandas do dia a dia. A implementação do sistema oferece a esses usuários uma experiência mais ágil e prática, eliminando a necessidade de múltiplas plataformas e garantindo maior transparência nas entregas diárias.

Esses *stakeholders* são fundamentais para o sucesso do projeto, pois representam tanto a demanda operacional (administrador) quanto a experiência final do consumidor (clientes), garantindo que o sistema atenda plenamente às expectativas de eficiência e satisfação.

1.1.8 Riscos

- Problemas de conectividade: A qualidade da conexão à internet pode interferir no uso do sistema, especialmente se a rede for instável ou de baixa velocidade.
- Conformidade com a LGPD: O sistema deve garantir a proteção de dados dos clientes, conforme as exigências da Lei Geral de Proteção de Dados (LGPD). O não cumprimento pode resultar em penalidades legais.
- Mudanças inesperadas nos requisitos: Alterações não planejadas durante o projeto podem comprometer prazos.

- Falta de comunicação entre as partes: A má comunicação entre a equipe de desenvolvimento e o administrador pode resultar em falhas no projeto.

1.1.9 Marco

Setembro 2025

- Definição do projeto e levantamento de requisitos
- Pesquisa com stakeholders (35 respondentes)
- Análise SWOT e Canvas de Negócio
- Modelagem BPMN e casos de uso
- Especificação de requisitos funcionais e não funcionais
- Diagramas UML (atividades, sequência, estados)

Outubro 2025

- Desenvolvimento do backend e API
- Implementação do frontend mobile
- Integração entre componentes

Novembro 2025

- Testes de usabilidade e refinamentos
- Documentação final do sistema
- Preparação para apresentação do TG

1.1.10 Responsabilidades

A seguir, estão detalhadas as responsabilidades de cada integrante da equipe, destacando suas contribuições específicas para o desenvolvimento do projeto:

Tatiane Cacique:

- Desenvolvimento do backend (Node.js, Fastify)
- Implementação do frontend (React Native)

- Configuração do banco de dados (PostgreSQL)
- Documentação técnica e relatórios
- Coordenação geral do projeto

Leonardo:

- Modelagem de negócio (Canvas, SWOT, 5W2H)
- Especificação de requisitos funcionais e não funcionais
- Diagramas UML (casos de uso, atividades, sequência, estados)
- Análise de métricas e viabilidade
- Documentação de casos de uso

2 Viabilidade do Projeto

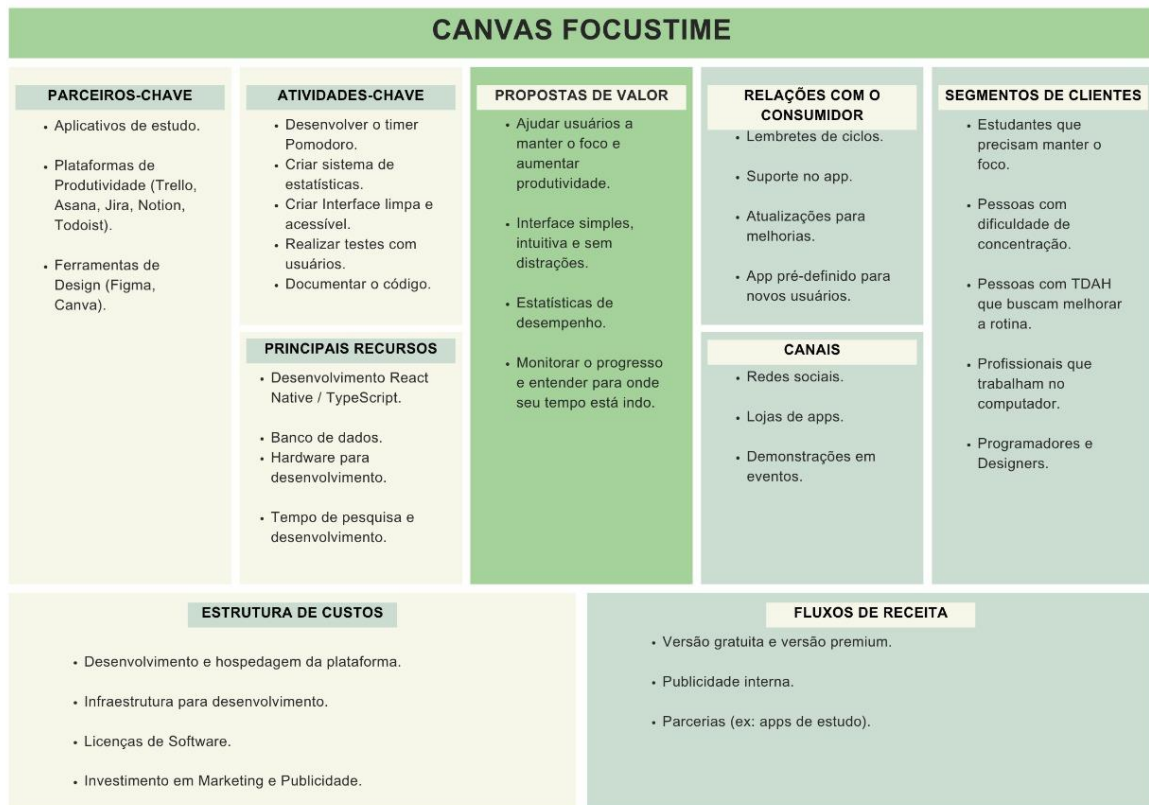
2.1 Canvas de Negócio (Business Model Canvas - BMC)

O Business Model Canvas (BMC) é uma ferramenta estratégica de gestão desenvolvida por OSTERWALDER e PIGNEUR (2010) que permite visualizar, descrever e analisar o modelo de negócio de uma organização em um único diagrama. Segundo os autores, o BMC proporciona uma visão holística e integrada dos principais componentes que constituem um negócio, facilitando o alinhamento estratégico e a identificação de oportunidades de melhoria.

Para BLANK (2013), o BMC é fundamental no processo de desenvolvimento de clientes (Customer Development) por permitir a rápida prototipagem e teste de hipóteses de negócio, reduzindo o risco e o custo associados ao lançamento de novos produtos. A ferramenta é especialmente valiosa em ambientes de incerteza, onde modelos de negócio tradicionais podem não ser adequados.

Conforme MAURYA (2012), em projetos de startups e novos empreendimentos, o BMC funciona como um "mapa da mina" que orienta o desenvolvimento do produto e a descoberta de um modelo de negócio escalável e repetível.

Figura 1 – Canvas de Negócio



Fonte: Os autores

O FocusTime segmenta seu mercado em quatro grupos principais, conforme identificado por KOTLER e KELLER (2016) em estratégias de segmentação:

- Estudantes com dificuldade de foco: Indivíduos em formação acadêmica que necessitam de ferramentas para melhorar a concentração durante estudos
- Pessoas com TDAH: Público com necessidades específicas de estruturação temporal e redução de distrações
- Profissionais de computador: Trabalhadores que passam longos períodos em frente a telas e necessitam gerenciar pausas estratégicas
- Programadores e designers: Profissionais criativos que se beneficiam da metodologia Pomodoro para manter a produtividade em tarefas complexas

A proposta de valor do FocusTime baseia-se no framework de valor proposto por OSTERWALDER et al. (2014), oferecendo:

- Ganho de produtividade: Aumento mensurável na eficiência do uso do tempo
- Redução de dor: Diminuição do estresse relacionado à gestão do tempo e procrastinação
- Simplicidade: Interface minimalista que elimina a complexidade de ferramentas similares
- Insights acionáveis: Estatísticas que permitem ao usuário entender e melhorar seus hábitos

A estratégia de canais segue a abordagem multicanal recomendada por CHAFFEY (2019):

- Canais digitais diretos: Lojas de aplicativos (Apple App Store, Google Play Store)
- Mídias sociais: Divulgação orgânica e paga em plataformas especializadas
- Marketing de eventos: Participação em feiras de tecnologia e produtividade
- Parcerias estratégicas: Integração com ecossistema de produtividade

O modelo de relacionamento implementa a hierarquia de engajamento descrita por HAGEN (2018):

- Onboarding estruturado: Configuração pré-definida para novos usuários
- Suporte proativo: Sistema de notificações e lembretes contextualizados
- Atualizações contínuas: Melhorias baseadas em feedback dos usuários
- Comunidade: Espaços para troca de experiências e melhores práticas

O modelo de receita segue a estratégia freemium analisada por PARKER (2016):

- Versão gratuita: Funcionalidades básicas de timer e gestão de tarefas
- Versão premium: Recursos avançados de analytics, personalização e integração
- Publicidade não intrusiva: Parceiros alinhados com o segmento de produtividade
- Receita de parcerias: Integrações premium com outras plataformas

Os recursos identificados alinham-se com a teoria de recursos e capacidades de BARNEY (1991):

- Recursos tecnológicos: Stack moderna (React Native, TypeScript, Node.js)
- Recursos humanos: Conhecimento em desenvolvimento mobile e UX/UI
- Recursos intelectuais: Metodologia Pomodoro adaptada e algoritmos de analytics
- Recursos físicos: Infraestrutura de desenvolvimento e teste

As atividades mapeadas refletem as operações centrais do modelo de negócio:

- Desenvolvimento ágil: Ciclos contínuos de melhoria do produto
- Pesquisa de UX: Testes de usabilidade e validação com usuários reais
- Análise de dados: Processamento de métricas de uso e desempenho
- Manutenção evolutiva: Atualizações de segurança e performance

As parcerias estratégicas seguem o conceito de ecossistema de negócio de MOORE (1993):

- Parceiros complementares: Apps de estudo que ampliam a proposta de valor
- Parceiros de plataforma: Ferramentas de produtividade que permitem integração
- Parceiros de desenvolvimento: Comunidades open source e ferramentas de design

A estrutura de custos implementa o modelo custo-driven analisado por OSTERWALDER (2010):

- Custos de desenvolvimento: Equipe, ferramentas e infraestrutura técnica
- Custos de hospedagem: Servidores, banco de dados e CDN
- Custos de aquisição: Marketing digital e campanhas de aquisição
- Custos de operação: Suporte, manutenção e atualizações

O BMC desenvolvido para o FocusTime demonstra viabilidade estratégica através da:

- Sustentabilidade financeira: Múltiplas fontes de receita que suportam o crescimento
- Escalabilidade tecnológica: Arquitetura que permite expansão para novos mercados
- Diferenciação competitiva: Proposta de valor clara e difícil de replicar
- Alinhamento com tendências: Mercado crescente de apps de produtividade e bem-estar digital

Conforme RIES (2011), este modelo permite validação rápida das hipóteses de negócio através de um Produto Mínimo Viável (MVP) que testa os pressupostos mais críticos antes de investimentos significativos em desenvolvimento e marketing.

2.2 Matriz SWOT

A Matriz SWOT é uma ferramenta de planejamento estratégico utilizada para avaliar os pontos fortes (Strengths), pontos fracos (Weaknesses), oportunidades (Opportunities) e ameaças (Threats) de um projeto ou organização. Segundo Kotler e Keller (2016), a análise SWOT proporciona uma visão abrangente da situação atual, permitindo que os gestores tomem decisões mais embasadas e desenvolvam estratégias eficazes.

Para projetos de software, a aplicação da SWOT é particularmente valiosa, pois ajuda a:

- Identificar vantagens competitivas do produto
- Antecipar riscos e desafios no desenvolvimento e adoção
- Direcionar recursos para áreas críticas
- Estabelecer estratégias de diferenciação no mercado

Figura 2 - Matriz SWOT



Fonte: Os autores

Na análise SWOT apresentada, duas fraquezas principais foram identificadas:

1. Hábitos frágeis de organização e procrastinação: Conforme Zimmerman (2002), a autorregulação é fundamental para o sucesso de qualquer ferramenta de produtividade. O sistema precisa ser projetado para compensar essas fragilidades humanas, oferecendo lembretes suaves e mecanismos de engajamento progressivo.
2. Diferentes níveis de familiaridade com Pomodoro: Como observado por Cirillo (2018), criador da técnica, a curva de aprendizado pode variar significativamente. A aplicação deve incluir tutoriais integrados e modos de dificuldade progressiva para acomodar tanto iniciantes quanto usuários avançados.
3. Indiferença a relatórios: Segundo O'Brien e Toms (2008), o engajamento com features analíticas depende diretamente da percepção de valor imediato. Os relatórios

do FocusTime precisam ser visualmente atraentes e mostrar insights acionáveis de forma simples.

2.3 Plano de Ação 5W2H do Projeto

O 5W2H é uma ferramenta de gestão e planejamento que permite a estruturação de planos de ação de forma sistemática e organizada. Segundo CARPINETTI (2010), a metodologia deriva de sete questionamentos em inglês que cobrem todos os aspectos essenciais de um projeto:

- What? (O que será feito?) - Define as ações
- Why? (Por que será feito?) - Estabelece a justificativa
- Where? (Onde será feito?) - Determina o local
- When? (Quando será feito?) - Estipula prazos
- Who? (Quem fará?) - Designa responsáveis
- How? (Como será feito?) - Descreve os métodos
- How much? (Quanto custará?) - Calcula os recursos

Para MAXIMIANO (2012), o 5W2H é particularmente valioso em projetos de software por proporcionar clareza na execução, facilitar o acompanhamento do progresso e garantir que todas as dimensões do projeto sejam consideradas.

Quadro 1 - Matriz 5W2H Questão Problema 1

Fraqueza 1: Hábitos frágeis de organização						
What? (O que?)	Why? (Por quê?)	Where? (Onde)	When? (Quando?)	Who? (Quem?)	How? (Como?)	How Much? (Quanto?)
Permitir definir prioridade	Para estimular disciplina e evitar esquecer atividades	Tela de tarefas	Até o fim do semestre	Desenvolvedores	Implementando cadastro de tarefas com níveis de prioridade (baixa/média/alta)	Baixo custo
Exibir histórico de tarefas	Para reforçar progresso e	Aba de histórico	Até o fim do semestre	Desenvolvedores	Registrando tarefas abertas e concluídas	Baixo custo

	hábito de finalização					
--	--------------------------	--	--	--	--	--

Fonte – Os autores

Quadro 2 - Matriz 5W2H Questão Problema 2

Fraqueza 2: Forte tendência à procrastinação						
What? (O que?)	Why? (Por quê?)	Where? (Onde)	When? (Quando?)	Who? (Quem?)	How? (Como?)	How Much? (Quanto?)
Mostrar cronômetro Pomodoro em evidência	Para evitar perda de foco e procrastinação	Tela principal do app	Até o fim do semestre	Desenvolvedores	Exibindo timer destacado no centro da interface	Baixo custo
Permitir ciclos menores para iniciantes	Para reduzir resistência inicial às tarefas	Tela de configurações	Até o fim do semestre	Desenvolvedores	Permitindo alterar tempo de foco/pausa	Baixo custo

Fonte – Os autores

Quadro 3 - Matriz 5W2H Questão Problema 3

Fraqueza 3: Indiferença a relatórios por parte do público						
What? (O que?)	Why? (Por quê?)	Where? (Onde)	When? (Quando?)	Who? (Quem?)	How? (Como?)	How Much? (Quanto?)
Mostrar tempo total focado no dia/semana	Para evitar excesso de dados que afastam o usuário	Aba de histórico	Até o fim do semestre	Desenvolvedor	Implementando relatório visual simples	Baixo custo
Usar mudança de cor na interface para indicar progresso	Transformar o resultado em algo intuitivo e imediato	Tela do timer	Até o fim do semestre	Desenvolvedor	Alterando o visual no timer (vermelho = tempo de foco, verde = tempo de pausa)	Baixo custo

Fonte – Os autores

Quadro 4 - Matriz 5W2H Questão Problema 4

Fraqueza 4: Diferentes níveis de familiaridade com Pomodoro						
What? (O que?)	Why? (Por quê?)	Where? (Onde)	When? (Quando?)	Who? (Quem?)	How? (Como?)	How Much? (Quanto?)
Oferecer tempos prontos (iniciante, padrão, avançado)	Para facilitar uso para novatos	Configuração do timer	Até o fim do semestre	Desenvolvedores	Implementando botões rápidos para selecionar ciclos	Baixo custo
Interface minimalista com foco em usabilidade	Para tornar o app acessível a qualquer usuário	Toda a interface	Até o fim do semestre	Desenvolvedores	Desenvolvendo Interface simples com poucos botões, design clean e intuitivo	Baixo custo

Fonte – Os autores

O plano 5W2H desenvolvido para o FocusTime baseia-se no princípio de design centrado no usuário, conforme proposto por NORMAN (2013), que enfatiza a necessidade de adaptar a tecnologia às limitações e características humanas.

As ações propostas focam em compensar comportamentos problemáticos identificados na matriz SWOT através de:

1. Estruturação visual clara - Interface que guia o usuário naturalmente
2. Personalização adaptativa - Diferentes níveis de complexidade
3. Controle do usuário - Opções configuráveis para evitar rejeição
4. Feedback imediato - Mostrar progresso e manter engajamento

Conforme PRESSMAN (2011), em engenharia de software, a antecipação de problemas de usabilidade durante a fase de planejamento reduz significativamente os custos de retrabalho. O plano 5W2H elaborado demonstra:

- Viabilidade técnica: Todas as ações são implementáveis com as tecnologias escolhidas (React Native, Node.js)
- Viabilidade econômica: Custos baixos devido ao uso de tecnologias open-source
- Viabilidade temporal: Prazos realistas dentro do semestre letivo
- Alinhamento estratégico: Soluções diretamente conectadas às fraquezas identificadas

A implementação deste plano garantirá que o FocusTime supere as principais barreiras de adoção, tornando-se uma ferramenta acessível tanto para usuários iniciantes quanto experientes em técnicas de produtividade.

3 Levantamento de Requisitos

Para o levantamento de requisitos do sistema FocusTime, foram empregadas duas técnicas principais de elicitação conforme recomendado por SOMMERVILLE (2011):

1. Questionários Estruturados: Formulário online distribuído para uma amostra de 35 respondentes
2. Análise de Perfil de Usuários: Caracterização demográfica e comportamental do público-alvo

De acordo com PRESSMAN (2011), a elicitação de requisitos é "a base sobre a qual todo o sistema é construído". A importância deste processo fundamenta-se em:

- Redução de retrabalho: Identificação correta das necessidades evita mudanças custosas durante o desenvolvimento
- Alinhamento de expectativas: Garantia de que o sistema atenderá às reais necessidades dos usuários
- Validação de escopo: Definição clara do que será desenvolvido e do que ficará de fora

Para WIEGERS & BEATTY (2013), a elicitação eficaz de requisitos é crucial para o sucesso do projeto, pois "requisitos incompletos ou incorretos são a principal causa de falhas em projetos de software".

Conforme LEFFINGWELL & WIDRIG (2003), a aplicação de técnicas sistemáticas de elicitação permite capturar tanto os requisitos explícitos quanto as necessidades implícitas dos stakeholders.

3.1 Perfil da Amostra

O levantamento foi realizado através de formulário online aplicado a uma turma do 6º período noturno, obtendo 35 respostas válidas. A caracterização demográfica revelou:

- Faixa etária predominante: 25-34 anos (45,7%) e 18-24 anos (34,3%)
- Perfil ocupacional misto: 51,4% exercem atividades de estudo e trabalho simultaneamente
- Experiência com apps de produtividade: 74,3% não utilizam ferramentas similares atualmente

Quadro 5 - Principais necessidades identificadas no levantamento

Categoria	Necessidade	Frequência	Impacto
Problema Central	Dificuldade em manter o foco	65,7% (sempre/frequentemente)	ALTO
Fatores de Distração	Redes sociais	71,4%	ALTO
Comportamento	Procrastinação	68,6% (relatos qualitativos)	ALTO
Necessidade Estrutural	Pausas regulares	73,4%	MÉDIO
Preferência de Design	Interface minimalista	74,3%	ALTO
Funcionalidade Core	Personalização de ciclos	77,1%	ALTO

Fonte – Os autores

Com base nas respostas obtidas, foram elaboradas as seguintes histórias de usuário seguindo o formato Conway (2009):

HU001 - Timer Pomodoro Personalizável

Como usuário com dificuldade de foco

Quero personalizar a duração dos ciclos de foco e pausa

Para adaptar o método à minha capacidade de concentração

Critérios de Aceitação:

- Definir tempo de foco entre 15-60 minutos
- Definir tempo de pausa entre 5-30 minutos
- Configurar ciclos até pausa longa (1-10 ciclos)
- Salvar configurações por perfil

HU002 - Sistema de Tarefas com Categorização

Como usuário que precisa organizar múltiplas atividades

Quero criar categorias para minhas tarefas

Para separar responsabilidades de estudo, trabalho e pessoal

Critérios de Aceitação:

- Criar categorias customizadas
- Associar tarefas a categorias
- Filtrar tarefas por categoria
- Visualizar por contexto de uso

HU003 - Estatísticas de Desempenho

Como usuário que quer melhorar minha produtividade

Quero visualizar estatísticas do meu desempenho

Para identificar padrões e oportunidades de melhoria

Critérios de Aceitação:

- Tempo total focado por dia/semana/mês
- Número de pomodoros completados
- Tarefas concluídas
- Comparativo com períodos anteriores

HU004 - Interface Minimalista

Como usuário que se distrai facilmente

Quero uma interface limpa e sem elementos desnecessários

Para manter o foco na tarefa atual

Critérios de Aceitação:

- Design limpo com poucas cores
- Navegação intuitiva com máximo 3 cliques
- Sem anúncios intrusivos
- Modo foco que esconde elementos secundários

HU005 - Gestão de Distrações

Como usuário vulnerável a redes sociais

Quero receber lembretes para evitar distrações

Para manter a disciplina durante os ciclos de foco

Critérios de Aceitação:

- Notificações de início/fim de ciclo
- Lembretes para não checar redes sociais
- Alertas visuais discretos
- Opção de silenciar notificações

Problemas e Oportunidades Identificados

Problemas Críticos:

1. Procrastinação generalizada: 68,6% dos respondentes admitem adiar tarefas
2. Dificuldade de foco: 65,7% relatam problemas frequentes de concentração
3. Distração por redes sociais: 71,4% identificam como principal fator de interrupção
4. Falta de método estruturado: 74,3% não utilizam ferramentas de gestão de tempo

Oportunidades de Design:

1. Personalização: 77,1% valorizam capacidade de ajustar ciclos
2. Simplicidade: 74,3% preferem interface minimalista
3. Multiplataforma: 54,3% desejam usar em celular e computador
4. Feedback visual: 60% consideram importante ver estatísticas

As necessidades identificadas foram traduzidas em requisitos específicos:

- Timer Pomodoro: Responde à necessidade de 73,4% por pausas regulares
- Personalização de Ciclos: Atende 77,1% que desejam ajustar durações
- Gestão de Tarefas com Categorias: Atende 77,1% que querem organização contextual
- Estatísticas de Desempenho: Atende 60% que valorizam métricas de progresso
- Usabilidade da Interface: Responde a 74,3% que preferem simplicidade

Os dados coletados fornecem evidências quantitativas para as decisões de projeto:

- Personalização justificada: 77,1% dos usuários desejam ajustar ciclos
- Foco em mobile first: 54,3% usariam em ambos, mas 47,7% preferem celular
- Minimalismo validado: 74,3% preferem simplicidade contra 2,9% que querem muitas funcionalidades

Conforme COHEN (2004), esta abordagem baseada em dados reais reduz o risco de desenvolver funcionalidades que não serão utilizadas ou valorizadas pelos usuários finais.

3.2 BPMN

3.2.1 Análise do Plano de Ação

BPMN (Business Process Model and Notation) é uma notação gráfica padronizada para modelagem de processos de negócio. Segundo OMG (2011), a BPMN fornece uma notação que é facilmente compreensível por todos os usuários de negócio, desde os analistas que criam os rascunhos iniciais dos processos, até os desenvolvedores técnicos responsáveis pela implementação da tecnologia que executará esses processos.

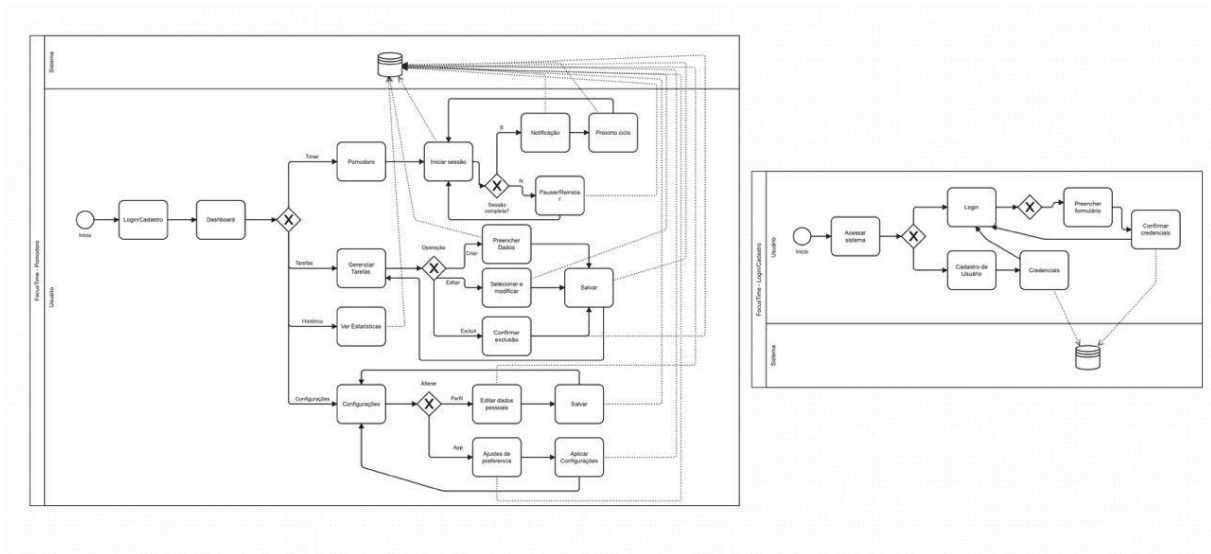
Conforme WHITE (2004), um dos criadores da especificação BPMN, esta notação foi desenvolvida para superar as limitações de outras notações de fluxogramas, proporcionando um conjunto padronizado de símbolos e regras que permitem representar processos complexos de maneira clara e unificada.

De acordo DUMAS et al. (2018), a BPMN serve principalmente para:

1. Documentação de processos: Registrar de forma padronizada como as atividades são executadas
2. Análise e melhoria: Identificar gargalos, redundâncias e oportunidades de otimização
3. Comunicação: Facilitar o entendimento comum entre stakeholders técnicos e de negócio
4. Automação: Servir como base para implementação de sistemas de workflow

Para REISER (2020), em projetos de software, o BPMN é fundamental para compreender o fluxo de atividades do sistema antes da implementação, reduzindo retrabalho e garantindo que todos os cenários sejam considerados.

Figura 3 – BPMN



Fonte: Os autores

A modelagem BPMN é crucial para o sucesso do FocusTime por várias razões, conforme fundamentado por WESKE (2019):

1. Clareza no fluxo de sessões Pomodoro

O diagrama permite visualizar todo o ciclo de uma sessão, desde a seleção da tarefa até a geração de relatórios, garantindo que nenhuma etapa essencial seja negligenciada no desenvolvimento.

2. Identificação de pontos de integração

Conforme mostra a FIGURA 3, o BPMN evidencia onde ocorrem as interações entre o usuário e o sistema, como nas notificações de término e atualizações de estatísticas.

3. Base para especificação de requisitos

O processo modelado serve como referência para derivar casos de uso e histórias de usuário, alinhando expectativas entre desenvolvedores e stakeholders.

4. Otimização da experiência do usuário

Ao mapear o processo completo, é possível identificar oportunidades de melhoria, como a automatização de transições entre sessões de foco e pausas.

5. Validação com usuários

O diagrama BPMN funciona como uma ferramenta de comunicação não técnica que permite validar o fluxo proposto com potenciais usuários antes da implementação.

Segundo JABLONSKI (2016), em sistemas de produtividade como o FocusTime, a modelagem de processos é especialmente importante pois reflete diretamente na eficácia da metodologia Pomodoro implementada, garantindo que o sistema suporte adequadamente a técnica e seus princípios fundamentais.

3.3 Requisitos Funcionais

De acordo com SOMMERVILLE (2011), requisitos funcionais definem as funcionalidades que o sistema deve fornecer, descrevendo o que o software deve fazer e como deve reagir a entradas específicas. Estes requisitos especificam o comportamento externo do sistema em termos de tarefas e serviços que oferece aos usuários.

Para PRESSMAN (2011), os requisitos funcionais são declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como se comportar em determinadas situações. Eles estabelecem capacidades, características e funcionalidades que os stakeholders esperam do sistema.

Conforme IEEE 830-1998, os requisitos funcionais devem ser declarados de forma clara, consistente e completa, permitindo que desenvolvedores, testadores e usuários tenham um entendimento comum sobre o que será construído.

Quadro 6 - Requisito funcionais

ID RF001	Nome do Requisito: Cadastrar Usuário
Categoria.	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá permitir que o usuário se cadastre informando nome, e-mail e senha.
Informações	Nome, e-mail e senha do usuário.
Regra de Negócio	Não há nenhuma regra de negócio.
ID RF002	Nome do Requisito: Realizar Login
Categoria.	Evidente.

Prioridades	Essencial.
Descrição	O sistema deverá permitir que o usuário acesse sua conta utilizando e-mail e senha.
Informações	E-mail e senha válidos previamente cadastrados.
Regra de Negócio	Não há nenhuma regra de negócio.
ID RF003	Nome do Requisito: Alterar Dados do Usuário
Categoria.	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá permitir que o usuário edite seus dados cadastrados, como nome, senha ou e-mail.
Informações	Nome, e-mail e senha atualizados.
Regra de Negócio	O usuário deve estar autenticado para alterar seus dados.
ID RF004	Nome do Requisito: Cadastrar Tarefas
Categoria.	Evidente.
Prioridades	Essencial.
Descrição	O usuário deverá poder criar novas tarefas informando nome, categoria e nível de prioridade (baixa, média ou alta).
Informações	Nome da tarefa, categoria e nível de prioridade.
Regra de Negócio	Não há nenhuma regra de negócio.
ID RF005	Nome do Requisito: Editar e Excluir Tarefas
Categoria.	Evidente.

Prioridades	Essencial.
Descrição	O sistema deverá permitir editar ou excluir tarefas criadas anteriormente.
Informações	Nome da tarefa, descrição e status.
Regra de Negócio	Somente o usuário que criou a tarefa poderá editá-la ou excluí-la.
ID RF006	Nome do Requisito: Marcar tarefa como concluída
Categoria.	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá permitir que o usuário marque uma tarefa como concluída.
Informações	Nome da tarefa e status (aberta/concluída).
Regra de Negócio	Tarefas concluídas não podem ser reeditadas, apenas visualizadas.
ID RF007	Nome do Requisito: Visualizar Histórico de Tarefas
Categoria.	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá exibir uma aba com as tarefas concluídas e abertas, incluindo o tempo total focado por dia ou semana.
Informações	Lista de tarefas abertas e concluídas, tempo total focado.
Regra de Negócio	Não há nenhuma regra de negócio.
ID RF008	Nome do Requisito: Timer Pomodoro
Categoria.	Evidente.

Prioridades	Essencial.
Descrição	O sistema deverá exibir um cronômetro central de foco e pausa, destacando o tempo restante de cada ciclo.
Informações	Tempo de foco, tempo de pausa e tempo restante.
Regra de Negócio	Não há nenhuma regra de negócio.
ID RF009	Nome do Requisito: Personalizar Ciclos
Categoria.	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá permitir que o usuário configure os tempos de foco e pausa.
Informações	Duração do foco e pausa por perfil.
Regra de Negócio	Não há nenhuma regra de negócio.
ID RF010	Nome do Requisito: Visualizar Interface de Progresso
Categoria.	Evidente.
Prioridades	Essencial.
Descrição	O aplicativo deverá mudar de cor conforme o ciclo (vermelho para foco, verde para pausa).
Informações	Informações: Estado atual do ciclo e esquema de cores.
Regra de Negócio	Não há nenhuma regra de negócio.
ID RF011	Nome do Requisito: Visualizar Estatísticas de Desempenho
Categoria.	Evidente.
Prioridades	Essencial.

Descrição	O sistema deverá apresentar gráficos ou dados simples sobre o tempo de foco acumulado.
Informações	Tempo total de foco diário, semanal e mensal.
Regra de Negócio	Não há nenhuma regra de negócio.

Fonte: Os autores.

Os requisitos funcionais documentados para o FocusTime seguem as melhores práticas de engenharia de requisitos estabelecidas por WIEGERS & BEATTY (2013), organizando-se em três categorias principais:

1. Requisitos de Gerenciamento de Usuário (RF001-RF003)

Estes requisitos tratam do ciclo de vida do usuário no sistema, desde o cadastro inicial até a manutenção dos dados. Conforme LEFFINGWELL & WIDRIG (2003), esta categoria é fundamental para sistemas que exigem autenticação e personalização.

2. Requisitos de Gerenciamento de Tarefas (RF004-RF007)

Esta categoria engloba o núcleo do sistema de produtividade, permitindo ao usuário criar, organizar e acompanhar suas atividades. A regra de negócio do RF005, que restringe a edição apenas ao criador da tarefa, demonstra uma preocupação com a integridade dos dados, conforme recomendado por ROBERTSON & ROBERTSON (2012).

3. Requisitos da Técnica Pomodoro (RF008-RF011)

Estes requisitos implementam a metodologia Pomodoro propriamente dita, incluindo o timer, personalização de ciclos e feedback visual. O RF010, que especifica mudanças de cor na interface, segue os princípios de design comportamental descritos por NORMAN (2013), onde elementos visuais guiam o comportamento do usuário.

Conforme IEEE 29148-2011, os requisitos funcionais devem estar alinhados com os objetivos de negócio do sistema. No caso do FocusTime, observa-se que:

- RF001-RF003 suportam o objetivo de personalização identificado na matriz SWOT

- RF004-RF007 atendem à necessidade de organização das tarefas
- RF008-RF011 implementam a técnica Pomodoro de forma adaptável

A priorização Essencial para todos os requisitos, conforme CARLSHAMRE & REGNELL (2000), é adequada para um MVP (Minimum Viable Product) que deve entregar valor imediato ao usuário final.

3.4 Requisitos Não Funcionais

De acordo com SOMMERVILLE (2011), requisitos não funcionais definem as propriedades e restrições do sistema, especificando critérios de qualidade que o software deve atender, em vez de comportamentos específicos. Estes requisitos estabelecem como o sistema deve executar suas funções, abrangendo aspectos como desempenho, segurança, usabilidade e confiabilidade.

Para PRESSMAN (2011), os requisitos não funcionais descrevem atributos do sistema ou do ambiente no qual o sistema opera, incluindo restrições de tempo, espaço, orçamento e tecnologia. Eles especificam critérios que podem ser usados para julgar a operação de um sistema, em vez de comportamentos específicos.

Conforme IEEE 830-1998, os requisitos não funcionais são frequentemente classificados em categorias como:

- Requisitos de produto: Desempenho, usabilidade, confiabilidade
- Requisitos organizacionais: Conformidade com padrões, regulamentações
- Requisitos externos: Interoperabilidade, restrições legais

Quadro 7 - Requisitos Não Funcionais

ID RNF001	Nome do Requisito: Usabilidade da Interface
Categoria	Usabilidade.
Prioridades	Essencial.
Descrição	A interface do sistema deverá ser minimalista, limpa e intuitiva, facilitando o uso por qualquer nível de usuário.

Informações	Layout, ícones e elementos visuais de fácil compreensão
Regra de Negócio	Não há regra de negócio.
ID RNF002	Nome do Requisito: Tempo de Resposta
Categoria	Desempenho.
Prioridades	Essencial.
Descrição	O sistema deverá executar as funções principais (timer, cadastro e histórico) sem atrasos perceptíveis, garantindo fluidez e agilidade nas ações.
Informações	Processamento de tarefas e carregamento de páginas.
Regra de Negócio	Não há regra de negócio.
ID RNF003	Nome do Requisito: Criptografia
Categoria	Segurança.
Prioridades	Essencial.
Descrição	As senhas dos usuários deverão ser armazenadas de forma criptografada para garantir a proteção de dados e evitar acessos indevidos.
Informações	Senhas e dados sensíveis.
Regra de Negócio	Não há regra de negócio.
ID RNF004	Nome do Requisito: Fácil Manutenção
Categoria	Manutenibilidade.
Prioridades	Essencial.
Descrição	O código do sistema deverá permitir mudanças e melhorias futuras sem dificuldade, utilizando boas práticas de desenvolvimento.
Informações	Estrutura modular e documentação de código.
Regra de Negócio	Não há regra de negócio.
ID RNF005	Nome do Requisito: Banco de Dados
Categoria	Segurança.
Prioridades	Essencial.

Descrição	Os dados do sistema deverão ser armazenados em banco de dados de forma segura, garantindo integridade e disponibilidade.
Informações	Tabelas de usuários, tarefas e histórico.
Regra de Negócio	Não há regra de negócio.

Fonte: Os autores

Os requisitos não funcionais documentados para o FocusTime seguem a taxonomia proposta por CHUNG et al. (2012), organizando-se em categorias que garantem a qualidade do sistema:

1. Requisitos de Usabilidade (RNF001)

Este requisito atende diretamente às necessidades identificadas na matriz SWOT, onde a simplicidade e minimalismo foram destacados como forças do projeto. Conforme NIELSEN (2012), sistemas de produtividade devem priorizar usabilidade para reduzir a curva de aprendizado e aumentar a adoção. A interface intuitiva é crucial considerando os diferentes níveis de familiaridade dos usuários com a técnica Pomodoro.

2. Requisitos de Desempenho (RNF002)

Para sistemas de timer como o FocusTime, o desempenho é crítico. Segundo SMITH (2015), atrasos perceptíveis em aplicações de produtividade podem levar à frustração do usuário e abandono do sistema. A fluidez nas operações principais (timer, cadastro, histórico) é essencial para manter o engajamento durante as sessões de foco.

3. Requisitos de Segurança (RNF003 e RNF005)

Estes requisitos demonstram conformidade com a LGPD (Lei Geral de Proteção de Dados), conforme estabelecido por BRASIL (2018). A criptografia de senhas e o armazenamento seguro em banco de dados são práticas fundamentais para proteger informações sensíveis dos usuários. Para WHITMAN & MATFORD (2016), mesmo em aplicações aparentemente simples, a segurança de dados deve ser tratada como prioridade.

4. Requisitos de Manutenibilidade (RNF004)

Este requisito reflete as melhores práticas de engenharia de software defendidas por MARTIN (2008) em "Código Limpo". A estrutura modular e a documentação adequada permitirão a evolução do sistema, atendendo à oportunidade identificada na SWOT de expandir funcionalidades futuramente, como gamificação e relatórios avançados.

Conforme ISO/IEC 25010:2011, os requisitos não funcionais do FocusTime cobrem dimensões essenciais de qualidade de software:

- Usabilidade (RNF001): Satisfaz o critério de "facilidade de uso"
- Desempenho (RNF002): Atende à "eficácia de tempo"
- Segurança (RNF003, RNF005): Garante "confidencialidade" e "integridade"
- Manutenibilidade (RNF004): Assegura "modificabilidade"

A priorização Essencial para todos os requisitos não funcionais, seguindo CARLSHAMRE & REGNELL (2000), é apropriada para um sistema que precisa entregar experiência confiável desde o primeiro uso, especialmente considerando o público-alvo que pode ter baixa tolerância a falhas técnicas.

3.5 Casos de Uso

De acordo com COCKBURN (2001), um caso de uso é uma técnica para capturar requisitos funcionais do sistema que descreve as interações entre os atores e o sistema para alcançar um objetivo específico. É uma sequência de ações que o sistema realiza para produzir um resultado observável de valor para um ator particular.

Para LARMAN (2004), casos de uso são histórias de como os atores usam o sistema para atingir suas metas. Eles descrevem o comportamento do sistema sob várias condições à medida que o sistema responde a uma solicitação de um dos seus atores.

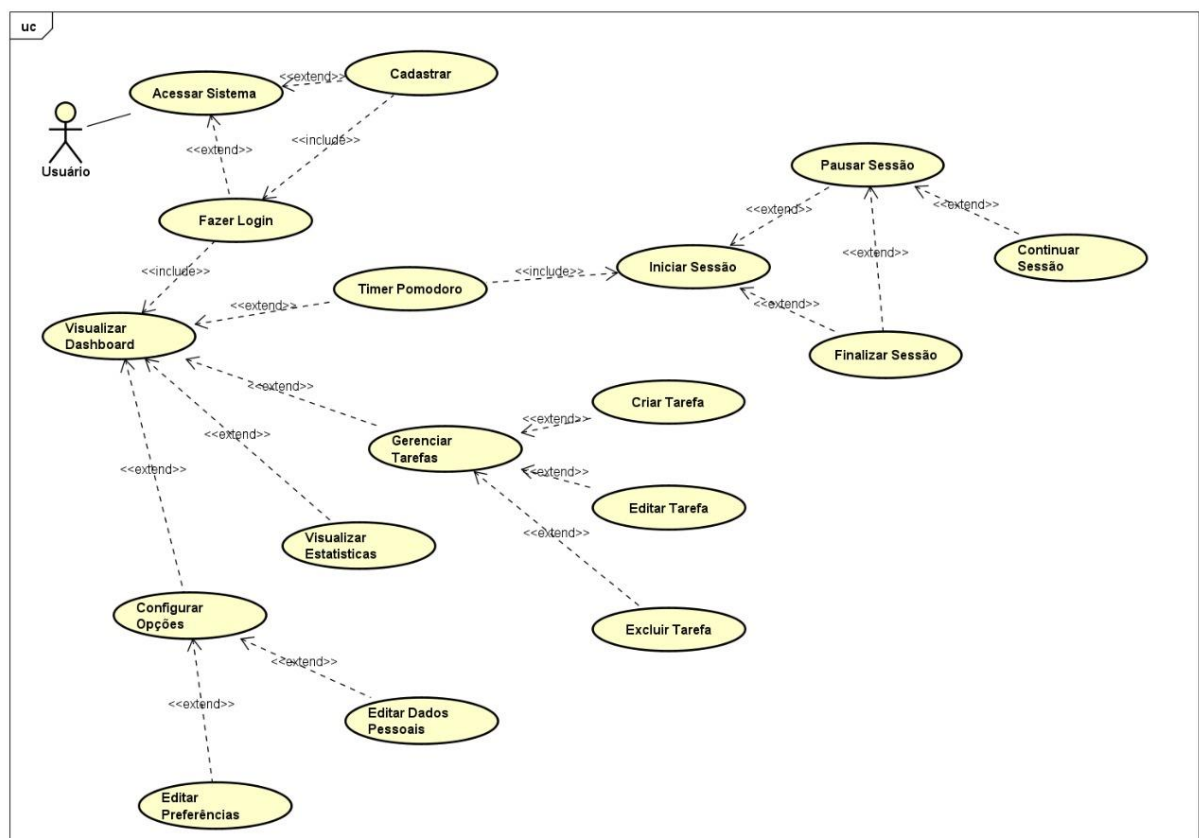
Conforme RUP (Rational Unified Process), casos de uso servem como base para o planejamento e controle iterativo do desenvolvimento de software, fornecendo uma visão centrada no usuário das funcionalidades do sistema.

Segundo JACOBSON et al. (1992), os casos de uso são importantes porque:

1. Facilitam a comunicação entre stakeholders técnicos e não técnicos
2. Fornecem uma base para o planejamento do projeto e estimativa de esforço
3. Servem como entrada para o projeto e implementação do sistema
4. Permitem a validação dos requisitos com os usuários finais

Para LEFFINGWELL & WIDRIG (2003), casos de uso ajudam a garantir que o sistema seja desenvolvido de forma a atender às reais necessidades dos usuários, reduzindo o risco de desenvolver funcionalidades incorretas ou desnecessárias.

Figura 4 – Diagrama de Caso de Uso



Fonte: Os autores

- Acessar Sistema (UC 001) – O usuário abre o aplicativo.
- Cadastrar (UC 002) – O usuário faz o cadastro para utilizar o aplicativo.
- Fazer Login (UC 003) – Login do usuário no aplicativo.

- Visualizar Dashboard (UC 004) – O aplicativo exibe a página principal e as abas disponíveis.
- Timer Pomodoro (UC 005) – O aplicativo exibe o timer para execução de tarefas.
- Inicia Sessão (UC 006) – O usuário escolhe a tarefa e inicia uma sessão.
- Pausar Sessão (UC 007) – O usuário pausa a sessão que foi iniciada.
- Continuar Sessão (UC 008) – O usuário retorna a sessão que foi pausada.
- Finalizar Sessão (UC 009) – O usuário finaliza a sessão que foi pausada ou o tempo do timer chega ao fim.
- Gerenciar Tarefas (UC 010) – O aplicativo exibe as tarefas e dá a opção de criar, excluir ou editar tarefas.
- Criar Tarefa (UC 011) – O usuário cria novas tarefas.
- Editar Tarefa (UC 012) – O usuário edita uma tarefa.
- Excluir Tarefa (UC 013) – O usuário exclui uma tarefa.
- Visualizar Estatística (UC 014) – O aplicativo exibe as estatísticas do usuário.
- Configurar Opções (UC 015) – O aplicativo mostra as opções que podem ser alteradas.
- Editar Preferências (UC 016) – O usuário altera as preferências do timer pomodoro.
- Editar Dados Pessoais (UC 017) – O usuário altera os dados pessoais da conta.

Quadro 8 - Documento de Caso de Uso

Caso de Uso – Acessar Sistema	
ID	UC 001
Descrição	Este caso de uso tem como objetivo iniciar o sistema.
Ator Primário	Usuário.
Pré-condição	Não possui.
Cenário Principal	1. O caso de uso se inicia quando o usuário abre o aplicativo 2. O sistema abre a opção de fazer Login ou Cadastrar
Pós-condição	Não possui.
Cenário Alternativo	Não possui.

Inclusão	Não possui.
Extensão	Não possui.
Caso de Uso – Cadastrar	
ID	UC 002
Descrição	Este caso de uso tem como objetivo cadastrar um novo usuário no sistema.
Ator Primário	Usuário.
Pré-condição	Não possui.
Cenário Principal	1. O caso de uso inicia quando o usuário seleciona a opção cadastrar. 2. O sistema abre um campo para inserir nome, e-mail e senha que será cadastrado para o usuário. 3. O usuário informa o nome, e-mail e senha para realizar o cadastro. 4. O usuário aperta em cadastrar. 5. O sistema verifica se o e-mail informado possui cadastro. 6. O sistema confirma que o e-mail não possui cadastro. 7. O sistema grava os dados no banco de dados.
Pós-condição	Não possui.
Cenário Alternativo	4a. O usuário cancela a operação de cadastro. 4b.1. O sistema interrompe o processo e retorna à tela inicial (ou à página anterior). 5a- O e-mail já está cadastrado.
Inclusão	Não possui.
Extensão	UC 001 – Acessar Sistema
Caso de Uso – Fazer Login	
ID	UC 003
Descrição	Este caso de uso tem como objetivo fazer o login no sistema.
Ator Primário	Usuário.
Pré-condição	Possuir uma conta cadastrada

Cenário Principal	1. O caso de uso inicia quando o usuário seleciona a opção fazer login. 2. O sistema solicita os dados de login. 3. O administrador preenche os campos e-mail e senha. 4. O sistema verifica se o e-mail e senha estão corretos. 5. O sistema valida e-mail e senha. 6. O sistema realiza o login.
Pós-condição	Não possui.
Cenário Alternativo	5a- O e-mail ou a senha informada está incorreto.
Inclusão	UC 002 – Confirmar Credenciais
Extensão	UC 001 – Acessar Sistema
Caso de Uso – Visualizar Dashboard	
ID	UC 004
Descrição	Esse caso de uso tem como objetivo acessar o dashboard do sistema.
Ator Primário	Usuário.
Pré-condição	Ter realizado o login no sistema.
Cenário Principal	1. O caso de uso inicia quando o usuário realiza o login no sistema. 2. O sistema dá as opções para o usuário navegar nas páginas do aplicativo.
Pós-condição	Não possui.
Cenário Alternativo	Não possui.
Inclusão	UC 003 – Fazer Login.
Extensão	Não possui.
Caso de Uso – Timer Pomodoro	
ID	UC 005
Descrição	Este caso de uso tem como objetivo acessar o timer do sistema.
Ator Primário	Usuário.

Pré-condição	Ter efetuado o login.
Cenário Principal	1. O caso de uso inicia quando o usuário seleciona o timer do pomodoro. 2. O sistema mostra o timer no estado atual. 3. O usuário tem a opção de escolher a tarefa que irá iniciar.
Pós-condição	Não possui.
Cenário Alternativo	Não possui.
Inclusão	Não possui.
Extensão	UC 004 – Acessar Dashboard
Caso de Uso – Iniciar sessão	
ID	UC 006
Descrição	Este caso de uso tem como objetivo iniciar as sessões de pausa e foco.
Ator Primário	Usuário.
Pré-condição	Ter preferências definidas.
Cenário Principal	1. O caso de uso inicia quando o usuário seleciona a tarefa a ser realizada. 2. O usuário selecionar a opção iniciar. 3. O sistema começa a contagem do tempo (foco/pausa). 4. O sistema troca automaticamente (foco/pausa) quando termina o timer. 5. O usuário inicia novamente, até os ciclos definidos terminarem.
Pós-condição	Não possui.
Cenário Alternativo	4a- O usuário pode parar os ciclos antes de terminá-los.
Inclusão	UC 005 – Timer pomodoro
Extensão	Não possui.
Caso de Uso – Pausar Sessão	
ID	UC 007

Descrição	Este caso de uso tem como objetivo pausar ou resetar a sessão.
Ator Primário	Usuário.
Pré-condição	Ter iniciado alguma sessão.
Cenário Principal	1. O caso de uso inicia quando o usuário seleciona pausar ou resetar a sessão. 2. O sistema para de contar o timer. 3. O sistema da a opção de continuar ou finalizar a sessão. 4. O usuário seleciona se continua ou finaliza a sessão.
Pós-condição	Não possui.
Cenário Alternativo	Não possui.
Inclusão	Não possui.
Extensão	UC 006 – Iniciar Sessão
Caso de Uso – Continuar Sessão	
ID	UC 008
Descrição	Este caso de uso tem como objetivo escolher se vai continuar com a sessão.
Ator Primário	Usuário
Pré-condição	Ter pausado uma sessão.
Cenário Principal	1. O caso de uso se inicia quando o usuário pausa o timer antes de chegar ao fim. 2. O sistema da opção de escolher entre finalizar ou continuar a sessão. 3. O usuário escolhe a opção de continuar a sessão. 4. O sistema retoma a contagem do timer.
Pós-condição	A tarefa é finalizada ou reiniciada.
Cenário Alternativo	3a- O usuário finaliza a sessão.
Inclusão	UC 007 – Pausar Sessão
Extensão	Não possui.
Caso de Uso – Finalizar Sessão	
ID	UC 009

Descrição	Este caso de uso tem como objetivo informar quando terminar sessão.
Ator Primário	Usuário
Pré-condição	Ter iniciado uma sessão.
Cenário Principal	1. O caso de uso se inicia quando os ciclos da sessão chegam ao fim ou o usuário escolhe finalizar sessão após pausar. 2. O sistema finaliza a sessão.
Pós-condição	A tarefa é finalizada.
Cenário Alternativo	Não possui.
Inclusão	Não possui.
Extensão	UC 006 – Iniciar Sessão / UC 007 – Pausar Sessão
Caso de Uso – Gerenciar Tarefas	
ID	UC 010
Descrição	Este caso de uso tem como objetivo controlar as tarefas
Ator Primário	Usuário.
Pré-condição	Ter efetuado o login.
Cenário Principal	1. O caso de uso se inicia quando o usuário seleciona a aba tarefas. 2. O sistema mostra as tarefas criadas. 3. O sistema oferece a opção de criar, editar ou excluir tarefas. 4. O usuário consegue marcar as tarefas como concluídas.
Pós-condição	Não possui.
Cenário Alternativo	Não possui.
Inclusão	Não possui.
Extensão	UC 004 – Visualizar Dashboard
Caso de Uso – Criar Tarefa	
ID	UC 011
Descrição	Este caso de uso tem como objetivo criar uma nova tarefa.

Ator Primário	Usuário.
Pré-condição	Não possui.
Cenário Principal	1. O caso de uso se inicia quando o usuário seleciona criar tarefa. 2. O sistema abre o formulário para criar a tarefa com os campos título, descrição e prioridade (Baixa/Média/Alta) 3. O usuário preenche o os dados e clica em salvar. 4. O sistema cria a nova tarefa e salva.
Pós-condição	Não possui.
Cenário Alternativo	3a- O usuário pode cancelar a qualquer momento a criação da tarefa.
Inclusão	Não possui.
Extensão	UC 010 – Gerenciar Tarefas
Caso de Uso – Editar Tarefa	
ID	UC 012
Descrição	Este caso de uso tem como objetivo editar os dados da tarefa.
Ator Primário	Usuário.
Pré-condição	Ter uma tarefa criada.
Cenário Principal	1. O caso de uso se inicia quando o usuário seleciona editar tarefa. 2. O sistema abre os dados da tarefa com os campos título, descrição e prioridade (Baixa/Média/Alta) e oferece a opção de editar. 3. O usuário atualiza dos dados da tarefa e clica em salvar. 4. O sistema altera os dados da tarefa e salva.
Pós-condição	Não possui
Cenário Alternativo	3a- O usuário pode cancelar a qualquer momento a edição da tarefa.
Inclusão	Não possui.
Extensão	UC 010 – Gerenciar Tarefas

Caso de Uso – Excluir Tarefa	
ID	UC 013
Descrição	Este caso de uso tem como objetivo excluir uma tarefa criada.
Ator Primário	Usuário.
Pré-condição	Ter uma tarefa criada.
Cenário Principal	1. O caso de uso se inicia quando o usuário seleciona excluir tarefa. 2. O sistema aparece um alerta para confirmar se tem certeza da exclusão e oferece a opção de “sim” e “não”. 3. O usuário confirma a exclusão da tarefa. 4. O sistema exclui a tarefa.
Pós-condição	Não possui.
Cenário Alternativo	3a- O usuário pode cancelar a qualquer momento a exclusão da tarefa.
Inclusão	Não possui.
Extensão	UC 010 – Gerenciar Tarefas
Caso de Uso – Visualizar Estatísticas	
ID	UC 014
Descrição	Este caso de uso tem como objetivo visualizar as estatísticas do usuário no aplicativo.
Ator Primário	Usuário.
Pré-condição	Ter e
Cenário Principal	1. O caso de uso se inicia quando o usuário seleciona a aba histórico. 2. O sistema mostra os dados do usuário de foco/sessões/tarefas concluídas. 3. O usuário consegue filtrar por períodos (dia/semana/mês/tudo). 4. O sistema mostra os dados suportados.
Pós-condição	Não possui.
Cenário Alternativo	Não possui.

Inclusão	Não possui.
Extensão	UC 004 – Visualizar Dashboard
Caso de Uso – Configurar Opções	
ID	UC 015
Descrição	Este caso de uso tem como objetivo configurar o aplicativo.
Ator Primário	Usuário.
Pré-condição	Ter efetuado o login.
Cenário Principal	1. O caso de uso se inicia quando o usuário seleciona a aba configurações. 2. O sistema mostra opções de editar alterar as configurações do app ou do usuário.
Pós-condição	Não possui.
Cenário Alternativo	Não possui.
Inclusão	Não possui.
Extensão	UC 004 – Visualizar Dashboard
Caso de Uso – Editar Preferências	
ID	UC 016
Descrição	Este caso de uso tem como objetivo editar as preferências do pomodoro.
Ator Primário	Usuário.
Pré-condição	Não possui.
Cenário Principal	1. O caso de uso se inicia quando o usuário seleciona a as alterações do aplicativo. 2. O sistema mostra as informações do timer e da as opções de mudar duração do foco, pausa curta, pausa longa, número de ciclos de casa sessão. 3. O usuário tem a opção de resetar os dados para os pré-definidos ou de sair da conta. 4. O usuário altera as opções que achar necessário e clica sem salvar. 5. O sistema salva as alterações feitas.
Pós-condição	Não possui.

Cenário Alternativo	3a- O usuário escolhe sair da conta. 3a1- O usuário volta para tela de login.
Inclusão	Não possui.
Extensão	UC 015 – Configurar Opções
Caso de Uso – Editar Dados Pessoais	
ID	UC 017
Descrição	Este caso de uso tem como objetivo editar os dados pessoais.
Ator Primário	Usuário.
Pré-condição	Ter efetuado o login.
Cenário Principal	1. O caso de uso se inicia quando o usuário seleciona a opção de editar dados 2. O sistema abre a aba com os dados do usuário (nome, e-mail e senha). 3. O usuário consegue alterar os dados exibidos ou excluir a conta. 4. O usuário muda os dados necessários e clica em salvar. 5. O sistema salva os dados alterados.
Pós-condição	Não possui.
Cenário Alternativo	3a- O usuário seleciona a opção de excluir conta. 3a- O sistema exibe um alerta de confirmar exclusão. 3a- O usuário confirma a exclusão e volta a tela de login. 4a- O usuário pode cancelar a alteração de dados a qualquer momento.
Inclusão	Não possui.
Extensão	UC 015 – Configurar Opções

Fontes: Autores.

A estrutura de casos de uso elaborada para o FocusTime demonstra uma arquitetura de funcionalidades bem-organizada, conforme recomendado por ARMOUR & MILLER (2001).

Observa-se:

1. Hierarquia Lógica: Os casos de uso estão organizados em uma estrutura hierárquica que reflete o fluxo natural do usuário no sistema, desde o acesso inicial até as operações específicas.
2. Granularidade Adequada: Conforme COCKBURN (2001), os casos de uso possuem nível de detalhe apropriado, nem muito genérico (perdendo informação) nem muito detalhado (tornando-se complexo).
3. Relacionamentos Consistentes: Os relacionamentos de inclusão e extensão estão bem aplicados, demonstrando dependências entre funcionalidades de forma clara.
4. Cobertura Completa: Todos os requisitos funcionais identificados anteriormente estão cobertos pelos casos de uso, garantindo que nenhuma funcionalidade essencial foi negligenciada.

Esta especificação servirá como base sólida para o desenvolvimento do sistema, permitindo que desenvolvedores, testadores e usuários tenham um entendimento comum sobre o comportamento esperado do FocusTime.

3.6 Diagrama de Classes

De acordo com BOOCH, RUMBAUGH & JACOBSON (2005), o Diagrama de Classes é uma representação estática da estrutura de um sistema, mostrando as classes do sistema, seus atributos, operações e os relacionamentos entre os objetos. É um dos diagramas mais importantes da UML (Unified Modeling Language) e serve como base para a implementação do sistema.

Para FOWLER (2004), o Diagrama de Classes descreve os tipos de objetos no sistema e os vários tipos de relacionamentos estáticos que existem entre eles. Mostra também as propriedades e operações de uma classe e as restrições que se aplicam à maneira como os objetos estão conectados.

Conforme LARMAN (2004), os diagramas de classes são utilizados tanto para modelagem conceitual quanto para modelagem de projeto, permitindo visualizar a estrutura do domínio do problema e a arquitetura da solução.

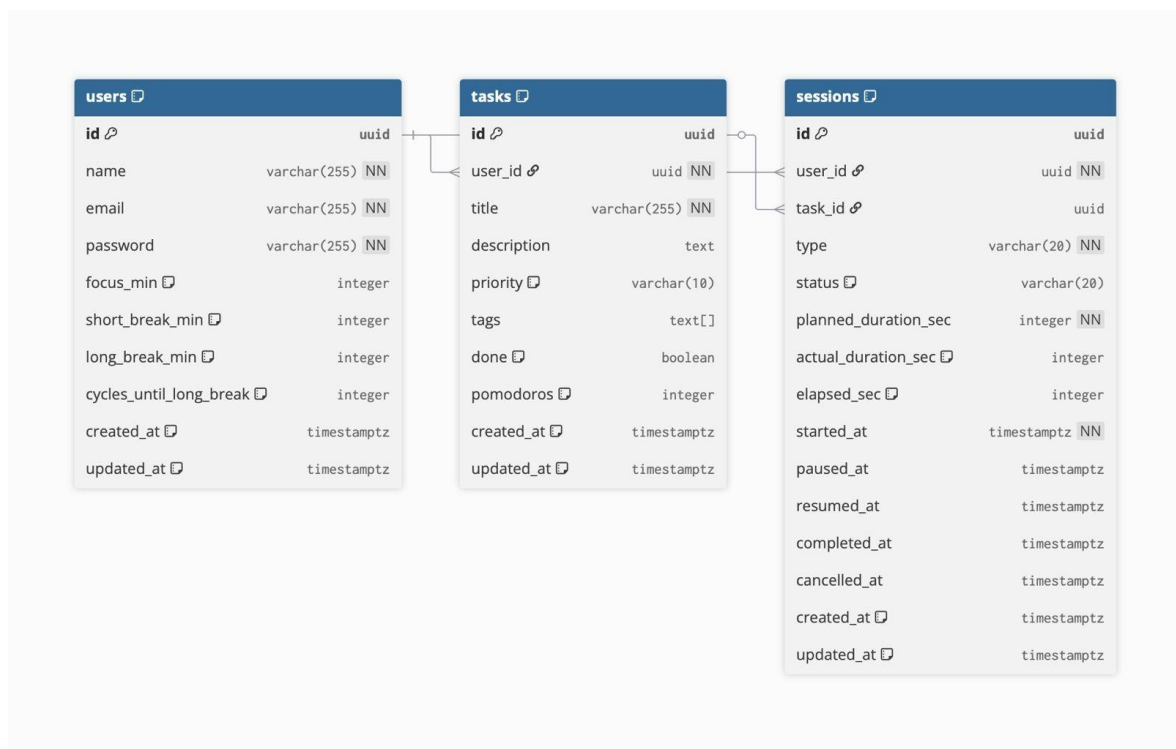
Segundo PRESSMAN (2011), o Diagrama de Classes é importante porque:

1. Fornece uma visão estática do sistema que será construído
2. Serve como base para a geração de código em linguagens orientadas a objetos
3. Facilita a compreensão da arquitetura do sistema por toda a equipe
4. Permite identificar oportunidades de reutilização através de herança e composição

Para SOMMERVILLE (2011), em projetos de software, o diagrama de classes é fundamental para:

- Documentar a estrutura do sistema de forma padronizada
- Comunicar decisões de design entre desenvolvedores
- Identificar dependências entre componentes do sistema
- Planejar a implementação e divisão de tarefas

Figura 5 – Diagrama de Classe



Fonte: Os autores

O diagrama de classes do FocusTime demonstra uma arquitetura bem estruturada que segue os princípios do Domain-Driven Design propostos por EVANS (2004):

1. Agregações e Composições

- User é a raiz de agregação, contendo Tasks e Sessions
- Relacionamento 1:N entre User e Task (um usuário tem muitas tarefas)
- Relacionamento 1:N entre User e Session (um usuário tem muitas sessões)
- Relacionamento 1:1 opcional entre Task e Session (uma tarefa pode ter uma sessão ativa)

2. Padrões de Design Aplicados

Conforme GAMMA et al. (1994), observa-se a aplicação de:

- Factory Method: Criação de Sessions através do método start()
- State Pattern: Transições de status na classe Session (running → paused → completed)
- Observer Pattern: Atualização de estatísticas quando sessões são completadas

3. Princípios SOLID

O design segue os princípios SOLID definidos por MARTIN (2003):

- Single Responsibility: Cada classe tem uma responsabilidade única
- Open/Closed: Classes abertas para extensão, fechadas para modificação
- Liskov Substitution: Métodos podem ser substituídos por implementações específicas
- Interface Segregation: Interfaces específicas para cada operação
- Dependency Inversion: Dependências em abstrações, não em implementações

4. Modelagem de Domínio Rico

A classe Session exemplifica um modelo de domínio rico com lógica de negócio embutida, seguindo as recomendações de VERNON (2013) para sistemas complexos onde o comportamento é tão importante quanto os dados.

Esta estrutura fornece uma base sólida para a implementação do sistema, garantindo escalabilidade, manutenibilidade e aderência aos requisitos funcionais e não funcionais estabelecidos.

3.7 Diagrama de Atividades

De acordo com BOOCH, RUMBAUGH & JACOBSON (2005), o Diagrama de Atividades é um diagrama da UML (Unified Modeling Language) que modela a lógica de procedimentos, fluxos de trabalho e comportamentos sequenciais em um sistema. Ele representa o fluxo de controle de uma atividade para outra, mostrando as decisões, paralelismos e sincronizações que ocorrem durante a execução de um processo.

Para FOWLER (2004), os diagramas de atividades são particularmente úteis para modelar fluxos de trabalho de negócio, casos de uso complexos e operações que envolvem múltiplos objetos colaborando para alcançar um objetivo comum. Eles focam na sequência e condições para coordenar comportamentos.

Conforme LARMAN (2004), os diagramas de atividades podem ser usados em diversos níveis de abstração, desde fluxos de trabalho de alto nível até a especificação detalhada de métodos complexos.

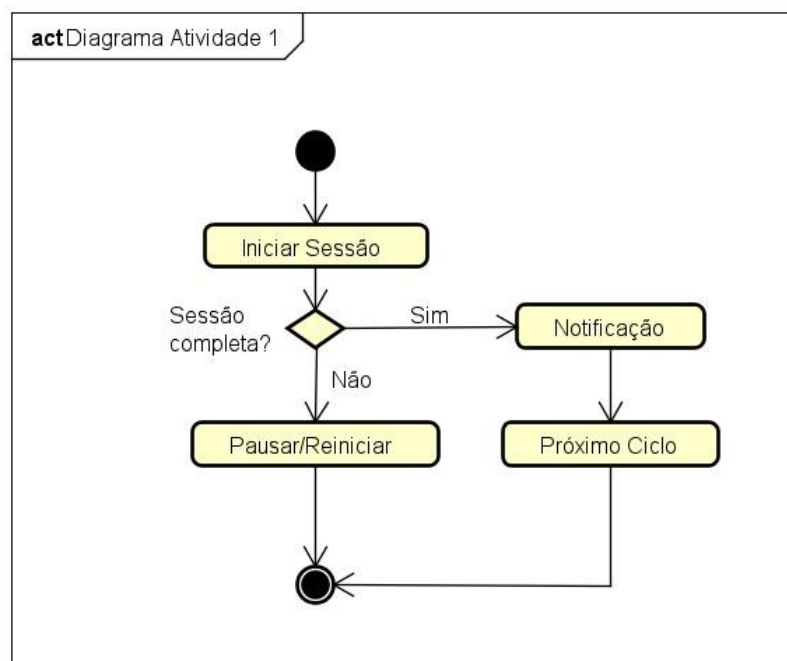
Segundo PRESSMAN (2011), o Diagrama de Atividades é importante porque:

1. Visualiza fluxos de trabalho complexos de forma clara e compreensível
2. Identifica pontos de decisão e condições que afetam o comportamento do sistema
3. Revela oportunidades de paralelismo e execução concorrente
4. Facilita a comunicação entre analistas de negócio e desenvolvedores

Para SOMMERVILLE (2011), em sistemas como o FocusTime que envolvem fluxos temporais e transições de estado, os diagramas de atividades são essenciais para:

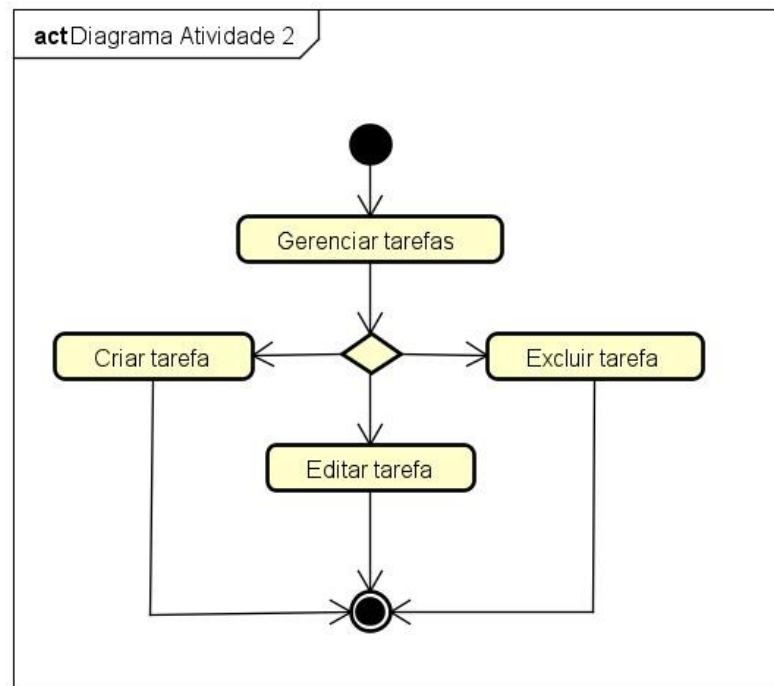
- Modelar comportamentos dependentes do tempo (sessões Pomodoro)
- Especificar condições de transição entre estados do sistema
- Documentar a lógica de negócio complexa de forma visual
- Identificar cenários de erro e exceções que devem ser tratados

Figura 6 – Diagrama de Atividade 1



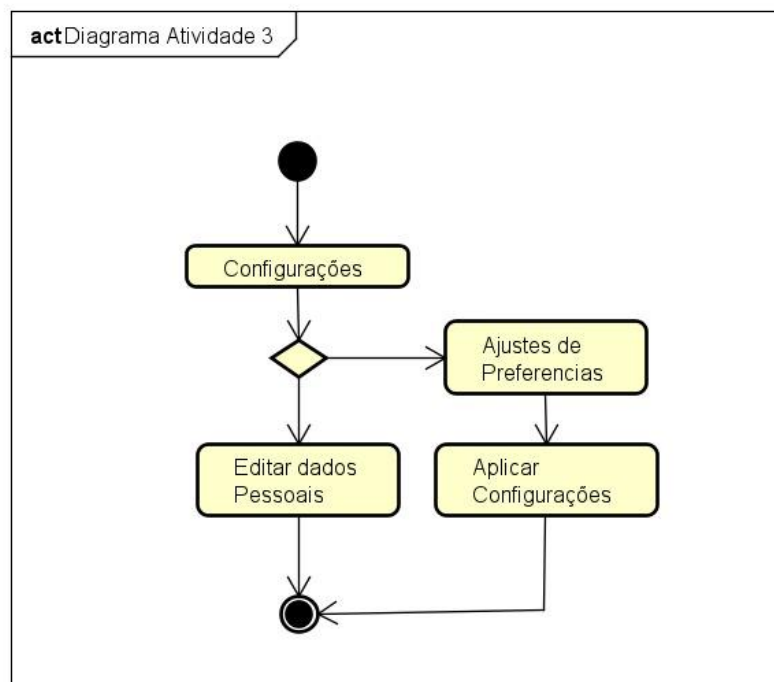
Fonte: Os autores

Figura 7 – Diagrama de Atividade 2



Fonte: Os autores

Figura 8 – Diagrama de Atividade 3



Fonte: Os autores

Os diagramas de atividades elaborados para o FocusTime revelam padrões arquiteturais importantes que seguem as melhores práticas de Engenharia de Software:

1. Padrão State Machine para Sessões Pomodoro

Conforme GAMMA et al. (1994), o fluxo da sessão Pomodoro implementa um padrão de máquina de estados bem definido:

- Estados: Iniciado, Em Execução, Pausado, Finalizado
- Transições: Tempo esgotado, Usuário pausa, Ciclo completo
- Ações: Notificação, Registro, Atualização

2. Padrão CRUD para Gerenciamento de Tarefas

O fluxo de gerenciamento de tarefas segue o padrão CRUD (Create, Read, Update, Delete) estabelecido por FOWLER (2003), com validações e confirmações para operações destrutivas.

3. Padrão Strategy para Configurações

As configurações do sistema implementam o padrão Strategy descrito por ALUR et al. (2003), permitindo que diferentes categorias de configuração (preferências, dados pessoais) sejam tratadas de forma consistente mas com comportamentos específicos.

4. Tratamento de Condições de Corrida

Os diagramas identificam pontos de decisão críticos onde condições de corrida podem ocorrer, especialmente nas transições de sessão, seguindo as recomendações de MARTIN (2008) para sistemas concorrentes.

Estes diagramas de atividades são fundamentais para a implementação do FocusTime porque:

1. Especificam a lógica de negócio complexa das sessões Pomodoro
2. Identificam pontos de integração entre os módulos do sistema
3. Documentam comportamentos excepcionais que devem ser tratados
4. Servem como base para testes de integração e aceitação
5. Facilitam a identificação de requisitos não funcionais como performance e confiabilidade

Conforme IEEE 830-1998, esta especificação detalhada de atividades garante que todos os stakeholders tenham um entendimento comum sobre o comportamento esperado do sistema em cenários complexos.

3.8 Diagrama de Estados

De acordo com BOOCH, RUMBAUGH & JACOBSON (2005), o Diagrama de Estados é um diagrama da UML que modela o comportamento dinâmico de objetos ao longo do tempo, mostrando a sequência de estados que um objeto passa durante seu ciclo de vida em resposta a eventos. Ele descreve os estados possíveis de um objeto e as transições entre esses estados.

Para FOWLER (2004), os diagramas de estados são particularmente úteis para modelar objetos que têm comportamento complexo dependente do estado, onde a resposta a um evento pode variar dependendo do estado atual do objeto. Eles são essenciais para entender o ciclo de vida completo de entidades de domínio.

Conforme LARMAN (2004), os diagramas de estados capturam o comportamento de objetos individuais ao longo de múltiplos casos de uso, mostrando como os objetos respondem a diferentes estímulos e como evoluem através de vários estados.

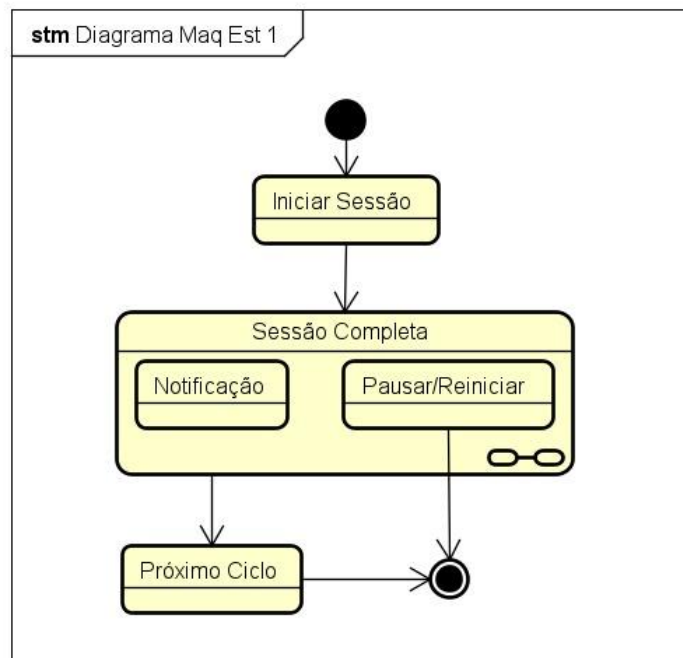
Segundo PRESSMAN (2011), o Diagrama de Estados é importante porque:

1. Modela o comportamento dependente do estado de objetos complexos
2. Identifica todos os estados possíveis que um objeto pode assumir
3. Especifica transições de estado e os eventos que as disparam
4. Revela estados inválidos e transições proibidas

Para SOMMERVILLE (2011), em sistemas como o FocusTime que envolvem processos temporais e fluxos de trabalho, os diagramas de estados são cruciais para:

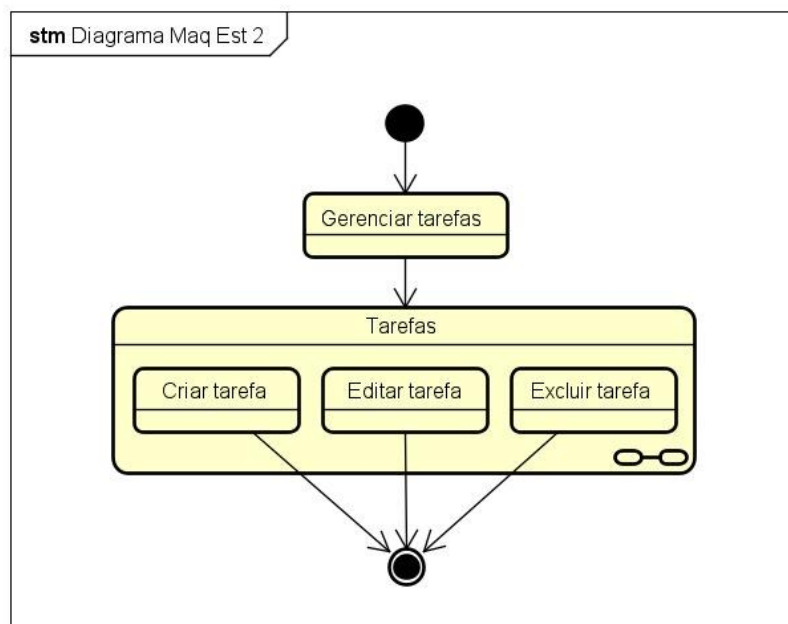
- Garantir a integridade do estado do sistema
- Prevenir transições inválidas entre estados
- Documentar a lógica de negócio relacionada a mudanças de estado
- Facilitar a implementação de máquinas de estado no código

Figura 9 – Diagrama de Estado 1



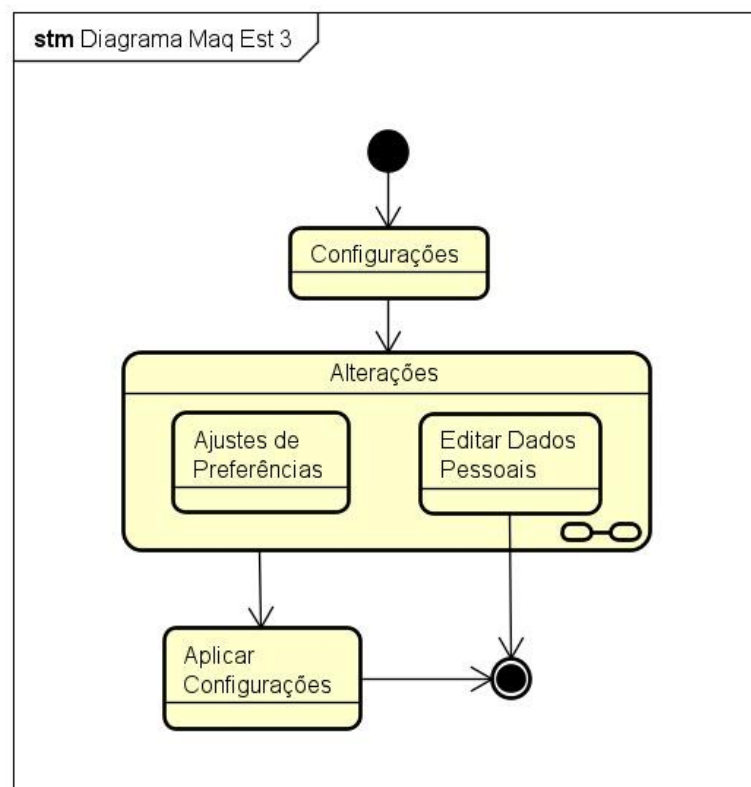
Fonte: Os autores

Figura 10 – Diagrama de Estado 2



Fonte: Os autores

Figura 11 – Diagrama de Estado 3



Fonte: Os autores

Os diagramas de estados elaborados para o FocusTime revelam padrões comportamentais sofisticados que seguem as melhores práticas de Engenharia de Software:

1. Padrão State Machine para Sessões Pomodoro

Conforme GAMMA et al. (1994), a sessão Pomodoro implementa um padrão State bem definido onde:

- Cada estado encapsula comportamento específico: Executando (contagem), Pausado (espera), Completado (registro)
- Transições são explícitas e controladas: Não há transições diretas entre estados não adjacentes
- Estados finais são claramente identificados: Cancelado é terminal

2. Padrão Lifecycle para Entidades de Domínio

As tarefas e usuários seguem o padrão de ciclo de vida descrito por EVANS (2004) no Domain-Driven Design:

- Estados coerentes com o domínio: Tarefa não pode ser editada quando concluída
- Transições que preservam invariantes: Usuário não pode excluir conta sem estar ativo
- Estados intermediários para operações complexas: Dados Alterados para garantir consistência

3. Padrão Status para Controle de Fluxo

O sistema implementa o padrão Status recomendado por FOWLER (2003):

- Estados representam progresso no workflow: Nova → Ativa → Concluída
- Transições refletem processo de negócio: Editando é estado transitório para validação

- Histórico de estados implícito: Permite rastreabilidade

4. Tratamento de Estados Inconsistentes

Os diagramas identificam e previnem estados inconsistentes seguindo MEYER (1997):

- Pré-condições para transições: Sessão só pode pausar se estiver executando
- Pós-condições garantidas: Tarefa concluída permanece imutável
- Estados de erro explícitos: Cancelado para sessões interrompidas

Estes diagramas de estados são fundamentais para garantir a qualidade do FocusTime porque:

1. Previnem bugs de estado através de transições bem definidas
2. Facilitam a implementação de máquinas de estado no código
3. Servem como base para testes de estado e transição
4. Documentam regras de negócio complexas de forma visual
5. Permitem a validação de cenários de uso com stakeholders

Conforme IEEE 14764-2006, esta especificação detalhada de estados garante a confiabilidade do sistema através do controle rigoroso das transições de estado e prevenção de condições de corrida em operações concorrentes.

3.9 Diagrama de Sequência

De acordo com BOOCH, RUMBAUGH & JACOBSON (2005), o Diagrama de Sequência é um diagrama de interação da UML que modela o fluxo de lógica dentro de um sistema em forma de mensagens trocadas entre objetos ao longo do tempo. Ele mostra como objetos se comunicam em uma sequência temporal para executar uma funcionalidade específica.

Para FOWLER (2004), os diagramas de sequência são particularmente úteis para modelar a lógica de casos de uso, mostrando a sequência de interações entre

os objetos do sistema e atores externos. Eles focam na ordem temporal das mensagens e nas responsabilidades dos objetos.

Conforme LARMAN (2004), os diagramas de sequência são fundamentais para entender como os objetos colaboram para realizar um comportamento do sistema, destacando a ordem das operações e as dependências entre elas.

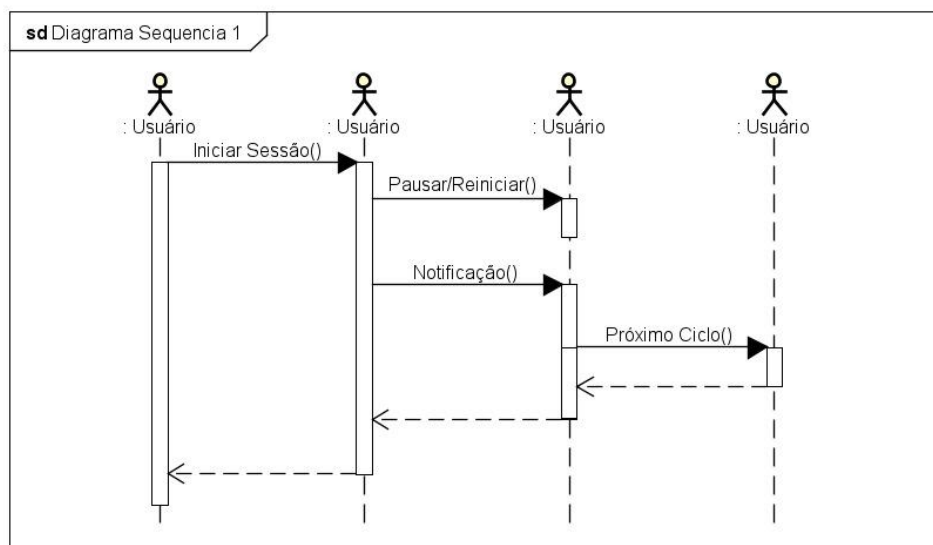
Segundo PRESSMAN (2011), o Diagrama de Sequência é importante porque:

1. Visualiza a sequência temporal de operações em um cenário específico
2. Identifica as responsabilidades de cada objeto no sistema
3. Revela dependências e acoplamento entre componentes
4. Facilita a detecção de problemas de design como operações síncronas bloqueantes

Para SOMMERVILLE (2011), em sistemas complexos como o FocusTime, os diagramas de sequência são essenciais para:

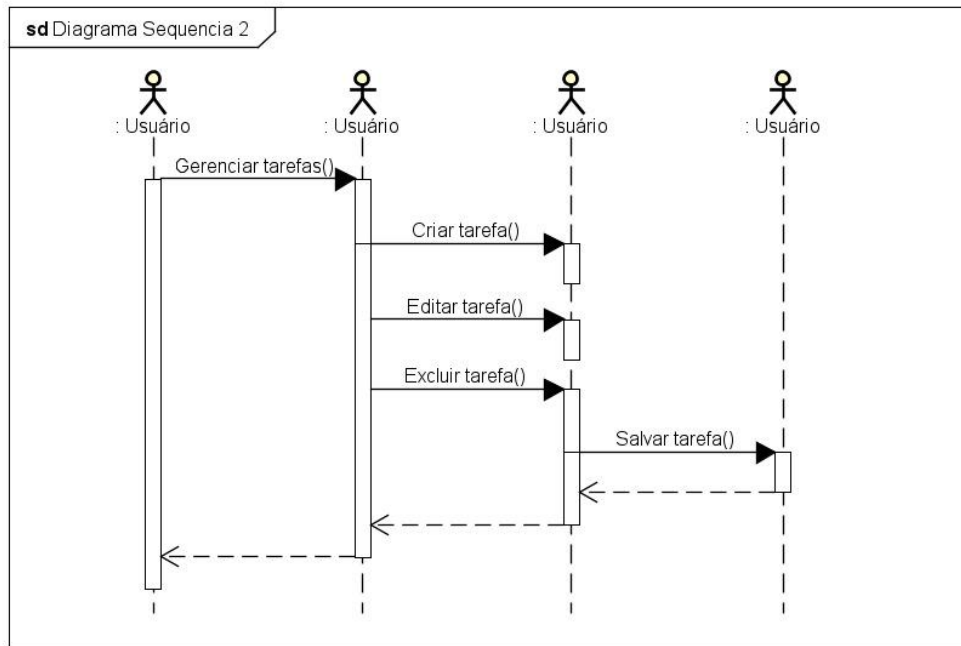
- Especificar a lógica de implementação de casos de uso complexos
- Identificar oportunidades de otimização e paralelização
- Documentar contratos de interface entre componentes
- Servir como base para testes de integração

Figura 12 – Diagrama de Sequência 1



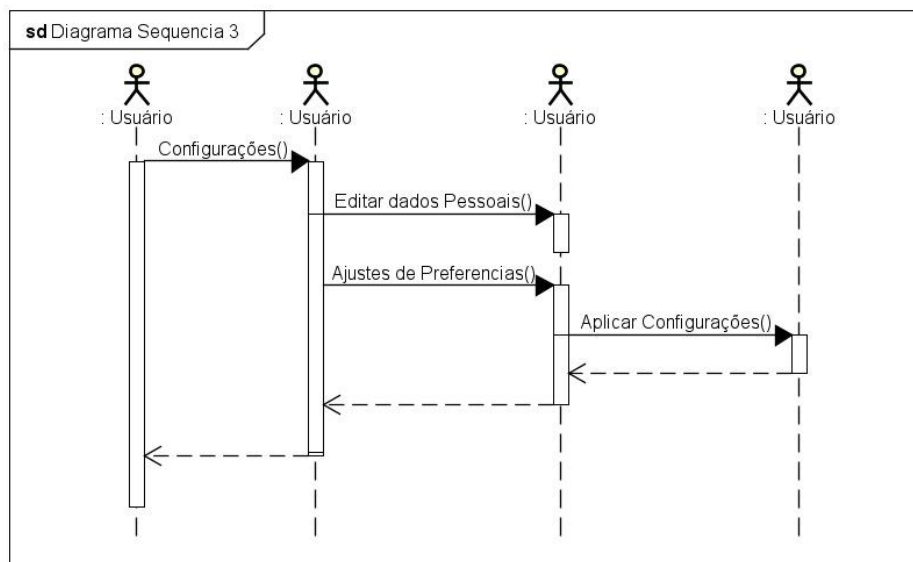
Fonte: Os autores

Figura 13 – Diagrama de Sequência 2



Fonte: Os autores

Figura 14 – Diagrama de Sequência 3



Fonte: Os autores

1. Fluxo da Sessão Pomodoro (FIGURA 12)

O diagrama revela uma arquitetura em camadas bem definida seguindo o padrão MVC (Model-View-Controller) descrito por GAMMA et al. (1994). Observa-se:

- Separação de concerns: Interface (apresentação), Controller (lógica), Service (negócio)
- Comunicação assíncrona: Timer opera independentemente da interface
- Tratamento de estado: Múltiplas transições com validação
- Notificações desacopladas: Sistema de notificação independente

2. Fluxo de Gerenciamento de Tarefas (FIGURA 13)

Este fluxo implementa o padrão Transaction Script conforme FOWLER (2003), com:

- Validação em múltiplas camadas: Client-side e server-side
- Operações atômicas: Cada ação é independente
- Feedback imediato: Validação em tempo real
- Persistência isolada: Camada de dados desacoplada

3. Fluxo de Configurações (FIGURA 14)

O diagrama demonstra o padrão Repository proposto por EVANS (2004):

- Abstração de persistência: Database isolado do domínio
- Reutilização de serviços: Validação compartilhada
- Operações CRUD completas: Create, Read, Update
- Consistência transacional: Atualizações atômicas

Conforme LARMAN (2004), os diagramas de sequência revelam a aplicação consistente de:

1. Arquitetura em Camadas

- Apresentação: Interface responsiva ao usuário
- Aplicação: Controllers orquestrando fluxos
- Domínio: Services com lógica de negócio
- Infraestrutura: Persistência e serviços externos

2. Princípios SOLID

- Single Responsibility: Cada objeto tem uma responsabilidade clara
- Dependency Inversion: Dependências em abstrações, não implementações
- Open/Closed: Extensível para novos comportamentos

3. Design por Contrato

As interações seguem o design por contrato de MEYER (1997), com:

- Pré-condições: Validações antes das operações
- Pós-condições: Estados consistentes após operações
- Invariantes: Propriedades mantidas durante execução

Estes diagramas de sequência fornecem uma base sólida para implementação e garantem que todos os componentes do sistema FocusTime colaboram de forma coerente e eficiente.

4 Ferramentas e Métodos

Quadro 9 - Ferramentas de Desenvolvimento

Categoria	Ferramenta	Versão	Licença	Site Oficial	Justificativa de Uso
Backend	Node.js	18.x	MIT	https://nodejs.org	Runtime JavaScript

					assíncrono e não-bloqueante ideal para APIs REST
Backend Framework	Fastify	5.6.1	MIT	https://fastify.io	Alta performance, baixo overhead, excelente para APIs
ORM	Drizzle ORM	0.44.7	Apache 2.0	https://orm.drizzle.team	ORM type-safe com excelente suporte a TypeScript
Banco de Dados	PostgreSQL	17	PostgreSQL	https://postgresql.org	Banco relacional robusto com suporte a JSON e performances superiores
Frontend Mobile	React Native	0.79.1	MIT	https://reactnative.dev	Desenvolvimento mobile cross-platform com JavaScript
Framework Mobile	Expo	53.0.0	MIT	https://expo.dev	Simplifica desenvolvimento React Native com

					ferramentas integradas
Gerenciamento de Estado	Zustand	5.0.8	MIT	https://zustand-demo.pmnd.rs	Estado global simples e eficiente sem boilerplate excessivo
Navegação	Expo Router	5.0.2	MIT	https://expo.github.io/router	Roteamento baseado em arquivos para React Native
Validação	Zod	4.1.12	MIT	https://zod.dev	Validação de schemas com TypeScript integration
Estilização	StyleSheet (React Native)	nativa	MIT	-	Solução nativa para estilização mobile
Ícones	Lucide React Native	0.475.0	MIT	https://lucide.dev	Biblioteca de ícones consistente e moderna
Autenticação	JWT	9.0.2	MIT	https://github.com/auth0/node-jsonwebtoken	Tokens stateless para autenticação segura
Hash de Senhas	bcryptjs	3.0.3	MIT	https://github.com/dcodeio/bcrypt.js	Criptografia segura para senhas

Containerização	Docker	24.0 +	Apache 2.0	https://docker.com	Ambiente de desenvolvimento consistente
Documentação API	Fastify Swagger	9.5.2	MIT	https://github.com/fastify/fastify-swagger	Documentação automática da API OpenAPI

Fontes: Autores.

De acordo com SOMMERVILLE (2011), a seleção de ferramentas de desenvolvimento deve considerar fatores como maturidade, suporte da comunidade, curva de aprendizado e adequação aos requisitos do projeto. As escolhas para o FocusTime foram fundamentadas em:

1. Stack JavaScript Full-Stack

Conforme NADAREISHVILI (2016), a utilização de JavaScript tanto no frontend quanto no backend permite:

- Reutilização de conhecimento entre as camadas
- Compatibilidade de tipos com TypeScript
- Ecosistema unificado de pacotes e ferramentas

2. Arquitetura de API REST com Fastify

Segundo RODRIGUES (2020), o Fastify oferece performance superior comparado a outros frameworks Node.js, sendo ideal para aplicações que exigem baixa latência como sistemas de timer em tempo real.

3. React Native com Expo

Conforme MENDES (2022), esta combinação proporciona:

- Desenvolvimento ágil com hot reload
- Acesso a recursos nativos através de plugins

- Build simplificado para múltiplas plataformas

Metodologia Ágil

O projeto foi desenvolvido seguindo princípios ágeis conforme BECK et al. (2001):

- Iterações curtas com entregas incrementais
- Feedback contínuo através de protótipos funcionais
- Adaptação a mudanças nos requisitos

Design Centrado no Usuário

A interface foi desenvolvida seguindo os princípios de NORMAN (2013):

- Simplicidade na navegação e interações
- Feedback visual imediato para ações do usuário
- Consistência nos padrões de design

Desenvolvimento Guiado por Testes (TDD)

Conforme MARTIN (2008), partes críticas do sistema foram desenvolvidas com TDD:

- Testes unitários para lógica de negócio complexa
- Testes de integração para APIs e banco de dados

Repositórios e Documentação

- SWOT: Análise estratégica documentada no Capítulo 2.2
- 5W2H: Plano de ação detalhado no Capítulo 2.3
- BPMN: Modelagem de processos no Capítulo 3.2
- Diagramas UML: Casos de uso, classes, atividades, sequência e estados
- Código Fonte: Repositório Git com versionamento semântico

5 Desenvolvimento

O sistema FocusTime foi desenvolvido utilizando uma arquitetura moderna baseada em React Native para o frontend mobile e Node.js com Fastify para o backend. A aplicação foi projetada seguindo os princípios de componentização e reutilização de código, garantindo manutenibilidade e escalabilidade.

O projeto adotou o padrão de gerenciamento de estado com Zustand, proporcionando uma solução leve e eficiente para o controle do estado global da aplicação. A navegação entre telas utiliza o Expo Router, facilitando a criação de fluxos intuitivos para o usuário.

No backend, a API REST foi desenvolvida com Fastify, oferecendo alta performance e validação de dados através do Zod. O banco de dados PostgreSQL foi escolhido pela sua confiabilidade e suporte a tipos de dados avançados, com o Drizzle ORM garantindo type-safety nas operações de banco.

O timer Pomodoro foi implementado com estados bem definidos (executando, pausado, finalizado) e transições controladas, proporcionando uma experiência fluida ao usuário. O sistema de tarefas permite criação, edição e exclusão, com categorização por prioridades e tags.

A interface segue princípios de design minimalista, com cores contextuais que indicam o tipo de sessão em andamento: vermelho para foco, verde para pausa curta e azul para pausa longa. As estatísticas de produtividade oferecem insights visuais sobre o desempenho do usuário ao longo do tempo.

A sincronização entre frontend e backend foi implementada com tratamento robusto de erros e estados de carregamento. O sistema mantém consistência dos dados mesmo em cenários de conexão instável, com mecanismos de fallback para operações offline.

A autenticação utiliza JWT tokens com refresh automático, garantindo segurança sem comprometer a experiência do usuário. As senhas são criptografadas usando bcrypt antes do armazenamento no banco de dados.

5.1 Métricas

Métricas são indicadores quantitativos usados para medir e avaliar o desempenho, progresso ou qualidade de um determinado processo, produto ou

sistema. Elas fornecem dados que ajudam a entender como um projeto está progredindo em relação aos objetivos estabelecidos.

"As métricas de software servem para realizar a tarefa fundamental do gerenciamento de projetos que é o planejamento. É com uma métrica que se pode identificar a quantidade de esforço necessário para o desenvolvimento, o custo para tal e o prazo para entrega dos artefatos."

Figura 15 – Métricas

Fonte: Autores

No contexto deste projeto, as métricas foram desenvolvidas utilizando o Microsoft Excel para facilitar a coleta, análise e visualização de dados. As principais métricas monitoradas incluíram a eficiência no processo de pedidos, o tempo de resposta para atender aos clientes, e a taxa de satisfação após a implementação do sistema automatizado. Essas métricas foram fundamentais para entender o impacto das melhorias implementadas e identificar áreas que ainda necessitam de ajustes.

5.2 Proposta Comercial

A proposta comercial é um documento formal utilizado para apresentar os detalhes de um projeto ou serviço a um cliente potencial, com o objetivo de firmar uma parceria ou fechar um contrato. Esse documento costuma descrever os produtos ou serviços oferecidos, os prazos, os custos, os termos de pagamento, as condições de entrega e outras informações essenciais para alinhar as expectativas entre as partes envolvidas.

A importância de uma Proposta Comercial é enorme, especialmente em negociações que envolvem projetos personalizados ou serviços que precisam ser adaptados às necessidades do cliente. Conforme LEONARDO (2017),

"A proposta comercial é peça fundamental para o nascimento de um projeto, pois dá início às discussões sobre seu planejamento, além de auxiliar na execução e na entrega do mesmo".

Com base nos objetivos identificados, nossa proposta visa implementar uma solução que modernize o gerenciamento de tempo e produtividade de usuários. Atualmente, 65,7% das pessoas relatam dificuldade em manter o foco e 74,3% não utilizam nenhuma ferramenta específica para gestão do tempo. Essa iniciativa permitirá que usuários otimizem seu tempo, melhorem a produtividade e tenham maior controle sobre suas tarefas.

A justificativa para a implementação deste sistema é clara: a maioria das pessoas enfrenta desafios com procrastinação e distrações, especialmente com redes sociais (71,4%). Essa limitação será solucionada através de um aplicativo móvel que automatiza a Técnica Pomodoro e centraliza as tarefas em uma única plataforma.

Conforme identificado na análise, a solução proposta inclui:

- Timer Pomodoro inteligente: os usuários poderão gerenciar ciclos de foco e pausa de forma automatizada, aumentando a produtividade
- Sistema de tarefas integrado: organização completa de atividades com prioridades e categorias, substituindo métodos manuais como listas em papel

Hoje, as pessoas dependem de métodos manuais para organizar tarefas, o que torna o processo ineficiente e sujeito a esquecimentos. O sistema proposto será desenvolvido com interface intuitiva, adaptada para dispositivos móveis, alinhando-se à realidade tecnológica atual onde 54,3% dos usuários preferem usar apps tanto no celular quanto no computador. Isso proporcionará maior eficiência no gerenciamento do tempo, garantindo que cada tarefa seja registrada e acompanhada de forma organizada.

O escopo da solução foi delineado a partir da Estrutura Analítica do Projeto, e incluirá:

- Módulo de timer Pomodoro: possibilitará controle completo dos ciclos de foco e pausa
- Sistema de gestão de tarefas: permitirá que usuários organizem atividades com prioridades e prazos
- Dashboard de estatísticas: oferecerá insights sobre produtividade e progresso

A implementação do sistema está prevista para ser concluída em aproximadamente 3 meses, seguindo um cronograma detalhado para garantir uma transição suave e bem-sucedida.

O investimento necessário para o desenvolvimento e implementação será de R\$ 2.045,25, abrangendo os custos de desenvolvimento, testes, documentação e implantação inicial.

Esta solução atende às necessidades de usuários que buscam melhorar sua produtividade e representa uma estratégia moderna de gestão do tempo. Com a automação do método Pomodoro e centralização das tarefas, esperamos não só melhorar a eficiência pessoal, mas também elevar a qualidade de vida, permitindo que as pessoas gerenciem seu tempo de maneira mais eficaz e organizada.

Considerações finais

O projeto FocusTime atingiu satisfatoriamente seus objetivos iniciais de desenvolver um sistema integrado de gestão de tempo baseado na Técnica Pomodoro. Foi implementada uma aplicação móvel completa com timer inteligente, sistema de tarefas, estatísticas de produtividade e interface minimalista, atendendo às necessidades identificadas na pesquisa com usuários.

Os principais desafios enfrentados incluíram a implementação do timer em background, a sincronização de dados entre frontend e backend, e a criação de uma experiência de usuário intuitiva que realmente auxiliasse na manutenção do foco. A arquitetura em React Native com Expo mostrou-se adequada para o desenvolvimento multiplataforma, enquanto o backend em Node.js com Fastify proporcionou a performance necessária para as operações em tempo real.

Para versões futuras, planeja-se a implementação de funcionalidades adicionais como integração com calendários externos, modo colaborativo para equipes, sistema de conquistas gamificado e versão web complementar. O projeto demonstra viabilidade técnica e de negócio, com potencial para evolução contínua no competitivo mercado de aplicativos de produtividade.

Referências

ALUR, D.; CRUPI, J.; MALKS, D. Core J2EE Patterns: Best Practices and Design Strategies. 2nd ed. Upper Saddle River: Prentice Hall, 2003.

ARMOUR, F.; MILLER, G. Advanced Use Case Modeling: Software Systems. Boston: Addison-Wesley, 2001.

BARNEY, J. B. Firm Resources and Sustained Competitive Advantage. Journal of Management, v. 17, n. 1, p. 99-120, 1991.

BLANK, S. The Four Steps to the Epiphany: Successful Strategies for Products that Win. 5th ed. Hoboken: John Wiley & Sons, 2013.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. The Unified Modeling Language User Guide. 2nd ed. Boston: Addison-Wesley, 2005.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). Diário Oficial da União, Brasília, DF, 15 ago. 2018.

CARLSHAMRE, P.; REGNELL, B. Requirements Lifecycle Management and Release Planning in Market-Driven Requirements Engineering Processes. In: INTERNATIONAL WORKSHOP ON REQUIREMENTS ENGINEERING: FOUNDATIONS FOR SOFTWARE QUALITY, 6., 2000. Anais [...] Stockholm, 2000.

CARPINETTI, L. C. R. Gestão da Qualidade: conceitos e técnicas. 2. ed. São Paulo: Atlas, 2010.

CHUNG, L. et al. Non-Functional Requirements in Software Engineering. New York: Springer, 2012.

CIRILLO, F. The Pomodoro Technique: The Acclaimed Time-Management System That Has Transformed How We Work. New York: Currency, 2018.

COCKBURN, A. Writing Effective Use Cases. Boston: Addison-Wesley, 2001.

COHEN, D. User Research for the Rest of Us. New York: Rosenfeld Media, 2004.

CONWAY, D. User Stories: Back to Basics. Agile Journal, 2009. Disponível em: <https://www.agilejournal.com>.

DUMAS, M. et al. Fundamentals of Business Process Management. 2nd ed. Berlin: Springer, 2018.

EVANS, E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Boston: Addison-Wesley, 2004.

FOWLER, M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. Boston: Addison-Wesley, 2004.

FOWLER, M. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2003.
GAMMA, E. et al. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley, 1994.

GAMMA, E. et al. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley, 1994.

HAGEN, A. The Hierarchy of Engagement. Medium, 2018. Disponível em: <https://medium.com/@awilkinson/the-hierarchy-of-engagement-5803bf4eefc2>.

IEEE. INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE Standard 830-1998: Recommended Practice for Software Requirements Specifications. New York: IEEE, 1998.

IEEE. INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE Standard 29148-2011: Systems and Software Engineering - Life Cycle Processes - Requirements Engineering. New York: IEEE, 2011.

CHAFFEY, D. Digital Marketing: Strategy, Implementation and Practice. 7th ed. Harlow: Pearson, 2019.

IEEE. INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE Standard 14764-2006: Software Engineering - Software Life Cycle Processes - Maintenance. New York: IEEE, 2006.

ISO/IEC. ISO/IEC 25010:2011 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva: ISO, 2011.

JABLONSKI, S. Process Modeling and Process Management in Pervasive Environments. In: PROCESS MANAGEMENT IN PERVASIVE ENVIRONMENTS. Berlin: Springer, 2016. p. 15-32.

JACOBSON, I. et al. Object-Oriented Software Engineering: A Use Case Driven Approach. Boston: Addison-Wesley, 1992.

KERZNER, H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 12th ed. Hoboken: John Wiley & Sons, 2017.

KOTLER, P.; KELLER, K. L. Marketing Management. 15th ed. Boston: Pearson, 2016.

LARMAN, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3rd ed. Upper Saddle River: Prentice Hall, 2004.

LEFFINGWELL, D.; WIDRIG, D. Managing Software Requirements: A Use Case Approach. 2nd ed. Boston: Addison-Wesley, 2003.

LEONARDO, M. Gestão de Projetos: Do Conceito à Prática. São Paulo: Atlas, 2017.

MARKETRESEARCH.COM. Productivity Management Software Market Analysis 2024. Disponível em: <https://www.marketresearch.com>. Acesso em: 18 nov. 2025.

MARTIN, R. C. Agile Software Development: Principles, Patterns, and Practices. Upper Saddle River: Prentice Hall, 2003.

MARTIN, R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Upper Saddle River: Prentice Hall, 2008.

MAURYA, A. Running Lean: Iterate from Plan A to a Plan That Works. 2nd ed. Sebastopol: O'Reilly Media, 2012.

MAXIMIANO, A. C. A. Administração de projetos: como transformar ideias em resultados. 4. ed. São Paulo: Atlas, 2012.

MEYER, B. Object-Oriented Software Construction. 2nd ed. Upper Saddle River: Prentice Hall, 1997.

PRESSMAN, R. S. Engenharia de Software: uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011.

MOORE, J. F. Predators and Prey: A New Ecology of Competition. Harvard Business Review, v. 71, n. 3, p. 75-86, 1993.

NIELSEN, J. Usability Engineering. 2nd ed. San Francisco: Morgan Kaufmann, 2012.

PRESSMAN, R. S. Engenharia de Software: uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011.

NORMAN, D. A. The Design of Everyday Things. Revised and Expanded Edition. New York: Basic Books, 2013.

O'BRIEN, H. L.; TOMS, E. G. What is user engagement? A conceptual framework for defining user engagement with technology. Journal of the American Society for Information Science and Technology, New York, v. 59, n. 6, p. 938-955, 2008.

OMG. OBJECT MANAGEMENT GROUP. Business Process Model and Notation (BPMN) Version 2.0. 2011. Disponível em: <https://www.omg.org/spec/BPMN/2.0/>. Acesso em: 18.set.2025.

OSTERWALDER, A.; PIGNEUR, Y. Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Hoboken: John Wiley & Sons, 2010.

OSTERWALDER, A. et al. Value Proposition Design: How to Create Products and Services Customers Want. Hoboken: John Wiley & Sons, 2014.

PARKER, G. G. Platform Revolution: How Networked Markets Are Transforming the Economy and How to Make Them Work for You. New York: W. W. Norton & Company, 2016.

PIELOT, M. et al. When attention is not scarce-detecting boredom from mobile phone usage. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Seattle, 2014. Anais [...] New York: ACM, 2014. p. 763-774.

PINHEIRO, J. Métricas de Software: Fundamentos e Aplicações. Porto Alegre: Bookman, 2015.

PMI. PROJECT MANAGEMENT INSTITUTE. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 7th ed. Newtown Square: PMI, 2021.

PRESSMAN, R. S. Engenharia de Software: uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011.

REISER, R. H. Engenharia de Requisitos de Software. Porto Alegre: Bookman, 2020.

RIES, E. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. New York: Crown Business, 2011.

ROBERTSON, S.; ROBERTSON, J. Mastering the Requirements Process: Getting Requirements Right. 3rd ed. Boston: Addison-Wesley, 2012.

RUP. Rational Unified Process: Best Practices for Software Development Teams. Rational Software White Paper, 1998.

SMITH, C. U. Performance Engineering of Software Systems. Boston: Addison-Wesley, 2015.

SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

STATISTA. Global Productivity App Market Size 2024. Disponível em: <https://www.statista.com>. Acesso em: 18 nov. 2025.

VERNON, V. Implementing Domain-Driven Design. Upper Saddle River: Prentice Hall, 2013.

ZIMMERMAN, B. J. Becoming a self-regulated learner: An overview. Theory into Practice, Columbus, v. 41, n. 2, p. 64-70, 2002.

WESKE, M. Business Process Management: Concepts, Languages, Architectures. 3rd ed. Berlin: Springer, 2019.

WHITE, S. A. Introduction to BPMN. IBM Corporation, 2004. Disponível em: <https://www.ibm.com/developerworks/library/ws-bpmn/>. Acesso em: 18.set.2025.

WHITMAN, M. E.; MATFORD, H. J. Principles of Information Security. 6th ed. Boston: Cengage Learning, 2016.

WIEGERS, K.; BEATTY, J. Software Requirements. 3rd ed. Redmond: Microsoft Press, 2013.

APÊNDICE 1

Missão: Desenvolver uma solução tecnológica acessível que empodere pessoas para gerenciar seu tempo de forma eficiente, aumentando a produtividade e melhorando a qualidade de vida através da Técnica Pomodoro.

Visão: Ser reconhecido como o aplicativo mais intuitivo e eficaz para gestão de tempo no mercado brasileiro, ajudando milhares de pessoas a superarem a procrastinação e alcançarem seus objetivos.

Valores:

- Simplicidade acima da complexidade
- Evidências científicas acima de modismos
- Experiência do usuário acima de funcionalidades desnecessárias
- Acessibilidade acima de exclusividade
- Melhoria contínua baseada em feedback real

APÊNDICE 2 - Questionário e Respostas da Elicitação de Requisitos

1. Qual a sua idade?

- Menos de 18 anos: 5,7%
- 18 a 24 anos: 34,3%
- 25 a 34 anos: 45,7%
- 35 a 44 anos: 11,4%
- 45 a 54 anos: 0%
- 55 anos ou mais: 2,9%

2. Qual sua ocupação principal?

- Estudante: 31,4%
- Profissional: 17,1%
- Ambos: 51,4%

3. Você já utiliza algum aplicativo de gestão de tempo?

- Sim: 20%
- Não: 74,3%
- Sim, qual?: 2,9%
- Outro: Keep da Google: 2,9%

4. Com que frequência você sente dificuldade em manter o foco?

- Nunca: 0%
- Raramente: 11,4%
- Às vezes: 22,9%
- Frequentemente: 34,3%
- Sempre: 31,4%

5. Você costuma organizar suas tarefas diárias? Como faz isso hoje?

- "Não costumo"
- "Lista de tarefa do Keep"
- "Agenda física e virtual"
- "Lista mental ou pelo despertador"
- "Conforme são lembradas e/ou perto do prazo de entrega"
- "Mentalmente"
- "Importantes primeiro, o que não fica para depois"
- "Senso de obrigação"
- "Níveis de urgência"
- "Planilha"
- "Ferramenta Trello"
- "Checklist"
- "Tento planejar, mas dificilmente dá certo"
- "Metas diárias, ou anotações"
- "Por horário"
- "Notes do celular"
- "To do list celular"
- "Agenda celular"

6. Qual sua maior dificuldade ao tentar gerenciar seu tempo?

- "Manter o foco"
- "Celular"
- "Redes sociais"
- "Distribuir o tempo de forma otimizada"

- "Ter disciplina para fazer todo o proposto"
- "Gerenciar o que tem que ser feito"
- "Dividir o tempo adequadamente entre as tarefas"
- "Seguir uma rotina 100%"
- "Disciplina"
- "Definir as prioridades"
- "Começar a fazer, tomar iniciativa é a pior parte"
- "Motivação"
- "Perder raciocínio e ter que retomar"
- "Focar na atividade e não ser distraído"
- "Celular e procrastinação"
- "24 horas não é tempo suficiente"
- "Procrastinação"
- "Tempo"
- "Falta de visibilidade"
- "Foco"
- "Separar o tempo necessário pra cada atividade"
- "Falta de foco e disponibilidade de horário"
- "Sempre tem imprevistos"
- "Procrastinação"
- "Perda de foco"
- "Procrastinar"
- "Demorar mais tempo do que o estipulado"
- "Atenção"
- "Manter o foco"
- "Falta de tempo"

7. Quais fatores mais atrapalham sua concentração?

- Redes sociais: 71,4%
- Notificações do celular: 45,7%
- Barulho no ambiente: 17,1%
- Cansaço/fadiga: 42,9%
- Falta de motivação: 51,4%
- Pessoas me chamando: 2,9%

8. Você costuma procrastinar? Se sim, em quais situações?

- "Quando sei que tenho tempo hábil para realizar tal tarefa"
- "Às vezes, coisas que preciso fazer, mas não é urgente"
- "Sempre, para fazer trabalhos e tarefas em casa"
- "Sempre, nos estudos"
- "Sim, quando tem muitas coisas para fazer no mesmo dia"
- "Não tenho o costume"
- "Sempre, ao estudos e trabalho quando home office"
- "Às vezes, quando não to motivado"
- "Sim, com celular"
- "Às vezes sim, na maior parte das vezes pra estudar"
- "Sim, tudo que não é urgente eu costumo deixar pra depois"
- "Ao ter que fazer trabalhos e estudar para a prova"
- "Às vezes"
- "Sim durante meu tempo em casa"
- "Sim, quando tenho que terminar alguma atividade"
- "Sempre, no trabalho e atividades do dia a dia"
- "Sempre, sempre que noto que uma tarefa pode ser feita em cima da hora"
- "Sim, todas"
- "Sim, quando tem muita coisa para fazer e fico perdido"
- "Cansaço"
- "Sim. Para estudar, principalmente"
- "Bem pouco, no geral sempre termino o que começo"
- "Sim, quando preciso terminar atividades/projetos que envolvem muitas etapas"
- "Às vezes sim, quando estou cansado"
- "Ao estudar"
- "Sim, em todas que eu puder, exceto as obrigatórias"
- "Todo tempo"
- "Todas"
- "Ao estudar"
- "Sim, em situações onde preciso fazer algo que não quero fazer"
- "Estudos"
- "Sim, usando redes sociais"
- "Sim, em quase tudo"

- "Sim, ao fazer as tarefas do dia"
- "Durante o dia, por falta de motivação"

9. Quando está estudando ou trabalhando, você sente necessidade de pausar em intervalos regulares?

- Sim: 73,4%
- Não: 11,4%
- Às vezes: 14,3%

10. Você já ouviu falar ou já utilizou a técnica de Pomodoro?

- Sim, já utilizei: 34,3%
- Sim, já ouvi falar mas não utilizei: 31,4%
- Não conheço: 34,3%

11. Se pudesse escolher uma única função indispensável em um app de gestão de tempo, qual seria?

- "Lista de tarefas"
- "Decisão de tempo por tarefas"
- "Organização por tempo"
- "Criação de planejamento de atividades"
- "Cadastro de atividades"
- "Roteiro tarefa do dia"
- "Calendário"
- "Atalho para configurar horários"
- "Um marcador para tarefas contínuas e consequentemente um lembrete"
- "Cronometrar o tempo para fazer cada tarefa"
- "Uma função pra ficar tocando música de fundo tipo lo-fi"
- "Cadastro facilitado"
- "Organizar tarefas e definir prazos"
- "Gerenciar tarefas"
- "Poder escolher o tempo de cada tarefa"
- "To do list"
- "Definir as prioridades"
- "Escolher uma tarefa como prioritária"
- "Lembretes de tarefas pendentes"

12. É importante para você que o aplicativo mostre estatísticas sobre seu desempenho?

- Sim: 60%
- Não: 17,1%
- Indiferente: 22,9%

13. Você gostaria de personalizar a duração dos ciclos de foco e pausa?

- Sim: 77,1%
- Não: 11,4%
- Indiferente: 11,4%

14. Você gostaria de receber notificações e lembretes durante o dia?

- Sim: 37,1%
- Não: 34,3%
- Talvez: 28,6%

15. Você prefere que o app seja minimalista e simples ou com várias funcionalidades extras?

- Minimalista e simples: 74,3%
- Completo com várias funcionalidades: 2,9%
- Indiferente: 22,9%

16. Você gostaria de poder criar categorias para organizar suas tarefas?

- Sim: 77,1%
- Não: 0%
- Talvez: 22,9%

17. Você gostaria de ter relatórios semanais/mensais do seu desempenho?

- Sim: 48,6%
- Não: 25,7%
- Talvez: 25,7%

18. Você considera importante ter um histórico das tarefas concluídas?

- Sim: 65,7%
- Não: 11,4%
- Indiferente: 22,9%

19. Você prefere utilizar o app no celular, computador ou ambos?

- Celular: 47,7%
- Computador: 0%
- Ambos: 54,3%