

Projeto - ISTECP Porto

WORKNEST

Plataforma para Gestão de Projetos e Colaboração

Modalidade: Académico

Diogo Lima Regadas

Número: 2022103

Orientador Científico: Prof. Doutor António Pereira

Instituto Superior de Tecnologias Avançadas do Porto
Rua de Silva Tapada, 115, 4200-501 Porto
Portugal

Porto, 12 de Junho de 2025

Resumo

Este projeto, de natureza académica, consistiu no desenvolvimento de uma plataforma digital denominada WorkNest, orientada para as necessidades de estudantes e pequenas equipas de trabalho. Inserido na área de Engenharia de Software, o projeto focou-se na criação de uma ferramenta colaborativa integrada, capaz de responder à fragmentação das soluções existentes para gestão de projetos, comunicação e organização de tarefas.

A plataforma foi concebida com uma estrutura hierárquica baseada em projetos, categorias e tópicos, permitindo organizar o trabalho de forma intuitiva. Dentro de cada tópico, os utilizadores podem interagir em tempo real através de um sistema de chat com suporte a encriptação AES, partilhar mensagens e ficheiros, bem como criar e acompanhar tarefas associadas com datas de entrega, responsáveis, estado de progresso e anexos encriptados.

O WorkNest integra ainda um sistema de notificações visuais, pedidos de amizade, gestão de colaboradores e estados online, promovendo um ambiente de trabalho coeso e dinâmico. Um calendário interativo oferece uma visão global de prazos e eventos, permitindo filtrar por tarefas ou projetos e consultar rapidamente os eventos do mês.

A implementação seguiu uma metodologia ágil, com ciclos iterativos de desenvolvimento, testes e validação. O projeto foi desenvolvido com recurso a tecnologias modernas, como React, Node.js, MongoDB e Firestore, aliando práticas de segurança, como encriptação ponto-a-ponto, à experiência do utilizador.

Este trabalho demonstrou a aplicação prática de conhecimentos adquiridos no curso de Engenharia Informática, desde o levantamento de requisitos até à entrega de uma solução funcional e testada. A plataforma contribui para a melhoria da colaboração e produtividade em equipas de pequeno porte, validando competências técnicas e metodológicas do estudante.

Palavras-chave: WorkNest, Plataforma colaborativa, Gestão de tarefas, Encriptação AES, Engenharia de Software, Comunicação em equipa.

Abstract

This academic project consisted in the development of a digital platform called **WorkNest**, aimed at the needs of students and small work teams. Within the field of Software Engineering, the project focused on creating an integrated collaborative tool capable of addressing the fragmentation of existing solutions for project management, communication, and task organization.

The platform was designed with a hierarchical structure based on projects, categories, and topics, allowing for intuitive work organization. Within each topic, users can interact in real time through a chat system with AES encryption support, share messages and files, as well as create and track tasks associated with due dates, assignees, progress status, and encrypted attachments.

WorkNest also integrates a system of visual notifications, friend requests, collaborator management, and online status indicators, fostering a cohesive and dynamic work environment. An interactive calendar offers a global view of deadlines and events, allowing filtering by tasks or projects and quick access to the month's events.

The implementation followed an agile methodology, with iterative development, testing, and validation cycles. The project was developed using modern technologies such as React, Node.js, MongoDB, and Firestore, combining security practices like end-to-end encryption with user experience.

This work demonstrated the practical application of knowledge acquired in the Computer Engineering course, from requirements gathering to the delivery of a functional and tested solution. The platform contributes to improved collaboration and productivity in small teams, validating the student's technical and methodological skills.

Keywords: WorkNest, Collaborative platform, Task management, AES encryption, Software Engineering, Team productivity.

Agradecimentos

A realização deste projeto representou não só um desafio académico, mas também uma oportunidade de desenvolvimento pessoal e técnico. Quero expressar a minha sincera gratidão a todos os que contribuíram, direta ou indiretamente, para a concretização deste trabalho.

Agradeço a todos os docentes do curso de Engenharia Informática, assim como aos docentes do Instituto Superior de Tecnologias Avançadas do Porto, pelo conhecimento, orientação e incentivo prestados ao longo do percurso académico.

Um agradecimento especial é devido ao meu orientador deste projeto, o Professor António Pereira, pela sua disponibilidade, orientação contínua, rigor técnico e motivação constante, que foram fundamentais para a estruturação e conclusão deste trabalho.

A todos os colegas, amigos e familiares que estiveram presentes ao longo deste percurso, deixo igualmente um sincero agradecimento pelo apoio e encorajamento incondicionais.

Siglário

Sigla	Significado
AES	<i>Advanced Encryption Standard</i> – Padrão de Encriptação Avançado
API	<i>Application Programming Interface</i> – Interface de Programação de Aplicações
CSS	<i>Cascading Style Sheets</i> – Folhas de Estilo em Cascata
CRUD	<i>Create, Read, Update, Delete</i> – Criar, Ler, Atualizar, Eliminar
DB	<i>Database</i> – Base de Dados
E2EE	<i>End-to-End Encryption</i> – Encriptação Ponto-a-Ponto
Express.js	Framework web para Node.js
Firestore	Base de dados em tempo real da Google (Firebase)
HRV	<i>Heart Rate Variability</i> – Variabilidade da Frequência Cardíaca
HTML	<i>HyperText Markup Language</i> – Linguagem de Marcação de Hipertexto
IDE	<i>Integrated Development Environment</i> – Ambiente de Desenvolvimento Integrado
ISTEC	Instituto Superior de Tecnologias Avançadas do Porto
JWT	<i>JSON Web Token</i> – Token de Autenticação no formato JSON
KPI	<i>Key Performance Indicator</i> – Indicador-Chave de Desempenho
MongoDB	Base de dados NoSQL orientada a documentos
MVC	<i>Model-View-Controller</i> – Arquitetura Modelo–Vista–Controlador
Node.js	Ambiente de execução JavaScript do lado do servidor
NoSQL	<i>Not Only SQL</i> – Base de Dados Não Relacional

Sigla	Significado
ODM	<i>Object Document Mapper</i> – Mapeador de Documentos Orientado a Objetos
React	Biblioteca JavaScript para construção de interfaces
REST	<i>Representational State Transfer</i> – Estilo Arquitetural de APIs
SCRUM	Metodologia Ágil para desenvolvimento iterativo
Socket.IO	Biblioteca para WebSockets com fallback automático
SPA	<i>Single Page Application</i> – Aplicação de Página Única
SRS	<i>Software Requirements Specification</i> – Especificação de Requisitos de Software
UI	<i>User Interface</i> – Interface do Utilizador
UX	<i>User Experience</i> – Experiência do Utilizador
UX/UI	Experiência e Interface do Utilizador
Vite	Bundler moderno para aplicações frontend
VS Code	<i>Visual Studio Code</i> – Editor de Código da Microsoft
WebSocket	Protocolo para comunicação bidirecional em tempo real

ÍNDICE GERAL

Resumo	III
Abstract	V
Agradecimentos	VII
Siglário	IX
PARTE I – RELATÓRIO DE PROJETO	
1. INTRODUÇÃO	1
1.1. Modalidade do Projeto	1
1.2. Contextualização do Tema / Descrição do Problema	2
1.3. Motivação	4
1.4. Objetivos	4
2. ESTADO DE ARTE	6
2.1. Over-tooling e a Fadiga Digital	6
2.2. Ferramentas Atuais	8
3. MÉTODOS E FERRAMENTAS	11
3.1. Formalização do Problema Tecnológico	11
3.2. Metodologia	12
4. DESENVOLVIMENTO E IMPLEMENTAÇÃO TÉCNICA	14
4.1. Arquitetura Geral	14
4.2. Backend – Tecnologias e Estrutura	15
4.3. Frontend – Tecnologias e Estrutura	19
4.4. Websockets e Atualização em Tempo Real	26
5. CONCLUSÕES	29
6. REFERÊNCIAS	31
PARTE II – SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)	33
1. Introdução	34
1.1. Objetivos	34
1.2. Público-Alvo	34
1.3. Âmbito do Projeto	34
1.4. Organização do Documento	34
2. Descrição Geral	35
2.1. Perspetiva do Projeto	35
2.2. Funcionalidades do Produto	35
2.3. Classe de Utilizadores	36

2.4.	Ambiente de Operação	36
2.5.	Restrições Técnicas e de Implementação	37
2.6.	Documentação e Apoio ao Utilizador	37
2.7.	Pressupostos e Dependências	38
3.	Stakeholders	39
4.	Metodologias	41
4.1.	Investigação	41
4.2.	Desenvolvimento	42
	4.2.1. Aplicação da Metodologia ao WorkNest	42
5.	Requisitos e Proposta de Solução	43
5.1.	Requisitos Funcionais	43
5.2.	Requisitos Não-Funcionais	58
5.3.	Requisitos de Software	58
5.3.1.	Ambiente do Cliente (Frontend)	59
5.3.2.	Requisitos do Servidor (Backend)	59
5.3.3.	Requisitos para a Base de Dados	60
5.3.4.	Configurações Recomendadas para Desenvolvimento	60
5.3.5.	Requisitos de Deploy (produção)	61
5.4.	Tabela de Rastreabilidade	61
5.5.	Tabela de Prioridade de Requisitos Funcionais (RF)	62
6.	Modelos do Sistema	65
6.1.	Diagrama de Classes	65
6.2.	Casos de Uso	67
6.2.1.	Casos de Uso Textuais	67
6.2.2.	Diagrama de Caso Uso (Plataforma Geral)	72
6.3.	Diagrama de Entidades	73
7.	Regras de Negócio	75
8.	Glossário	76
9.	Plano de Validação e Verificação	78
9.1.	Estratégia Geral	78
9.2.	Validação de Requisitos Funcionais	79
9.3.	Validação de Requisitos Não-Funcionais	80
9.4.	Validação com Utilizadores	80
10.	Gestão de Riscos	81
11.	Propostas de Futuras Melhorias	82
	PARTE III – ARTIGO CIENTÍFICO	83

Resumo	83
Abstract	83
1. Introdução	84
2. Estado de Arte	84
3. Metodologia	87
3.1. Objetivo da Investigação	87
3.2. Abordagem Metodológica	88
3.3. Fontes e Técnicas de Recolha de Dados	88
3.4. Processo de Desenvolvimento da Plataforma WorkNest	88
3.5. Testes Exploratórios e Validação Inicial	89
3.6. Limitação do Estudo	89
4. Apresentação e Discussão de Resultados	89
4.1. Resultados Preliminares	89
4.2. Discussão	90
4.3. Limitações e Próximos Passos	90
5. Conclusão	90
6. Referências	91
BIBLIOGRAFIA GERAL	93
Índice de Tabelas	94
Índice de Figuras	94

PARTE I – RELATÓRIO DE PROJETO

1. INTRODUÇÃO

Os estudantes e pequenas equipas de trabalho enfrentam dificuldades recorrentes na gestão de projetos e na organização das suas atividades devido à fragmentação das ferramentas disponíveis no mercado. Este problema traduz-se na necessidade de utilizar múltiplas plataformas para comunicação, gestão de tarefas, partilha de ficheiros e organização de prazos, criando barreiras a uma colaboração eficiente e produtiva.

Com base na experiência adquirida desde o 10.º ano, ao realizar trabalhos que envolveram tanto desenvolvimento como pesquisa, identifiquei a importância de dispor de uma solução integrada. Este contacto contínuo com ferramentas não especializadas revelou lacunas significativas no suporte às necessidades específicas de estudantes e pequenos grupos. Essa constatação serviu de motivação para o desenvolvimento de uma plataforma que reunisse as funcionalidades essenciais para uma experiência de trabalho mais fluida e eficaz, especialmente no contexto universitário.

Neste projeto, foi desenvolvida uma plataforma de colaboração unificada que utilizou categorias e tópicos para estruturar e organizar projetos, comunicação e tarefas. A solução incluiu funcionalidades como partilha de ficheiros, gestão de tarefas, calendário e um chat centralizado, organizado por categorias e tópicos. Estas ferramentas foram concebidas para otimizar o fluxo de trabalho, reduzir a complexidade da gestão de projetos e melhorar a comunicação entre os membros da equipa.

1.1. Modalidade do Projeto

O presente projeto será desenvolvido na modalidade de trabalho académico. Este formato permite-me aplicar os conhecimentos adquiridos ao longo do curso, focando-me no desenvolvimento de uma solução inovadora que aborda problemas reais enfrentados por estudantes e pequenas equipas de trabalho.

O projeto é inteiramente académico, não estando associado a nenhuma empresa ou entidade externa, e foi concebido com o objetivo de explorar e validar as competências teóricas e práticas que adquiri durante o meu percurso académico.

1.2. Contextualização do Tema / Descrição do Problema

O desenvolvimento de plataformas de colaboração digital desempenhou um papel central na transformação da forma como as equipas trabalham, comunicam e partilham conhecimento. Estas plataformas, inseridas na área da Engenharia de Software, evoluíram significativamente, com impactos notáveis em diversos contextos, como ambientes de trabalho, comunidades de aprendizagem e instituições educativas.

Internacionalmente, as plataformas de colaboração digital foram amplamente reconhecidas como ferramentas essenciais para equipas, permitindo não só a comunicação em tempo real, mas também a organização de tarefas, gestão de prazos e partilha de informação. Exemplos como Slack, Microsoft Teams e Trello ilustram o crescimento deste mercado. (Maican et al., 2021; Zamiri & Esmaeili, 2024) No entanto, essas ferramentas são frequentemente generalistas, não respondendo de forma específica às necessidades de grupos mais pequenos e de estudantes que requerem soluções integradas e acessíveis.

Em Portugal, o uso de ferramentas digitais em contextos educativos e de trabalho cresceu, mas ainda existiam lacunas na sua adaptação às especificidades de equipas de pequeno porte e ao ambiente académico. As soluções existentes eram fragmentadas, exigindo que os utilizadores utilizassem várias ferramentas para comunicação, gestão de tarefas e partilha de ficheiros. Este cenário refletia um problema recorrente que afetava a produtividade e a eficiência dessas equipas.

Esta realidade não é exclusiva do contexto nacional. Um estudo publicado pela plataforma Classter identificou que a fragmentação de sistemas digitais nas universidades irlandesas contribui diretamente para a fadiga digital entre estudantes e professores, devido à necessidade constante de alternar entre múltiplas plataformas para tarefas essenciais, como partilha de conteúdos, comunicação e avaliação (Arissa Shanganlall, 2025). O estudo reforça a importância de soluções centralizadas, destacando que a integração digital é um dos fatores-chave para melhorar a experiência académica e reduzir o esforço cognitivo associado à dispersão tecnológica.

Este projeto inseriu-se nas áreas de Engenharia de Software, Interação Humano-Computador, Ferramentas de Produtividade Digital e Educação e Colaboração Digital. Teve como objetivo desenvolver uma plataforma integrada e intuitiva, desenhada para atender às necessidades específicas de estudantes e pequenas equipas de trabalho. A abordagem baseou-se em princípios de usabilidade, design centrado no utilizador e soluções tecnológicas modernas que facilitassem o fluxo de trabalho, a comunicação e a organização.

A motivação para este projeto surgiu da minha própria experiência desde o ensino secundário, durante a qual realizei diversos trabalhos académicos que envolviam pesquisa e desenvolvimento. Nesse percurso, identifiquei a ausência de uma solução integrada que facilitasse a gestão do trabalho em equipa e percebi como uma plataforma desse tipo poderia ter sido uma mais-valia no meu percurso académico, particularmente no ensino superior.

Problema

O problema tecnológico que este projeto aborda é a fragmentação de ferramentas disponíveis para estudantes e pequenas equipas de trabalho na gestão de projetos e na colaboração. Atualmente, estas equipas enfrentam desafios significativos ao utilizar múltiplas plataformas para comunicação, partilha de ficheiros, organização de tarefas e gestão de prazos.

Esta fragmentação resulta em perda de eficiência, confusão na gestão de informação e dificuldade em manter a comunicação alinhada com as tarefas e objetivos definidos. Apesar da existência de ferramentas amplamente utilizadas, como Slack, Trello ou Google Drive (Torres et al., n.d.), estas não são projetadas para responder de forma integrada e personalizada às necessidades de equipas de pequeno porte ou do contexto académico. A falta de uma solução unificada representa um entrave significativo à produtividade e ao sucesso dos projetos desenvolvidos por este público.

O desafio principal reside, portanto, na criação de uma plataforma que centralize as funcionalidades essenciais para equipas de trabalho pequenas e estudantes, eliminando a necessidade de alternar entre várias ferramentas e promovendo uma colaboração mais eficiente e estruturada.

1.3. Motivação

A motivação para a realização deste projeto surgiu da minha experiência pessoal desde o 10.º ano, ao longo da qual participei em diversos trabalhos de desenvolvimento e pesquisa. Durante este percurso académico, enfrentei várias dificuldades relacionadas com a ausência de uma solução integrada que facilitasse a gestão de tarefas, a comunicação e a organização dos projetos em equipa.

Enquanto estudante de Engenharia Informática, reconheci a relevância deste problema, que evidencia desafios na interligação entre tecnologia e produtividade. O desenvolvimento de uma plataforma capaz de responder a estas necessidades permitiu-me validar os conhecimentos técnicos adquiridos ao longo do curso e, simultaneamente, criar uma solução com impacto real no quotidiano de estudantes e equipas, ao simplificar os processos colaborativos.

Além disso, este projeto revelou-se particularmente enriquecedor do ponto de vista académico, pois possibilitou a aplicação e integração de conceitos fundamentais da área da Engenharia Informática, como o desenvolvimento de software, o design de interfaces intuitivas, a experiência do utilizador (UX) e as metodologias ágeis. A concretização de uma solução prática, acessível e orientada para problemas reais refletiu os objetivos da formação, ao mesmo tempo que contribuiu para a inovação no domínio das ferramentas de colaboração digital.

1.4. Objetivos

O principal objetivo deste projeto foi desenvolver uma aplicação web integrada que funcionasse como uma plataforma de colaboração destinada a estudantes e pequenas equipas de trabalho. A aplicação teve como finalidade facilitar a organização e a produtividade dessas equipas, oferecendo funcionalidades como gestão de tarefas, comunicação por tópicos, partilha de ficheiros e um calendário para monitorização de prazos.

Este projeto teve como propósito central criar uma solução prática e funcional que respondesse às necessidades identificadas no contexto académico e profissional, aplicando as competências técnicas adquiridas ao longo do curso e explorando abordagens tecnológicas modernas.

Objetivos específicos:

- Identificar as necessidades específicas de estudantes e equipas de pequeno porte através do levantamento de requisitos.
- Projetar uma aplicação com uma interface intuitiva e centrada no utilizador, organizada em categorias, tópicos e funcionalidades integradas.
- Implementar as principais funcionalidades da plataforma, incluindo:
 - Gestão de tarefas vinculadas a tópicos;
 - Sistema de mensagens com organização por categorias e tópicos;
 - Ferramentas de partilha de ficheiros;
 - Um calendário funcional para monitorizar prazos e eventos.
- Incorporar objetivos tecnológicos avançados, tais como:
 - Implementação de criptografia E2EE (end-to-end encryption) para mensagens e ficheiros, garantindo segurança e privacidade.
 - Utilização combinada de MongoDB e Firebase para a gestão de dados, explorando a escalabilidade e a eficiência de ambas as tecnologias.
 - Suporte para troca de mensagens em tempo real, assegurando uma comunicação fluida e eficiente entre os utilizadores.
- Realizar testes com utilizadores reais para validar a funcionalidade, usabilidade e impacto da plataforma no contexto de trabalho colaborativo.
- Documentar todo o processo de desenvolvimento, incluindo os requisitos, design, implementação, validação e resultados alcançados.

2. ESTADO DE ARTE

2.1. Over-tooling e a Fadiga Digital

Os ambientes contemporâneos de trabalho e de educação são crescentemente moldados pela integração generalizada de ferramentas digitais, em particular **software de gestão de projetos**. Estas ferramentas são amplamente reconhecidas como essenciais para potenciar a eficiência, a colaboração e a produtividade. Proporcionam plataformas centralizadas para o acompanhamento de tarefas, colaboração em tempo real, comunicação, planeamento de atividades e gestão de recursos. Em 2025, o mercado apresenta uma ampla diversidade de soluções, adaptadas a diferentes dimensões de equipas, graus de complexidade dos projetos e necessidades setoriais, desde gestores de tarefas simples para pequenas equipas até sistemas empresariais robustos. Entre as funcionalidades mais valorizadas encontram-se: visualizações diversas de tarefas (Kanban, Gantt, cronogramas), automatizações, colaboração (chat, partilha de ficheiros, validação), dashboards, controlo de tempo e recursos, bem como opções de integração (Jon Darbyshire, 2025; Ok, n.d.; Praburam Srinivasan, 2025).

No entanto, a adoção rápida e frequentemente descoordenada de múltiplas plataformas digitais, fenômeno designado por **over-tooling**, tem originado um problema crescente: a **fadiga digital**. Este conceito refere-se a um estado de exaustão mental e desengajamento resultante da utilização prolongada e excessiva de dispositivos e plataformas digitais. Caracteriza-se por sentimentos de sobrecarga, traduzindo-se em cansaço mental, emocional e físico, causado pela interação constante com ferramentas digitais e pela sobrecarga informacional associada. Esta realidade agrava-se em ambientes fragmentados, onde as ferramentas não se integram de forma fluida, obrigando os utilizadores a saltar continuamente entre sistemas distintos (Ripla Andre, 2025; Supriyadi et al., 2025).

De acordo com a literatura analisada, os principais mecanismos através dos quais a fadiga digital afeta a produtividade e o bem-estar incluem:

Sobrecarga cognitiva e redução da performance mental: O envolvimento constante com múltiplas ferramentas e notificações esgota a capacidade limitada de processamento do cérebro, resultando em fadiga mental e menor desempenho cognitivo (Supriyadi et al., 2025).

Perda de tempo e ineficiência: A fragmentação dos sistemas exige uma navegação constante entre diferentes plataformas, levando a perdas de tempo significativas, estimadas em cerca de 1,8

horas por dia ou mais de 9 horas por semana em alguns casos (Arissa Shanganlall, 2025; Supriyadi et al., 2025).

Troca de contexto e perda de foco: A necessidade de alternar frequentemente entre ferramentas impõe um custo cognitivo conhecido como *troca de contexto*, que interrompe o fluxo de trabalho e deixa resíduos de atenção que dificultam a concentração na tarefa atual (Supriyadi et al., 2025).

Sobrecarga informacional e fadiga decisional: A proliferação de ferramentas e notificações dispersas conduz a um ambiente saturado de decisões micro e fragmentação da informação, contribuindo para exaustão mental, redução da qualidade das decisões e maior tendência à procrastinação.

Maior propensão a erros: A sobrecarga e a perda de foco aumentam a probabilidade de cometer falhas.

Contributo para o burnout e desmotivação: A exposição contínua à sobrecarga digital eleva os níveis de stress, favorece o burnout e reduz o envolvimento com o trabalho, uma dinâmica associada ao conceito de *tecnostress* (Supriyadi et al., 2025).

Impacto negativo no bem-estar: A fadiga digital está associada a ansiedade, frustração, insatisfação profissional, má qualidade do sono, dores de cabeça e outros efeitos físicos e psicológicos.

Redução da criatividade e inovação: A constante fragmentação digital dificulta o pensamento profundo e bloqueia a capacidade de resolução criativa de problemas.

Absenteísmo e rotatividade: Ambientes digitais mal estruturados e o stress acumulado favorecem a **rotatividade** e o absenteísmo, dificultando a retenção de talentos (Haiilo, 2025; Supriyadi et al., 2025).

Enfrentar a fadiga digital exige uma abordagem **multidimensional**, que combine estratégias individuais e organizacionais. A nível individual, recomenda-se a prática de bem-estar digital, o uso de técnicas de gestão de tempo (como o método Pomodoro) e a consciencialização dos limites cognitivos. No plano organizacional, as soluções passam por consolidar ferramentas, adotar design centrado no utilizador, garantir formação adequada, cultivar uma cultura que privilegie o bem-estar dos colaboradores, selecionar plataformas estrategicamente e assegurar integrações eficazes que minimizem interrupções no fluxo de trabalho.

Métodos de investigação emergentes como o **NeuroIS** têm vindo a ser utilizados para avaliar as respostas fisiológicas a ambientes digitais, utilizando métricas como a **variabilidade da frequência cardíaca (HRV)** para identificar sinais de fadiga cognitiva e emocional. Resultados preliminares sugerem que a colaboração digital pode provocar a redução da atividade do sistema nervoso parassimpático, um potencial indicador de fadiga (Supriyadi et al., 2025).

2.2. Ferramentas Atuais

Perspetivas futuras apontam para tendências tecnológicas que visam mitigar estas questões através da **consolidação de plataformas, melhoria nas integrações**, uso de **inteligência artificial** para simplificação de fluxos de trabalho, e soluções **low-code/no-code** que permitem maior personalização com menor complexidade técnica. O conceito relacionado de **aprendizagem fragmentada**, frequente no contexto educativo digital, reforça também a necessidade de estruturas de gestão de conhecimento mais sistemáticas e de orientação pedagógica adequada (Supriyadi et al., 2025).

A investigação recente tem destacado um vasto leque de ferramentas digitais dedicadas à gestão de projetos e à colaboração em equipa, muitas delas amplamente utilizadas em ambientes empresariais e educativos. Entre as mais referenciadas encontram-se **Asana, Slack, Trello, Jira e Microsoft Teams**, cada uma com abordagens distintas à organização de tarefas, comunicação, partilha de recursos e acompanhamento de progresso (Dahire et al., 2024; Pawłowski & Plechawska-Wójcik, 2022).

Segundo (Dahire et al., 2024), o **Jira** tem sido amplamente adotado em contextos técnicos, sobretudo por equipas que seguem metodologias ágeis como Scrum ou Kanban. A sua força reside na possibilidade de definir fluxos de trabalho complexos, associar tarefas a bugs ou versões de software e configurar automatizações. No entanto, os autores alertam que a ausência de um sistema

de comunicação nativo (como chat ou chamadas) obriga à integração com outras plataformas, o que pode agravar o problema da fragmentação digital.

De acordo com (Pawlowski & Plechawska-Wójcik, 2022), **Asana** é uma das ferramentas mais equilibradas no que respeita à interface, acessibilidade e versatilidade. Permite a gestão visual de projetos em múltiplas vistas (lista, cronograma, Kanban), bem como a definição de dependências e datas de entrega. Contudo, a sua dependência de integrações externas para funcionalidades como reuniões, comunicação síncrona ou gestão de ficheiros pode limitar a centralização do trabalho colaborativo, especialmente em ambientes académicos com menos apoio técnico.

Slack, como destaca (Yankah et al., 2023), é reconhecido como uma das plataformas mais eficazes para comunicação em tempo real, especialmente em equipas distribuídas. Permite organizar canais por temas, integrar bots e conectar-se a centenas de outras ferramentas. No entanto, o seu foco na comunicação instantânea não é complementado por funcionalidades estruturadas de gestão de projetos, tornando-o dependente de soluções paralelas como Trello ou Asana.

Por sua vez, **Microsoft Teams** beneficia de uma forte integração com o ecossistema Office 365, sendo frequentemente usado em instituições educativas e organizações de grande escala (Pawlowski & Plechawska-Wójcik, 2022). Ainda assim, a sua rigidez na personalização dos fluxos de trabalho e a complexidade na gestão de tarefas para utilizadores iniciantes podem limitar a sua eficácia em grupos académicos informais.

Finalmente, **Trello** é descrita por (Monnin, 2020) como uma plataforma intuitiva e visual, ideal para utilizadores com pouca experiência em ferramentas de gestão. Baseada em quadros Kanban, permite organizar tarefas em colunas e acompanhar o progresso com facilidade. No entanto, a ausência de funcionalidades avançadas, como gestão de recursos, relatórios ou automatizações complexas, pode tornar-se um entrave em projetos de maior dimensão ou com múltiplas equipas envolvidas.

Em síntese, apesar das suas qualidades, estas plataformas apresentam limitações que se tornam críticas no contexto do ensino superior, onde os projetos envolvem equipas temporárias, prazos curtos e recursos limitados. A complexidade, a fragmentação funcional e os custos associados constituem barreiras à adoção efetiva e sustentada destas ferramentas, evidenciando a necessidade

de soluções integradas, intuitivas e acessíveis especificamente concebidas para ambientes académicos. É neste contexto que se propõe a plataforma **WorkNest**, desenvolvida como resposta concreta às lacunas identificadas e orientada para as realidades e necessidades das comunidades universitárias.

3. MÉTODOS E FERRAMENTAS

3.1. Formalização do Problema Tecnológico

O problema tecnológico que este projeto abordou foi a fragmentação das ferramentas e recursos utilizados por estudantes e pequenas equipas de trabalho para a colaboração. Era comum que os utilizadores recorressem a várias aplicações independentes para realizar tarefas interligadas, como comunicação, gestão de tarefas, partilha de ficheiros e monitorização de prazos. Esta fragmentação conduzia a problemas como duplicação de esforço, perda de dados, dificuldades na organização e falhas na comunicação entre membros da equipa.

Durante a fase de levantamento de requisitos, foram identificadas as seguintes questões fundamentais para guiar o desenvolvimento da solução:

Como integrar funcionalidades dispersas numa única plataforma?

Garantir que a solução combinasse funcionalidades essenciais de forma intuitiva, acessível e eficiente.

De que forma a fragmentação afeta a produtividade e a organização das equipas?

Avaliar o impacto da utilização de várias ferramentas desconectadas no fluxo de trabalho, com base em experiências reais e feedback de utilizadores.

Quais foram os desafios técnicos enfrentados na criação de uma plataforma unificada?

Identificar barreiras como a integração de dados em tempo real, a escalabilidade da aplicação, a sincronização entre serviços e a garantia de segurança.

Como assegurar que a plataforma respondesse às necessidades específicas de estudantes e pequenas equipas?

Focar na usabilidade, simplicidade, acessibilidade e adaptabilidade ao contexto de trabalho académico e colaborativo.

3.2. Metodologia

A fase inicial da investigação foi conduzida através de pesquisa bibliográfica e documental. Foram analisados artigos científicos, relatórios técnicos e outras fontes relevantes que exploravam o problema da fragmentação de ferramentas digitais, os seus impactos em contextos académicos e as soluções propostas por diferentes autores. Este levantamento permitiu identificar padrões, boas práticas e requisitos técnicos essenciais para o desenvolvimento da plataforma.

Para o desenvolvimento da aplicação, foi adotada a metodologia ágil SCRUM, uma framework amplamente utilizada no desenvolvimento de software. Esta abordagem permitiu organizar o trabalho em sprints iterativos, com objetivos definidos e entregas regulares, garantindo a evolução contínua do projeto. Foram realizadas reuniões de acompanhamento (dailies) e sessões de revisão de sprint, o que possibilitou a deteção precoce de problemas e a incorporação de melhorias com base em feedback.

A adoção do SCRUM revelou-se particularmente eficaz para este projeto, dado o seu carácter académico e exploratório, permitindo flexibilidade, adaptação contínua e foco no utilizador final.

Ferramentas e Tecnologias

- **Visual Studio Code:** Ambiente de desenvolvimento integrado (IDE) para criação da solução.
- **Arquitetura MVC:** Para separar a interface, lógica de negócios e gestão de dados, promovendo organização e manutenção do código.
- **MongoDB:** Base de dados NoSQL para armazenamento escalável e flexível.
- **Firebase Real-time Database:** Para sincronização de dados em tempo real entre utilizadores.
- **Firebase Functions:** Para implementar lógica de back-end eficiente e baseada em eventos.

Técnicas de Validação e Testes: Para validar a eficácia da plataforma, foram aplicadas as seguintes técnicas de verificação e análise:

- **Testes funcionais:** Garantiram o correto funcionamento das principais funcionalidades implementadas, como criação de projetos, envio de mensagens, criação de tarefas e partilha de ficheiros.
- **Testes de usabilidade:** Realizados com utilizadores reais, permitiram recolher feedback sobre a interface, navegação e experiência geral da aplicação.
- **Análise quantitativa:** Foram recolhidos dados sobre o desempenho da plataforma e a percepção de impacto na organização e comunicação em ambiente académico, com especial foco em tempo economizado e clareza na gestão de tarefas.

4. DESENVOLVIMENTO E IMPLEMENTAÇÃO TÉCNICA

4.1. Arquitetura Geral

O WorkNest foi desenvolvido com base numa arquitetura web moderna do tipo **cliente-servidor**, que separa de forma clara a interface do utilizador (frontend) da lógica de negócio e persistência de dados (backend). O sistema foi concebido para ser escalável, modular e compatível com atualizações em tempo real.

Componentes principais:

- **Frontend:** Aplicação React que corre no navegador do utilizador e interage com o backend via HTTP e WebSockets.
- **Backend:** Servidor Node.js com Express, que expõe uma API REST e gere regras de negócio, autenticação, permissões e comunicação com a base de dados.
- **Base de Dados:**
 - **MongoDB Atlas** para dados estruturados (utilizadores, projetos, categorias, tópicos, tarefas, etc.).
 - **Firebase Firestore** para mensagens do chat em tempo real.
- **Comunicação em tempo real:** WebSockets (via Socket.IO) são utilizados para garantir sincronização imediata entre os membros do projeto.
- **Segurança:**
 - Autenticação via **JWT** com tokens guardados no frontend.
 - Encriptação ponto-a-ponto (E2EE) no chat por tópico, com chaves AES geradas e guardadas localmente.

Fluxo de comunicação:

1. O utilizador interage com a interface React (ex: cria um projeto).
2. A interface comunica com o backend via API.
3. O backend valida e processa a ação, atualiza a base de dados e, se necessário, emite eventos WebSocket.
4. Os utilizadores ligados ao mesmo projeto recebem atualizações em tempo real.
5. Em chats, as mensagens são encriptadas no frontend, enviadas para o Firestore e sincronizadas via WebSocket.

Esta arquitetura garante uma **experiência fluida, segura e colaborativa**, mesmo em ambientes com múltiplos utilizadores em simultâneo.

4.2. Backend – Tecnologias e Estrutura

O backend do WorkNest foi desenvolvido com recurso ao **Node.js** e ao framework **Express**, adotando uma arquitetura modular e escalável. A principal responsabilidade do backend é gerir a lógica de negócio, autenticação, permissões e persistência de dados, fornecendo uma **API RESTful** segura e estável para o frontend.

Tecnologias Utilizadas

- **Node.js**: Plataforma baseada em JavaScript utilizada para desenvolver o servidor backend.
- **Express.js**: Framework que facilita a criação de rotas, middleware e controladores.
- **MongoDB Atlas**: Base de dados NoSQL usada para armazenar os dados estruturais do sistema.
- **Mongoose**: Biblioteca ODM (Object Data Modeling) para modelar os dados MongoDB em JavaScript.
- **Socket.IO**: Biblioteca para WebSockets que permite a comunicação em tempo real entre cliente e servidor.
- **Firebase Firestore**: Utilizado exclusivamente para o armazenamento em tempo real das mensagens dos chats.
- **JSON Web Token (JWT)**: Utilizado para autenticação segura dos utilizadores.
- **bcryptjs + crypto (Node)**: Para hashing, verificação e gestão segura de passwords.

```
backend > package.json > ...
1  {
2    "dependencies": {
3      "axios": "^1.9.0",
4      "bcryptjs": "^3.0.2",
5      "compression": "^1.8.0",
6      "cors": "^2.8.5",
7      "dotenv": "^16.4.7",
8      "express": "^4.21.2",
9      "express-validator": "^7.2.1",
10     "gridfs-stream": "^1.1.1",
11     "helmet": "^8.0.0",
12     "joi": "^17.13.3",
13     "jsonwebtoken": "^9.0.2",
14     "mongoose": "^8.12.1",
15     "morgan": "^1.10.0",
16     "multer": "^1.4.5-lts.1",
17     "multer-gridfs-storage": "^5.0.2",
18     "nodemailer": "^6.10.0",
19     "socket.io": "^4.8.1"
20   },
21   "devDependencies": {
22     "dotenv-cli": "^8.0.0",
23     "jest": "^29.7.0",
24     "nodemon": "^3.1.9",
25     "supertest": "^7.0.0"
26   }
}
```

Figura 1 - Backend Package.json

Organização do Código Backend

O código foi organizado em diferentes pastas por responsabilidade, seguindo os princípios de separação de preocupações (SoC):

- */models/* : Contém os esquemas mongoose (User, Projeto, Categoria, Tópico, Tarefa, Mensagem, Pedido).
- */services/* : Camada com a lógica de negócio (CRUDs, validações, integrações, regras).
- */controllers/* : Recebem os pedidos HTTP, validam os inputs e delegam para os serviços.
- */middlewares/* : Incluem autenticação por JWT e verificação de permissões de acesso.
- */routes/* : Ficheiros de configuração das rotas REST associadas a cada recurso (projetos, utilizadores, etc.).

- `/utils/` : Helpers reutilizáveis, como validações de campos, geração de identificadores, formatações.

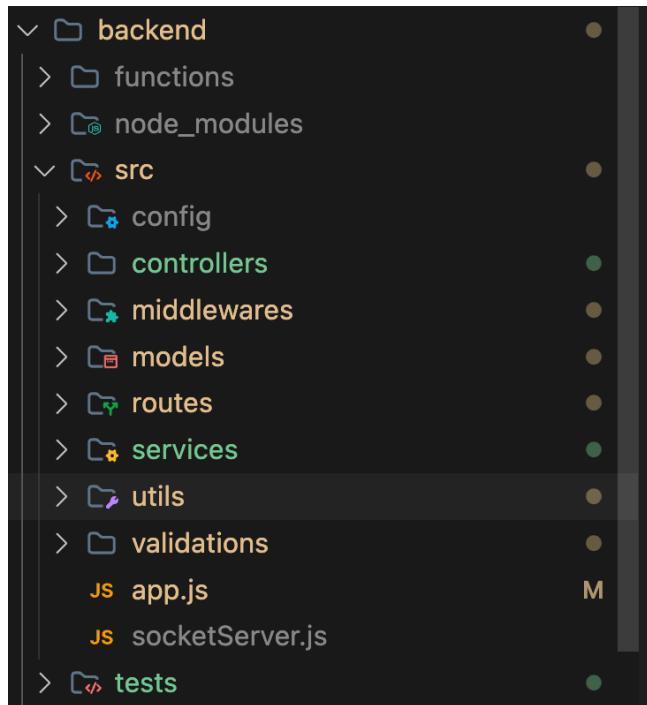


Figura 2 - Estrutura Backend Files

Exemplos de Endpoints REST

A API segue a convenção REST com uso de verbos HTTP:

- `POST /api/users/register` → Regista um novo utilizador.
- `POST /api/users/login` → Autentica um utilizador e devolve um token JWT.
- `GET /api/projetos/:id` → Obtém os dados completos de um projeto.
- `POST /api/categorias` → Cria uma nova categoria.
- `PATCH /api/topicos/:id` → Atualiza o nome ou estado de um tópico.
- `DELETE /api/tarefas/:id` → Elimina uma tarefa específica.
- `POST /api/pedidos` → Envia um pedido de amizade.
- `PATCH /api/projetos/transferir/:id` → Transfere a posse de um projeto para outro colaborador.

Autenticação e Segurança

- Após o login, o sistema gera um **JWT** que é enviado ao cliente e guardado localmente.
- Todos os endpoints protegidos utilizam o middleware authMiddleware que verifica a validade do token.
- Algumas rotas exigem permissões específicas (ex: apenas o owner pode eliminar projetos).
- As passwords são encriptadas com **bcrypt** no frontend e verificadas no backend antes de autenticar.

```
// Enviar novo pedido (amizade ou projeto)
router.post('/enviar', authMiddleware, PedidoController.enviarPedido);
```

Figura 3 - Exemplo de Route Protegida

Regras de Negócio no Backend

As principais **regras de negócio** são verificadas diretamente nos serviços:

- Verificar se um projeto tem categorias antes de permitir eliminação.
- Garantir que apenas o owner pode transferir propriedade.
- Validar se o nome de categorias/tópicos é único dentro do mesmo projeto.
- Impedir que se criem tarefas em tópicos arquivados.
- Verificar se o utilizador é colaborador antes de permitir acesso ao projeto

WebSocket (Socket.IO) no Backend

- O backend mantém uma lista de utilizadores ligados e os projetos em que estão.
- Quando uma alteração ocorre (ex: nova categoria), o backend emite um evento via WebSocket com os dados atualizados.
- O frontend escuta estes eventos e atualiza automaticamente o estado da aplicação sem reload.

4.3. Frontend – Tecnologias e Estrutura

O frontend do WorkNest foi desenvolvido como uma **SPA (Single Page Application)** utilizando **React.js**, com foco em modularidade, escalabilidade e experiência de utilizador moderna. A interface foi pensada para facilitar a organização colaborativa de projetos, com navegação fluida, estrutura intuitiva e resposta em tempo real a alterações no sistema.

Tecnologias Utilizadas

- **React.js**: Biblioteca principal para construção da interface.
- **Vite**: Ferramenta de bundling moderna, escolhida pela sua rapidez no desenvolvimento e compilação.
- **Context API + useReducer**: Gestão de estado global para dados como o utilizador autenticado, projeto atual, categorias e notificações.
- **Socket.IO Client**: Ligação WebSocket ao backend para sincronização em tempo real.
- **Web Crypto API (AES-GCM)**: Encriptação de mensagens diretamente no browser, antes do envio.
- **CSS Modules**: Estilização modular por componente, permitindo reutilização e isolamento de estilos.
- **LocalStorage**: Armazenamento local de tokens JWT, identificadores e chaves AES por tópico.

```
private _create,
"dependencies": {
  "@fontsource/inter": "^5.2.5",
  "@testing-library/dom": "^10.4.0",
  "@testing-library/jest-dom": "^6.6.3",
  "@testing-library/react": "^16.2.0",
  "@testing-library/user-event": "^13.5.0",
  "axios": "^1.8.4",
  "bcryptjs": "^3.0.2",
  "classnames": "^2.5.1",
  "crypto-js": "^4.2.0",
  "firebase": "^11.8.0",
  "framer-motion": "^12.5.0",
  "jwt-decode": "^4.0.0",
  "lottie-react": "^2.4.1",
  "lucide-react": "^0.487.0",
  "react": "^19.0.0",
  "react-calendar": "^6.0.0",
  "react-dom": "^19.0.0",
  "react-icons": "^5.5.0",
  "react-router-dom": "^7.4.0",
  "react-scripts": "5.0.1",
  "react-scroll": "^1.9.3",
  "recharts": "^2.15.3",
  "socket.io-client": "^4.8.1",
  "web-vitals": "^2.1.4"
}
```

Figura 4 - Frontend Package.json

Organização da Estrutura de Código

A aplicação está organizada de forma funcional, separando claramente componentes reutilizáveis, páginas principais e lógica de contexto:

- `/pages/`: Representa cada página principal do sistema. Ex:
 - `SignInPage.jsx`, `SignUpPage.jsx`, `HomePage.jsx`, `InsideProjectPage.jsx`
- `/components/`: Componentes visuais reutilizáveis e funcionais. Ex:
 - `Navbar`, `Sidebar`, `ChatBox`, `TaskModal`, `TopSelector`, `DropdownSugestoes`
- `/contexts/`: Define os contextos globais:
 - `AuthContext`, `ProjectContext`, `AlertContext`, `SocketContext`
- `/sockets/`: Contém os `socketHandlers.js` e eventos emitidos/recebidos.
- `/services/api.js`: Centraliza as chamadas REST ao backend com `axios`.
- `/styles/`: CSS modularizado por componente e por secção visual.
- `/utilis/`: Lógicas utilitárias como truncamento de texto, validações, encriptação local.

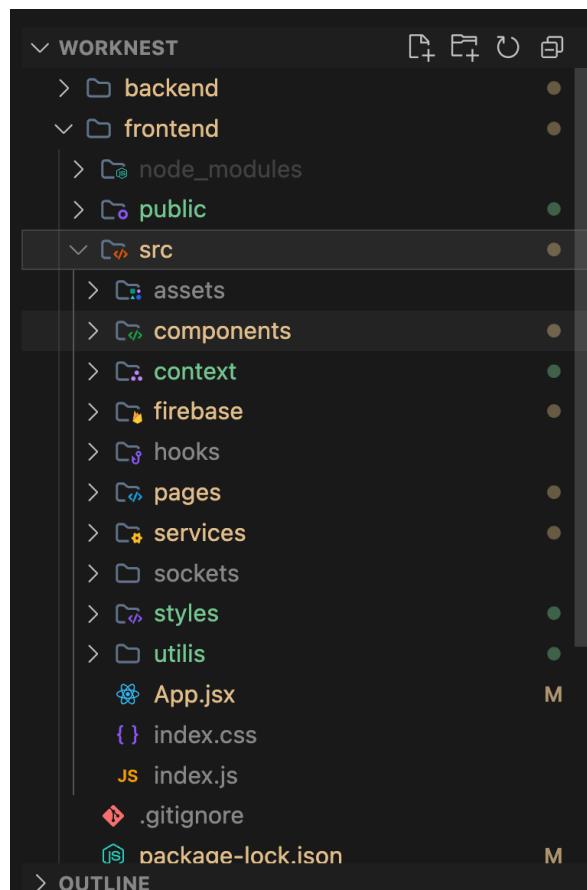


Figura 5 - Estrutura Frontend Files

Gestão de Estado

Utilizou-se React Context e useReducer para gerir o estado da aplicação de forma previsível e reativa:

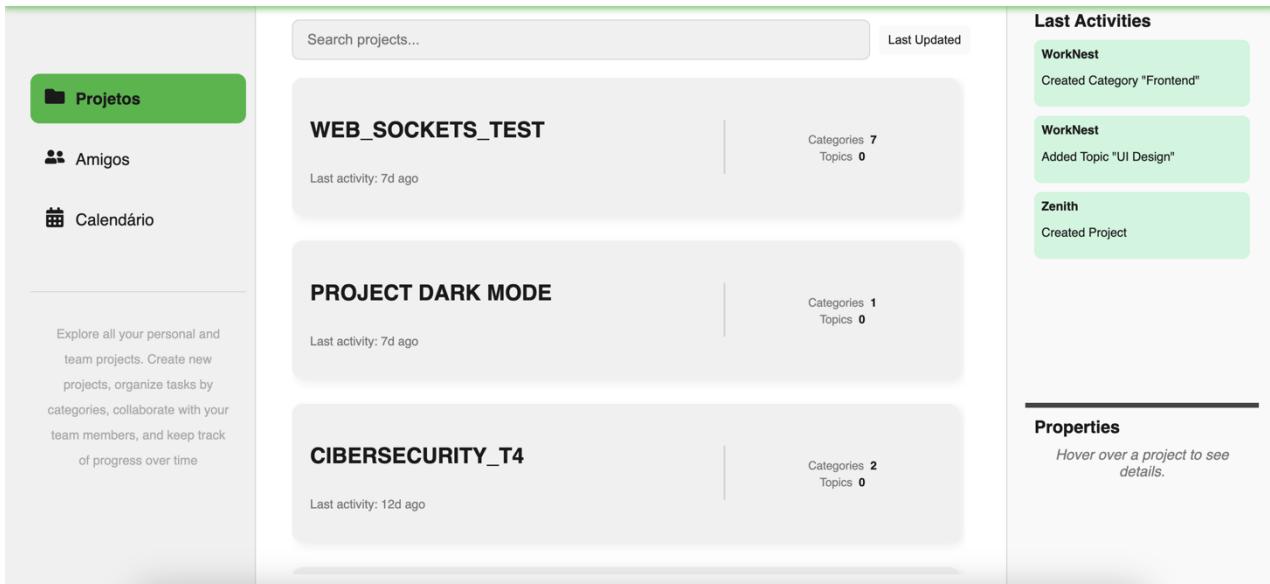
- AuthContext : Guarda o utilizador atual, o token JWT e permite login/logout.
- ProjectContext : Guarda o projeto atual, categorias, tópicos, tarefas.
- AlertContext : Controla o sistema global de alertas e confirmações (visuais e modais).

O estado é persistido entre sessões usando localStorage, incluindo o token JWT e as chaves de encriptação AES para cada tópico.

Funcionalidades Principais no Frontend

- Autenticação e Registo:
 - Páginas com validação visual e forte feedback de password.
 - Identificador gerado a partir dos últimos caracteres do `_id`.
- HomePage (painel principal):
 - Lista de projetos com botão de criação e cards dinâmicos.
 - Notificações visuais com contadores (bolinhas vermelhas).
 - Secção de amigos com pedidos recebidos e lista de contatos.
 - Calendário com eventos de tarefas e projetos.

WORKNEST



WORKNEST

Search projects...

Last Updated

WEB_SOCKETS_TEST
Last activity: 7d ago
Categories 7
Topics 0

PROJECT DARK MODE
Last activity: 7d ago
Categories 1
Topics 0

CIBERSECURITY_T4
Last activity: 12d ago
Categories 2
Topics 0

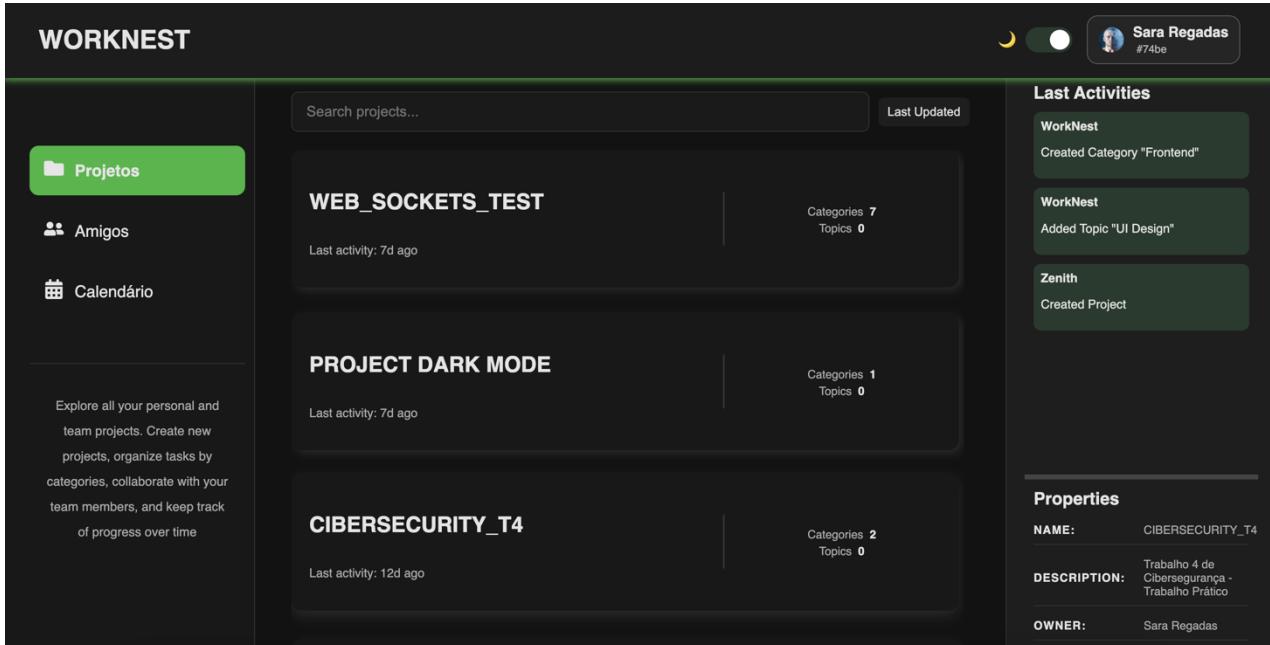
Last Activities

- WorkNest Created Category "Frontend"
- WorkNest Added Topic "UI Design"
- Zenith Created Project

Properties
Hover over a project to see details.

Sara Regadas #74be

Figura 6 - HomePage - LightMode



WORKNEST

Search projects...

Last Updated

WEB_SOCKETS_TEST
Last activity: 7d ago
Categories 7
Topics 0

PROJECT DARK MODE
Last activity: 7d ago
Categories 1
Topics 0

CIBERSECURITY_T4
Last activity: 12d ago
Categories 2
Topics 0

Last Activities

- WorkNest Created Category "Frontend"
- WorkNest Added Topic "UI Design"
- Zenith Created Project

Properties

NAME:	CIBERSECURITY_T4
DESCRIPTION:	Trabalho 4 de Cibersegurança - Trabalho Prático
OWNER:	Sara Regadas

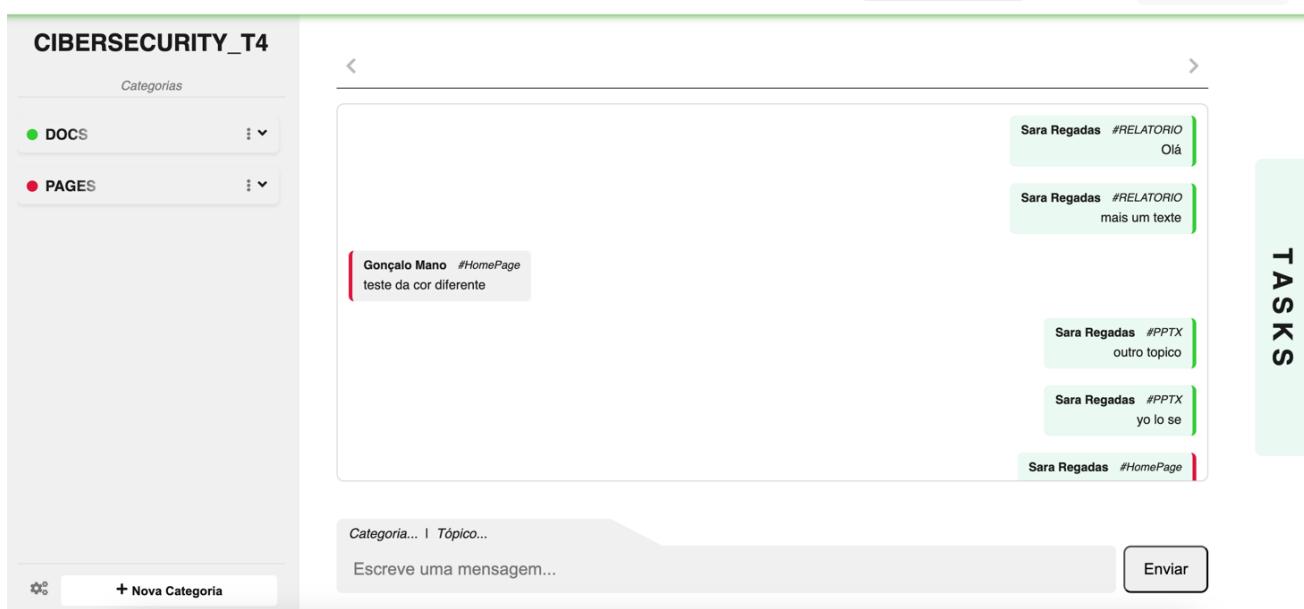
Sara Regadas #74be

Figura 7 - HomePage - DarkMode

- **InsideProjectPage:**

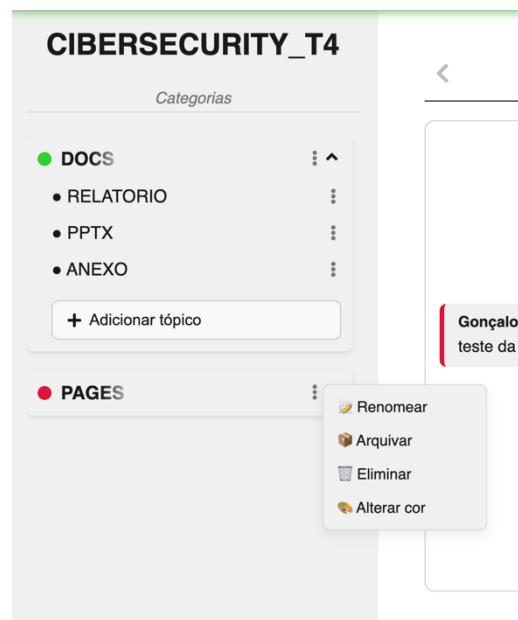
- Estrutura em três painéis:
 - Navbar superior com opções de logout, ajuda e colaboradores.
 - Sidebar interativa com categorias, tópicos e menus por item.
 - Área principal com chat, tarefas e tópicos visuais.
- Criação dinâmica de categorias e tópicos (com animações).
- Sistema de dropdowns de três pontos (renomear, arquivar, eliminar, mudar de cor (apenas em Categorias)).

WORKNEST



The screenshot shows the InsideProject interface. On the left, there's a sidebar titled 'CIBERSECURITY_T4' with sections for 'Categorias' (containing 'DOCS' and 'PAGES'), 'Nova Categoria', and a 'Categoria...' / 'Tópico...' input field. The main area displays a list of messages from users like Gonçalo Mano and Sara Regadas, each with a timestamp and a small preview of the content. A vertical bar on the right is labeled 'TASKS'. At the bottom, there's a message input field and an 'Enviar' button.

Figura 8 – InsideProject



This screenshot shows the 'DOCS' section of the sidebar. It lists items like 'RELATORIO', 'PPTX', and 'ANEXO', along with a 'Adicionar tópico' button. A context menu is open over one of the items, showing options: 'Renomear', 'Arquivar', 'Eliminar', and 'Alterar cor'.

Figura 9 - Opções nos Tópicos e Categorias

- **Chat por Tópico com E2EE:**

- ChatBox ligado ao tópico selecionado.
- Encriptação AES-GCM no frontend com chaves por tópico.
- Comandos: /CAT, @nome com sugestões e filtros visuais no topo do chat.
- Interface com suporte a mensagens, anexos e histórico.

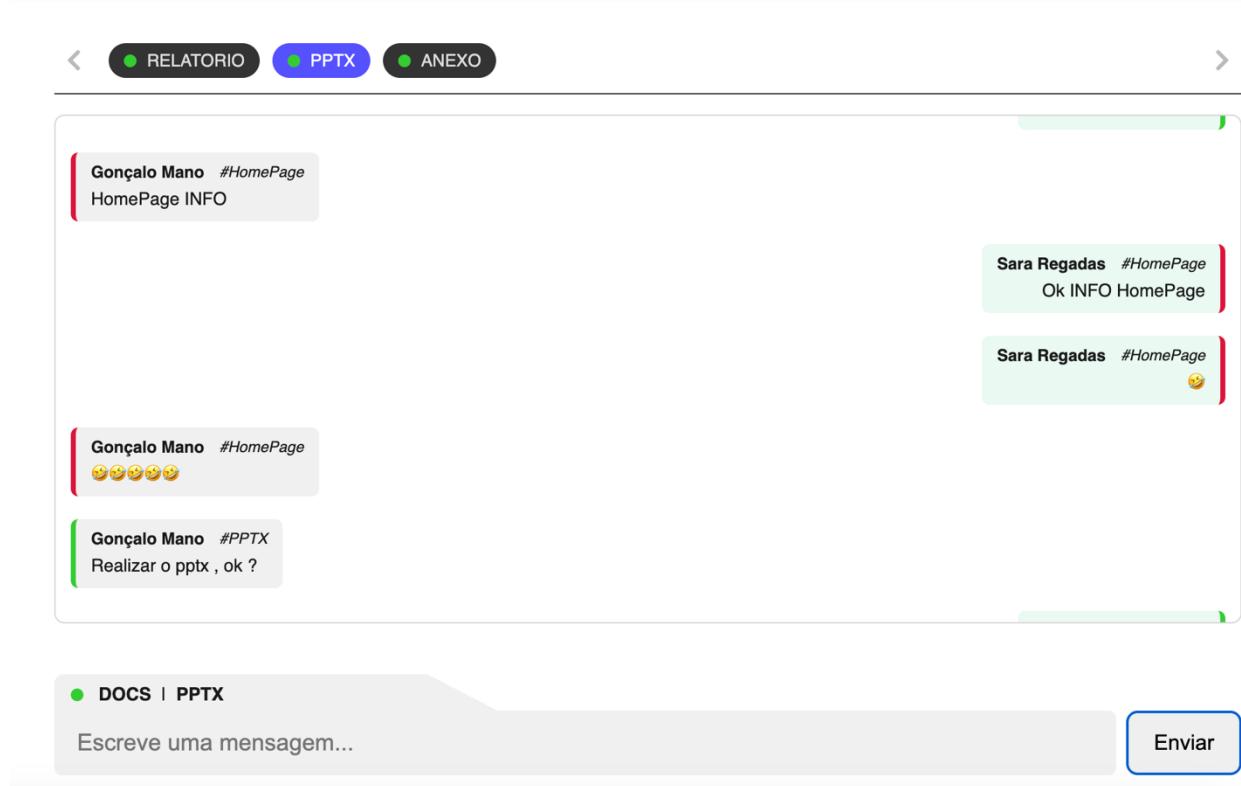


Figura 10 - ChatBox

- **Tarefas:**

- Modal lateral com visualização e criação de tarefas.
- Cada tarefa tem título, descrição, data-limite, responsável e estado.

- Vinhetas interativas e ordenação por estado.

TASKS , your project's to-do list, simplified

TÍTULO	CATEGORIA	TÓPICO	ENTREGA	STATUS
task123	● DOCS	#RELATORIO	04/06/2025	● Done
TAREFA1	● DOCS	#PPTX	10/06/2025	● Done
TASK12423	● DOCS	#PPTX	03/06/2025	● Done
TAREFA78	● DOCS	#PPTX	21/06/2025	● To Do
TASK4	● DOCS	#ANEXO	02/06/2025	● To Do
101	● DOCS	#ANEXO	26/06/2025	● To Do
TASK 2	● PAGES	#HomePage	04/06/2025	● To Do
TASK2	● PAGES	#HomePage	10/06/2025	● To Do

CHAT

Criar Tarefa

Figura 11 - Tarefas

- **WebSocket:**

- Ligação ativa ao backend com handlers organizados.
- Re却是ão e emissão de eventos como:
 - *projetoAtualizado, categoriaEliminada, topicoCriado, novaMensagem, etc.*
- Atualizações automáticas da UI sem reload (live sync).

```
⚠ Projeto atualizado via WebSocket: socketHandlers.js:23
  {_id: '6834543887e4286606aae9ac', nome: 'CIBERSECURITY_T4', d
  ►escricao: 'Trabalho 4 de Cibersegurança – Trabalho Prático',
  owner: {...}, listaUtilizadores: Array(1), ...}
```

Figura 12 - Atualização Websocket

Sistema de Alerta Global e Modais

- Sistema de AlertContext permite:
 - Alertas visuais temporários (✓ ou ✗ com barra de tempo).
 - Confirmações com botões “Sim” / “Não”.
 - Exibição centralizada e animada no topo da interface.
- Todos os modais da aplicação (criar projeto, adicionar amigo, ver colaboradores) seguem um padrão comum visual.

Design e Estilo

- Interface inspirada em sistemas modernos (ex: IDE, Notion, Figma).
- Animações suaves com CSS para menus, modais e input fields.
- Adaptação visual por secção (landing page diferente da página interna).
- Foco na clareza, feedback visual e simplicidade funcional.

4.4. Websockets e Atualização em Tempo Real

Uma das componentes-chave do WorkNest é a sua capacidade de **atualização em tempo real**, permitindo que múltiplos utilizadores vejam alterações nos seus projetos de forma instantânea e sincronizada, sem necessidade de recarregar a página. Esta funcionalidade é implementada através da biblioteca **Socket.IO**, que utiliza WebSockets como principal meio de comunicação entre cliente e servidor.

Objetivo da Integração de WebSockets

O objetivo principal é garantir que **qualquer alteração feita por um membro de um projeto seja refletida de imediato nos restantes membros** com acesso a esse projeto. Isto inclui:

- Criação, renomeação ou eliminação de categorias e tópicos.
- Arquivamento e desarquivamento de itens.
- Entrada ou saída de membros.
- Atualização da lista de tarefas.
- Recepção de mensagens de chat em tempo real.

Funcionamento Técnico

1. Ligação inicial

- Quando um utilizador acede ao projeto, o frontend inicia uma ligação com o servidor Socket.IO.
- É enviado o ID do projeto e do utilizador, que são registados numa estrutura interna do servidor (`socket.projectId`, `socket.userId`).

2. Emissão de eventos

Sempre que o backend realiza uma alteração relevante, emite um evento via Socket.IO:

Tabela 1 - Eventos Websocket

Evento	Disparado por...	Ação no frontend
categoriaCriada	Criação de nova categoria	Adiciona categoria à sidebar
topicoRenomeado	Renomeação de tópico	Atualiza o nome do tópico na interface
projetoAtualizado	Mudança no nome/descrição do projeto	Atualiza título do projeto
novaMensagem	Envio de nova mensagem encriptada	Exibe-a no chat em tempo real
pedidoAmizade	Envio de pedido	Adiciona pedido ao painel de notificações

3. Recepção e gestão de eventos

- No frontend, o ficheiro `socketHandlers.js` organiza todos os listeners de forma modular.
- Cada evento recebido é processado para atualizar diretamente o estado da aplicação, via `dispatch()` dos contextos correspondentes (ex: `ProjectContext` ou `AlertContext`).
- A ligação é **mantida ativa** durante toda a sessão do utilizador, e reconectada automaticamente em caso de falha.

Vantagens Práticas da Atualização em Tempo Real

- **Colaboração simultânea:** vários utilizadores podem criar tópicos ou tarefas sem pisar o trabalho uns dos outros.

- **Feedback instantâneo:** alterações feitas por um membro são visíveis para todos os outros sem recarregar a página.
- **Experiência fluida e moderna,** comparável a plataformas profissionais como Slack, Trello ou Notion.

Gestão de Reconexões e Desempenho

- O sistema deteta automaticamente se o utilizador perdeu a ligação e tenta reconectar.
- Durante reconexões, o estado é atualizado através de chamadas à API REST para garantir que não houve perdas de dados.
- O número de sockets ativos é controlado por projeto, evitando emissões desnecessárias.

5. CONCLUSÕES

O cenário digital contemporâneo, marcado pela intensificação do trabalho e da educação remota e colaborativa, tornou as ferramentas digitais indispensáveis, mas simultaneamente expôs um problema crescente: a **fadiga digital**, resultante da fragmentação informacional e do "over-tooling". A análise aprofundada da literatura e das plataformas existentes, como Jira, Asana, Slack, Microsoft Teams e Trello, revelou que, embora valiosas, carecem de integração funcional e adaptabilidade a contextos flexíveis e com recursos limitados, como o universitário.

Foi neste contexto que o projeto **WorkNest** estabeleceu como objetivo inicial o desenvolvimento de uma plataforma que **centralizasse as funcionalidades base essenciais** para a gestão de projetos e colaboração, visando mitigar a dispersão e a sobrecarga cognitiva experienciada pelos estudantes. A implementação do WorkNest seguiu uma metodologia Agile Scrum, permitindo um desenvolvimento incremental e a validação contínua com utilizadores reais.

Os resultados obtidos na fase exploratória confirmam o sucesso na concretização dos objetivos iniciais de ter as funcionalidades base implementadas e a sua pertinência. O WorkNest integra, com sucesso, funcionalidades cruciais como:

- Criação e gestão de projetos colaborativos, organizados hierarquicamente em categorias e tópicos.
- Gestão de tarefas com prazos, responsáveis e capacidade para anexos.
- Um sistema de mensagens vinculado aos tópicos para comunicação estruturada.
- Um calendário integrado para visualização de prazos.
- Suporte a anexos com encriptação AES por tópico, garantindo segurança.
- Funcionalidades de amizade, convites e estados online/offline, promovendo a interação social.

Os testes exploratórios com 32 participantes, que utilizaram a plataforma para simular projetos interdisciplinares, revelaram um **interesse genuíno na centralização funcional** proporcionada pelo WorkNest. Os 47 comentários espontâneos validam a percepção da **redução da necessidade de alternar entre múltiplas plataformas** e a valorização da clareza na estrutura organizacional. Com uma taxa de retorno de 81% e um envolvimento ativo de 69% dos utilizadores com as funcionalidades de tarefas e mensagens, estes dados demonstram que as funcionalidades base implementadas não só foram bem aceites, como também começam a cumprir o seu propósito de unificar e otimizar o fluxo de trabalho colaborativo.

Embora esta fase inicial, com as suas inerentes limitações exploratórias (amostra reduzida e feedback qualitativo primário), não permita conclusões definitivas sobre a total mitigação da fadiga digital, os resultados são encorajadores. Eles justificam a continuidade do desenvolvimento, com o alargamento da amostra de utilizadores, a realização de testes em contextos curriculares reais, e a eventual integração de métricas quantitativas e neurofisiológicas para uma avaliação mais aprofundada do impacto.

Em suma, o WorkNest não é apenas uma prova de conceito; é uma plataforma funcional que atingiu os seus objetivos iniciais de centralização e integração de ferramentas colaborativas. Representa um passo significativo na construção de soluções digitais mais humanas, coesas e eficientes para o ambiente universitário, demonstrando o seu potencial em combater a fragmentação e a fadiga digital através da simplificação e unificação do ecossistema de trabalho.

6. REFERÊNCIAS

- Arissa Shanganlall. (2025, April 29). *How Fragmented Systems Are Causing Digital Fatigue in Irish Universities*. <https://www.classter.com/blog/edtech/how-fragmented-systems-are-causing-digital-fatigue-in-irish-universities/>
- Arissa Shanganlall. (2025, April 29). *How Fragmented Systems Are Causing Digital Fatigue in Irish Universities*. https://www.classter.com/blog/edtech/how-fragmented-systems-are-causing-digital-fatigue-in-irish-universities/?utm_source=chatgpt.com
- Dahire, A., Dhamale, T., Dhanke, S., Gaikwad, S., & Parate, Ms. R. (2024). Team Tracker: A Project Management Tool. *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 08(10), 1–7. <https://doi.org/10.55041/IJSREM38265>
- Haiilo. (2025, March 6). *Digital Fatigue: How Fragmented Tools Are Hurting Your Team*. https://blog.haiilo.com/blog/digital-fatigue-how-fragmented-tools-are-hurting-your-team/?_gl=1*9ipjz2*_gcl_au*MTI2MTAwODgzOS4xNzQ5NDY2MTE2
- Ji, H., & Zhong, Z. (2023). Challenges and Countermeasures of Fragmented Learning to College Mathematics Teaching in the Era of Mobile Internet. *Advances in Educational Technology and Psychology*, 7(9). <https://doi.org/10.23977/aetp.2023.070905>
- Jon Darbyshire. (2025, April 28). *10 Best Project Management Tools & Software In 2025 [Reviewed]*. https://www.smartsuite.com/blog/project-management-software?105755a5_page=2
- Laker Benjamin. (2023). *How to fight digital fatigue at work*. <https://www.henley.ac.uk/news/2023/how-to-fight-digital-fatigue-at-work>
- Maican, C., Cazan, A. M., Lixandroiu, R., Dovleac, L., & Maican, M. A. (2021). Exploring the factors influencing the use of communication and collaboration applications. *Journal of Organizational and End User Computing*, 33(4), 1–27. <https://doi.org/10.4018/JOEUC.20210701.0a5>
- Monnin, C. (2020). Product review. *Journal of the Canadian Health Libraries Association*, 42(3), 129–129. <https://doi.org/10.29173/jchla29509>
- Ok, E. (2024). *Collaborative Learning Tools*. <https://www.researchgate.net/publication/386250907>
- Pawłowski, P., & Plechawska-Wójcik, M. (2022). *A comparative analysis of tools dedicated to project management Analiza porównawcza narzędzi dedykowanych zarządzaniu projektami*.
- Praburam Srinivasan. (2025, May 5). *15 Best Project Management Tools in 2025*. <https://clickup.com/blog/best-project-management-tools/>

- Que, C. G. L., Adonis, L. V. C., Casus, A. I. A., Hugo, A. J. D., Sino-ag, M. E. L., Tariman, J. M. O., & Angoluan, M. A. S. (2023). Effectiveness of Pomodoro Technique on Memory Retention among Psychology Students in a University in Quezon City, Philippines. *European Modern Studies Journal*, 7(5), 270–273. [https://doi.org/10.59573/emsj.7\(5\).2023.24](https://doi.org/10.59573/emsj.7(5).2023.24)
- Ripla Andre. (2025, February 19). *The “Digital Fatigue” Epidemic: How Over-Tooling Kills Employee Productivity*. <https://www.linkedin.com/pulse/digital-fatigue-epidemic-how-over-tooling-kills-ripla-pgcert-pgdip-mnwue>
- Stangl, F. J., Riedl, R., Weitzl, W. J., & Martin, S. (2024). Fatigue and Stress Levels in Digital Collaboration: A Pilot Study with Video Conferencing and the Metaverse. *Lecture Notes in Information Systems and Organisation*, 68, 89–103. https://doi.org/10.1007/978-3-031-58396-4_9
- Supriyadi, T., Sulistiasih, S., Rahmi, K. H., Pramono, B., & Fahrudin, A. (2025). The impact of digital fatigue on employee productivity and well-being: A scoping literature review. In *Environment and Social Psychology* (Vol. 10, Issue 2). Arts and Science Press Pte. Ltd. <https://doi.org/10.59429/esp.v10i2.3420>
- Torres, R., Laboissiere, R., Camargos, S., Nunes, C., Melo, M., & Liu, P. (n.d.). *GUIA PRÁTICO PARA USO DE PLATAFORMAS VIRTUAIS NO ENSINO REMOTO*.
- Vartika Kashyap. (2025, March 3). *15 Best Project Management Tools in 2025*. <https://www.proofhub.com/articles/top-project-management-tools-list>
- Yankah, J. E., Adjei, K. O., Bonney, S. O., Kotey, S., & Tieru, C. K. (2023). APPRAISAL OF MOBILE APPS FOR COMMUNICATION AND COLLABORATION AMONG CONSTRUCTION PROJECT TEAMS. *African Journal of Applied Research*, 9(2), 144–170. <https://doi.org/10.26437/ajar.31.10.2023.10>
- Zamiri, M., & Esmaeili, A. (2024). Methods and Technologies for Supporting Knowledge Sharing within Learning Communities: A Systematic Literature Review. In *Administrative Sciences* (Vol. 14, Issue 1). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/admisci14010017>



PARTE II – SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)

**WorkNest
Plataforma de Gestão e Colaboração**

**ISTEC Porto - Projeto
Diogo Lima Regadas – 2022103**

1. Introdução

1.1. Objetivos

O principal objetivo desta plataforma é proporcionar um ambiente colaborativo para estudantes e pequenos grupos de trabalho, facilitando a criação, organização e execução de projetos. Além disso, a plataforma procura integrar ferramentas de comunicação e organização em um único local, promovendo maior eficiência no trabalho em equipe. Com funcionalidades como calendário, gerenciamento de tarefas e mensagens em tópicos, o sistema visa melhorar a produtividade e reduzir os atritos comuns em projetos colaborativos.

1.2. Público-Alvo

Este documento destina-se principalmente à equipa de desenvolvimento responsável pela implementação da plataforma WorkNest. Também poderá ser consultado por membros da equipa de testes, analistas de requisitos e gestores de projeto, com o objetivo de compreender as funcionalidades, restrições e requisitos técnicos do sistema.

1.3. Âmbito do Projeto

A plataforma permitirá que os utilizadores:

- Criem e giram projetos.
- Adicionem categorias e tópicos para organização interna.
- Enviem mensagens e giram tarefas relacionadas a tópicos.
- Giram eventos e deadlines através de um calendário.

1.4. Organização do Documento

Este documento está estruturado para detalhar os aspectos essenciais da plataforma. Inicialmente, apresentamos os objetivos gerais e o público-alvo da solução, seguidos pela descrição geral das funcionalidades. Em seguida, detalhamos os requisitos funcionais e não funcionais em tabelas organizadas, além de diagramas que ilustram a arquitetura do sistema. Finalmente, abordamos considerações futuras para melhorias.

2. Descrição Geral

2.1. Perspetiva do Projeto

O **WorkNest** é um sistema de apoio à colaboração académica, concebido especificamente para contextos educativos como trabalhos de grupo universitários. O produto distingue-se por reunir num único ambiente várias funcionalidades que normalmente estariam dispersas por plataformas diferentes como Slack, Trello, Notion e Discord.

O sistema apresenta uma arquitetura **cliente-servidor** baseada em tecnologias web modernas:

- **Frontend em React**, responsável pela interface do utilizador e lógica interativa.
- **Backend em Node.js com Express**, que processa as regras de negócio, autenticação e integração com bases de dados.
- **Base de dados MongoDB** para dados estruturados (projetos, utilizadores, tarefas, etc.).
- **Firestore (Firebase)** para persistência de mensagens em tempo real com baixa latência.
- **WebSockets** para comunicação instantânea entre os utilizadores de um projeto.
- **Encriptação E2EE por tópico** usando Web Crypto API para garantir confidencialidade.

Este produto foi pensado para ser usado tanto em computadores pessoais como em tablets, funcionando unicamente em ambiente web, sem necessidade de instalação.

2.2. Funcionalidades do Produto

O WorkNest oferece um conjunto coeso e moderno de funcionalidades, entre as quais:

- **Criação e gestão de projetos:** os utilizadores podem criar projetos com nome e descrição, convidar amigos e gerir permissões de acesso.
- **Hierarquia lógica:** cada projeto contém categorias (semelhantes a pastas), e dentro destas existem tópicos. Cada tópico agrupa tarefas e mensagens.
- **Chat contextualizado:** os utilizadores podem enviar mensagens dentro de um tópico. O chat suporta comandos (@nome, /Cat) e apresenta sugestões dinâmicas.
- **Encriptação de mensagens (E2EE):** cada tópico possui uma chave AES única. As mensagens são encriptadas no frontend e só podem ser lidas por quem tiver a chave.

- **Gestão de tarefas:** os utilizadores podem criar tarefas com título, descrição, data de entrega, responsável e anexos.
- **Sistema de amizades:** os utilizadores podem procurar amigos, enviar pedidos de amizade, aceitar ou recusar, e ver amigos em comum.
- **Estado online/offline:** cada utilizador mostra um indicador em tempo real, visível no modal de colaboradores e no chat.
- **Gestão de categorias e tópicos:** é possível criar, renomear, arquivar e eliminar elementos. Arquivados podem ser geridos através de um modal centralizado.
- **Notificações visuais e pedidos:** sistema visual de alertas e bolinhas de notificação para eventos como convites ou novos pedidos.

2.3. Classe de Utilizadores

O sistema distingue claramente diferentes perfis de utilizadores:

- **Utilizador comum (estudante):** Regista-se na plataforma, cria ou entra em projetos, comunica com colegas, organiza o trabalho em tópicos e tarefas. Pode convidar amigos, alterar o seu perfil e sair de projetos onde não seja o proprietário.
- **Proprietário do projeto:** É o criador do projeto e tem controlo total sobre a sua estrutura e membros. Pode transferir a posse, remover colaboradores, arquivar conteúdos e apagar tópicos ou categorias sem dependências.
- **Colaborador externo:** É convidado para um projeto. Pode participar no chat, ver tarefas e categorias às quais tem acesso, mas não pode alterar elementos fora da sua permissão.
- **Administrador da plataforma:** É responsável pela gestão técnica da infraestrutura. Garante o bom funcionamento do sistema, segurança, backups e intervenções quando necessário.

2.4. Ambiente de Operação

O WorkNest funciona exclusivamente como uma aplicação web, suportada por navegadores modernos. O ambiente de operação exige:

Do lado do cliente:

- Navegadores modernos com suporte a WebSockets, Web Crypto API, LocalStorage e CSS moderno (Chrome, Firefox, Edge).

- Dispositivos desktop ou portáteis com resolução $\geq 1366 \times 768$ (responsividade suportada até tablets).

Do lado do servidor:

- Node.js (v18+), Express.js
- MongoDB Atlas (cloud) + Firestore (Firebase)
- Deploy em serviços como Vercel (frontend), Render ou Railway (backend)

O sistema está preparado para escalabilidade horizontal, garantindo performance em múltiplas sessões simultâneas.

2.5. Restrições Técnicas e de Implementação

Durante o desenvolvimento do WorkNest foram identificadas e assumidas algumas restrições:

- **Persistência híbrida:** o uso de duas bases de dados (MongoDB e Firestore) impõe sincronização lógica e segurança adicional.
- **Mensagens E2EE:** as chaves AES são armazenadas localmente; se perdidas, não há forma de recuperar as mensagens.
- **Validação offline:** por performance, algumas validações (nomes duplicados, campos vazios) são feitas localmente antes de contacto com o servidor.
- **Interface baseada em React:** o carregamento inicial pode ser superior a 1 segundo em ligações lentas, devido à natureza SPA (Single Page Application).

2.6. Documentação e Apoio ao Utilizador

Atualmente, o WorkNest inclui apenas elementos de apoio direto ao utilizador:

- Tooltips interativos nos botões e ícones
- Layout consistente e baseado em hierarquia visual
- Mensagens de erro e sucesso contextuais
- Sistema de alertas globais (topo do ecrã, com ícones e tempo de desaparecimento)

Está prevista para fases futuras a integração de:

- Tutoriais interativos passo a passo
- Página de FAQ e ajuda rápida
- Sistema de onboarding inicial para novos utilizadores

2.7. Pressupostos e Dependências

Os utilizadores terão acesso regular à Internet e dispositivos compatíveis com navegadores modernos.

O backend será executado num ambiente Node.js com ligação à base de dados MongoDB Atlas e Firebase.

A autenticação via Firebase deverá estar operacional para permitir login e registo.

A aplicação assume que os utilizadores compreendem conceitos básicos de colaboração digital, como categorias, mensagens e tarefas.

As chaves de encriptação devem ser armazenadas localmente com segurança pelo utilizado.

3. Stakeholders

O sistema WorkNest é direcionado para diferentes tipos de utilizadores, cada um com responsabilidades, objetivos e níveis de envolvimento distintos. Compreender estes stakeholders é essencial para garantir que os requisitos do sistema respondem às suas reais necessidades.

Estudantes Universitários

Os estudantes são os principais utilizadores e beneficiários da plataforma. Representam grupos de trabalho académico que usam o WorkNest para colaborar em projetos curriculares. A plataforma permite-lhes criar projetos, adicionar colegas, estruturar o conteúdo por categorias e tópicos, comunicar através de mensagens em tempo real e atribuir tarefas com prazos. Estes utilizadores valorizam uma interface intuitiva, rapidez na troca de mensagens e funcionalidades que facilitem a organização e o acompanhamento do progresso. Por exemplo, um grupo de três estudantes pode criar um projeto para uma disciplina, dividir o trabalho por tópicos e manter toda a comunicação dentro da aplicação.

Docentes ou Orientadores

Os docentes ou orientadores académicos podem utilizar o WorkNest como ferramenta de acompanhamento e supervisão dos grupos de trabalho. Apesar de não serem utilizadores tão ativos como os estudantes, podem ser convidados para aceder a projetos específicos, consultar o histórico de mensagens, verificar tarefas atribuídas e acompanhar o progresso geral de cada equipa. O seu interesse está na transparência do processo de trabalho e na facilidade de avaliação do desempenho colaborativo. Por exemplo, um docente pode consultar um projeto antes de uma entrega para perceber o envolvimento de cada elemento.

Membros Externos das Equipas

Nem todos os projetos são compostos apenas por colegas de turma. Os grupos podem incluir utilizadores externos, como estudantes de outras turmas ou cursos, colegas de outras instituições ou até elementos convidados de forma informal para colaborar. Estes utilizadores precisam de acesso prático e limitado às funcionalidades principais do projeto: ver tópicos, enviar mensagens, participar

nas tarefas e eventos. O sistema deve garantir uma gestão clara de permissões, para que estes stakeholders não accedam a conteúdos fora do âmbito do projeto em que participam.

Administrador da Plataforma

O administrador tem como função garantir o bom funcionamento técnico do sistema. Atua nos bastidores, com acesso privilegiado à infraestrutura do WorkNest: base de dados, autenticação, desempenho do servidor, logs de erro e estatísticas de utilização. Este stakeholder é responsável por manter o sistema estável, seguro e atualizado. Espera que o sistema seja modular, com boas práticas de segurança e fácil de manter. Por exemplo, poderá necessitar de intervir em situações de falha de login, erros em tempo real no chat ou abuso de funcionalidades.

Equipa Técnica / Desenvolvedores

A equipa que desenvolve e mantém o WorkNest é também um stakeholder importante. Tem interesse direto na evolução contínua da plataforma, na recolha de feedback dos utilizadores e na adaptação a novas necessidades ou tecnologias. Estes stakeholders interagem com o sistema em ambiente de desenvolvimento, testes e produção, e precisam que o código esteja organizado, documentado e testável. Para além disso, são responsáveis por integrar novas funcionalidades como notificações em tempo real, encriptação de mensagens e comandos interativos no chat.

4. Metodologias

Para este projeto foram utilizadas metodologias tanto de **investigação** como de **desenvolvimento**, com o objetivo de sustentar teoricamente o problema identificado e aplicar práticas modernas e eficazes na construção do sistema WorkNest.

4.1. Investigação

Durante a fase de investigação, foram consultadas diversas fontes académicas e técnicas sobre plataformas colaborativas, produtividade em grupo, sistemas de gestão de tarefas e metodologias ágeis. Esta pesquisa teve como finalidade contextualizar o problema da fragmentação digital enfrentado por estudantes e equipas de projeto, e fundamentar as soluções propostas.

A investigação incluiu:

Pesquisa Bibliográfica, focada em:

- Impacto da utilização de múltiplas ferramentas na produtividade das equipas.
- Boas práticas em aplicações colaborativas centradas no utilizador.
- Soluções de comunicação síncrona e assíncrona aplicadas a contextos académicos.

Estudos de Caso, analisando:

- Ferramentas populares como Trello, Microsoft Teams, Notion e Slack.
- Plataformas de código aberto e sistemas integrados utilizados em ambientes académicos e profissionais.
- Limitações de plataformas não especializadas para contextos educativos.

Estas fontes ajudaram a identificar padrões, práticas recomendadas e lacunas nas soluções existentes, o que orientou diretamente o levantamento de requisitos do WorkNest.

4.2. Desenvolvimento

Para a fase de desenvolvimento foi adotada a **metodologia ágil Scrum**, devido à sua flexibilidade e foco em entregas iterativas com base em feedback constante.

Scrum permitiu dividir o trabalho em **sprints** quinzenais, onde cada ciclo teve como objetivo o desenvolvimento e validação de funcionalidades específicas, desde a criação de contas de utilizador até à comunicação em tempo real com WebSockets.

O processo incluiu:

- **Planeamento de Sprints** com definição de prioridades e funcionalidades;
- **Daily Scrum informal**, com monitorização do progresso e ajustes;
- **Revisão de Sprint**, onde as funcionalidades eram testadas e ajustadas;
- **Retrospectiva**, onde foram identificadas melhorias no processo.

4.2.1. Aplicação da Metodologia ao WorkNest

Sprint 1 – Autenticação e Interface Base

- Criação das páginas Sign In e Sign Up
- Integração com sistema JWT e frontend React

Sprint 2 – Estrutura de Projetos

- CRUDs de projeto, categoria e tópico
- Primeiras interações com base de dados MongoDB

Sprint 3 – Comunicação e Colaboração

- Implementação do chat por tópicos
- Introdução de tarefas e sistema de estados

Sprint 4 – Funcionalidades Sociais

- Sistema de amigos, pedidos, notificações e colaboração

- Gestão de utilizadores por projeto

Sprint 5 – Melhorias visuais, comandos e segurança

- Encriptação E2EE, comandos interativos, estado online
- Otimização visual e modularização da lógica

5. Requisitos e Proposta de Solução

5.1. Requisitos Funcionais

Tabela 2 - Requisitos Funcionais

Identificador	RF.01 - CriarProjeto
Aplicação	WEB
Nome	Criar Projeto
Descrição	O utilizador cria um novo projeto, fornecendo nome, descrição e colaboradores iniciais.
Pré-Condições	O utilizador deve estar autenticado.
Pós-Condições	O projeto é adicionado à lista de projetos do utilizador e aparece para os colaboradores adicionados.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede à página de "Criar Projeto". 2. Clica no botão "Criar". 3. Insere o nome, descrição e, opcionalmente, seleciona colaboradores iniciais. 4. O sistema valida os dados e cria o projeto. 5. O projeto é exibido na lista de projetos do utilizador.
Req. não-funcionais	<ul style="list-style-type: none"> • Operação deve ser concluída em até 2 segundos. • Somente utilizadores autenticados podem criar projetos. • Validação para evitar duplicação de projetos com o mesmo nome dentro do mesmo utilizador. • Deve suportar centenas de projetos por utilizador sem perda de desempenho.

Identificador	RF.02 - AdicionarColaboradoresProjeto
Aplicação	WEB
Nome	Adicionar Colaboradores ao Projeto
Descrição	O criador do projeto adiciona novos colaboradores, atribuindo permissões e categorias específicas.
Pré-Condições	O projeto deve existir, e o utilizador deve ser o criador ou possuir permissões administrativas no projeto.
Pós-Condições	Os novos colaboradores podem visualizar e interagir no projeto conforme as permissões concedidas.
Percorso Normal	<ol style="list-style-type: none"> O utilizador acede às configurações do projeto. Seleciona a opção "Adicionar Colaboradores". Insere os emails dos colaboradores ou escolhe da lista de utilizadores existentes. Define as permissões para cada colaborador. Clica em "Adicionar". O sistema valida os dados e adiciona os colaboradores ao projeto.
Req. não-funcionais	<ul style="list-style-type: none"> Somente o dono do projeto ou administradores podem adicionar colaboradores. Deve haver um controle de permissões (visualizar, editar, excluir). O colaborador recebe uma notificação após ser adicionado ao projeto. A adição de colaboradores deve ocorrer em menos de 3 segundos. Suporte para centenas de colaboradores por projeto.

Identificador	RF.03 - CriarCategorias
Aplicação	WEB
Nome	Criar Categorias
Descrição	O utilizador cria uma categoria dentro de um projeto, fornecendo um nome e descrição.
Pré-Condições	O projeto deve existir, e o utilizador deve ter permissão no projeto.

Pós-Condições	A categoria é exibida como parte da organização do projeto.
Percorso Normal	<ol style="list-style-type: none"> 1. Utilizador acede a um projeto existente. 2. Navega para a seção de categorias. 3. Clica em "Adicionar Categoria". 4. Insere o nome e descrição da categoria. 5. Clica em "Guardar". 6. A nova categoria é exibida na organização do projeto.
Req. não-funcionais	<ul style="list-style-type: none"> • Apenas utilizadores com permissão podem criar categorias. • Deve haver validação para evitar categorias duplicadas dentro do mesmo projeto. • A criação da categoria deve ocorrer em até 1 segundo. • Deve exibir um feedback visual ao criar a categoria (ex.: animação, alerta de sucesso).

Identificador	RF.04 - CriarTopicos
Aplicação	WEB
Nome	Criar Tópicos
Descrição	O utilizador cria um tópico dentro de uma categoria, com título e descrição.
Pré-Condições	A categoria deve existir, e o utilizador deve ter acesso à categoria.
Pós-Condições	O tópico é exibido na lista de tópicos da categoria.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede a uma categoria dentro de um projeto. 2. Clica em "Criar Tópico". 3. Insere o título e descrição do tópico. 4. Clica em "Salvar". 5. O tópico aparece na lista de tópicos da categoria.
Req. não-funcionais	<ul style="list-style-type: none"> • Apenas utilizadores com permissão na categoria podem criar tópicos. • O tempo de criação do tópico não pode ultrapassar 2 segundos. • Deve permitir milhares de tópicos por categoria sem afetar a performance.

Identificador	RF.05 - EnviarMensagens
Aplicação	WEB

Nome	Enviar Mensagens
Descrição	O utilizador envia mensagens em um tópico.
Pré-Condições	O tópico deve existir, e o utilizador deve ter acesso ao tópico.
Pós-Condições	A mensagem aparece no fluxo de mensagens do tópico.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede a um tópico dentro de uma categoria. 2. Digita a mensagem no campo de entrada. 3. Opcionalmente, anexa arquivos à mensagem. 4. Clica no botão "Enviar". 5. A mensagem é exibida na lista de mensagens do tópico.
Req. não-funcionais	<ul style="list-style-type: none"> • Apenas membros do projeto podem enviar mensagens. • A mensagem deve ser entregue instantaneamente (menos de 1 segundo). • O sistema deve suportar grandes volumes de mensagens simultaneamente. • Deve exibir mensagens em tempo real sem necessidade de atualizar a página.

Identificador	RF.06 - EditarPerfil
Aplicação	WEB
Nome	Editar Perfil
Descrição	O utilizador atualiza informações como nome, email e foto de perfil.
Pré-Condições	O utilizador deve estar autenticado.
Pós-Condições	As informações atualizadas são refletidas imediatamente no perfil.
Percorso Normal	<ol style="list-style-type: none"> 1. utilizador acede à página de perfil. 2. Clica em "Editar Perfil". 3. Altera os campos desejados (nome, email, foto de perfil, etc.). 4. Clica em "Salvar". 5. sistema valida as alterações e atualiza o perfil.

Req. não-funcionais	<ul style="list-style-type: none"> • Apenas o próprio utilizador pode editar seus dados. • Senhas devem ser armazenadas de forma encriptada. • Deve permitir o upload de imagens de perfil e recorte da foto. • As alterações devem ser aplicadas em menos de 2 segundos.
---------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Identificador	RF.07 - GerirTarefas
Aplicação	WEB
Nome	Gerir Tarefas
Descrição	O utilizador cria e atualiza o status de tarefas associadas a tópicos.
Pré-Condições	O tópico deve existir, e o utilizador deve ter acesso ao tópico.
Pós-Condições	As tarefas são exibidas no respetivo tópico.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede a um tópico dentro de uma categoria. 2. Selecciona a opção "Criar Tarefa". 3. Insere o título, descrição, data de entrega e responsável. 4. Clica em "Salvar". 5. A tarefa é exibida na lista de tarefas do tópico.
Req. não-funcionais	<ul style="list-style-type: none"> • Apenas utilizadores com permissão podem criar, editar ou excluir tarefas. • As alterações de tarefas devem ser refletidas em até 2 segundos. • Deve suportar centenas de tarefas por tópico sem comprometer a performance.

Identificador	RF.08 - GerirEventos
Aplicação	WEB
Nome	Gerir Eventos
Descrição	O utilizador cria e edita eventos associados a projetos ou pessoais.
Pré-Condições	O utilizador deve estar autenticado.
Pós-Condições	O evento aparece no calendário associado ao utilizador ou ao projeto

Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede ao calendário. 2. Clica em "Adicionar Evento". 3. Insere o título, descrição, data e, opcionalmente, associa o evento a um projeto. 4. Clica em "Salvar". 5. O evento é exibido no calendário.
Req. não-funcionais	<ul style="list-style-type: none"> • Apenas o dono do evento ou administradores podem editá-lo. • Utilizadores devem receber notificações automáticas sobre novos eventos. • O evento deve ser visível instantaneamente após a criação.

Identificador	RF.09 - VisualizarCalendario
Aplicação	WEB
Nome	Visualizar Calendário
Descrição	O utilizador visualiza deadlines de tarefas e eventos associados a seus projetos.
Pré-Condições	O utilizador deve estar autenticado.
Pós-Condições	O calendário exibe todas as tarefas e eventos relacionados ao utilizador.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede a página do calendário. 2. O sistema exibe eventos e deadlines de tarefas associados ao utilizador. 3. O utilizador pode filtrar por projetos, datas ou tipo de evento.
Req. não-funcionais	<ul style="list-style-type: none"> • O calendário deve carregar em menos de 2 segundos. • Deve ter opções de filtro por data, projeto e tipo de evento. • O sistema deve suportar muitos eventos simultaneamente sem perder desempenho.

Identificador	RF.10 - EnviarNotificacoesCalendario
Aplicação	WEB
Nome	Enviar Notificações do Calendário
Descrição	O sistema envia notificações automáticas para informar eventos ou deadlines.
Pré-Condições	Notificações devem estar ativadas para o utilizador.
Pós-Condições	O utilizador recebe a notificação na sua plataforma.

Percorso Normal	<ol style="list-style-type: none"> 1. O sistema deteta eventos que necessitam de notificação (ex.: novo evento ou deadline iminente). 2. Prepara uma mensagem de notificação com os detalhes do evento. 3. Envia a notificação para o dispositivo configurado pelo utilizador.
Req. não-funcionais	<ul style="list-style-type: none"> 4. O utilizador é notificado em tempo real. • Notificações devem ser entregues em menos de 5 segundos. • Deve suportar muitos utilizadores recebendo notificações simultaneamente. • Notificações devem ter um link direto para o evento relacionado.

Identificador	RF.11 – Remover Categoria
Aplicação	WEB
Nome	Remover Categoria
Descrição	O utilizador remove uma categoria existente num projeto, desde que esta não contenha tópicos ativos.
Pré-Condições	O utilizador deve estar autenticado e ter permissões no projeto; a categoria não pode conter tópicos.
Pós-Condições	A categoria é eliminada da estrutura do projeto e deixa de estar visível no projeto.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede à sidebar do projeto. 2. Clica nos três pontos da categoria pretendida. 3. Selecciona a opção "Eliminar". 4. O sistema verifica se a categoria está vazia (sem tópicos). 5. Em caso afirmativo, mostra confirmação. 6. O utilizador confirma e a categoria é removida.
Req. não-funcionais	<ul style="list-style-type: none"> • A operação deve ser concluída em menos de 2 segundos. • Apenas utilizadores com permissão de edição podem remover categorias. • A remoção deve acionar um update via WebSocket para todos os membros do projeto.

Identificador	RF.12 – Remover Tópico
Aplicação	WEB
Nome	Remover Tópico
Descrição	O utilizador pode remover um tópico de uma categoria, desde que este não tenha mensagens ou tarefas associadas.
Pré-Condições	O utilizador deve ter acesso ao projeto e permissões na categoria; o tópico não pode conter dados.
Pós-Condições	O tópico é eliminado da respetiva categoria e deixa de ser listado.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede à categoria. 2. Clica nos três pontos do tópico a eliminar. 3. Selecciona "Eliminar". 4. O sistema verifica se não existem mensagens nem tarefas no tópico. 5. Caso positivo, mostra confirmação. 6. O utilizador confirma e o tópico é eliminado.
Req. não-funcionais	<ul style="list-style-type: none"> • O tempo de eliminação não deve ultrapassar 2 segundos. • A ação deve acionar um update em tempo real via WebSocket. • Apenas utilizadores com permissão na categoria podem executar esta ação.

Identificador	RF.13 – Renomear Categoria ou Tópico
Aplicação	WEB
Nome	Renomear Categoria ou Tópico
Descrição	Permite ao utilizador alterar o nome de uma categoria ou tópico, desde que tenha permissões para tal.
Pré-Condições	O utilizador deve ter permissão de edição sobre o item pretendido.
Pós-Condições	O novo nome é atualizado e refletido em toda a interface para todos os utilizadores.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede à sidebar do projeto. 2. Clica nos três pontos de um tópico ou categoria. 3. Selecciona "Renomear". 4. Um campo de input é exibido. 5. O utilizador insere o novo nome. 6. O sistema valida (não vazio e não duplicado) e aplica a mudança.

Req. não-funcionais	<ul style="list-style-type: none"> • Alterações devem ser aplicadas em menos de 1 segundo. • Validação automática local + envio para backend. • O novo nome deve ser emitido via WebSocket para sincronização imediata.
---------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Identificador	RF.14 – Arquivar/Desarquivar Categorias ou Tópicos
Aplicação	WEB
Nome	Arquivar ou Desarquivar Categorias e Tópicos
Descrição	Permite ao utilizador ocultar temporariamente categorias ou tópicos, sem os apagar, podendo ser restaurados mais tarde.
Pré-Condições	O utilizador deve ter permissão no projeto ou categoria.
Pós-Condições	A categoria ou tópico deixa de aparecer na visualização padrão e passa a estar acessível apenas no modo de gestão.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador clica nos três pontos da categoria ou tópico. 2. Seleciona "Arquivar". 3. O item é movido para o estado arquivado e ocultado da sidebar. 4. Através do modal de gestão, o utilizador pode visualizar os arquivados. 5. Ao clicar em "Desarquivar", o item volta ao estado ativo.
Req. não-funcionais	<ul style="list-style-type: none"> • Alteração de estado deve ocorrer instantaneamente. • Deve ser possível filtrar apenas elementos arquivados. • O sistema deve emitir atualização por WebSocket.

Identificador	RF.15 – Gerir Estado Online dos Utilizadores
Aplicação	WEB
Nome	Gerir Estado Online
Descrição	O sistema identifica e atualiza em tempo real o estado online/offline dos utilizadores nos projetos.
Pré-Condições	O utilizador deve estar autenticado e com sessão ativa.

Pós-Condições	A presença online é refletida nos avatares e nos menus de seleção de utilizadores.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador entra no projeto. 2. O sistema regista a sua presença como "online". 3. Quando o utilizador fecha ou perde conexão, o estado muda para "offline". 4. O estado é atualizado em tempo real para todos os participantes dos projetos.
Req. não-funcionais	<ul style="list-style-type: none"> • Detecção e emissão de estado deve ocorrer em menos de 3 segundos. • O sistema deve ser capaz de escalar para centenas de utilizadores simultâneos. • Integração com WebSocket para notificações de presença.

Identificador	RF.16 – Enviar Pedido de Amizade
Aplicação	WEB
Nome	Enviar Pedido de Amizade
Descrição	Permite a um utilizador procurar outro e enviar-lhe um pedido de amizade com base no nome ou identificador único.
Pré-Condições	O utilizador deve estar autenticado.
Pós-Condições	O pedido de amizade fica pendente e visível para o destinatário aceitar ou recusar.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador abre o modal "Adicionar Amigo". 2. Pesquisa o utilizador pelo nome ou identificador. 3. Selecciona o utilizador e clica em "Enviar Pedido". 4. O sistema valida e regista o pedido como pendente.
Req. não-funcionais	<ul style="list-style-type: none"> • Os resultados da pesquisa devem ser apresentados em tempo real. • O pedido deve ser enviado em menos de 2 segundos.

Identificador	RF.17 – Aceitar ou Recusar Pedido de Amizade
Aplicação	WEB

Nome	Aceitar ou Recusar Pedido de Amizade
Descrição	O utilizador pode visualizar, aceitar ou recusar pedidos de amizade recebidos através do painel de notificações.
Pré-Condições	O utilizador deve estar autenticado e ter pedidos pendentes.
Pós-Condições	O pedido é movido para o estado "aceite" (amigos) ou "recusado" (removido).
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede ao painel de pedidos de amizade. 2. Visualiza a lista de pedidos pendentes. 3. Clica em "Aceitar" ou "Recusar". 4. O sistema regista a ação e atualiza o estado do pedido. 5. A interface é atualizada e o número de amigos é ajustado.
Req. não-funcionais	<ul style="list-style-type: none"> • A resposta deve ser imediata (<1 segundo). • O utilizador deve receber feedback visual da aceitação/recusa.

Identificador	RF.18 – Gerir Lista de Amigos
Aplicação	WEB
Nome	Gerir Lista de Amigos
Descrição	Permite ao utilizador visualizar, pesquisar e remover amigos adicionados.
Pré-Condições	O utilizador deve ter amigos adicionados.
Pós-Condições	A lista de amigos é atualizada de acordo com as ações do utilizador.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede à secção "Amigos". 2. Visualiza a lista atual de amigos com nome, foto e identificador. 3. Pode pesquisar por nome. 4. Pode clicar em "Remover" para eliminar um amigo.
Req. não-funcionais	<ul style="list-style-type: none"> • Pesquisa e filtragem devem ser instantâneas. • As ações de remoção devem ser confirmadas. • A lista deve ser atualizada automaticamente.

Identificador	RF.19 – Convidar Amigos para Projetos
Aplicação	WEB
Nome	Convidar Amigos para Projetos
Descrição	Permite ao utilizador convidar amigos da sua lista para colaborar num projeto em que tenha permissões.
Pré-Condições	O utilizador deve ser membro do projeto e ter amigos na sua lista.
Pós-Condições	O convite é enviado e aparece como pendente até ser aceite ou recusado.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador entra no modal de colaboradores. 2. Selecciona um amigo da lista apresentada à direita. 3. Clica em "Convidar". 4. O sistema envia o convite, visível na secção de notificações do destinatário.
Req. não-funcionais	<ul style="list-style-type: none"> • O convite deve ser enviado em tempo real. • A UI deve mostrar visualmente o estado "pendente".

Identificador	RF.20 – Transferir Propriedade de Projeto
Aplicação	WEB
Nome	Transferir Propriedade de Projeto
Descrição	O utilizador que é proprietário de um projeto pode transferir a propriedade para outro colaborador.
Pré-Condições	O utilizador deve ser o proprietário do projeto e o destino da transferência deve já estar adicionado ao projeto.
Pós-Condições	O novo proprietário assume todas as permissões e o anterior passa a colaborador.
Percorso Normal	<ol style="list-style-type: none"> 1. O proprietário acede ao modal de colaboradores. 2. Visualiza a lista de utilizadores com uma estrela junto a cada um. 3. Clica na estrela de um colaborador para transferir a propriedade. 4. O sistema pede confirmação. 5. Ao confirmar, o projeto é atualizado com o novo owner.

Req. não-funcionais	<ul style="list-style-type: none"> A transição deve ocorrer em menos de 3 segundos. Só o proprietário pode executar esta ação.
---------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------

Identificador	RF.21 – Sair de Projeto
Aplicação	WEB
Nome	Sair de Projeto
Descrição	Um utilizador pode sair de um projeto, exceto se for o proprietário, caso em que terá de transferir a posse primeiro.
Pré-Condições	O utilizador deve ser colaborador do projeto.
Pós-Condições	O utilizador é removido da lista de participantes e perde acesso ao conteúdo do projeto.
Percorso Normal	<ol style="list-style-type: none"> O utilizador abre o modal de colaboradores. Clica em "Sair do Projeto". O sistema verifica se não é o proprietário. Se for, solicita transferência de posse antes de permitir saída. Se for colaborador, confirma saída e remove o utilizador.
Req. não-funcionais	<ul style="list-style-type: none"> A operação deve ser instantânea. A UI deve atualizar a lista de participantes em tempo real. O utilizador deixa de receber notificações relativas ao projeto.

Identificador	RF.22 – Usar Chat com Troca de Tópicos
Aplicação	WEB
Nome	Usar Chat com Troca de Tópicos
Descrição	O chat permite aos utilizadores enviar e visualizar mensagens de diferentes tópicos organizados por categoria.
Pré-Condições	O utilizador deve estar autenticado e dentro de um projeto.
Pós-Condições	As mensagens são guardadas por tópico e ficam acessíveis aos membros do projeto.
Percorso Normal	<ol style="list-style-type: none"> O utilizador entra num projeto. A interface do chat mostra mensagens agrupadas por tópico. O utilizador pode alternar entre tópicos clicando ou através

	<p>do comando /CAT.</p> <ol style="list-style-type: none"> 4. O chat atualiza dinamicamente o histórico de mensagens. 5. O envio de mensagens é ligado ao tópico ativo.
Req. não-funcionais	<ul style="list-style-type: none"> • As mensagens devem aparecer em tempo real (delay < 1s). • A interface deve indicar claramente o tópico ativo. • A navegação entre tópicos deve ser fluida e visualmente destacada.

Identificador	RF.23 – Enviar Mensagens Encriptadas por Tópico
Aplicação	WEB
Nome	Enviar Mensagens Encriptadas por Tópico
Descrição	As mensagens enviadas no chat são encriptadas no frontend com uma chave AES única por tópico, garantindo segurança ponto-a-ponto (E2EE).
Pré-Condições	O utilizador deve estar autenticado e com acesso à chave de encriptação do tópico atual.
Pós-Condições	A mensagem é armazenada no backend encriptada e é desencriptada apenas por utilizadores autorizados.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador entra no projeto e seleciona um tópico. 2. O sistema obtém localmente a chave AES associada ao tópico. 3. O utilizador escreve e envia a mensagem. 4. A mensagem é encriptada no frontend. 5. É enviada para o backend (MongoDB/Firebase) encriptada. 6. Os outros utilizadores desencriptam-na com a mesma chave local.
Req. não-funcionais	<ul style="list-style-type: none"> • A encriptação e desencriptação devem ser transparentes ao utilizador. • A chave AES nunca é transmitida pela rede. • O tempo de encriptação não deve ultrapassar 0,5 segundos.

Identificador	RF.24 – Usar Comandos no Chat (@ e /)
Aplicação	WEB

Nome	Usar Comandos no Chat (@ e /)
Descrição	O input do chat suporta comandos como @nome para menções e /CAT ou /Tópico para mudar rapidamente de contexto.
Pré-Condições	O utilizador deve estar autenticado e num projeto.
Pós-Condições	O comando é interpretado e executado, alterando a interface ou a funcionalidade do chat.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador escreve @ ou / no campo de mensagem. 2. O sistema mostra um dropdown de sugestões (utilizadores ou categorias/tópicos). 3. O utilizador seleciona uma opção com teclado ou rato. 4. O chat interpreta o comando: 5. @ envia notificação à pessoa mencionada. 6. /CAT muda o foco do chat para a categoria/tópico selecionado.
Req. não-funcionais	<ul style="list-style-type: none"> • O dropdown deve aparecer em menos de 200 ms. • A execução do comando deve ocorrer imediatamente. • A UI deve indicar visualmente o resultado da ação (nova seleção, menção, etc.).

Identificador	RF.25 – Remover Categoria
Aplicação	WEB
Nome	Remover Categoria
Descrição	O utilizador remove uma categoria existente num projeto, desde que esta não contenha tópicos ativos.
Pré-Condições	O utilizador deve estar autenticado e ter permissões no projeto; a categoria não pode conter tópicos.
Pós-Condições	A categoria é eliminada da estrutura do projeto e deixa de estar visível no projeto.
Percorso Normal	<ol style="list-style-type: none"> 1. O utilizador acede à sidebar do projeto. 2. Clica nos três pontos da categoria pretendida. 3. Selecciona a opção "Eliminar". 4. O sistema verifica se a categoria está vazia (sem tópicos). 5. Em caso afirmativo, mostra confirmação. 6. O utilizador confirma e a categoria é removida.

Req. não-funcionais	<ul style="list-style-type: none"> • A operação deve ser concluída em menos de 2 segundos. • Apenas utilizadores com permissão de edição podem remover categorias. • A remoção deve acionar um update via WebSocket para todos os membros do projeto.
----------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.2. Requisitos Não-Funcionais

Usabilidade

A interface deve ser moderna e de fácil utilização, proporcionando uma navegação intuitiva para utilizadores com diferentes níveis de experiência.

Fiabilidade

O sistema deve garantir um tempo de atividade elevado, reduzindo falhas e interrupções para assegurar a continuidade das atividades dos utilizadores.

Desempenho

Todas as interações da plataforma, como carregar páginas, enviar mensagens e salvar tarefas, devem ser concluídas em poucos segundos para garantir uma experiência fluida.

Segurança e Privacidade

As informações armazenadas devem ser protegidas contra acessos não autorizados, garantindo que apenas utilizadores autorizados tenham permissão para visualizar ou modificar dados sensíveis.

Manutenção

O sistema deve ser projetado para facilitar atualizações sem necessidade de interrupção, garantindo que novas funcionalidades possam ser adicionadas sem impactar a disponibilidade.

5.3. Requisitos de Software

Esta secção descreve os requisitos técnicos necessários para o correto funcionamento do sistema WorkNest, incluindo as configurações recomendadas para o ambiente de desenvolvimento, servidor e cliente. São considerados os requisitos do lado do utilizador (frontend), do lado do servidor (backend) e da base de dados.

5.3.1. Ambiente do Cliente (Frontend)

O sistema foi desenvolvido para funcionar em navegadores modernos e dispositivos com acesso à Internet.

Requisitos mínimos:

- Navegador compatível: Google Chrome (v100+), Mozilla Firefox (v95+), Microsoft Edge (v100+)
- Resolução mínima recomendada: 1366x768
- JavaScript ativado
- Suporte a WebSockets
- Suporte a LocalStorage e Web Crypto API

Tecnologias utilizadas:

- **React.js** com Vite como bundler
- **CSS Modules** + Tailwind CSS (animações e responsividade)
- WebSocket client (socket.io-client)
- Criptografia com Web Crypto API (AES-GCM por tópico)
- Armazenamento local temporário de chaves de encriptação

5.3.2. Requisitos do Servidor (Backend)

O servidor da aplicação lida com autenticação, lógica de negócio, persistência e comunicação em tempo real.

Requisitos mínimos:

- Node.js 18 ou superior
- Sistema operativo: Ubuntu 20.04 ou superior / Windows 10 / macOS 11+
- MongoDB 6.0+
- Firebase (Firestore e Authentication) para integração híbrida

Tecnologias utilizadas:

- **Node.js** com Express
- **MongoDB** com Mongoose (ORM)
- **JWT** para autenticação segura
- **Socket.io** para comunicação em tempo real
- **Firebase Admin SDK** para integração com Firestore
- **dotenv** para gestão de variáveis de ambiente

5.3.3. Requisitos para a Base de Dados

O sistema utiliza duas soluções em conjunto: MongoDB para dados estruturados e Firestore para mensagens em tempo real.

MongoDB (dados principais):

- Armazena: utilizadores, projetos, categorias, tópicos, tarefas, pedidos de amizade
- Índices criados para performance em buscas por ID e relacionamentos

Firebase Firestore (mensagens de chat):

- Armazena: mensagens organizadas por tópico
- Comunicação assíncrona e escalável
- Permite atualizações em tempo real com latência mínima

Considerações de Segurança:

- Mensagens encriptadas no frontend (E2EE)
- Tokens JWT com expiração e validação por middleware
- Validação de permissões para cada ação (criar, editar, eliminar)

5.3.4. Configurações Recomendadas para Desenvolvimento

Editor: Visual Studio Code com extensões ESLint, Prettier e REST Client

Sistema de controlo de versões: Git + GitHub

Ambiente isolado com .env para configuração local

Uso de nodemon e concurrently para desenvolvimento em tempo real

5.3.5. Requisitos de Deploy (produção)

Serviço de alojamento: Render, Vercel, Railway, Heroku ou servidor VPS

Build do frontend: npm run build (Vite)

Build do backend: Servidor Express + instância MongoDB online (Mongo Atlas)

Domínio personalizado: recomendado para publicação

5.4. Tabela de Rastreabilidade

Tabela 3 - Tabela de Rastreabilidade

ID do Requisito	Tipo	Nome	Casos de Uso Relacionados	Módulo/Função
RF.01	Funcional	Criar Projeto	CU.01	Projetos
RF.02	Funcional	Adicionar Colaboradores	CU.02, CU.11	Projetos / Colaboradores
RF.03	Funcional	Criar Categorias	CU.03	Sidebar / Categorias
RF.04	Funcional	Criar Tópicos	CU.04	Sidebar / Tópicos
RF.05	Funcional	Enviar Mensagens	CU.05, CU.15	Chat
RF.06	Funcional	Enviar Anexos	CU.05, CU.18	Chat / Tarefas
RF.07	Funcional	Gerir Tarefas	CU.06, CU.18	Tarefas
RF.08	Funcional	Ver Estado Online	CU.07	Presença / Utilizadores
RF.09	Funcional	Enviar Pedido de Amizade	CU.08	Amigos
RF.10	Funcional	Aceitar Pedido de Amizade	CU.09	Amigos
RF.11	Funcional	Gerir Lista de Amigos	CU.10	Amigos
RF.12	Funcional	Convidar Amigos p/ Projeto	CU.11	Projetos / Amigos

ID do Requisito	Tipo	Nome	Casos de Uso Relacionados	Módulo/Função
RF.13	Funcional	Sair de Projeto	CU.12	Projetos
RF.14	Funcional	Transferir Propriedade	CU.13	Projetos / Colaboradores
RF.15	Funcional	Arquivar/Desarquivar Itens	CU.14	Sidebar / Gestão
RF.16	Funcional	Estado Online/Offline	CU.07	Presença / WebSocket
RF.17	Funcional	Sistema de Notificações	CU.16	Notificações
RF.18	Funcional	Ver Calendário de Tarefas	CU.17	Tarefas / Calendário
RF.19	Funcional	Ver e Gerir Tarefas	CU.18	Tarefas / Tópicos
RF.20	Funcional	Suporte Offline Parcial	CU.19	Frontend / LocalStorage
RF.21	Funcional	Ajuda / Onboarding	CU.20	UI / Onboarding

5.5. Tabela de Prioridade de Requisitos Funcionais (RF)

Tabela 4 - Tabela de Prioridade de Requisitos Funcionais

ID	Nome do Requisito	Prioridade	Justificação
RF.01	Criar Projeto	● Alta	Funcionalidade central da plataforma.
RF.02	Adicionar Colaboradores	● Alta	Essencial para permitir trabalho em equipa.
RF.03	Criar Categorias	● Alta	Necessário para organização da estrutura do projeto.
RF.04	Criar Tópicos	● Alta	Elemento base para mensagens e tarefas.
RF.05	Enviar Mensagens	● Alta	Comunicação é parte crítica da colaboração.
RF.06	Enviar Anexos	● Média	Útil, mas não essencial para o MVP.

ID	Nome do Requisito	Prioridade	Justificação
RF.07	Gerir Tarefas	● Alta	Planeamento e divisão de trabalho dependem desta função.
RF.08	Ver Estado Online	● Média	Aumenta a percepção de presença, mas não é funcionalidade crítica.
RF.09	Enviar Pedido de Amizade	● Média	Facilita a conexão entre utilizadores.
RF.10	Aceitar Pedido de Amizade	● Média	Complementar ao RF.09.
RF.11	Gerir Lista de Amigos	● Média	Essencial para manter rede de contactos, mas não central ao funcionamento.
RF.12	Convidar Amigos para Projeto	● Alta	Ligaçao direta entre social e colaboração.
RF.13	Sair de Projeto	● Média	Garante autonomia do utilizador, mas não afeta funcionalidades principais.
RF.14	Transferir Propriedade de Projeto	● Média	Importante para continuidade, mas pode ser gerida manualmente no início.
RF.15	Arquivar/Desarquivar Itens	● Média	Ajuda na organização, mas pode ser adicionado posteriormente.
RF.16	Estado Online/Offline	● Média	Informativo, mas não afeta funcionalidades essenciais.
RF.17	Sistema de Notificações	● Média	Melhora a experiência do utilizador.
RF.18	Ver Calendário de Tarefas	● Média	Ajuda visual adicional, mas não indispensável.
RF.19	Ver e Gerir Tarefas	● Alta	Complementa o RF.07; faz parte da gestão diária.
RF.20	Suporte Offline Parcial	● Baixa	Funcionalidade avançada, ideal para futuras versões.

ID	Nome do Requisito	Prioridade	Justificação
RF.21	Ajuda / Onboarding	● Baixa	Útil para iniciantes, mas não essencial ao funcionamento.

6. Modelos do Sistema

6.1. Diagrama de Classes

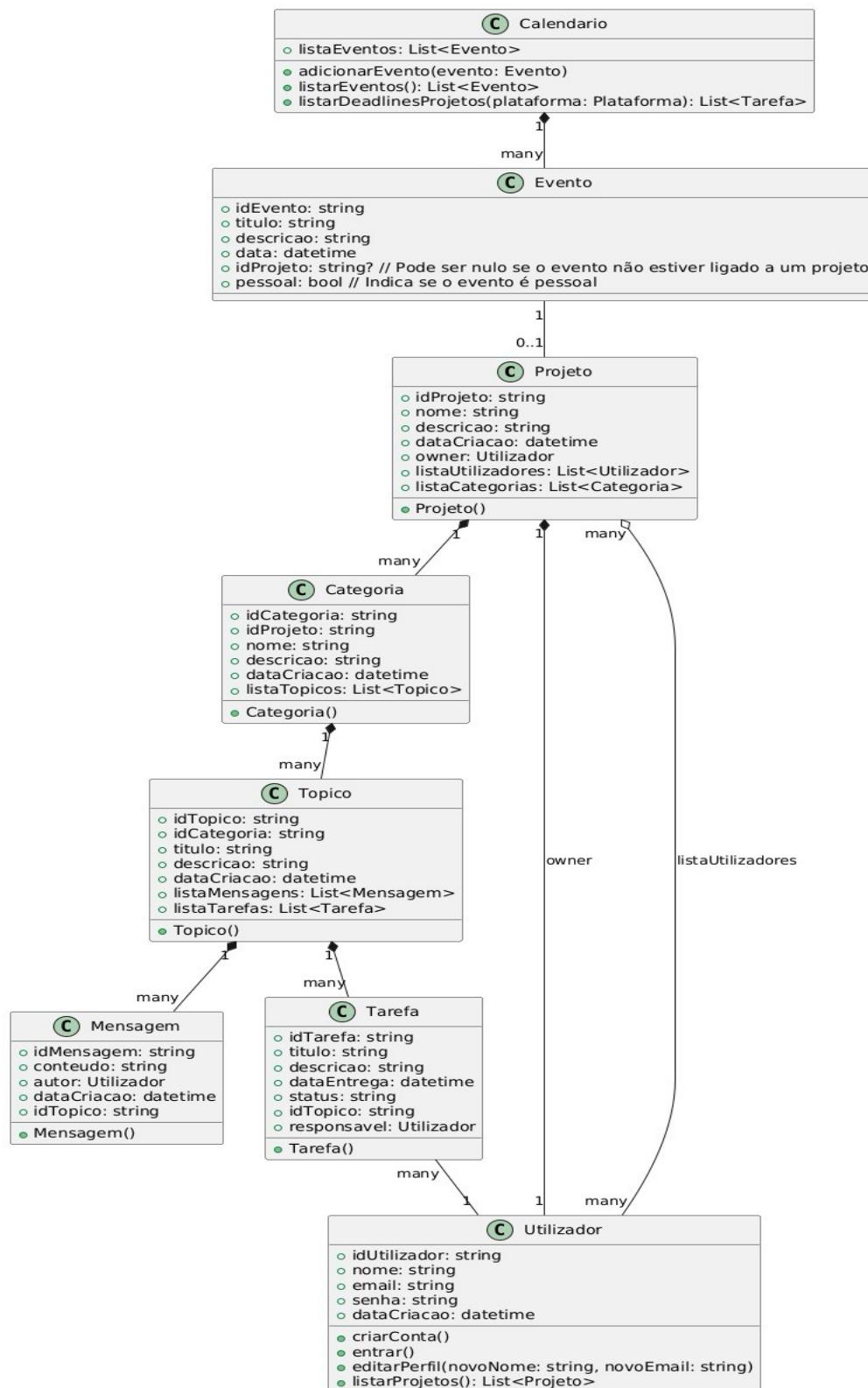


Figura 13 - Diagrama de Classes

Tabela 5 - Tabela de Atributos

Entidade	Descrição	Atributos	Métodos	Relações
Utilizador	Representa um utilizador da plataforma	idUtilizador, nome, email, senha, dataCriacao	criarConta(), entrar(), editarPerfil(), listarProjetos()	1:1 com Projeto (dono), N:N com Projeto (colaborador via Projeto_Utilizador)
Projeto	Representa um projeto criado na plataforma	idProjeto, nome, descricao, dataCriacao, owner, listaUtilizadores, listaCategorias	adicionarUtilizador(), removerUtilizador(), listarUtilizadores(), adicionarCategoria(), listarCategorias(), procurarCategoria(), listarDeadlines()	1:1 com Utilizador (owner), N:N com Utilizador, 1:N com Categoria
Categoria	Categoria dentro de um projeto	idCategoria, idProjeto, nome, descricao, dataCriacao, listaTopics	adicionarTopico(), listarTopics(), procurarTopico(), listarUtilizadoresAutorizados()	N:1 com Projeto, 1:N com Topico
Topico	Tópico dentro de uma categoria	idTopico, idCategoria, titulo, descricao, dataCriacao, listaMensagens, listaTarefas	adicionarMensagem(), listarMensagens(), procurarMensagem(), adicionarTarefa(), listarTarefas()	N:1 com Categoria, 1:N com Mensagem e Tarefa
Mensagem	Mensagem dentro de um tópico	idMensagem, conteudo, autor, dataCriacao	editarMensagem(), excluirMensagem()	N:1 com Topico
Tarefa	Tarefa dentro de um tópico	idTarefa, titulo, descricao, dataEntrega, status, responsavel	atribuirResponsavel(), atualizarStatus()	N:1 com Topico, 1:1 com Utilizador (responsável)
Evento	Evento associado a um projeto ou pessoal	idEvento, titulo, descricao, data, idProjeto (opcional), pessoal	criarEvento(), ListarEventos()	1:1 com Projeto (opcional)

6.2. Casos de Uso

6.2.1. Casos de Uso Textuais

Caso de Uso 01 – Criar Projeto (versão simplificada)

O utilizador inicia sessão na plataforma e acede à área principal. Ao clicar no botão “Novo Projeto”, é apresentado um formulário onde pode inserir o nome do projeto, uma descrição e, se desejar, escolher colaboradores da sua lista de amigos. Após submeter o formulário, o sistema valida a informação e cria o projeto. O projeto aparece de imediato na sua área de trabalho, e os colaboradores convidados recebem acesso automático. O utilizador torna-se o proprietário desse projeto.

Caso de Uso 02 – Adicionar Colaboradores ao Projeto

Dentro de um projeto criado, o proprietário pode abrir o modal de gestão de colaboradores. A partir daí, pesquisa amigos por nome ou identificador único e seleciona quem deseja convidar. Pode também definir permissões específicas, se necessário. Ao enviar o convite, o sistema regista esse pedido e o utilizador convidado passa a vê-lo na sua secção de notificações. Se o convite for aceite, o novo colaborador ganha acesso ao projeto e às respetivas funcionalidades.

Caso de Uso 03 – Criar Categoria

Dentro de um projeto, o utilizador pode clicar no botão “Nova Categoria” localizado na barra lateral. Surge um campo de input onde pode introduzir o nome da nova categoria. Após confirmar, o sistema valida se o nome é válido e único dentro do projeto. A nova categoria é imediatamente adicionada à lista e pode começar a ser utilizada para organizar tópicos. Os restantes membros do projeto veem automaticamente a nova categoria na interface, graças à atualização em tempo real via WebSocket.

Caso de Uso 04 – Criar Tópico

Ao aceder a uma categoria já existente dentro de um projeto, o utilizador pode criar um novo tópico ao clicar no botão correspondente. Um campo aparece para que escreva o nome do tópico. Depois de confirmar, o sistema verifica se o nome é válido e não está duplicado dentro da mesma categoria.

O novo tópico é então adicionado à lista e fica disponível para mensagens e tarefas. A criação é partilhada em tempo real com os restantes membros do projeto.

Caso de Uso 05 – Enviar Mensagem

Num tópico selecionado, o utilizador pode escrever uma mensagem no campo de texto do chat. Antes de ser enviada, a mensagem é automaticamente encriptada no frontend com a chave AES do tópico. Depois, é armazenada na base de dados (Firestore) e distribuída em tempo real para os outros membros com acesso ao tópico. As mensagens aparecem imediatamente na interface dos utilizadores, com suporte a anexos e comandos como @nome para menções ou /Cat para trocar de categoria.

Caso de Uso 06 – Criar Tarefa

Dentro de um tópico, o utilizador pode abrir o modal de tarefas e clicar em “Nova Tarefa”. É apresentado um formulário onde pode preencher o título, descrição, data de entrega e, se desejar, atribuir a tarefa a outro membro do projeto. Após confirmar, o sistema valida os dados e adiciona a nova tarefa ao tópico. Esta tarefa aparece imediatamente na lista de tarefas como uma vinheta clicável. Todos os membros do projeto com acesso ao tópico conseguem visualizá-la em tempo real.

Caso de Uso 07 – Ver Estado Online dos Membros

Ao aceder à página do projeto ou ao modal de colaboradores, o utilizador pode ver o estado online ou offline dos outros membros. Uma bolinha verde viva indica que o utilizador está online, enquanto uma bolinha verde escura representa o estado offline. Esta informação é atualizada automaticamente sempre que um utilizador entra ou sai da aplicação, graças à ligação via WebSocket.

Caso de Uso 08 – Enviar Pedido de Amizade

Na secção de amigos, o utilizador pode abrir o modal “Adicionar Amigo” e usar a barra de pesquisa para encontrar pessoas por nome ou identificador único. Ao clicar em “Adicionar”, o sistema envia um pedido de amizade para o utilizador selecionado. O pedido fica visível na secção “Friend

Requests” do destinatário. Enquanto estiver pendente, o pedido pode ser aceite ou recusado a qualquer momento.

Caso de Uso 09 – Aceitar Pedido de Amizade

Quando um utilizador recebe um pedido de amizade, pode aceder à secção “Friend Requests” no painel de amigos. Lá encontra os pedidos pendentes, com o nome, avatar e identificador do remetente. Ao clicar em “Aceitar”, o sistema atualiza automaticamente ambas as listas de amigos e remove o pedido da secção de pendentes. A interface mostra o novo amigo na lista principal, com data de início da amizade.

Caso de Uso 10 – Gerir Lista de Amigos

Na secção principal de amigos, o utilizador pode visualizar todos os amigos atuais como cartões com foto, nome, data e amigos em comum. A interface permite também aceder ao modal de “Adicionar Amigo”, enviar novos pedidos, ou remover um amigo atual. A pesquisa por nome e identificador ajuda a localizar rapidamente pessoas dentro da rede pessoal. Todas as ações são refletidas visualmente em tempo real.

Caso de Uso 11 – Convidar Amigos para Projeto

Dentro de um projeto, o proprietário pode aceder ao modal de gestão de colaboradores e consultar a lista de amigos já adicionados à sua conta. A partir dessa lista, pode selecionar amigos e enviar convites diretos para o projeto. Os amigos convidados recebem uma notificação na secção de pedidos e, ao aceitar, ganham acesso imediato ao projeto com permissões definidas. O processo é rápido e visa facilitar a colaboração entre utilizadores que já têm relação estabelecida.

Caso de Uso 12 – Sair de Projeto

Um membro de um projeto pode decidir sair do mesmo a qualquer momento. Ao clicar na opção “Sair do Projeto”, o sistema pede confirmação. Se o utilizador for apenas colaborador, é removido do projeto e perde acesso. No caso de ser o proprietário, o sistema solicita que transfira primeiro a

posse para outro membro ativo. Apenas após essa transferência é possível abandonar o projeto. A interface é atualizada em tempo real para todos os membros.

Caso de Uso 13 – Transferir Propriedade de Projeto

O proprietário de um projeto pode decidir passar o controlo para outro membro. No modal de gestão de colaboradores, pode clicar no ícone de estrela ao lado de um colaborador para o definir como novo proprietário. O sistema pede confirmação antes de aplicar a mudança. Após a transferência, o novo proprietário passa a ter acesso a todas as permissões de gestão, e o antigo proprietário mantém-se como colaborador comum.

Caso de Uso 14 – Arquivar e Desarquivar Categorias/Tópicos

O utilizador pode arquivar uma categoria ou tópico caso já não esteja em uso ativo. Ao clicar no botão de opções (três pontos), escolhe “Arquivar” e o item desaparece da lista principal. Arquivar não elimina o conteúdo, apenas o oculta da interface. Para gerir elementos arquivados, o utilizador pode aceder ao modal de gestão e usar os filtros “Ver Arquivados”. Lá, pode ativar um switch para desarquivar a qualquer momento. Esta funcionalidade mantém o espaço de trabalho limpo sem perda de dados.

Caso de Uso 15 – Usar Comandos no Chat

Durante uma conversa num tópico, o utilizador pode usar comandos especiais para agilizar ações. Por exemplo, ao escrever /CAT seguido do nome de uma categoria, a aplicação mostra os tópicos dessa categoria no topo do chat, permitindo mudar de tópico rapidamente. Ao escrever @ seguido de texto, são sugeridos utilizadores para mencionar. Estes comandos tornam a navegação e a comunicação mais eficiente, simulando atalhos dentro do chat.

Caso de Uso 16 – Ver e Gerir Notificações

O utilizador pode consultar notificações no topo da interface ou nas secções de pedidos e convites. Sempre que recebe um pedido de amizade ou um convite para projeto, é exibida uma bolinha vermelha com número. Ao clicar na secção correspondente, são apresentados os pedidos com opções

de aceitar ou recusar. Após interação, as notificações desaparecem automaticamente. Este sistema garante que o utilizador não perca eventos importantes dentro da plataforma.

Caso de Uso 17 – Visualizar Calendário e Deadlines

Na área principal de cada projeto, o utilizador pode abrir o calendário de tarefas. Este calendário mostra, em formato mensal ou semanal, todas as tarefas com prazos atribuídos dentro dos tópicos do projeto. O utilizador pode clicar numa tarefa no calendário para ver detalhes, editar, mudar a data ou alterar o responsável. Esta funcionalidade facilita a visualização da carga de trabalho e a gestão de tempo da equipa.

Caso de Uso 18 – Ver e Gerir Tarefas

Ao aceder ao botão de tarefas dentro de um tópico, o utilizador vê todas as tarefas existentes como vinhetas organizadas por estado ou prioridade. Pode clicar numa tarefa para abrir o seu detalhe, onde pode editar o título, descrição, data de entrega, anexos e responsável. Também é possível marcar a tarefa como concluída ou apagá-la, caso tenha permissões. Todas as alterações são aplicadas em tempo real e visíveis para os restantes membros do projeto.

Caso de Uso 19 – Usar o Sistema Offline (modo limitado)

Se o utilizador perder temporariamente a ligação à Internet, pode continuar a interagir com algumas funcionalidades locais (como escrever mensagens ou preencher tarefas). Quando a ligação for restabelecida, o sistema sincroniza automaticamente os dados com o servidor, garantindo que nenhuma informação se perde. Esta funcionalidade oferece uma experiência mais estável mesmo em ambientes com ligações instáveis.

6.2.2. Diagrama de Caso Uso (Plataforma Geral)

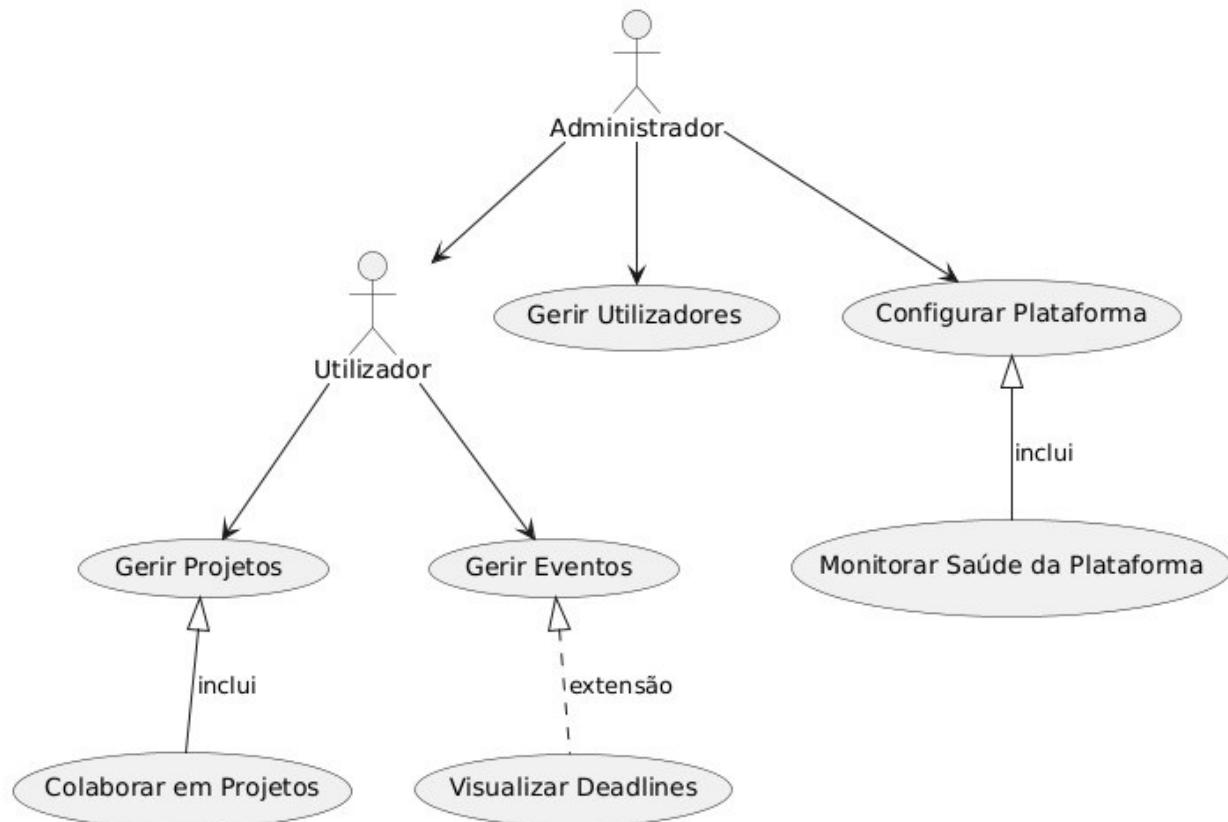


Figura 14 - Diagrama de Caso Uso (Plataforma Geral)

6.3. Diagrama de Entidades

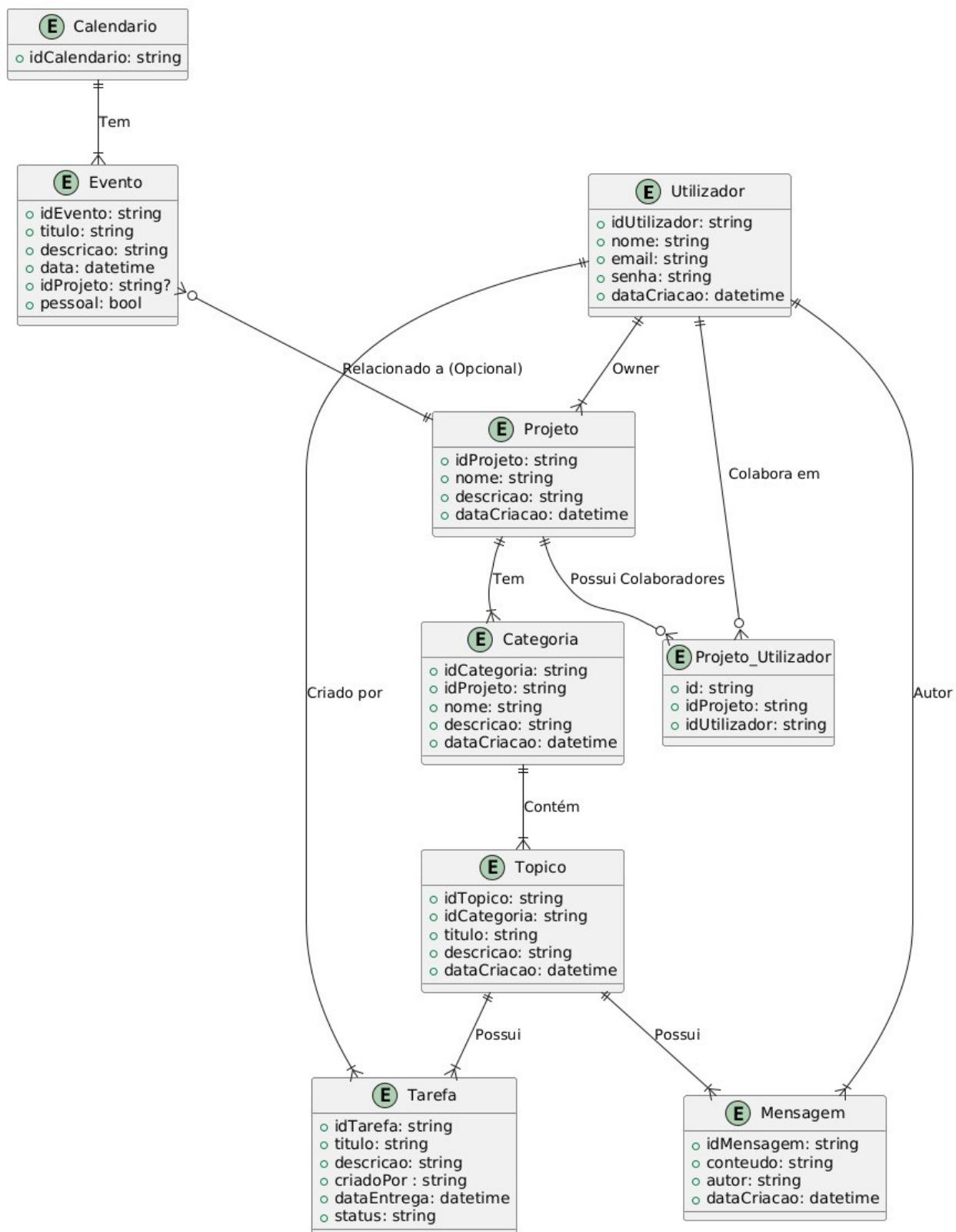


Figura 15 - Diagrama de Entidades

Tabela 6 - Tabela de Entidades

Entidade (Tabela)	Descrição	Atributos Principais	Relações
t_Users	Representa os utilizadores da plataforma	idUtilizador, nome, email, senha, dataCriacao	1:1 com t_Projeto (dono), N:N com t_Projeto via t_Projeto_Utilizador
t_Projeto	Projeto criado na plataforma	idProjeto, nome, descricao, dataCriacao, ownerId	1:1 com t_Users (dono), N:N com t_Users via t_Projeto_Utilizador, 1:N com t_Categoria, Opcional com t_Evento
t_Projeto_Utilizador	Relação N:N entre Projeto e Utilizador	idProjeto, idUtilizador	N:1 com t_Projeto, N:1 com t_Users
t_Categoria	Categoria dentro de um projeto	idCategoria, idProjeto, nome, descricao, dataCriacao	N:1 com t_Projeto, 1:N com t_Topicos
t_Topicos	Tópico dentro de uma categoria	idTopico, idCategoria, titulo, descricao, dataCriacao	N:1 com t_Categoria, 1:N com t_Mensagem, 1:N com t_Tarefa
t_Mensagem	Mensagem postada num tópico	idMensagem, conteudo, dataCriacao, idTopico	N:1 com t_Topicos, N:1 com t_Users (autor)
t_Tarefa	Tarefa associada a um tópico	idTarefa, titulo, descricao, dataEntrega, status, idTopico, responsavelId	N:1 com t_Topicos, N:1 com t_Users (responsável)
t_Calendario	Calendário onde eventos são armazenados	idCalendario	1:N com t_Evento
t_Evento	Evento dentro do calendário	idEvento, titulo, descricao, data, idProjeto (opcional), pessoal	N:1 com t_Calendario, opcional com t_Projeto

7. Regras de Negócio

Tabela 7 - Regras de Negócio

Nº	Regra de Negócio
1	Cada utilizador deve ter um email e identificador únicos para efeitos de autenticação e pesquisa.
2	Um projeto pode ser criado sem categorias nem tópicos, mas será considerado incompleto até os ter.
3	Não existe um limite máximo de colaboradores por projeto.
4	Cada projeto deve ter apenas um proprietário (owner) ativo em qualquer momento.
5	Um utilizador pode participar em múltiplos projetos em simultâneo, com diferentes permissões.
6	Um projeto só pode ser apagado se não tiver categorias, tópicos ou colaboradores associados.
7	Apenas o proprietário pode transferir a posse do projeto para outro membro.
8	Não é permitido criar duas categorias com o mesmo nome dentro de um mesmo projeto.
9	Apenas o proprietário pode remover colaboradores de um projeto.
10	Apenas o proprietário pode eliminar permanentemente o projeto.
11	Os nomes de tópicos dentro de uma mesma categoria devem ser únicos.
12	Tópicos arquivados não permitem envio de novas mensagens ou criação de tarefas.
13	Tarefas só podem ser atribuídas a membros ativos do projeto.
14	Só utilizadores com permissão de escrita podem criar tópicos numa categoria.
15	Não é permitido enviar pedidos de amizade duplicados entre os mesmos dois utilizadores.
16	Pedidos de amizade que não forem aceites expiram automaticamente após 30 dias.
17	O proprietário não pode ser removido da lista de colaboradores por outros membros.
18	Cada tópico deve pertencer obrigatoriamente a uma categoria e um projeto.
19	Tarefas com datas de entrega no passado não podem ser reabertas como pendentes.

Nº Regra de Negócio

- 20 Um projeto sem proprietário não pode ser acedido — o sistema exige transferência antes da saída do owner.
- 21 Arquivar uma categoria oculta todos os tópicos que ela contém.
- 22 Apenas administradores do projeto podem desarquivar categorias ou tópicos.
- 23 As mensagens são visíveis apenas para membros ativos do projeto.
- 24 As chaves AES de encriptação dos tópicos são guardadas localmente no frontend, nunca no backend.
- 25 Mensagens enviadas para utilizadores offline são sincronizadas assim que voltarem a ligar-se.

8. Glossário

Termo	Definição
WorkNest	Plataforma colaborativa para gestão de projetos académicos, com funcionalidades de chat, tarefas, organização por tópicos e categorias.
Projeto	Unidade principal de trabalho colaborativo. Cada projeto possui um nome, descrição, membros e estrutura interna própria.
Categoria	Subdivisão dentro de um projeto, usada para organizar os tópicos. Equivale a uma “pasta” temática.
Tópico	Elemento dentro de uma categoria que agrupa mensagens e tarefas relacionadas a um subtema específico.
Mensagem	Texto enviado dentro de um tópico, visível em tempo real pelos membros do projeto. Pode incluir anexos.
Tarefa	Item de trabalho associado a um tópico, com título, descrição, data de entrega e responsável.
Estado Online	Indicação visual do estado atual de presença dos utilizadores (online ou offline), exibida sobre os avatares.
Encriptação E2EE	(End-to-End Encryption) Sistema de encriptação ponto-a-ponto onde apenas os membros com a chave local podem ler as mensagens do tópico.

Termo	Definição
Comando /CAT	Comando de texto utilizado no chat para trocar rapidamente de categoria ou tópico.
Comando @nome	Comando de menção usado no chat para referenciar diretamente um utilizador.
WebSocket	Tecnologia que permite comunicação bidirecional em tempo real entre cliente e servidor.
JWT	(JSON Web Token) Mecanismo de autenticação usado para manter sessões seguras dos utilizadores.
MongoDB	Base de dados NoSQL usada para armazenar projetos, utilizadores, categorias, tópicos, tarefas e permissões.
Firestore	Base de dados da Google usada para armazenar e sincronizar mensagens em tempo real com baixa latência.
Frontend	Parte do sistema com a qual o utilizador interage, desenvolvida em React.js com animações e estrutura modular.
Backend	Lógica do servidor, desenvolvida em Node.js e Express, responsável pelo tratamento de dados, permissões e API.
Sprint	Ciclo de desenvolvimento curto (2-3 semanas) usado na metodologia Scrum para implementar e validar funcionalidades incrementais.
Stakeholder	Qualquer pessoa ou grupo com interesse direto no sistema, como estudantes, docentes, colaboradores ou a equipa técnica.
SPA	(Single Page Application) Aplicação web que carrega uma única página e atualiza dinamicamente o conteúdo, sem recarregar a página por completo.
Notificação	Alerta visual que informa o utilizador sobre pedidos de amizade, convites para projetos ou deadlines de tarefas.
Modal	Janela flutuante usada para apresentar informações ou interações sem sair da página principal.
Dropdown de Sugestões	Componente do chat que mostra sugestões contextuais ao escrever comandos como @ ou /.

Termo	Definição
AES-GCM	Algoritmo de encriptação simétrica usado para proteger mensagens no frontend com alta performance e segurança.

9. Plano de Validação e Verificação

Este plano tem como objetivo assegurar que os requisitos definidos neste documento sejam corretamente **implementados, testados e aceites**. Divide-se em dois grandes eixos:

- **Verificação:** Processo técnico que confirma se o sistema foi construído corretamente (ex: testes).
- **Validação:** Processo que confirma se o sistema cumpre os objetivos e necessidades dos utilizadores (ex: testes com utilizadores finais).

9.1. Estratégia Geral

Tabela 8 - Tipos de Testes

Tipo de Teste	Objetivo
Testes Unitários	Verificar o correto funcionamento de funções isoladas no backend/frontend.
Testes de Integração	Garantir que os componentes comunicam corretamente entre si.
Testes de Interface (UI)	Verificar se a interação do utilizador com a aplicação decorre como esperado.
Testes Funcionais	Confirmar que os requisitos funcionais são cumpridos com dados reais.
Testes de Aceitação	Validar com utilizadores se o sistema responde às suas necessidades.

9.2. Validação de Requisitos Funcionais

Tabela 9 - Validação de Requisitos Funcionais (RF)

ID	Requisito	Tipo de Validação	Estratégia
RF.01	Criar Projeto	Funcional / UI	Testar criação com e sem dados válidos. Validar aparecimento imediato na lista.
RF.02	Adicionar Colaboradores	Funcional / UI	Testar convites com diferentes utilizadores e verificação de permissões.
RF.03	Criar Categorias	Funcional / UI	Verificar criação de categoria visível em tempo real para todos.
RF.05	Enviar Mensagens	Funcional / Integração	Testar envio entre diferentes utilizadores com WebSocket ativo.
RF.06	Enviar Anexos	Funcional / UI	Verificar envio de PDFs, imagens e visualização dos ficheiros.
RF.07	Gerir Tarefas	Funcional / UI	Testar criação, edição e atribuição de tarefas com datas.
RF.09	Enviar Pedido de Amizade	Funcional / UI	Verificar que o pedido aparece corretamente no destinatário.
RF.12	Convidar Amigos para Projeto	Funcional / UI	Testar convites diretos e validação de acesso pós-aceitação.
RF.15	Arquivar e Desarquivar	Funcional / UI	Testar ocultação da sidebar e visibilidade no modal de gestão.
RF.21	Ajuda / Onboarding	Funcional / UI	Verificar se é apresentado a novos utilizadores e se responde a dúvidas comuns.

9.3. Validação de Requisitos Não-Funcionais

Tabela 10 - Validação de Requisitos Não-Funcionais

ID	Requisito	Tipo de Validação	Estratégia
RNF.01	Compatibilidade com navegadores	Técnica / UI	Testar em Chrome, Firefox e Edge com diferentes resoluções.
RNF.02	Tempo de resposta em ações básicas	Técnica	Medir tempo de carregamento de mensagens, tarefas e projetos.
RNF.03	Segurança de dados	Técnica / Código	Validar encriptação AES no frontend e proteção JWT no backend.
RNF.04	Acessibilidade	UI / Manual	Testar com teclado apenas, contraste de cores e tooltips.
RNF.05	Escalabilidade	Técnica / Teórica	Verificar arquitetura modular e suporte a múltiplos projetos simultâneos.

9.4. Validação com Utilizadores

Serão realizados **testes de aceitação** com 2 a 3 utilizadores reais (ex: colegas de curso).

Cada participante irá realizar tarefas como:

- Criar projeto
- Adicionar colaborador
- Trocar mensagens
- Atribuir tarefas
- Arquivar conteúdos

A validação será considerada **positiva** se os participantes conseguirem realizar as tarefas **sem assistência direta** e com feedback positivo quanto à interface e organização.

10. Gestão de Riscos

Esta secção identifica possíveis riscos associados ao desenvolvimento, funcionamento e adoção do sistema WorkNest, bem como as respetivas estratégias de mitigação.

Tabela 11 - Tabela de Riscos

Risco Identificado	Impacto	Probabilidade	Estratégia de Mitigação
Perda da chave de encriptação local (mensagens E2EE)	Alto	Média	Alertar o utilizador e permitir backup seguro da chave local.
Quebra de ligação WebSocket afeta comunicação em tempo real	Médio	Alta	Implementar reconexão automática e fallback para carregamento manual.
Utilizador abandona projeto sem transferir propriedade	Médio	Baixa	Forçar transferência antes de permitir a saída do proprietário.
Crescimento rápido sem otimização de base de dados	Alto	Baixa	Monitorizar escalabilidade e usar índices em MongoDB.
Erros de validação local não refletidos no backend	Médio	Média	Revalidar sempre os dados no backend com mensagens claras de erro.
Má compreensão do sistema por novos utilizadores	Médio	Alta	Integrar sistema de onboarding simples e ajuda rápida.
Dependência de serviços externos (Firestore, Firebase)	Médio	Média	Usar fallback local e prever exportação de dados.

11. Propostas de Futuras Melhorias

Estas sugestões não estão incluídas no escopo atual de desenvolvimento, mas poderão ser consideradas em versões futuras da plataforma WorkNest:

Tabela 12 - Futuras Melhorias

Área	Melhoria Proposta
Interface	Modo escuro completo, personalização de cores por projeto
Comunicação	Reações rápidas em mensagens, respostas encadeadas, pin de mensagens importantes
Tarefas	Subtarefas, etiquetas coloridas, prioridade visual
Integrações	Ligaçāo com Google Calendar, Google Drive, OneDrive
Automação	Comandos automáticos (ex: /deadline amanhā cria tarefa com data)
Estatísticas	Relatórios de produtividade por projeto, tarefas por membro
Mobile	Aplicação mobile nativa (React Native ou Flutter)
Acessibilidade	Leitura por voz, modo de alto contraste, suporte a leitores de ecrā
IA / Smart Features	Sugestāo de tópicos, previsão de tarefas atrasadas, resumo automático de tópicos

PARTE III – ARTIGO CIENTÍFICO

WorkNest: Centralização Funcional como Resposta à Fadiga Digital Universitária

Diogo Regadas diogo.regadas.2022103@my.istec.pt

Instituto Superior de Tecnologias Avançadas do Porto

Resumo

A fragmentação digital decorrente da utilização simultânea de múltiplas ferramentas, nomeadamente plataformas de comunicação, gestão de tarefas e partilha de ficheiros, tem originado um fenómeno cada vez mais relevante: a fadiga digital. Este conceito refere-se à sobrecarga cognitiva e à perda de produtividade resultantes da dispersão informacional e do excesso de notificações e sistemas paralelos. O artigo explora as causas subjacentes a este fenómeno, com destaque para o overtooling, analisando também o impacto da fragmentação nos ambientes colaborativos modernos. Através de uma abordagem qualitativa com base empírica, recorre-se à revisão de literatura e à análise de práticas digitais no ensino superior para contextualizar o problema. Na segunda parte do artigo, é discutida a forma como plataformas de gestão de projetos e colaboração, quando bem integradas, podem representar uma solução eficaz. Por fim, apresenta-se a proposta de um modelo funcional adaptado ao contexto académico, com base nas características observadas nas plataformas empresariais existentes, mas orientado para a realidade universitária, com foco na centralização, usabilidade e acessibilidade.

Palavras-chave: fadiga digital, fragmentação informacional, overtooling, gestão de projetos, plataformas colaborativas, produtividade digital, ferramentas integradas.

Abstract

Digital fragmentation, resulting from the simultaneous use of multiple independent tools such as communication platforms, task managers, and file-sharing systems, has contributed to the growing phenomenon of digital fatigue. This condition refers to cognitive overload and productivity loss caused by dispersed information and excessive notifications across disconnected systems. This article explores the underlying causes of digital fatigue, with a particular focus on overtooling, and examines its impact on modern collaborative environments. Adopting a qualitative, empirically grounded approach, the study is supported by a literature review and an analysis of digital practices within higher education. In the second part of the article, the discussion focuses on how well-integrated project management and collaboration platforms can offer an effective solution. The article concludes by presenting a functional model tailored to academic contexts, inspired by features found in enterprise platforms but designed to meet the specific needs of university settings, with an emphasis on centralization, usability, and accessibility.

Keywords: digital fatigue, information fragmentation, overtooling, project management, collaborative platforms, digital productivity, integrated tools

1. Introdução

No contexto atual do trabalho e da educação, as ferramentas e plataformas digitais tornaram-se elementos indispensáveis do quotidiano, sendo amplamente utilizadas para promover a eficiência, a colaboração e a produtividade. Soluções tecnológicas como software de gestão de projetos, aplicações colaborativas e sistemas de comunicação são hoje parte integrante da dinâmica profissional e académica. No entanto, esta adoção rápida e generalizada, muitas vezes feita de forma desarticulada e sem planeamento integrado, tem vindo a dar origem a um fenómeno crescente: a **fadiga digital** (Haiilo, 2025; Ripla Andre, 2025; Vartika Kashyap, 2025). A fadiga digital, também conhecida como *technostress* ou *cyber fatigue*, caracteriza-se por um estado de exaustão mental, emocional e física, resultante da exposição prolongada a múltiplos dispositivos e ambientes digitais. Esta condição manifesta-se frequentemente em contextos onde a interação constante com diferentes sistemas, aliada à sobrecarga de informação e à ausência de integração entre ferramentas, impõe um esforço cognitivo excessivo. Entre os fatores mais críticos estão as exigências de navegação em interfaces complexas, a pressão da conectividade permanente, que esbate as fronteiras entre vida pessoal e trabalho, e o fenómeno do *overtooling*, isto é, a multiplicação de aplicações que deveriam simplificar processos, mas acabam por os complicar Click or tap here to enter text..(Ok, 2024; Ripla Andre, 2025; Supriyadi et al., 2025).

Esta fragmentação de sistemas digitais revela-se especialmente problemática, tanto em ambientes empresariais como académicos. Utilizadores são frequentemente obrigados a alternar entre diversas aplicações para comunicar, partilhar ficheiros, gerir tarefas e coordenar projetos. Essa fragmentação

conduz a perdas de tempo, duplicação de informação, quebras de foco e fricção constante no fluxo de trabalho. A consequência é um aumento da carga mental e uma sensação contínua de dispersão, que afeta diretamente a produtividade e o bem-estar (Arissa Shanganlall, 2025; Ripla Andre, 2025).

Com efeito, a fadiga digital não deve ser encarada como um mero incômodo individual, mas sim como uma questão sistémica com impacto real na motivação, no desempenho e na saúde mental dos utilizadores. Estudos recentes apontam para ligações entre ambientes digitais desorganizados e níveis elevados de stress, frustração e até rotatividade em contextos profissionais. No caso das instituições de ensino superior, onde o trabalho colaborativo e o uso intensivo de tecnologia são cada vez mais centrais, compreender e mitigar esta realidade é fundamentalClick or tap here to enter text..(Arissa Shanganlall, 2025; Haiilo, 2025; Laker Benjamin, 2023).

Neste artigo, analisa-se o fenómeno da fadiga digital associado à fragmentação de ferramentas e discute-se o papel das plataformas unificadas como resposta potencial. Através de uma abordagem qualitativa com base empírica, são exploradas práticas atuais em contexto académico e apresentada a proposta da plataforma **WorkNest**, concebida para integrar num único espaço digital as funcionalidades essenciais de colaboração e gestão de projetos, com foco na usabilidade, centralização e acessibilidade.

2. Estado de Arte

Os ambientes contemporâneos de trabalho e de educação são crescentemente moldados pela integração generalizada de ferramentas

digitais, em particular **software de gestão de projetos**. Estas ferramentas são amplamente reconhecidas como essenciais para potenciar a eficiência, a colaboração e a produtividade. Proporcionam plataformas centralizadas para o acompanhamento de tarefas, colaboração em tempo real, comunicação, planeamento de atividades e gestão de recursos. Em 2025, o mercado apresenta uma ampla diversidade de soluções, adaptadas a diferentes dimensões de equipas, graus de complexidade dos projetos e necessidades setoriais, desde gestores de tarefas simples para pequenas equipas até sistemas empresariais robustos. Entre as funcionalidades mais valorizadas encontram-se: visualizações diversas de tarefas (Kanban, Gantt, cronogramas), automatizações, colaboração (chat, partilha de ficheiros, validação), dashboards, controlo de tempo e recursos, bem como opções de integração (Jon Darbyshire, 2025; Ok, 2024; Praburam Srinivasan, 2025).

No entanto, a adoção rápida e frequentemente descoordenada de múltiplas plataformas digitais, fenómeno designado por **overtooling**, tem originado um problema crescente: a **fadiga digital**. Este conceito refere-se a um estado de exaustão mental e desengajamento resultante da utilização prolongada e excessiva de dispositivos e plataformas digitais. Caracteriza-se por sentimentos de sobrecarga, traduzindo-se em cansaço mental, emocional e físico, causado pela interação constante com ferramentas digitais e pela sobrecarga informacional associada. Esta realidade agrava-se em ambientes fragmentados, onde as ferramentas não se integram de forma fluida, obrigando os utilizadores a saltar continuamente entre sistemas distintos.

De acordo com a literatura analisada, os principais mecanismos através dos quais a

fadiga digital afeta a produtividade e o bem-estar incluem:

Sobrecarga cognitiva e redução da performance mental: O envolvimento constante com múltiplas ferramentas e notificações esgota a capacidade limitada de processamento do cérebro, resultando em fadiga mental e menor desempenho cognitivo (Supriyadi et al., 2025).

Perda de tempo e ineficiência: A fragmentação dos sistemas exige uma navegação constante entre diferentes plataformas, levando a perdas de tempo significativas, estimadas em cerca de 1,8 horas por dia ou mais de 9 horas por semana em alguns casos (Arissa Shanganlall, 2025; Supriyadi et al., 2025)

Troca de contexto e perda de foco: A necessidade de alternar frequentemente entre ferramentas impõe um custo cognitivo conhecido como *troca de contexto*, que interrompe o fluxo de trabalho e deixa resíduos de atenção que dificultam a concentração na tarefa atual (Supriyadi et al., 2025).

Sobrecarga informacional e fadiga decisional: A proliferação de ferramentas e notificações dispersas conduz a um ambiente saturado de decisões micro e fragmentação da informação, contribuindo para exaustão mental, redução da qualidade das decisões e maior tendência à procrastinação.

Maior propensão a erros: A sobrecarga e a perda de foco aumentam a probabilidade de cometer falhas.

Contributo para o burnout e desmotivação: A exposição contínua à sobrecarga digital

eleva os níveis de stress, favorece o burnout e reduz o envolvimento com o trabalho, uma dinâmica (Ripla Andre, 2025; Supriyadi et al., 2025) associada ao conceito de *technostress* (Supriyadi et al., 2025).

Impacto negativo no bem-estar: A fadiga digital está associada a ansiedade, frustração, insatisfação profissional, má qualidade do sono, dores de cabeça e outros efeitos físicos e psicológicos.

Redução da criatividade e inovação: A constante fragmentação digital dificulta o pensamento profundo e bloqueia a capacidade de resolução criativa de problemas.

Absenteísmo e rotatividade: Ambientes digitais mal estruturados e o stress acumulado favorecem a rotatividade e o absenteísmo, dificultando a retenção de talentos. (Haiilo, 2025; Supriyadi et al., 2025).

Enfrentar a fadiga digital exige uma abordagem **multidimensional**, que combine estratégias individuais e organizacionais. A nível individual, recomenda-se a prática de bem-estar digital, o uso de técnicas de gestão de tempo (como o método Pomodoro (Que et al., 2023)) e a consciencialização dos limites cognitivos. No plano organizacional, as soluções passam por consolidar ferramentas, adotar design centrado no utilizador, garantir formação adequada, cultivar uma cultura que privilegie o bem-estar dos colaboradores, selecionar plataformas estratégicamente e assegurar integrações eficazes que minimizem interrupções no fluxo de trabalho.

Métodos de investigação emergentes como o **NeuroIS** têm vindo a ser utilizados para avaliar as respostas fisiológicas a ambientes digitais, utilizando métricas como a **variabilidade da frequência cardíaca**

(**HRV**) para identificar sinais de fadiga cognitiva e emocional. Resultados preliminares sugerem que a colaboração digital pode provocar a redução da atividade do sistema nervoso parassimpático — um potencial indicador de fadiga (Supriyadi et al., 2025).

Perspetivas futuras apontam para tendências tecnológicas que visam mitigar estas questões através da **consolidação de plataformas, melhoria nas integrações, uso de inteligência artificial** para simplificação de fluxos de trabalho, e soluções **low-code/no-code** que permitem maior personalização com menor complexidade técnica. O conceito relacionado de **aprendizagem fragmentada**, frequente no contexto educativo digital, reforça também a necessidade de estruturas de gestão de conhecimento mais sistemáticas e de orientação pedagógica adequada (Supriyadi et al., 2025).

A investigação recente tem destacado um vasto leque de ferramentas digitais dedicadas à gestão de projetos e à colaboração em equipa, muitas delas amplamente utilizadas em ambientes empresariais e educativos. Entre as mais referenciadas encontram-se **Asana, Slack, Trello, Jira** e **Microsoft Teams**, cada uma com abordagens distintas à organização de tarefas, comunicação, partilha de recursos e acompanhamento de progresso (Dahire et al., 2024; Pawłowski & Plechawska-Wójcik, 2022).

Segundo (Dahire et al., 2024), o **Jira** tem sido amplamente adotado em contextos técnicos, sobretudo por equipas que seguem metodologias ágeis como Scrum ou Kanban. A sua força reside na possibilidade de definir fluxos de trabalho complexos, associar tarefas a bugs ou versões de software e configurar automatizações. No entanto, os autores alertam que a ausência de um sistema de comunicação nativo (como chat ou chamadas)

obriga à integração com outras plataformas, o que pode agravar o problema da fragmentação digital.

De acordo com (Pawlowski & Plechawska-Wójcik, 2022), **Asana** é uma das ferramentas mais equilibradas no que respeita à interface, acessibilidade e versatilidade. Permite a gestão visual de projetos em múltiplas vistas (lista, cronograma, Kanban), bem como a definição de dependências e datas de entrega. Contudo, a sua dependência de integrações externas para funcionalidades como reuniões, comunicação síncrona ou gestão de ficheiros pode limitar a centralização do trabalho colaborativo, especialmente em ambientes académicos com menos apoio técnico.

Slack, como destaca (Yankah et al., 2023), é reconhecido como uma das plataformas mais eficazes para comunicação em tempo real, especialmente em equipas distribuídas. Permite organizar canais por temas, integrar bots e conectar-se a centenas de outras ferramentas. No entanto, o seu foco na comunicação instantânea não é complementado por funcionalidades estruturadas de gestão de projetos, tornando-o dependente de soluções paralelas como Trello ou Asana.

Por sua vez, **Microsoft Teams** beneficia de uma forte integração com o ecossistema Office 365, sendo frequentemente usado em instituições educativas e organizações de grande escala (Pawlowski & Plechawska-Wójcik, 2022). Ainda assim, a sua rigidez na personalização dos fluxos de trabalho e a complexidade na gestão de tarefas para utilizadores iniciantes podem limitar a sua eficácia em grupos académicos informais.

Finalmente, **Trello** é descrita por (Monnin, 2020) como uma plataforma intuitiva e visual, ideal para utilizadores com pouca experiência em ferramentas de gestão. Baseada em quadros Kanban, permite organizar tarefas em

colunas e acompanhar o progresso com facilidade. No entanto, a ausência de funcionalidades avançadas, como gestão de recursos, relatórios ou automatizações complexas, pode tornar-se um entrave em projetos de maior dimensão ou com múltiplas equipas envolvidas.

Em síntese, apesar das suas qualidades, estas plataformas apresentam limitações que se tornam críticas no contexto do ensino superior, onde os projetos envolvem equipas temporárias, prazos curtos e recursos limitados. A complexidade, a fragmentação funcional e os custos associados constituem barreiras à adoção efetiva e sustentada destas ferramentas, evidenciando a necessidade de soluções integradas, intuitivas e acessíveis especificamente concebidas para ambientes académicos. É neste contexto que se propõe a plataforma **WorkNest**, desenvolvida como resposta concreta às lacunas identificadas e orientada para as realidades e necessidades das comunidades universitárias.

3. Metodologia

3.1. Objetivo da Investigação

Esta investigação centra-se na análise da fragmentação digital e da fadiga associada ao uso excessivo e descoordenado de múltiplas plataformas digitais em contextos académicos. O problema foi identificado a partir da observação empírica de práticas comuns no ensino superior, onde estudantes lidam diariamente com ferramentas distintas para comunicação, gestão de tarefas, partilha de ficheiros e organização de projetos. Esta realidade contribui para uma sobrecarga cognitiva significativa, dispersão de foco e perda de eficiência (Ji & Zhong, 2023; Ok, 2024; Stangl et al., 2024).

Neste contexto, foi concebida a plataforma WorkNest como uma resposta integrada, prática e centrada nas necessidades dos estudantes, procurando unificar funcionalidades essenciais num único ambiente de trabalho colaborativo digital. O objetivo da metodologia é, assim, avaliar a pertinência, aplicabilidade e impacto preliminar desta solução unificada na mitigação dos problemas observados.

3.2. Abordagem Metodológica

A investigação segue uma abordagem qualitativa com base empírica, articulando revisão de literatura, observação de práticas reais em contexto académico e o desenvolvimento prático de uma solução digital. Trata-se de uma investigação aplicada e exploratória, combinando fundamentos teóricos com uma componente prática focada na conceção e teste da plataforma WorkNest. O desenvolvimento da plataforma foi conduzido com recurso à metodologia Agile Scrum, permitindo uma abordagem incremental e iterativa. As funcionalidades foram desenvolvidas em sprints curtos, testadas progressivamente com utilizadores reais e ajustadas com base no feedback recolhido. Esta estratégia facilitou uma validação contínua da solução em ambientes académicos reais, promovendo melhorias rápidas e orientadas para as necessidades dos estudantes.

3.3. Fontes e Técnicas de Recolha de Dados

As fontes de informação e técnicas de recolha de dados utilizadas incluem:

- **Revisão de literatura científica** sobre os temas da fadiga digital, fragmentação cognitiva (e.g., overload e distração em multitool environments), gestão de projetos em contexto académico, ferramentas colaborativas e metodologias digitais de aprendizagem;
- **Observação prática informal** de métodos de trabalho e comunicação entre estudantes de diferentes cursos e faculdades;
- **Análise comparativa informal** de plataformas populares (ex: Trello, Slack, Microsoft Teams, Discord, Notion, Asana), com especial atenção às suas limitações em cenários de uso académico, como redundância de notificações, dispersão de dados, falta de integração entre módulos e carga cognitiva imposta pela fragmentação.

3.4. Processo de Desenvolvimento da Plataforma WorkNest

O desenvolvimento da plataforma **WorkNest** teve início com a **identificação de necessidades recorrentes** e dificuldades relatadas por estudantes em contextos reais. A partir dessas observações, foi elaborado um conjunto de requisitos funcionais e não funcionais, focado na **redução da fragmentação digital e melhoria da colaboração entre pares**.

As funcionalidades principais desenvolvidas incluem:

- Criação e gestão de projetos colaborativos;
- Organização hierárquica em **categorias e tópicos**;
- Gestão de tarefas com prazos, responsáveis e anexos;

- Sistema de **mensagens** vinculado a tópicos;
- **Calendário integrado** com visualização de prazos;
- Suporte a **anexos com encriptação AES por tópico**;
- Funcionalidades de **amizade, convites e estados online/offline**.

3.5. Testes Exploratórios e Validação Inicial

Para validação inicial da plataforma, foram conduzidos **testes exploratórios com utilizadores reais**, nomeadamente estudantes de diferentes áreas (ex: Engenharia Informática, Psicologia, Design). Os testes incidiram sobre a usabilidade, clareza da interface, utilidade prática das funcionalidades integradas, e impacto percebido na redução de carga cognitiva.

O **feedback recolhido** foi predominantemente qualitativo, com base em sessões de utilização assistida, entrevistas informais e observação direta. Os resultados apontaram para uma melhoria na percepção de foco, na organização de tarefas e na redução do cansaço associado ao uso de múltiplas plataformas dispersas.

3.6. Limitação do Estudo

Apesar dos resultados promissores, este estudo apresenta limitações relevantes:

- **Número reduzido de participantes** e testes exploratórios sem amostragem representativa;
- **Ambiente controlado e limitado ao meio académico**, sem extração para contextos empresariais;
- **Ausência de métricas quantitativas sistemáticas**, como tempo de

- execução de tarefas, número de cliques ou carga mental medida objetivamente;
- Utilização de **feedback informal e não padronizado**, limitando a comparabilidade de resultados.

4. Apresentação e Discussão de Resultados

4.1. Resultados Preliminares

Os testes envolveram **32 participantes** de cursos como Engenharia Informática, Psicologia, Design, Gestão e Ciências da Comunicação, com idades compreendidas entre os **19 e os 28 anos**. Durante um período de **duas semanas**, os estudantes utilizaram a plataforma para simular a criação e gestão de projetos interdisciplinares, incluindo a organização de tarefas, comunicação em tópicos temáticos e partilha de ficheiros. A análise qualitativa das interações revelou um **interesse genuíno** dos utilizadores na adoção de práticas colaborativas mais organizadas, especialmente pela possibilidade de integrar num só espaço funcionalidades que anteriormente estavam dispersas em várias ferramentas. Foram recolhidos **47 comentários espontâneos** via formulário e fórum interno, com destaque para:

- a redução da necessidade de alternar entre múltiplas plataformas (ex: WhatsApp, Google Drive, Trello);
- a valorização da clareza na estrutura de projetos, categorias e tarefas;
- sugestões de funcionalidades adicionais como alertas visuais e sincronização com calendários externos.

Do ponto de vista quantitativo, registou-se um tempo médio de utilização de **5,1 horas por utilizador**, com uma **taxa de retorno de 81%**, indicando um elevado nível de **engagement** tendo em conta o caráter ainda experimental da plataforma. **69% dos utilizadores** interagiramativamente com o sistema de tarefas e mensagens, demonstrando adesão às funcionalidades centrais.

4.2.Discussão

Os resultados obtidos reforçam a **pertinência do modelo unificado** proposto pela WorkNest, em contraste com a realidade frequentemente fragmentada do ecossistema digital académico:

- Enquanto muitas soluções empresariais assumem equipas estáveis e infraestruturas centralizadas, WorkNest revelou-se eficaz em contextos mais **flexíveis e informais**, como os grupos de trabalho académico.
- A combinação de tarefas, tópicos e calendário num só ambiente demonstrou ser especialmente útil em equipas multidisciplinares, onde a **clareza organizacional** é determinante para o sucesso da colaboração.
- A plataforma evidenciou potencial para promover hábitos de **gestão autónoma de projeto**, mesmo entre estudantes com pouca experiência prévia com ferramentas estruturadas.

Por outro lado, os dados qualitativos revelaram alguns **desafios** a considerar em futuras iterações:

- Vários utilizadores referiram a necessidade de **tutoriais mais visuais** ou onboarding inicial, sobretudo para

compreender a lógica de categorias e tópicos.

- Alguns estudantes levantaram dúvidas sobre a escalabilidade da plataforma em projetos com equipas muito grandes, sugerindo melhorias na navegação e personalização do espaço de trabalho.

4.3.Limitações e Próximos Passos

Importa reconhecer que esta fase exploratória foi limitada a uma amostra reduzida e maioritariamente composta por estudantes com **interesse em ferramentas digitais**, o que pode enviesar os níveis de aceitação observados. Estudos futuros deverão incluir **testes em contextos curriculares reais**, com turmas completas e prazos de entrega autênticos.

Planeia-se também realizar uma **análise longitudinal**, de forma a compreender como os utilizadores interagem com a plataforma ao longo de um semestre letivo e como se formam dinâmicas de colaboração sustentada entre estudantes de diferentes áreas.

5. Conclusão

A crescente fragmentação digital no ensino superior, potenciada pela utilização descoordenada de múltiplas plataformas para tarefas distintas, tem vindo a gerar impactos significativos na experiência dos estudantes, como a fadiga digital, a sobrecarga cognitiva e a perda de eficiência colaborativa. Estes desafios exigem soluções que respondam não apenas do ponto de vista tecnológico, mas também pedagógico e organizacional.

Neste contexto, a plataforma **WorkNest** foi concebida como uma proposta integrada e

centrada no utilizador, visando consolidar num único ambiente funcionalidades essenciais como gestão de projetos, comunicação estruturada, tarefas, calendário e partilha segura de ficheiros. Através de um desenvolvimento incremental baseado na metodologia Agile Scrum, a plataforma foi construída e ajustada com base em observações práticas e feedback de estudantes reais, permitindo validar a sua pertinência e usabilidade em cenários académicos reais.

Os resultados preliminares indicam que a unificação das ferramentas de trabalho num único ecossistema digital pode contribuir de forma decisiva para reduzir a fragmentação, melhorar a organização dos grupos académicos e aliviar a carga mental associada à gestão de múltiplas interfaces. Embora os testes realizados sejam exploratórios e limitados em escala, os indícios são positivos e justificam o aprofundamento desta linha de investigação.

No futuro, será essencial alargar a amostra de utilizadores, implementar mecanismos formais de avaliação da carga cognitiva e comparar diretamente a eficácia da WorkNest com outras soluções existentes. Além disso, a integração de métricas neurofisiológicas poderá permitir uma análise mais rigorosa do impacto real na fadiga digital. A investigação aqui apresentada constitui, assim, um contributo relevante para o debate em torno da saúde digital no ensino superior e para a construção de soluções tecnológicas mais humanas, coesas e centradas na experiência académica.

6. Referências

- Arissa Shanganlall. (2025, April 29). *How Fragmented Systems Are Causing Digital Fatigue in Irish Universities*. <https://www.classter.com/blog/edtech/how-fragmented-systems-are-causing-digital-fatigue-in-irish-universities/>
- Arissa Shanganlall. (2025, April 29). *How Fragmented Systems Are Causing Digital Fatigue in Irish Universities*. https://www.classter.com/blog/edtech/how-fragmented-systems-are-causing-digital-fatigue-in-irish-universities/?utm_source=chatgpt.com
- Dahire, A., Dhamale, T., Dhanke, S., Gaikwad, S., & Parate, Ms. R. (2024). Team Tracker: A Project Management Tool. *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 08(10), 1–7. <https://doi.org/10.55041/IJSREM38265>
- Haiilo. (2025, March 6). *Digital Fatigue: How Fragmented Tools Are Hurting Your Team*. https://blog.haiilo.com/blog/digital-fatigue-how-fragmented-tools-are-hurting-your-team/?_gl=1*9ipjz2*_gcl_au*MTI2MTAwODgzOS4xNzQ5NDY2MTE2
- Ji, H., & Zhong, Z. (2023). Challenges and Countermeasures of Fragmented Learning to College Mathematics Teaching in the Era of Mobile Internet. *Advances in Educational Technology and Psychology*, 7(9). <https://doi.org/10.23977/aetp.2023.070905>
- Jon Darbyshire. (2025, April 28). *10 Best Project Management Tools & Software In 2025 [Reviewed]*. https://www.smartsuite.com/blog/project-management-software?105755a5_page=2
- Laker Benjamin. (2023). *How to fight digital fatigue at work*. <https://www.henley.ac.uk/news/2023/how-to-fight-digital-fatigue-at-work>
- Maican, C., Cazan, A. M., Lixandroiu, R., Dovleac, L., & Maican, M. A. (2021).

- Exploring the factors influencing the use of communication and collaboration applications. *Journal of Organizational and End User Computing*, 33(4), 1–27. <https://doi.org/10.4018/JOEUC.20210701.0a5>
- Monnin, C. (2020). Product review. *Journal of the Canadian Health Libraries Association*, 42(3), 129–129. <https://doi.org/10.29173/jchla29509>
- Ok, E. (2024). *Collaborative Learning Tools*. <https://www.researchgate.net/publication/386250907>
- Pawłowski, P., & Plechawska-Wójcik, M. (2022). *A comparative analysis of tools dedicated to project management*. *Analiza porównawcza narzędzi dedykowanych zarządzaniu projektami*.
- Praburam Srinivasan. (2025, May 5). *15 Best Project Management Tools in 2025*. <https://clickup.com/blog/best-project-management-tools/>
- Que, C. G. L., Adonis, L. V. C., Casus, A. I. A., Hugo, A. J. D., Sino-ag, M. E. L., Tariman, J. M. O., & Angoluan, M. A. S. (2023). Effectiveness of Pomodoro Technique on Memory Retention among Psychology Students in a University in Quezon City, Philippines. *European Modern Studies Journal*, 7(5), 270–273. [https://doi.org/10.59573/emsj.7\(5\).2023.24](https://doi.org/10.59573/emsj.7(5).2023.24)
- Ripla Andre. (2025, February 19). *The “Digital Fatigue” Epidemic: How Over-Tooling Kills Employee Productivity*. <https://www.linkedin.com/pulse/digital-fatigue-epidemic-how-over-tooling-kills-riplapgcert-pgdip-mnwue>
- Stangl, F. J., Riedl, R., Weitzl, W. J., & Martin, S. (2024). Fatigue and Stress Levels in Digital Collaboration: A Pilot Study with Video Conferencing and the Metaverse. *Lecture Notes in Information Systems and Organisation*, 68, 89–103. https://doi.org/10.1007/978-3-031-58396-4_9
- Supriyadi, T., Sulistiasih, S., Rahmi, K. H., Pramono, B., & Fahrudin, A. (2025). The impact of digital fatigue on employee productivity and well-being: A scoping literature review. In *Environment and Social Psychology* (Vol. 10, Issue 2). Arts and Science Press Pte. Ltd. <https://doi.org/10.59429/esp.v10i2.3420>
- Torres, R., Laboissiere, R., Camargos, S., Nunes, C., Melo, M., & Liu, P. (n.d.). *GUIA PRÁTICO PARA USO DE PLATAFORMAS VIRTUAIS NO ENSINO REMOTO*.
- Vartika Kashyap. (2025, March 3). *15 Best Project Management Tools in 2025*. <https://www.proofhub.com/articles/top-project-management-tools-list>
- Yankah, J. E., Adjei, K. O., Bonney, S. O., Kotey, S., & Tieru, C. K. (2023). APPRAISAL OF MOBILE APPS FOR COMMUNICATION AND COLLABORATION AMONG CONSTRUCTION PROJECT TEAMS. *African Journal of Applied Research*, 9(2), 144–170. <https://doi.org/10.26437/ajar.31.10.2023.10>
- Zamiri, M., & Esmaeili, A. (2024). Methods and Technologies for Supporting Knowledge Sharing within Learning Communities: A Systematic Literature Review. In *Administrative Sciences* (Vol. 14, Issue 1). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/admsci14010017>

BIBLIOGRAFIA GERAL

- Express.js. (2024). *Instalação e Guia Inicial*. Disponível em:
<https://expressjs.com/en/starter/installing.html>
- Meta. (2024). *React Documentation – Learn React*. Disponível em: <https://react.dev/learn>
- OpenJS Foundation. (2024). *Node.js API Documentation*. Disponível em:
<https://nodejs.org/docs/latest/api/>
- Auth0 / NPM. (2024). *jsonwebtoken – JSON Web Token for Node.js*. Disponível em:
<https://www.npmjs.com/package/jsonwebtoken>
- MongoDB Inc. (2024). *MongoDB Documentation*. Disponível em:
<https://www.mongodb.com/docs/>
- Google. (2024). *Firebase Documentation*. Disponível em: <https://firebase.google.com/docs>

Índice de Tabelas

Tabela 1 - Eventos Websocket	27
Tabela 2 - Requisitos Funcionais	43
Tabela 3 - Tabela de Rastreabilidade	61
Tabela 4 - Tabela de Prioridade de Requisitos Funcionais	62
Tabela 5 - Tabela de Atributos	66
Tabela 6 - Tabela de Entidades	74
Tabela 7 - Regras de Negócio	75
Tabela 8 - Tipos de Testes	78
Tabela 9 - Validação de Requisitos Funcionais (RF)	79
Tabela 10 - Validação de Requisitos Não-Funcionais	80
Tabela 11 - Tabela de Riscos	81
Tabela 12 - Futuras Melhorias	82

Índice de Figuras

Figura 1 - Backend Package.json	16
Figura 2 - Estrutura Backend Files	17
Figura 3 - Exemplo de Route Protegida	18
Figura 4 - Frontend Package.json	19
Figura 5 - Estrutura Frontend Files	20
Figura 6 - HomePage - LightMode	22
Figura 7 - HomePage - DarkMode	22
Figura 8 – InsideProject	23
Figura 9 - Opções nos Tópicos e Categorias	23
Figura 10 - ChatBox	24
Figura 11 - Tarefas	25
Figura 12 - Atualização Websocket	25
Figura 13 - Diagrama de Classes	65
Figura 14 - Diagrama de Caso Uso (Plataforma Geral)	72
Figura 15 - Diagrama de Entidades	73