



UNIVERSIDADE
LUSÓFONA

Aplicação Web MentHA Digital

Trabalho Final de curso

Relatório Final

Tomás Nave, a22208623

André Jesus, a22207061

Nome do Orientador: Lúcio Studer Ferreira

Coorientador: Martim Mourão

Coorientador: Pedro Machado Santos

Entidade Externa: ASMAL, GIRA, Crinabel

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

27 de abril de 2025

www.ulusofona.pt

Direitos de cópia

Aplicação MentHA, Copyright de Tomás Nave e André Jesus, ULHT.

A Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

A realização deste trabalho não teria sido possível sem o apoio, orientação e dedicação de várias pessoas que, ao longo de todo o percurso, contribuíram de forma essencial para o seu desenvolvimento e concretização.

Gostaríamos de expressar um especial agradecimento ao **Professor Lúcio Studer**, ao **Professor Pedro Machado** e ao **Professor Martim Mourão**, pelo acompanhamento próximo, pela paciência e pela constante disponibilidade demonstrada em todas as sessões de reunião ao longo do projeto. A sua orientação técnica, os seus conselhos valiosos e o incentivo permanente foram determinantes para nos ajudar a ultrapassar os diversos desafios que surgiram em cada fase deste trabalho.

A sua experiência e compromisso pedagógico permitiram-nos não só encontrar soluções práticas para os problemas técnicos mais complexos, como também consolidar aprendizagens importantes que certamente levaremos para o nosso futuro profissional. O apoio prestado, mesmo perante as dificuldades naturais de um projeto desta dimensão e com as limitações de tempo que enfrentámos, foi sempre marcado por uma postura de compreensão, motivação e incentivo à superação.

A todos, o nosso sincero reconhecimento e profundo agradecimento pelo contributo decisivo para o sucesso deste trabalho.

Resumo

O Projeto MentHA (Mental Health and Aging) tem como objetivo desenvolver e otimizar uma aplicação Web destinada a melhorar o bem-estar e a qualidade de vida de adultos mais velhos com défice cognitivo, demência leve ou transtornos dissociativos de identidade, bem como apoiar os seus cuidadores formais e informais, promovendo a saúde mental na comunidade. Esta plataforma é composta por três aplicações distintas: MentHA COG, Protocolo MentHA e MentHA CARE. Este Trabalho Final de Curso (TFC) focar-se-á principalmente na melhoria contínua e na correção de falhas destas aplicações. Numa fase inicial, o objetivo será o levantamento de requisitos e a identificação de problemas existentes. Posteriormente, com o feedback das entidades parceiras, a prioridade será dada ao desenvolvimento de novas funcionalidades e à resolução dos problemas identificados, de forma a garantir uma experiência otimizada e alinhada com as necessidades do utilizador final.

Palavras-chave:

- Aplicação Web
- Saúde Mental
- Treino Cognitivo
- Avaliação Neuropsicológica
- Qualidade de Vida

Abstract

The MentHA Project (Mental Health and Aging) aims to develop and optimize a web application designed to enhance the well-being and quality of life of older adults with cognitive deficits, mild dementia, or dissociative identity disorders, as well as to support their formal and informal caregivers by promoting mental health within the community. This platform consists of three distinct applications: MentHA COG, MentHA Protocol, and MentHA CARE. This Final Course Project (FCP) will primarily focus on continuous improvement and bug correction within these applications. In the initial phase, the main goal will be to gather requirements and identify existing issues. Subsequently, with feedback from partner entities, the priority will shift to developing new features and resolving identified issues to ensure an optimized and final user-aligned experience.

Key-words:

- Web Application
- Mental Health
- Cognitive Training
- Neuropsychological Assessment
- Quality of Life

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	3
1.2.1	Objetivo Inicial	3
1.2.2	Mudança de Foco e Objetivos Até à Data	4
1.2.3	Objetivo Final	5
1.3	Estrutura do Documento	6
2	Pertinência e Viabilidade	7
2.1	Pertinência	7
2.2	Viabilidade	7
2.3	Análise Comparativa com Soluções Existentes	8
2.3.1	Soluções existentes	8
2.3.2	Análise de benchmarking	9
2.4	Proposta de inovação e mais-valias	9
2.5	Identificação de oportunidade de negócio	10
3	Especificação e Modelação	11
3.1	Análise de Requisitos	11
3.1.1	Enumeração de Requisitos	11
3.1.2	Descrição detalhada dos requisitos principais	20
3.1.3	Casos de Uso/ <i>User Stories</i>	21
3.2	Modelação	31
3.3	Protótipos de Interface	33
4	Solução Desenvolvida	36
4.1	Apresentação	36
4.2	Arquitetura	38
4.3	Tecnologias e Ferramentas Utilizadas	38
4.4	Ambientes de Teste e de Produção	39
4.4.1	Ambiente de Testes	39
4.4.2	Ambiente de Produção	40
4.5	Abrangência	42

4.6	Desafios Enfrentados e Como os Ultrapassamos	42
4.6.1	Adaptação a um Sistema Existente	42
4.6.2	Manutenção e Melhoria do Projeto	43
4.6.3	Tecnologia e Infraestrutura	45
4.6.4	Colaborar com Diferentes Áreas	47
4.7	Componentes.....	47
4.7.1	Exportação de Avaliações Risk em Ficheiro CSV	48
4.7.2	Criação de Grupos de Sessão com Filtros Multivariados (Reformulação da View)	
49		
4.7.3	Criação de Protótipo para Remodelação Visual do Website MentHA Digital.....	52
4.7.4	Desenvolvimento da Nova Home Page do Website MentHA Digital	55
4.8	Interfaces	57
5	Testes e Validação	58
5.1	Testes de Funcionalidades.....	58
5.1.1	Teste de Instalação e Execução em Ambiente Local com Docker Compose.....	58
5.1.2	Teste de Validação de Alterações na Criação de Grupos – MentHA COG	59
5.1.3	Teste de Exportação de Avaliações – MentHA Eval	59
5.2	Teste de CI/CD e Deployment com GitHub Actions.....	60
5.3	Testes Unitários	61
5.4	Testes da Interface do Utilizador (UI/UX).....	61
5.5	Teste de Entrada de Dados	62
5.6	Testes de Utilizador	62
6	Método e Planeamento	64
6.1	Planeamento inicial	64
6.1.1	Sprints Realizados.....	67
6.2	Análise Crítica ao Planeamento	73
6.2.1	Planeamento Inicial e Previsão de Sprints	73
6.2.2	Desvios ao Planeamento	73
6.2.3	Adaptação da Estratégia e Repositionamento das Prioridades	74
6.2.4	Fase de Retoma Funcional e Modernização Visual	74
6.2.5	Fase Final – Consolidação Técnica e Avaliação do Planeamento	74
6.2.6	Avaliação Geral e Aprendizagem.....	75
7	Resultados	79

7.1	Resultados dos Testes.....	79
7.1.1	Instalação e Execução do Ambiente de Desenvolvimento com Docker Compose	79
7.1.2	Validação da Criação de Grupos – MentHA COG	81
7.1.3	Exportação de Avaliações – MentHA EVAL	83
7.1.4	CI/CD com GitHub Actions.....	85
7.1.5	Testes UI/UX.....	85
7.1.6	Entrada de Dados no Protocolo MentHA.....	86
7.2	Cumprimento de requisitos	87
7.2.1	Considerações Finais da Secção	99
8	Conclusão	100
8.1	Conclusão.....	100
8.1.1	Grau de concretização do plano.....	100
8.1.2	Diferenças entre a Solução Proposta Inicialmente e a Solução Desenvolvida....	101
8.1.3	Maiores Dificuldades na Realização do TFC	101
8.1.4	Evolução do Trabalho e Conhecimentos ao Longo do TFC	102
8.2	Trabalhos Futuros	103
8.2.1	Prioridades Técnicas Essenciais.....	104
8.2.2	Etapas de Continuidade.....	105
8.2.3	Sugestões de Valorização e Inovação.....	105
	Bibliografia	107
	Anexo A – Aplicações MentHA Digital.....	109
	A.1 MentHA COG – Programa de Estimulação Cognitiva para Adultos Mais Velhos.....	109
	A.2 Protocolo MentHA – Avaliação Neuropsicológica para Validação de Treino Cognitivo	111
	A.3 MentHA CARE – Programa de Apoio Psicoeducativo para Cuidadores	113
	Anexo B – Casos de Uso	116
	Anexo C – Modelo ERD MentHA COG / MentHA CARE.....	119
	Anexo D - Workflow de Desenvolvimento	122
	D.1 Estrutura do Workflow	122
	D.2 Fluxo de Trabalho	123
	Anexo E - Documentação	126
	E.1 A importância da documentação	126
	E.2 Pesquisa sobre padrões de documentação.....	126

E.2.1 Geradores automáticos de documentação.....	126
E.2.2 Ferramentas de Documentação com IA Integrada	127
E.3 Definição do padrão de documentação adotado.....	127
E.3.1 Explicação do modelo de documentação.....	128
E.3.2 Exemplo aplicado	129
E.4 Documentação de Instalação e Execução	129
E.5 Conclusão Final da Documentação	130
Anexo F - Importância do gitignore.....	131
F.1 Problemas Encontrados no Projeto.....	131
F.2 Problemas Comuns em Projetos Django com .gitignore.....	131
F.3 Melhor Prática: Utilizar Presets de .gitignore	132
Anexo G – Configuração do Ambiente de Desenvolvimento com Docker e Docker Compose	133
G.1 Introdução	133
G.2 O que é o Docker Compose?	133
G.3 Estrutura e Configuração do Docker Compose no Projeto.....	133
G.4 Estrutura de Ficheiros Envoltos.....	134
G.5 Conclusão.....	136

Lista de Figuras

Figura 1 - Logótipos dos nossos Parceiros	1
Figura 2 - Visão Geral Trelio Epicos , Features e User Stories	21
Figura 3 - Modelo E/R Da Aplicação MentHA CARE/COG.....	32
Figura 4 - Modelo E/R Da Aplicação Protocolo MentHA	32
Figura 5 - Mapa Aplicacional do MentHA CARE	33
Figura 6 - Mapa aplicacional do Protocolo MentHA.....	34
Figura 7 - Mapa aplicacional do MentHA COG	35
Figura 8 - Arquitetura Proposta para o MentHA Digital	38
Figura 9 - View Exportação Risk.....	49
Figura 10 - View Criação de Grupo	50
Figura 11 - Validação Condicional da Criação de Grupo via JavaScript	51
Figura 12 - Website MentHA Digital 1	52
Figura 13 - Website MentHA Digital 2	53
Figura 14 - Website MentHA Digital	53
Figura 15 - Protótipo Website MentHA 1	54
Figura 16 - Protótipo Website MentHA 2	54
Figura 17 - Protótipo Website MentHA 3	55
Figura 18 - Home Page Screen 1	56
Figura 19 - Home Page Screen 2	56
Figura 20 - Imagem do Trello: Product Backlog e o Quadro Kanban.....	64
Figura 21 - Imagem Trello Sprints Realizados 1	66
Figura 22 - Imagem Trello Sprints Realizados 2	66
Figura 23 - Imagem Trello Spritns Realizados 3.....	66
Figura 24 - Diagrama de Gantt Final	72
Figura 25 - Diagrama de Gantt Intercalar 1ºSemeste.....	77
Figura 26 - Diagrama de Gantt Intercalar 2ºSemeste.....	78
Figura 27 - Projeto a correr numa máquina nova	80
Figura 28 - Docker compose máquina nova	80
Figura 29 - Web site a correr máquina nova	81
Figura 30 - Criar Grupo COG Pré Alterações.....	82
Figura 31 - Criar Grupo COG Pós Alterações	82
Figura 32 - View Exportação CSV Parte 1	83
Figura 33 - View Exportação CSV Parte 2	84
Figura 34 - CSV Protocolo MentHA 1.....	84
Figura 35 - CSV Protocolo MentHA 2.....	85
Figura 36 - Página principal do MentHA COG.....	109
Figura 37 - Página de Grupo do MentHA COG	110
Figura 38 - Página de Sessões do Grupo MentHA COG	110
Figura 39 - Página de Sessão do MentHA COG	110
Figura 40 - Página de uma Atividade de uma Sessão do MentHA COG	111
Figura 41 - Página de perfil de participantes do Protocolo MentHA.....	112
Figura 42 - Página de Avaliação MentHA-Risk do MentHA COG	112
Figura 43 - Página de Visão Geral do Protocolo MentHA.....	112
Figura 44 - Página de Questionário de uma Avaliação do Protocolo MentHA	113
Figura 45 - Página principal do MentHA CARE	114
Figura 46 - Página do Grupo do MentHA CARE	114
Figura 47 - Página de Sessões do Grupo MentHA CARE.....	115

Figura 48 - Página de Sessão do MentHA CARE	115
Figura 49 - Página de uma Atividade de uma Sessão do MentHA CARE	115
Figura 50 - Casos de Uso do Dinamizador MentHA CARE	116
Figura 51 - Casos de Uso do Administrador no MentHA CARE.....	117
Figura 52 - Casos de Uso relativos ao Protocolo MentHA	118
Figura 53 - Diagrama ERD MentHA CARE/COG	119
Figura 54 – Diagrama ERD parte 1.....	120
Figura 55 - Diagrama ERD parte 2	120
Figura 56 - Diagrama ERD parte 3	121
Figura 57 - WorkFlow 1	123
Figura 58 - WorkFlow 2	123
Figura 59 - WorkFlow 3	124
Figura 60 - WorkFlow 4	124
Figura 61 - WorkFlow 5	124
Figura 62 - WorkFlow 6	125
Figura 63 - WorkFlow 7	125
Figura 64 - Template de Documentação	128
Figura 65 - Exemplo de Documentação.....	129
Figura 66 - README Guia de Instalação e Execução	129
Figura 67 - Ficheiro .gitignore.....	132
Figura 68 - Docker compose.yml Parte 1.....	134
Figura 69 - Docker compose.yml Parte 2.....	135
Figura 70 - Dockerfile	135

Lista de Tabelas

Tabela 1 - Tabela de Análise de Benchmarking	9
Tabela 2 - Tabela de features	23
Tabela 3 - User Stories referentes á feature E1FMD (MentHA Digital Web Site)	25
Tabela 4 - User Stories referentes á feature E1FPT (Protocolo MentHA)	26
Tabela 5 - User Stories referentes á feature E1FCOG (MentHA COG).....	27
Tabela 6 - User Stories referentes á feature E2DR (Depuração do Repositório)	29
Tabela 7 - User Stories referentes á feature E2AD (Ambiente de Desenvolvimento)	29
Tabela 8 - User Stories referentes á feature E2AP (Ambiente de Produção).....	30
Tabela 9 - Lista de Requisitos Referentes ao MentHA Digital	87
Tabela 10 - Lista de Requisitos Referentes ao Protocolo MentHA.....	88
Tabela 11 - Lista de Requisitos Referentes ao MentHA COG	91
Tabela 12 - Lista de Requisitos Referentes à Depuração do Repositório	93
Tabela 13 - Lista de Requisitos Referentes ao Ambiente de Desenvolvimento	95
Tabela 14 - Lista de Requisitos Referentes ao Ambiente de Produção.....	96

Lista de Siglas

TFCs	Trabalhos Finais de Curso
HTML	Linguagem de Marcação de Hipertexto
CSS	Cascading Style Sheets
AJAX	Assynchronous JavaScript and XML
LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
Git	Sistema de Controlo de Versões

1 Introdução

1.1 Enquadramento

O Projeto MentHA (Mental Health and Aging) foi um empreendimento colaborativo de investigadores e instituições público-privadas, sem fins lucrativos, que decorreu entre os anos de 2015 e 2017 que teve como finalidade avaliar os efeitos da reabilitação neuropsicológica em adultos mais velhos com défice cognitivo, demência leve ou transtornos dissociativos de identidade, bem como apoiar os seus cuidadores formais e informais.

Através de um concurso promovido pela DGS em novembro de 2021, foi considerado interessante e propício os docentes Lúcio Studer Ferreira e Pedro Machado Santos avançarem, em nome da Universidade Lusófona - Centro Universitário Lisboa com uma candidatura, a qual dava o conhecimento do projeto Digital MentHA – “Mental Health and Aging”, com o objetivo de fazer a transformação digital dos produtos MentHA. A candidatura foi vencedora, tendo o resultado sido comunicado em janeiro de 2022.

Este projeto é uma iniciativa conjunta de investigadores do ICS (Instituto de Serviço Social) e do DEISI-COPELABS (Departamento de Engª Informática e Sistemas Informação) e tem vindo a ser desenvolvido no âmbito de Trabalhos Final de Cursos (TFC) dos alunos de DEISI da Universidade Lusófona de Humanidades e Tecnologia nos últimos 5 anos letivos. Com este projeto pretende-se que a DGS disponibilize a Plataforma aplicacional, em todo o país, às instituições e técnicos da área, potencializando a viabilidade deste projeto em grande escala.

Neste projeto foram realizados oito protocolos de parceria com entidades renomeadas na área:

- **ADEB** [Adeb24] [Adeb24][Adeb24][Adeb24] – Associação de Apoio aos Doentes Depressivos e Bipolares
- **ASMAL** [Asmal24] – Associação de Saúde Mental do Algarve
- **Elo Social** [Elo24] – Associação para a Integração e Apoio ao Deficiente jovem e adulto
- **CVP** [Cvp24] – Cruz Vermelha Portuguesa
- **FamiliarMente** [Fame24] – Federação Portuguesa das Associações das famílias de pessoas com experiência de Doença Mental
- **GIRA** [Gira24] – Grupo de Intervenção e Reabilitação Ativa
- **Crinabel** [Crl24] – Lar residencial e Centro de Atividades Ocupacionais
- **APDP** [Apdp24] – Associação Protetora dos diabéticos de Portugal

Os logótipos dos nossos parceiros estão representados na **Figura 1**.



Figura 1 - Logótipos dos nossos Parceiros

Este conjunto de parceiros assegura uma amostra vasta de representatividade que se estende no território nacional do grupo-alvo do projeto. Pretende-se a continuidade de colaboração entre estas seis parcerias e de futuras parcerias, de diferentes localidades em todo o território nacional.

A plataforma MentHA é composta por 3 aplicações que estão explicadas ao detalhe no **Anexo A – Aplicações MentHA Digital Motivação e Identificação do Problema**

O envelhecimento da população é uma realidade crescente em muitos países, incluindo Portugal, trazendo consigo desafios significativos para o sistema de saúde e para a sociedade como um todo. Entre esses desafios, destaca-se o aumento da prevalência de défices cognitivos e demências em adultos mais velhos, que afetam diretamente a qualidade de vida dos indivíduos e de seus cuidadores.

A transformação digital tem emergido como uma solução promissora para enfrentar esses problemas, promovendo maior acessibilidade, eficiência e abrangência nas intervenções de saúde mental e estimulação cognitiva. É nesse contexto que surge o projeto MentHA Digital – uma evolução do projeto MentHA original, desenvolvido entre 2015 e 2017, que visava a reabilitação neuropsicológica e o apoio a cuidadores. A versão digital tem como objetivo ampliar o alcance e o impacto dessas intervenções, permitindo que instituições e técnicos da área utilizem uma plataforma integrada e escalável em todo o território nacional.

As várias aplicações da plataforma MentHA tem vindo a ser desenvolvidas no âmbito de Trabalhos Final de Cursos (TFC) dos alunos de DEISI da Universidade Lusófona de Humanidades e Tecnologia nos últimos 5 anos letivos estando já as aplicações desenvolvidas de forma avançada e já se encontrando integradas. Segue-se a linha cronológica do desenvolvimento das aplicações ao longo dos últimos cinco anos.

- **2020/2021 - Desenvolvimento inicial da Plataforma MentHA**
 - **Plataforma MentHA:** [TFC21a] Afonso Rodrigues desenvolveu a base da Plataforma MentHA, estabelecendo as fundações de todas as aplicações.
- **2021/2022 - Desenvolvimento das Aplicações MentHA CARE e do Protocolo MentHA**
 - **Protocolo MentHA:** [TFC22a] Nuno Rasteiro deu continuidade ao desenvolvimento da Aplicação Web do Protocolo MentHA.
 - **MentHA CARE:** [TFC22b] Inês Rocha e João Santos deram continuidade ao desenvolvimento da Aplicação MentHA CARE.
- **2022/2023 - Expansão das Aplicações MentHA CARE e do Protocolo MentHA**
 - **MentHA CARE:** [TFC23a] Tiago Silva e Rodrigo Eira aprimoraram a Aplicação MentHA CARE, incorporando funcionalidades adicionais e melhorias.
 - **Protocolo MentHA:** [TFC23b] Bernardo Roque e Alexandre Godinho avançaram no desenvolvimento da Aplicação Web do Protocolo MentHA, aumentando a sua capacidade de análise de questionários.
- **2023/2024 - Consolidação da Plataforma MentHA**
 - **Plataforma MentHA:** [TFC24a] Miguel Cardoso e António Ventura trabalharam na Plataforma MentHA, consolidando as diferentes aplicações (MentHA COG,

MentHA CARE e Protocolo MentHA) em um ambiente integrado.

- **2024/2025 - Consolidação da Plataforma MentHA (Atual)**

- **Plataforma MentHA:** Nós, Tomás Nave e André Jesus, temos o objetivo de continuar a consolidar as diferentes aplicações (MentHA COG, MentHA CARE e Protocolo MentHA).

A motivação para o desenvolvimento deste trabalho está profundamente enraizada na relevância social e na necessidade prática de criar ferramentas tecnológicas que enfrentem os desafios da saúde mental na sociedade atual. Paralelamente, há uma motivação técnica e acadêmica, já que o projeto oferece uma oportunidade única de aplicar conhecimentos de Engenharia informática para resolver problemas concretos, contribuindo tanto para o bem-estar social quanto para o avanço da tecnologia aplicada à saúde. Além disso, ambos carregamos uma motivação pessoal, fruto de experiências vividas com familiares, idosos ou não, que enfrentaram doenças cognitivas. Essas vivências permitiram-nos testemunhar de perto o impacto profundamente negativo dessas condições na vida das pessoas, reforçando o nosso compromisso em desenvolver soluções que façam a diferença.

O problema identificado está relacionado à necessidade de corrigir falhas e desenvolver funcionalidades adicionais para melhorar a experiência do utilizador e dar continuidade ao aprimoramento de todas as aplicações. Embora as diferentes componentes da plataforma – MentHA COG, MentHA CARE e Protocolo MentHA – já estejam desenvolvidas e integradas, como descrito na linha cronológica, elas ainda apresentam limitações em termos de usabilidade, eficiência e erros, com destaque para o MentHA COG, que é a aplicação menos desenvolvida e será o foco principal deste trabalho.

1.2 Objetivos

Após a apresentação do problema, na **Secção 1.1**, é descrito o objetivo do MentHA Digital. O projeto MentHA visa melhorar a qualidade de vida de adultos mais velhos com défices cognitivos e doenças mentais crónicas, bem como a dos seus cuidadores informais, promovendo o acesso remoto a intervenções psicoeducativas, protocolos de avaliação neuropsicológica e programas de estimulação cognitiva. O objetivo global do projeto MentHA Digital é desenvolver soluções que contribuam para o bem-estar e autonomia dos pacientes, com base em evidências científicas, proporcionando também suporte educativo e emocional aos cuidadores para uma melhor gestão de quadros clínicos específicos.

Além do objetivo geral do projeto, cada uma das três aplicações que o compõem possui um propósito específico e direcionado. Os objetivos detalhados de cada aplicação estão descritos no **Anexo A – Aplicações MentHA Digital**

1.2.1 Objetivo Inicial

Inicialmente, neste projeto, assumimos como principal responsabilidade a **correção de erros existentes** e a **melhoria contínua** das várias componentes das aplicações do projeto MentHA. Durante a utilização das versões anteriores, foram detetados erros que comprometiam

diretamente a experiência do utilizador. Destacamos alguns erros críticos, como a **impossibilidade de efetuar logout**, o que levanta sérias preocupações a nível de segurança e privacidade de dados.

Na aplicação MentHA COG, verificaram-se falhas ao tentar atribuir um grupo a um participante sem grupo, resultando num erro que impede a gestão adequada dos utilizadores. Além disso, não é possível visualizar os membros que integram um grupo, nem consultar os materiais necessários para cada tipo de sessão, o que dificulta o planeamento e a execução das atividades.

Já na aplicação MentHA CARE, identificou-se que não é possível **editar as informações dos cuidadores**, o que limita a atualização e correção de dados relevantes. Estes problemas reforçam a importância da intervenção proposta neste projeto, destacando a necessidade urgente de uma solução mais robusta, funcional e fiável.

Ficou então decidido que o nosso **foco principal** seria a aplicação MentHA COG, pois é aquela que apresenta atualmente o maior número de desafios. Sendo a aplicação menos desenvolvida e ainda não testada com utilizadores finais no contexto das parcerias, requer uma intervenção profunda.

O nosso objetivo é garantir que o MentHA COG atinja um nível de maturidade técnica e funcional que permita a sua utilização eficaz por cuidadores e profissionais de saúde. Isto inclui a correção de erros críticos, a otimização do desempenho, a melhoria da interface e a implementação de novas funcionalidades com base nas necessidades identificadas durante as sessões de parceria.

1.2.2 Mudança de Foco e Objetivos Até à Data

No entanto, durante o desenvolvimento do projeto, surgiram diversos obstáculos técnicos e organizacionais, que obrigaram a uma mudança significativa de foco e redirecionamento dos nossos esforços. Ao contrário do que se esperava num projeto de TFC tradicional, não partimos de um sistema novo, mas sim de um conjunto de aplicações complexas já existentes, desenvolvidas ao longo de várias gerações de equipas.

A herança de um sistema de grandes dimensões, com código desorganizado, falta de documentação, problemas no repositório e inconsistências entre o código em produção e o código presente no Git, impôs **dificuldades substanciais logo nas fases iniciais**. Foram identificados problemas como:

- Impossibilidade de executar corretamente a aplicação localmente;
- Configurações desatualizadas e dependências obsoletas;
- Código em produção com alterações não refletidas no repositório;
- Complexidade técnica que dificultou a introdução de novas funcionalidades.

Estas dificuldades exigiram uma **redefinição profunda do nosso papel no projeto**, que passou de simples manutenção e evolução das aplicações, para uma reestruturação e estabilização técnica da base existente.

Uma descrição detalhada de todos os problemas enfrentados e das soluções adotadas ao longo do desenvolvimento encontra-se na **Secção 4.6** deste relatório.

Até ao momento, o nosso foco esteve centrado principalmente na **resolução destes problemas estruturais**, criando uma base técnica mais organizada, funcional e robusta, de forma a garantir que qualquer programador consiga, a partir de agora, clonar o repositório e iniciar o desenvolvimento sem erros e sem ter de enfrentar as mesmas dificuldades por que passámos.

Hoje, a maioria desses problemas encontra-se resolvida, e consideramos que o projeto se encontra **numa fase significativamente mais estável do ponto de vista técnico**.

Apesar do grande esforço dedicado à estabilização, não perdemos de vista o objetivo inicial, tendo sido possível realizar algumas melhorias pontuais na aplicação MentHA COG e desenvolver uma nova funcionalidade no **Protocolo MentHA**.

1.2.3 Objetivo Final

Com a maior parte dos problemas estruturais resolvidos e o repositório consolidado numa versão estável e funcional, os esforços recentes centraram-se em duas vertentes essenciais de evolução do projeto.

Em primeiro lugar, concretizámos com sucesso a reformulação da interface visual da plataforma MentHA Digital, tornando-a mais moderna, intuitiva e visualmente apelativa. Esta modernização, baseada nos protótipos previamente desenvolvidos, seguiu as melhores práticas de User Experience (UX) e Design Centrado no Utilizador, contribuindo para uma melhor experiência dos utilizadores finais e aumentando o potencial de adesão ao projeto por parte de novos parceiros e instituições.

Paralelamente, foi igualmente desenvolvido e implementado um pipeline automatizado de deploy em produção, alinhado com práticas modernas de **DevOps** e com princípios de **Integração Contínua (CI)** e **Entrega Contínua (CD)**. Este pipeline, suportado por **GitHub Actions**, permite que todas as alterações submetidas ao repositório sejam automaticamente sujeitas a um conjunto de testes automatizados em ambiente isolado, assegurando a validação contínua da qualidade do código e a deteção precoce de regressões. Uma vez aprovadas, as alterações são imediatamente integradas no ambiente de produção de forma controlada, auditável e segura. Esta abordagem reduz significativamente a intervenção manual, diminui o risco de erro humano e assegura a rastreabilidade total de cada atualização efetuada no sistema.

Do ponto de vista técnico, a infraestrutura do projeto foi consolidada segundo o paradigma de **Infraestrutura como Código (IaC)**, com uma arquitetura robusta baseada em **Docker** e **Docker Compose**. Esta abordagem permitiu o isolamento de serviços, a gestão centralizada de variáveis de ambiente, a criação de volumes persistentes de dados e a replicação fiável dos ambientes de desenvolvimento, testes e produção. Adicionalmente, foi implementada uma camada de proteção reforçada com um **reverse proxy NGINX**, complementado por um servidor **WSGI profissional (Gunicorn)** para gestão eficiente de múltiplos pedidos concorrentes.

A componente de segurança operacional foi também uma preocupação central, com a implementação de práticas como a gestão de segredos (através do GitHub Secrets), a separação de ambientes, e a proteção de endpoints HTTP com headers de segurança e compressão aplicados via NGINX.

Este projeto proporcionou-nos uma aprendizagem prática e contínua na aplicação dos princípios de **cultura DevOps**, promovendo a colaboração entre desenvolvimento e operações, a automação de processos críticos, a melhoria contínua e a entrega rápida de valor com qualidade assegurada. Adquirimos experiência concreta na gestão de **ciclos de vida completos de aplicações web** em ambientes colaborativos, na orquestração de servidores remotos, e na implementação de pipelines de **Continuous Delivery** alinhados com padrões profissionais.

A documentação exaustiva dos procedimentos e ferramentas implementadas proporciona uma base sólida para que futuras equipas possam dar continuidade ao projeto, com a possibilidade de expandir ou migrar a infraestrutura, tirando partido das soluções de automação e escalabilidade já desenvolvidas.

1.3 Estrutura do Documento

O presente documento estrutura-se da seguinte forma:

- **Secção 1 - Introdução**
- **Secção 2 - Pertinência e Viabilidade**
- **Secção 3 - Especificação e Modelação**
- **Secção 4 - Solução**
- **Secção 5 - Testes e Validação**
- **Secção 6 - Método e Planeamento**
- **Secção 7 - Resultados**
- **Secção 8 - Conclusão**
- **Anexo A – Aplicações MentHA Digital**
- **Anexo B – Casos de Uso**
- **Anexo C – Modelo ERD MentHA COG / MentHA CARE**
- **Anexo D - Workflow de Desenvolvimento**
- **Anexo E - Documentação**
- **Anexo F - Importância do gitignore**
- **Anexo G – Configuração do Ambiente de Desenvolvimento com Docker e Docker Compose**

2 Pertinência e Viabilidade

2.1 Pertinência

O projeto “MentHA Digital” surgiu como uma candidatura no âmbito do Programa Nacional para a Saúde Mental, com o objetivo de garantir o acesso equitativo a cuidados de qualidade para adultos idosos com problemas de saúde mental em Portugal. A candidatura foi coordenada pela COFAC e liderada pelos docentes **Pedro Machado dos Santos** e **Lúcio Studer Ferreira**, tendo sido aprovada em janeiro de 2022. Este projeto consiste na transformação digital do MentHA Digital, integrado na candidatura anterior, visando disponibilizar a plataforma em todo o território nacional para instituições e profissionais da área de saúde mental. A viabilidade do projeto é reforçada pela experiência acumulada em quatro projetos anteriores já aprovados, mencionados na **Secção 1.2**. Além disso, a parceria com diversas entidades de bastante referência na área da saúde mental reforça a solidez e a relevância prática desta iniciativa.

O projeto é considerado pertinente devido à sua importância para a comunidade que apoia pessoas com défice cognitivo. O interesse e a curiosidade demonstrados pelas organizações parceiras, reforça a pertinência deste projeto destacando a necessidade de ajustar o projeto com base nas críticas e sugestões obtidas. O Projeto MentHA Digital apesar de já ter alguns anos de desenvolvimento continua a ser considerado pertinente pois a aplicação ainda se encontra com alguns erros, e, das três aplicações principais, o MentHA COG é o que apresenta mais problemas e desafios, pois é uma aplicação inacabada, pouco desenvolvida e nunca testada.

Em suma, o projeto “MentHA” Digital destaca-se pela sua relevância e pelo seu potencial impacto na melhoria dos cuidados em saúde mental para adultos idosos com défice cognitivo em Portugal. A combinação da experiência acumulada em projetos anteriores, a colaboração com entidades de referência e o compromisso com a transformação digital garantem uma base sólida para o seu desenvolvimento contínuo. Apesar dos desafios existentes, especialmente no MentHA COG, a aplicação apresenta uma oportunidade significativa de inovação ao focar-se na correção de erros e no desenvolvimento de funcionalidades ainda não testadas. A pertinência do projeto reside na sua capacidade de se adaptar às necessidades das instituições parceiras e dos utilizadores finais, promovendo a inclusão digital, o acesso equitativo e a melhoria da qualidade de vida dos seus beneficiários.

2.2 Viabilidade

A viabilidade do projeto MentHA Digital está sustentada em critérios técnicos, económicos, sociais e ambientais que garantem a sua implementação e continuidade para além do âmbito académico.

Em termos de **viabilidade técnica**, a solução proposta baseia-se na transformação digital de projetos anteriores bem-sucedidos, como o MentHA COG, MentHA CARE e o Protocolo MentHA. A experiência acumulada em quatro projetos pré-aprovados, conforme descrito na linha cronológica, e a utilização de tecnologias já validadas reforçam a capacidade técnica para a execução do projeto. Testes realizados já com pacientes em anos anteriores têm demonstrado potencial para melhorar a experiência do utilizador final.

Na questão de **viabilidade económica** o projeto foi inicialmente financiado pela Direção-Geral de Saúde (DGS), o que assegura os recursos necessários para o seu desenvolvimento e implementação. Além disso, o apoio de várias organizações de referência reforça a viabilização da implementação da solução em instituições de todo o território nacional, objetivo do nosso projeto não é ser comercializado é apenas ser usado pela Direção-Geral de Saúde (DGS).

No que toca a **viabilidade social** o interesse demonstrado pelas organizações parceiras reflete a necessidade e o apoio à implementação do projeto. A inclusão de feedback de utilizadores finais, como o fornecido pela ADEB no ano anterior, assegura que as melhorias e funcionalidades desenvolvidas respondem às expectativas e preferências do público-alvo.

A propósito do alinhamento com os Objetivos de Desenvolvimento Sustentável (ODS), o projeto alinha-se diretamente com os ODS 3 (Saúde de Qualidade) e ODS 10 (Redução das Desigualdades). A disponibilização de uma solução tecnológica inclusiva promove o acesso equitativo a cuidados de saúde mental, contribuindo para a redução das desigualdades regionais e socioeconómicas em Portugal.

Ao garantir uma solução técnica robusta, economicamente viável e socialmente relevante, o projeto “MentHA Digital” tem potencial para transcender os limites de um trabalho académico, consolidando-se como uma ferramenta essencial na importância e cuidado da saúde mental em Portugal.

2.3 Análise Comparativa com Soluções Existentes

2.3.1 Soluções existentes

Neste capítulo analisámos alguns possíveis concorrentes presentes no mercado que apresentam soluções existentes para o problema proposto pelo nosso projeto.

Foram identificadas algumas organizações que já trabalham com doentes cognitivos:

- A organização Cogweb [**COWE24**] que disponibiliza aos utilizadores com défice cognitivo vários planos de treino que estimulam os pacientes, os treinos são concebidos e orientados por profissionais da área através da plataforma online.
- A instituição Humana Mente [**HUME24**] dedica-se a promover atividades e ocupações destinadas a pessoas idosas com demência, com o objetivo de melhorar a sua saúde e qualidade de vida social. Para tal, recorre a atividades como montagem de puzzles, aquapaints e outros passatempos que estimulam e exercitam as capacidades cognitivas e sensoriais dos pacientes.
- A Sioslife [**SILI24**] dedica-se à realização de atividades cognitivas online direcionadas a indivíduos com demência, com o objetivo de atenuar e retardar os efeitos da doença.
- A CogniFit [**COFI24**] oferece aos seus utilizadores uma ampla variedade de exercícios digitais, concebidos para estimular as capacidades cognitivas e potenciar o desenvolvimento cerebral. A plataforma permite ainda realizar o acompanhamento personalizado dos progressos dos utilizadores.

2.3.2 Análise de benchmarking

Conseguimos concluir que a aplicação MentHA Digital é mais dinâmica e que se diferencia das outras organizações referidas neste capítulo, pois agrupa todas as funcionalidades principais de cada plataforma apresentada e ainda acrescentando a funcionalidade de gerar Relatórios Automáticos sobre as atividades realizadas.

Tabela 1 - Tabela de Análise de Benchmarking

	Atividades Individuais	Atividades em Grupo	Acompanhamento durante as sessões	Acompanhamento da evolução dos pacientes	Gratuito	Relatórios Automáticos
MentHA Digital	X	X	X	X	X	X
Sioslife					X	
Cogweb	X		X	X		
CogniFit	X				X	

2.4 Proposta de inovação e mais-valias

O projeto **MentHA Digital** distingue-se pela sua proposta inovadora de integração tecnológica e pela capacidade de trazer benefícios significativos para a área da saúde mental em Portugal. Entre as principais inovações e mais-valias destacam-se:

- **Transformação Digital de Programas Existentes:**
 - O projeto aposta na digitalização de um conjunto de programas previamente aprovados e testados, como o MentHA COG, MentHA CARE e Protocolo MentHA. Esta transformação digital não apenas otimiza os processos existentes, como também expande o alcance dos serviços, tornando-os mais eficientes e acessíveis.
- **Disponibilização a Nível Global:**
 - Embora inicialmente desenvolvido para instituições em território nacional, o potencial de escalabilidade do MentHA Digital é uma mais-valia. A plataforma está desenhada para ser adaptável e replicável em outros contextos internacionais, promovendo o acesso global a uma solução tecnológica inovadora para a saúde mental.
- **Inovação no Enfoque Multidimensional:**
 - A plataforma combina abordagens baseadas em **interatividade digital** focada numa experiência de utilizador otimizada e inclusiva, **integração de feedback** que possui adaptações contínuas com base nas necessidades reais de pacientes, cuidadores e profissionais de saúde e por fim, **personalização de cuidados** que engloba ferramentas ajustáveis às particularidades de cada utilizador e contexto.
- **Impacto Social e Digital:**

- O projeto reforça o conceito de **inclusão digital**, permitindo que populações vulneráveis, como idosos com défices cognitivos, acedam a ferramentas modernas de apoio à saúde mental. A transformação digital proposta não apenas melhora a experiência do utilizador final, mas também facilita o trabalho de cuidadores formais e informais.
- **Alinhamento com os Objetivos de Desenvolvimento Sustentável (ODS)**
 - Contribuição direta para o ODS 3 (Saúde de Qualidade), assegurando acesso equitativo a cuidados de saúde mental.
 - Apoio ao ODS 10 (Redução das Desigualdades), reduzindo as barreiras socioeconómicas e regionais para o acesso a serviços essenciais.

2.5 Identificação de oportunidade de negócio

O projeto **MentHA Digital** identifica-se como uma iniciativa de **empreendedorismo social**, cuja principal missão é promover a inclusão digital e melhorar o acesso a cuidados de saúde mental em Portugal. Embora o projeto não tenha como objetivo ser comercializado, apresenta uma oportunidade significativa para impactar positivamente a sociedade e transformar a forma como os cuidados de saúde mental são prestados.

O MentHA Digital foi concebido para ser disponibilizado gratuitamente, com foco exclusivo no uso por instituições públicas e organizações da área da saúde mental, sob coordenação da Direção-Geral de Saúde (DGS). Este modelo reforça o compromisso com uma missão não comercial, colocando o impacto social como prioridade. A plataforma responde a necessidades reais da população, promovendo inclusão e melhoria na qualidade de vida, especialmente para os mais vulneráveis.

Além disso, o projeto destaca-se pelo seu potencial de impacto. Responde a uma necessidade essencial de acesso a soluções tecnológicas inclusivas, oferecendo suporte a populações com défices cognitivos e aos seus cuidadores formais e informais. A cooperação com organizações de referência e entidades públicas representa uma oportunidade para ampliar o alcance do projeto. Apesar de ter sido desenvolvido no contexto português, o MentHA Digital apresenta características de escalabilidade e replicabilidade, permitindo a sua adaptação a outros países e contextos, o que reforça a sua relevância a nível global.

A sustentabilidade do projeto é garantida por um modelo sem fins lucrativos, baseado em financiamento público e em parcerias estratégicas. Apoios institucionais, como o financiamento inicial da DGS, asseguram os recursos necessários para a sua implementação e manutenção, eliminando a dependência de receitas comerciais. Além disso, a sua integração com políticas públicas nacionais alinha o MentHA Digital com as estratégias para a saúde mental, contribuindo para soluções sustentáveis a longo prazo e fortalecendo os cuidados de saúde mental em Portugal.

3 Especificação e Modelação

3.1 Análise de Requisitos

Este capítulo tem como objetivo a análise dos requisitos identificados pela equipa para garantir o desenvolvimento e o sucesso do projeto. Os requisitos apresentados resumem as funcionalidades e necessidades da aplicação, servindo como base fundamental para o seu desenvolvimento e constituindo fatores cruciais a serem considerados ao longo de todo o processo. Nesta fase inicial, os requisitos descritos nas secções seguintes foram divididos entre aqueles levantados no ano anterior, no projeto **[TFC24a]**, e os identificados através de um levantamento preliminar de requisitos. Atualmente, estes requisitos estão maioritariamente centrados na identificação e correção de erros nas aplicações existentes. No entanto, a médio e longo prazo, o objetivo é realizar um levantamento mais concreto, incluindo reuniões presenciais com as entidades parceiras e, sempre que possível, com os seus pacientes, para compreender melhor os desafios enfrentados pelos cuidadores no uso da aplicação e recolher feedback direto durante as interações com os doentes.

3.1.1 Enumeração de Requisitos

Os requisitos são identificados e codificados com as mesmas codificações das user stories correspondentes a cada requisito (Ex: **E1FCOGU1** – Épico 1, Feature COG, UserStory 1). As user stories correspondentes a estes requisitos e a maneira como estão estruturadas estão explicados á frente na **Secção 3.1.2**. Para cada requisito estão inclusas uma descrição e uma pré-condição caso exista. Também se colocou o estado do requisito para sabermos em qual trabalho e a prioridade do mesmo, foi adicionado também às colunas uma relação com o tempo e se o requisito já se encontra implementado ou não.

Nas secções seguintes estão apresentados de forma separada e organizada os requisitos referentes a cada uma das features. Para cada requisito estão presentes uma descrição, a importância, a estimativa de esforço e o estado atual do requisito.

3.1.1.1 Lista de Requisitos Referentes ao MentHA Digital

- **E1FMDU1 – Erro a fazer log-out:**
 - **Descrição:** Na aplicação do MentHA, quando se inicia sessão não se consegue fazer Log- out.
 - **Importância:** Este requisito é de importância "Must have" e requer o trabalho de 1 pessoa e a duração prevista de 3 horas.
 - **Estado:** Concluído

- **E1FMDU2 – Corrigir erros ortográficos na web site MentHA:**
 - **Descrição:** No website MentHA, deve ser garantido que não existam erros ortográficos em qualquer conteúdo exibido nas páginas do site.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 4 horas.
 - **Estado:** Concluído

- **E1FMDU3 - Criação de Protótipo em Figma como sugestão de melhoria visual do web site MentHA digital:**
 - **Descrição:** A equipa deve desenvolver um protótipo de interface no Figma, com o objetivo de propor melhorias visuais e de usabilidade para o website MentHA Digital, respeitando a identidade visual do projeto e as necessidades dos utilizadores.
 - **Importância:** Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.
 - **Estado:** Concluído
- **E1FMDU4 – Alteração da Interface Visual do Website MentHA Digital com base no Protótipo**
 - **Descrição:**

A equipa deve implementar alterações na interface visual do website MentHA Digital, com base no protótipo desenvolvido no Figma. Estas alterações devem refletir as melhorias de usabilidade e estética propostas, assegurando a coerência com a identidade visual do projeto e a acessibilidade para os utilizadores.
 - **Importância:** Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 6 horas.
 - **Estado:** Concluído
- **E1FMDU5 – Adaptação Responsiva da Página Inicial para Resoluções Reduzidas e Dispositivos Móveis**
 - **Descrição:**

A equipa deve adaptar a página inicial do website MentHA Digital para garantir uma apresentação visual adequada em dispositivos com resoluções reduzidas, como portáteis com ecrãs pequenos, tablets e telemóveis. A implementação deve assegurar uma navegação fluida e responsiva, respeitando as boas práticas de design mobile-first, mantendo a coerência com o layout do protótipo original e garantindo acessibilidade e legibilidade em todos os tamanhos de ecrã.
 - **Importância:** Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 5 horas.
 - **Estado:** Concluído

3.1.1.2 Lista de Requisitos Referentes ao Protocolo MentHA

- **E1FPTU1 – Registar com o mesmo nome de utilizador deve dar um erro com alguma informação:**
 - **Descrição:** Na aplicação do MentHA EVAL, ao registar um Utilizador, se registamos um Participante com o mesmo Username de outro Participante, este dá apenas um prompt JavaScript.
 - **Importância:** Este requisito é de importância "Should have" e requer o trabalho de 1 pessoa e a duração prevista de 3 horas.
 - **Estado:** Pendente
- **E1FPTU2 – Relatório Gerado corretamente no MentHA EVAL:**
 - **Descrição:** Na aplicação do MentHA EVAL, o relatório deve ser gerado corretamente com os dados todos corretos.
 - **Importância:** Este requisito é de importância "Must have" e requer o trabalho de 1 pessoa e a duração prevista de 3 horas.
 - **Estado:** Pendente
- **E1FPTU3 – No Protocolo MentHA adicionar uma parte relacionada a um consentimento autorizado:**
 - **Descrição:** No Protocolo MentHA deve-se realizar uma parte relacionada a um consentimento autorizado para os pacientes.
 - **Importância:** Este requisito é de importância "Should Have" e exige o trabalho de 2 pessoas e a duração prevista de 4 horas.
 - **Estado:** Pendente
- **E1FPTU4 – Os campos opcionais no formulário de registo de participantes devem estar vazios por padrão e usar labels:**
 - **Descrição:** No Protocolo MentHA, os campos opcionais não devem ter valores predefinidos, devendo ser apresentados como campos vazios e devidamente identificados com labels.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.
 - **Estado:** Pendente
- **E1FPTU5 – No Protocolo MentHA, adicionar um aviso visual na avaliação MentHA Risk de dados em falta:**
 - **Descrição:** No Protocolo MentHA, deve ser exibido um aviso visual claro na lista de avaliações MentHA Risk indicando que o teste foi submetido com campos por preencher e está incompleto.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.
 - **Estado:** Pendente

- **E1FPTU6 – Corrigir erro ao gerar relatório de avaliação no Protocolo MentHA:**
 - **Descrição:** No Protocolo MentHA, ao tentar gerar o relatório de uma avaliação, ocorre um erro que resulta na exibição de uma página não disponível. Deve ser corrigido este erro, garantindo que o sistema consiga gerar o relatório de forma eficaz e sem falhas, assegurando a continuidade da funcionalidade.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 6 horas.
 - **Estado:** Pendente
- **E1FPTU7 – No Protocolo MentHA, corrigir a barra de progresso do questionário sociodemográfico no MentHA Risk:**
 - **Descrição:** No Protocolo MentHA, deve ser corrigida a barra de progresso do questionário sociodemográfico no MentHA Risk para garantir que ela funcione corretamente e acompanhe o progresso do utilizador durante a avaliação.
 - **Importância:** Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.
 - **Estado:** Pendente
- **E1FPTU8 – Exportação de Dados Estatísticos em CSV:**
 - **Descrição:** O sistema deve permitir aos utilizadores com perfil de cuidador aceder e exportar, em formato CSV, dados estatísticos referentes a todas as avaliações realizadas pelos pacientes, no âmbito do protocolo MentHA.
 - **Importância:** Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 2 horas.
 - **Estado:** Concluído
- **E1FPTU9 – Inclusão da Informação da Instituição no CSV de Análise de Dados:**
 - **Descrição:** O sistema deve incluir, no ficheiro CSV de análise de dados exportado pelos cuidadores, a informação relativa à instituição a que cada paciente pertence, de forma a permitir uma análise segmentada por instituição.
 - **Importância:** Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.
 - **Estado:** Pendente

3.1.1.3 Lista de Requisitos Referentes ao MentHA COG

- **E1FCOGU1 - Visualização do Grupo COG:**
 - **Descrição:** No MentHA COG, ao criar um Grupo não se consegue visualizar os Constituintes de cada sessão.
 - **Importância:** Este requisito é de importância "Should have" e requer o trabalho de 1 pessoa e duração prevista de 3 horas.
 - **Estado:** Pendente

- **E1FCOGU2 – MentHA COG, Materiais da sessão não são mostrados com sucesso:**
 - **Descrição:** No MentHA COG, os materiais da sessão não aparecem de todo, mas já se consegue mostrar os materiais dentro de outra página.
 - **Importância:** Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração de 8 horas.
 - **Estado:** Parcial
- **E1FCOGU3 – MentHA COG, barra de progresso não funciona:**
 - **Descrição:** Na aplicação MentHA COG, a barra de progresso no sociodemográfico não funciona corretamente.
 - **Importância:** Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração de 8 horas.
 - **Estado:** Pendente
- **E1FCOGU4 – Caderno de cotação MentHA COG não finalizado:**
 - **Descrição:** Na aplicação, O caderno de estímulos deve ser implementado (não está na aplicação).
 - **Importância:** Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração prevista de 10 horas.
 - **Estado:** Pendente
- **E1FCOGU5 – Criar avaliações no MentHA COG:**
 - **Descrição:** Na aplicação criar uma opção que permitirá o Orientador das sessões criar e submeter avaliações do MentHA COG.
 - **Importância:** Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração prevista de 8 horas.
 - **Estado:** Pendente
- **E1FCOGU6 – Feedback ao guardar no Diário de Bordo:**
 - **Descrição:** Na aplicação, o Dinamizador deve receber algum tipo de feedback para saber se as alterações foram realizadas com sucesso ou não.
 - **Importância:** Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração prevista de 9 horas.
 - **Estado:** Pendente
- **E1FCOGU7 – Submeter avaliações do MentHA COG com sucesso:**
 - **Descrição:** As avaliações do COG têm de ser submetidas com sucesso para estar totalmente funcional
 - **Importância:** Este requisito é de importância “Should Have” e exige o trabalho de 2 pessoas e a duração prevista de 10 horas.
 - **Estado:** Pendente

- **E1FCOGU8 – No Protocolo MentHA COG, corrigir o enquadramento de imagens e botões nos exercícios das Sessões:**
 - **Descrição:** No Protocolo MentHA COG, deve ser corrigido o enquadramento das imagens e botões no quadro de exercícios das Sessões, garantindo que todos os elementos visuais e interativos estejam devidamente alinhados e funcionando corretamente.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoa, com a duração prevista de 10 horas.
 - **Estado:** Pendente
- **E1FCOGU9 – No Protocolo MentHA COG, corrigir erro de código ao atribuir grupo a participantes sem grupo:**
 - **Descrição:** No Protocolo MentHA COG, ao clicar no botão de "Participantes sem grupo" e tentar atribuir um grupo a um participante, o sistema deve funcionar sem gerar erro de código. A funcionalidade deve permitir ao administrador atribuir o grupo corretamente a um participante sem grupo, sem falhas técnicas.
 - **Importância:** Este requisito é de importância "Should Have" e exige o trabalho de 2 pessoas, com a duração prevista de 4 horas.
 - **Estado:** Pendente
- **E1FCOGU10 – Corrigir erro de Pesquisar pessoa pelo nome. Erro na barra de pesquisa:**
 - **Descrição:** No Protocolo MentHA COG, deve ser corrigida a funcionalidade de pesquisa de participantes, de forma que, ao pesquisar pelo nome de uma pessoa, os resultados sejam corretamente exibidos.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.
 - **Estado:** Pendente
- **E1FCOGU11 – No Protocolo MentHA COG, corrigir o quadro de exercícios da Sessão 4:**
 - **Descrição:** No Protocolo MentHA COG, deve ser corrigido o quadro de exercícios da Sessão 4 garantindo que o exercício esteja correto.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 2 horas.
 - **Estado:** Pendente
- **E1FCOGU12 -Reformulação de View de Criação de Grupo:**
 - **Descrição:** Disponibilização de um formulário intuitivo e funcional que permita aos utilizadores com perfil de cuidador criar grupos COG. O formulário deve ser submetido sem erros garantindo que é submetido sem campos obrigatórios em falta e garantindo que todas os dados e estruturas de dados são guardados e criados de maneira correta.

- **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas e a duração prevista de 10 horas.
- **Estado:** Concluído

3.1.1.4 Lista de Requisitos Referentes à Depuração do Repositório

- **E1DR1 - Resolução de Problemas relativos ao requirements.txt, análise completa de versões e ficheiros desnecessários:**
 - **Descrição:** A equipa deve realizar uma análise completa ao ficheiro requirements.txt, com o objetivo de identificar e resolver problemas relacionados com versões incompatíveis, dependências obsoletas e ficheiros desnecessários no projeto, garantindo a estabilidade e portabilidade do ambiente de desenvolvimento.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas e a duração prevista de 3 horas.
 - **Estado:** Concluído
- **E1DR2 - Analise de Metodologia de Documentação de Código:**
 - **Descrição:** A equipa deve analisar e definir uma metodologia eficaz para a documentação do código e dos processos do projeto, com base numa pesquisa sobre abordagens tradicionais e ferramentas modernas, esta metodologia deverá promover clareza, consistência, manutenibilidade e integração fluida com o fluxo de desenvolvimento da equipa.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 5 horas.
 - **Estado:** Concluído
- **E1DR3 - Resolução de problemas de coerência entre a versão publicada no servidor e versão no repositório remoto git:**
 - **Descrição:** A equipa deve identificar e resolver quaisquer problemas de incoerência entre a versão do código publicada no servidor de produção e a versão armazenada no repositório remoto Git, garantindo que ambas as versões estejam alinhadas e atualizadas.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 5 horas.
 - **Estado:** Concluído
- **E1DR4 - Resolução de Problemas relativos ao repositório, mais especificamente problemas com o ambiente virtual e configuração do gitignore:**
 - **Descrição:** A equipa deve identificar e resolver problemas relacionados com o repositório, especificamente em relação à configuração do ambiente virtual e

ao ficheiro `.gitignore`, garantindo que dependências desnecessárias não sejam incluídas no repositório e que o ambiente de desenvolvimento esteja corretamente configurado.

- **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 6 horas.
- **Estado:** Concluído
- **E2DR5 – Atualização do README para Inclusão do Guia Completo de Infraestrutura, Configuração do Ambiente de Desenvolvimento e Deploy em Produção em Nova VM**
 - **Descrição:** A equipa deve atualizar o ficheiro README do projeto para incluir documentação detalhada sobre a infraestrutura geral da aplicação MentHA Digital. Esta atualização deve contemplar instruções claras para configurar o ambiente de desenvolvimento local, bem como um guia passo a passo para realizar o deploy da aplicação numa nova máquina virtual (VM), garantindo que futuros membros da equipa consigam replicar o ambiente com facilidade.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.
 - **Estado:** Concluído

3.1.1.5 Lista de Requisitos Referentes ao Ambiente de Desenvolvimento

- **E2AD1 - Adição de Docker compose ao projeto, com o objetivo de resolver problemas relativos ao ambiente de desenvolvimento:**
 - **Descrição:** A equipa deve integrar o Docker Compose no projeto, com o objetivo de resolver problemas relacionados com a configuração e consistência do ambiente de desenvolvimento, garantindo que o ambiente seja facilmente replicável e isolado, independentemente do sistema operativo utilizado pelos membros da equipa.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 3 horas.
 - **Estado:** Concluído
- **E2AD2 - Resolução de Problemas relativos á configuração do Docker Compose:**
 - **Descrição:** A equipa deve identificar e resolver problemas relacionados com a configuração do Docker Compose, garantindo que todos os serviços definidos sejam corretamente configurados, funcionando de forma integrada e sem conflitos no ambiente de desenvolvimento.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 4 horas.
 - **Estado:** Concluído

- **E2AD3 - Extração de base de dados em produção, para ser utilizada localmente para testes:**
 - **Descrição:** A equipa deve desenvolver um processo seguro e eficiente para extrair a base de dados em produção e permitir que seja utilizada localmente, garantindo que a cópia da base de dados seja atualizada regularmente para fins de teste e desenvolvimento, sem comprometer a segurança ou a integridade dos dados.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 2 horas.
 - **Estado:** Concluído

3.1.1.6 Lista de Requisitos Referentes ao Ambiente de Produção

- **E2AP1 – Criação de Ambiente de Produção com Docker Compose**
 - **Descrição:** A equipa deve configurar um ambiente de produção para o projeto MentHA Digital utilizando Docker Compose, garantindo isolamento, portabilidade e consistência entre ambientes. Esta configuração deve incluir todos os serviços essenciais (base de dados, aplicação Django, volumes persistentes, etc.), preparados para execução em contexto real.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 15 horas.
 - **Estado:** Concluído
- **E2AP2 – Adaptação do Ambiente de Produção com NGINX conteinerizado como Reverse Proxy**
 - **Descrição:** A equipa deve adaptar o ambiente de produção para utilizar o NGINX em container como reverse proxy. Esta configuração permite gerir o tráfego HTTP, servir ficheiros estáticos e aplicar cabeçalhos de segurança, melhorando a performance e a robustez da infraestrutura.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 4 horas.
 - **Estado:** Concluído
- **E2AP3 – Implementação de CI/CD Automatizado com GitHub Actions**
 - **Descrição:** A equipa deve desenvolver um pipeline automatizado de integração e deployment contínuo (CI/CD) com GitHub Actions, garantindo que qualquer alteração na branch principal (`main`) desencadeie automaticamente testes e deploy para o ambiente de produção.
 - **Importância:** Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 15 horas.
 - **Estado:** Concluído

- **E2AP4 – Criação de Ambiente de Staging na DigitalOcean com Domínio Dedicado**
 - **Descrição:** A equipa deve configurar uma instância de staging na plataforma DigitalOcean, simulando o ambiente de produção. Este ambiente deve utilizar o mesmo domínio e configuração do futuro servidor oficial, permitindo testes e validações antes do deploy definitivo.
 - **Importância:** Este requisito é de importância “Should Have” e exige o trabalho de 2 pessoas, com a duração prevista de 5 horas.
 - **Estado:** Concluído
- **E2AP5 – Deploy Automatizado na Versão de Staging (DigitalOcean)**
 - **Descrição:** A equipa deve configurar o processo de deploy automatizado para a instância de staging, garantindo que o código seja copiado, reconstruído e iniciado de forma automática via GitHub Actions, simulando o funcionamento da produção real.
 - **Importância:** Este requisito é de importância “Should Have” e exige o trabalho de 2 pessoas, com a duração prevista de 10 horas.
 - **Estado:** Concluído
- **E2AP6 – Testes de Validação da Configuração no Ambiente de Staging**
 - **Descrição:** A equipa deve realizar testes funcionais e técnicos no ambiente de staging para validar a infraestrutura, comportamento da aplicação, configuração de rede e disponibilidade do domínio, assegurando que o ambiente está pronto para deploy final.
 - **Importância:** Este requisito é de importância “Must Have” e exige o trabalho de 2 pessoas, com a duração prevista de 3 horas.
 - **Estado:** Concluído
- **E2AP7 – Separação de Base de Dados de Produção e Base de Dados de Teste**
 - **Descrição:** A equipa deve garantir a separação clara entre a base de dados de produção e a de testes, assegurando que os dados reais nunca sejam utilizados ou modificados em ambientes de desenvolvimento ou staging. Esta separação deve ser garantida através de variáveis de ambiente e configuração de volumes distintos.
 - **Importância:** Este requisito é de importância “Must Have” e exige o trabalho de 1 pessoa, com a duração prevista de 2 horas.
 - **Estado:** Concluído

3.1.2 Descrição detalhada dos requisitos principais

O nosso objetivo tem sido levantar requisitos maioritariamente centrados na identificação e correção de erros nas aplicações existentes, e no repositório.

Estes requisitos não necessitam de uma estrutura muito detalhada, logo consideramos que a maneira como eles estão apresentados e descritos na **Secção 3.1.1** é detalhada o suficiente.

Nesta Secção já apresentamos por requisito:

- Descrição detalhada do requisito
- Importância
- Estimativa de Esforço
- Estado Atual

3.1.3 Casos de Uso/User Stories

O desenvolvimento do projeto MentHA segue uma abordagem **Agile**, focada na adaptação contínua e na entrega incremental de valor. Utilizamos o Trello como ferramenta para organizar o **Product Backlog**, onde as tarefas estão categorizadas em **épicos**, **features** e **user stories**. Essa abordagem permite gerenciar tanto requisitos de correção de erros quanto a implementação de novas funcionalidades.

A maior parte das **user stories** até o momento corresponde a pequenas melhorias e correções de componentes, sendo rápidas de implementar. No entanto, à medida que o projeto avança, prevemos a criação de componentes maiores e mais complexos. Nesses casos, os requisitos serão subdivididos em **sub-requisitos** e, consequentemente, em **sub-user stories**, garantindo maior granularidade no planejamento e execução.

Na **Figura 2** está a apresentada uma visão geral do nosso Trello:



Figura 2 - Visão Geral Trello Épicos , Features e User Stories

3.1.3.1 Épicos

Os épicos a seguir apresentam as diferentes áreas de desenvolvimento do projeto MentHA Digital, divididos conforme as funcionalidades e objetivos específicos a serem alcançados. Cada épico aborda uma parte essencial do projeto, permitindo a organização e o direcionamento adequado das tarefas. Os épicos são organizados nas seguintes áreas:

- **E1 – Aplicações MentHA Digital:**
Este épico foca no aperfeiçoamento e expansão das funcionalidades das três principais aplicações da plataforma MentHA: MentHA COG, Protocolo MentHA (MentHA EVAL) e MentHA CARE.
- **E2 – Infraestrutura e Repositório:**
Este épico visa melhorar a organização interna do projeto, com organização do repositório, à estrutura do projeto e à resolução de problemas técnicos relacionados com a manutenção e depuração do ambiente de desenvolvimento, problemas estruturais do repositório git. A intenção é garantir a estabilidade e a escalabilidade da plataforma, criando um ambiente de desenvolvimento sólido e sustentável.
- **E3 – Tarefas Exploratório-Preparatórias:**
Este épico abrange tarefas preparatórias, como o levantamento de requisitos e a familiarização com a plataforma. Essas atividades foram essenciais para garantir que as necessidades dos usuários fossem compreendidas de maneira eficaz e para promover a integração da equipe nas funcionalidades existentes, antes de iniciar a implementação de novos componentes.

3.1.3.2 Features

As features do projeto estão organizadas em três categorias principais, de acordo com sua natureza e finalidade. A seguir estão as features associadas a cada épico:

Features Relacionadas com Aplicações da Plataforma MentHA (Épico 1)

Estas features correspondem às funcionalidades específicas das aplicações da plataforma MentHA:

- **MentHA COG:** Programa de estimulação cognitiva para adultos mais velhos com sinais de demência leve.
- **MentHA EVAL (Protocolo MentHA):** Protocolo de avaliação neuropsicológica para adultos com déficit cognitivo e condições de saúde mental.
- **MentHA CARE:** Programa psicoeducativo destinado a apoiar cuidadores de adultos com demência.
- **MentHA Digital (Website):** Interface web centralizada para acesso às soluções MentHA.

Feature Relacionada com Infraestrutura e Ambiente de Desenvolvimento (Épico 2)

Esta feature está relacionada à organização e manutenção do repositório de código e do ambiente de desenvolvimento:

- **Depuração do Re却itório (E1DR):** Organização e depuração do repositório de código, para facilitar o desenvolvimento futuro e melhorar a estrutura do projeto.

- **Ambiente de Desenvolvimento (E2AD):** Integração do Docker Compose para uniformizar e isolar o ambiente de desenvolvimento entre os membros da equipa, resolução de problemas técnicos com os containers e processos, e extração segura da base de dados de produção para testes locais.
- **Ambiente de Produção e Staging (E2AP):** Criação de um ambiente de produção completo utilizando Docker Compose, configuração de NGINX conteinerizado como reverse proxy, implementação de pipelines CI/CD com GitHub Actions e configuração de um ambiente de staging na DigitalOcean com domínio próprio para testes prévios ao deploy definitivo.

Estes problemas relacionados ao repositório e as soluções que criamos para os mesmos, apesar de estarem enumerados em user stories mais à frente na **Secção 3.1.3.3**, estão explicados em mais detalhe mais à frente na **Secção 4.6**

Features Preparatórias e Exploratórias (Épico 3)

Estas features não resultam diretamente na implementação de funcionalidades, mas foram essenciais para o levantamento de requisitos e ambientação à plataforma:

- **Levantamento de Requisitos (E1FLV):** Levantamento de necessidades e análise dos requisitos das aplicações MentHA.
- **Familiarização com a Plataforma (E1FP):** Compreensão detalhada das funcionalidades já desenvolvidas e o propósito de cada aplicação na plataforma.
- **Relatório (E1FR):** Elaboração do relatório final do projeto, documentando o progresso, as funcionalidades desenvolvidas e as melhorias realizadas.

As features são identificadas e codificadas desta maneira: (**E1FCOG** – Épico 1, Feature COG)

Na **Tabela 2**, estão representados as features tendo para cada uma delas: codificação/título e uma descrição sobre a feature.

Tabela 2 - Tabela de features

ID da Feature	Descrição
E1FCOG – MentHA COG	O objetivo é continuar o desenvolvimento do MentHA COG, um programa de estimulação cognitiva com 14 sessões para adultos mais velhos com sinais de demência leve. Este programa busca fortalecer a autonomia, melhorar a autoestima e fomentar a interação social dos participantes, contribuindo para sua conexão com o meio
E1FPT – Protocolo MentHA (MentHA EVAL)	O objetivo é continuar o desenvolvimento do MentHA Eval, um protocolo de avaliação neuropsicológica para adultos com défice cognitivo e condições de saúde mental. Esta funcionalidade visa permitir uma análise precisa e abrangente das funções cognitivas, emocionais e comportamentais dos participantes

Tabela 2 - Tabela de features

E1FCAR – MentHA CARE	O objetivo é continuar o desenvolvimento do MentHA CARE, um programa psicoeducativo destinado a apoiar cuidadores de adultos com demência. Composto por 13 sessões, o programa busca reduzir a sobrecarga emocional dos cuidadores, auxiliando-os na gestão do estresse e promovendo boas práticas de cuidado
E1FMD – MentHA Digital (Web Site)	O objetivo é continuar o desenvolvimento e aprimoramento do MentHA Digital, com foco na melhoria da acessibilidade, responsividade e navegação intuitiva. O objetivo é tornar o website mais atraente e funcional para os utilizadores, garantindo que a experiência de navegação seja agradável e otimizada.
E1DR – Depuração do Repositório	O objetivo é utilizar métodos de documentação e depuração de código de maneira a deixar o código mais organizado e documentado, para facilitar o desenvolvimento futuro
E2AD – Criação de ambiente de desenvolvimento	Criação de ambiente de desenvolvimento local conteinerizado com Docker Compose, incluindo base de dados PostgreSQL, volumes persistentes, dados anónimos de testes e suporte a hot reload para desenvolvimento ágil e seguro
E2AP – Criação de ambiente de produção	Criação de ambiente de produção conteinerizado com Docker Compose, utilizando Unicorn, NGINX e PostgreSQL, integrado com pipeline CI/CD automático via GitHub Actions para deploy contínuo em servidor remoto com variáveis de ambiente seguras
E1FLV – Levantamento de Requisitos	O objetivo é continuar o levantamento de requisitos e a deteção de erros nas aplicações MentHA COG, MentHA Eval e MentHA CARE, garantindo que as funcionalidades estejam alinhadas com as necessidades dos utilizadores e que a plataforma opere de forma confiável e eficiente
E1FP - Familiarização da Plataforma	Familiarização da plataforma MentHA Digital, analisando e compreendendo em detalhe as funcionalidades já desenvolvidas e o propósito de cada aplicação.
E1FR - Relatório	O objetivo é elaborar o relatório final do projeto, documentando de forma detalhada o progresso, as funcionalidades desenvolvidas, as melhorias realizadas e os resultados obtidos.

3.1.3.3 User Stories

A seguir, apresentam-se as tabelas que organizam as **User Stories** identificadas para cada uma das **features** do projeto. Cada User Story está codificada com a estrutura: **(E#FXXXU#)**, onde:

- **E#** representa o épico correspondente,

- **FXXX** representa a sigla da feature (por exemplo, COG para MentHA COG),
- **U#** indica o número sequencial da User Story dentro dessa feature.

Exemplo: (E1FCOGU1 – Épico 1, Feature COG, User Story 1)

Abaixo encontra-se a lista de referências às tabelas que contêm as User Stories por feature:

- **Tabela 3 - User Stories referentes á feature E1FMD (MentHA Digital Web Site)**
- **Tabela 4 - User Stories referentes á feature E1FPT (Protocolo MentHA)**
- **Tabela 5 - User Stories referentes á feature E1FCOG (MentHA COG)**
- **Tabela 6 - User Stories referentes á feature E2DR (Depuração do Repositório)**
- **Tabela 7 - User Stories referentes á feature E2AD (Ambiente de Desenvolvimento)**
- **Tabela 8 - User Stories referentes á feature E2AP (Ambiente de Produção)**

As features **E1FLV – Levantamento de Requisitos**, **E1FP – Familiarização com a Plataforma** e **E1FF – Familiarização com Funcionalidades** não apresentam User Stories associadas. Estas foram consideradas tarefas preparatórias e de exploração, com foco em levantamento de necessidades e ambientação à plataforma, não resultando diretamente na implementação de componentes funcionais do software. Por esta razão não apresentamos estas tarefas em forma de user stories aqui no relatório, mas estão contidas no Trello.

Tabela 3 - User Stories referentes á feature E1FMD (MentHA Digital Web Site)

ID da User Story	Descrição
E1FMDU1	Como cuidador, quero conseguir fazer log-out da minha conta, para que possa encerrar a sessão de forma segura e proteger os meus dados pessoais.
E1FMDU2	Como utilizador do website MentHA, Quero que não existam quaisquer erros ortográficos no conteúdo do site, Para que a informação seja clara, profissional e de fácil compreensão, melhorando a credibilidade e a experiência do utilizador.
E1FMDU3	Como utilizador do website MentHA, quero que seja criado um protótipo de interface no Figma, para sugerir melhorias visuais e de usabilidade para o website MentHA Digital, mantendo a identidade visual do projeto.
E1FMDU4	Como utilizador do website MentHA, quero que sejam aplicadas no website MentHA Digital as melhorias visuais definidas no protótipo Figma, para garantir uma interface mais atrativa, funcional e alinhada com as necessidades.

Tabela 4 - User Stories referentes á feature E1FPT (Protocolo MentHA)

ID da User Story	Descrição
E1FPTU1	Como cuidador, quero receber uma mensagem de erro clara e informativa ao tentar registar um participante com um username já existente, para que seja possível identificar o problema e corrigir rapidamente o erro durante o processo de registo.
E1FPTU2	Como cuidador, quero gerar relatórios corretamente com todos os dados apresentados de forma precisa, para que possa analisar e partilhar as informações completas e corretas sem erros.
E1FPTU3	Como cuidador, quero adicionar uma secção relacionada com consentimento autorizado, para que possa garantir que os pacientes fornecem o seu consentimento de forma clara e em conformidade com os requisitos legais e éticos.
E1FPTU4	Como um administrador, quero que os campos opcionais no formulário de registo de participantes estejam vazios por padrão e utilizem labels claras, para garantir a precisão dos dados inseridos e melhorar a experiência do utilizador.
E1FPTU5	Como um administrador, quero que as avaliações MentHA Risk submetidas com campos por preencher exibam um aviso visual claro na lista de avaliações, para que eu possa identificar facilmente quais testes estão incompletos e voltar a editá-los quando necessário.
E1FPTU6	Como administrador do Protocolo MentHA, quero gerar relatórios de avaliações sem encontrar erros de página indisponível, para que eu possa aceder e rever os relatórios de avaliação de forma eficiente e sem interrupções.
E1FPTU7	Como administrador do Protocolo MentHA, quero ver a barra de progresso funcionar corretamente no questionário sociodemográfico no MentHA Risk, para que eu possa acompanhar o meu progresso enquanto preencho a avaliação de forma eficiente e sem confusão.

Tabela 4 - User Stories referentes á feature E1FPT (Protocolo MentHA)

E1FPTU8	Como cuidador, quero puder ter acesso a dados estatísticos em formato csv sobre todas as avaliações feitas pelos pacientes, para dessa forma conseguir ter acesso ao estado geral do protocolo
E1FPTU9	Como cuidador, quero ter acesso às instituições dos pacientes no csv de analise de dados para dessa maneira conseguir fazer uma análise mais detalhada filtrando por instituições, podendo assim então reportar às mesmas

Tabela 5 - User Stories referentes á feature E1FCOG (MentHA COG)

ID da User Story	Descrição
E1FCOGU1	Como cuidador, quero visualizar os constituintes de cada sessão ao criar um grupo, para que possa gerir e acompanhar facilmente os participantes associados a cada sessão.
E1FCOGU2	Como cuidador, quero aceder e visualizar corretamente os materiais das sessões diretamente na página da sessão, para que possa utilizá-los de forma eficiente e sem necessidade de navegação adicional.
E1FCOGU3	Como cuidador, quero que a barra de progresso no sociodemográfico funcione corretamente, para que possa acompanhar de forma clara e precisa o meu progresso durante o preenchimento.
E1FCOGU4	Como cuidador, quero que o caderno de estímulos seja implementado na aplicação, para que possa utilizá-lo como parte das sessões e garantir que todas as funcionalidades necessárias estão disponíveis.
E1FCOGU5	Como cuidador, quero que o caderno de estímulos seja implementado na aplicação, para que possa utilizá-lo como parte das sessões e garantir que todas as funcionalidades necessárias estão disponíveis.

Tabela 5 - User Stories referentes á feature E1FCOG (MentHA COG)

E1FCOGU6	Como cuidador, quero ter uma opção para criar e submeter avaliações, para que possa personalizar e registar avaliações adaptadas às necessidades dos participantes.
E1FCOGU7	Como cuidador, quero submeter avaliações com sucesso, para que as avaliações sejam processadas corretamente e os resultados sejam guardados adequadamente.
E1FCOGU8	Como cuidador, quero que as imagens e os botões no quadro de exercícios das Sessões sejam corrigidos e corretamente enquadrados, para que eu tenha uma experiência de uso mais fluída e visualmente agradável ao interagir com as atividades.
E1FCOGU9	Como administrador, quero poder clicar no botão de "Participantes sem grupo" e atribuir um grupo a um participante sem que ocorra erro de código, para que eu possa gerenciar e organizar os participantes no sistema sem problemas técnicos.
E1FCOGU10	Como administrador do MentHA COG, quero poder pesquisar participantes pelo nome corretamente, para que eu possa encontrar facilmente as informações dos participantes e gerenciar o sistema de forma eficiente.
E1FCOGU11	Como Cuidador do Protocolo MentHA COG, quero que o quadro de exercícios da Sessão 4 seja corrigido, para que eu possa realizar os exercícios corretamente, sem erros de exibição ou funcionalidades, garantindo uma experiência de uso eficiente e sem problemas.
E1FCOGU12	Como cuidador, quero poder preencher um formulário que me permita criar grupos COG sem erros e onde todos os dados sejam guardados corretamente, para que possa começar sessões sem risco de erros

Tabela 6 - User Stories referentes á feature E2DR (Depuração do Repositório)

ID da User Story	Descrição
E2DR1	Como membro da equipa de desenvolvimento, quero analisar e corrigir o ficheiro requirements.txt, para garantir que todas as dependências são compatíveis, atualizadas e essenciais para o projeto.
E2DR2	Como equipa de desenvolvimento, queremos definir uma metodologia clara e eficaz para a documentação do código, para garantir consistência, clareza e facilitar a manutenção futura do projeto.
E2DR3	Como membro da equipa de desenvolvimento, quero garantir que a versão do código no servidor de produção está alinhada com o repositório Git, para evitar conflitos, bugs ou divergências nas funcionalidades implementadas.
E2DR4	Como membro da equipa de desenvolvimento, quero corrigir a configuração do ambiente virtual e o ficheiro .gitignore, para garantir que o repositório esteja limpo e funcional para todos os membros da equipa.

Tabela 7 - User Stories referentes á feature E2AD (Ambiente de Desenvolvimento)

ID da User Story	Descrição
E2DR1	Como equipa de desenvolvimento, queremos integrar Docker Compose no projeto, para garantir a consistência e replicabilidade do ambiente de desenvolvimento em qualquer máquina.
E2DR2	Como membro da equipa de desenvolvimento, quero corrigir problemas de configuração no Docker Compose, para assegurar que os serviços funcionam de forma integrada e sem erros.
E2DR3	Como membro da equipa de desenvolvimento, quero extrair uma cópia da base de dados de produção para o ambiente local, para testar funcionalidades com dados reais sem comprometer a segurança.

Tabela 8 - User Stories referentes á feature E2AP (Ambiente de Produção)

ID da User Story	Descrição
E2AP1	Como membro da equipa de desenvolvimento, quero configurar o ambiente de produção utilizando Docker Compose, para garantir consistência, portabilidade e facilidade de gestão dos serviços essenciais da aplicação.
E2AP2	Como membro da equipa de desenvolvimento, quero corrigir problemas de configuração no Docker Compose, para assegurar que os serviços funcionam de forma integrada e sem erros.
E2AP3	Como membro da equipa de DevOps, quero configurar um pipeline de CI/CD com GitHub Actions, para automatizar o processo de testes e deploy sempre que houver alterações na branch principal.
E2AP4	Como equipa de desenvolvimento, queremos criar um ambiente de staging na DigitalOcean com domínio próprio, para realizar testes realistas antes do deploy definitivo em produção.
E2AP5	Como membro da equipa de DevOps, quero configurar deploy automatizado para o ambiente de staging via GitHub Actions, para garantir rapidez, consistência e validação contínua das versões mais recentes.
E2AP6	Como equipa de desenvolvimento, queremos realizar testes técnicos e funcionais no ambiente de staging, para garantir que a infraestrutura está correta e a aplicação pronta para produção.
E2AP7	Como membro da equipa de desenvolvimento, quero separar claramente a base de dados de produção da base de dados de teste, para evitar riscos de perda de dados reais durante o desenvolvimento ou testes.

3.1.3.4 Casos de Uso

Como referido anteriormente, o nosso papel neste projeto é focado na continuidade e consolidação das aplicações. Assim, os **Casos de Uso** foram previamente elaborados pelos colegas Miguel Cardoso e António Ventura, no âmbito do TFC desenvolvido no ano passado, conforme ilustrado no **Anexo B – Casos de Uso**

3.2 Modelação

Este projeto é composto por diversas entidades que organizam e estruturam a informação de maneira eficiente. Essas entidades representam objetos essenciais para o armazenamento e gestão de dados, sendo fundamentais para o bom funcionamento da aplicação e para assegurar uma comunicação de dados abrangente. A base de dados, criada para sustentar a aplicação, já estava estabelecida devido ao fato de o projeto MentHA ser uma continuidade de um TFC iniciado em anos letivos anteriores.

Com a continuidade e consolidação das aplicações MentHA COG, MentHA CARE e Protocolo MentHA, a estrutura já foi cuidadosamente concebida e desenvolvida em anos anteriores. Portanto,

O nosso papel no projeto até o momento tem se concentrado principalmente na resolução de problemas em componentes existentes, o que resultou em poucas alterações aos modelos já estabelecidos e nenhuma criação de novas classes. O foco do nosso trabalho tem sido garantir que os dados sejam carregados corretamente nos modelos existentes, sem erros, assegurando a integridade e o bom funcionamento do sistema, logo o modelo Entidade-Relação atual continua bastante semelhante ao do ano passado **[TFC24a]**,

Dado que o MentHA CARE e o MentHA COG compartilham uma modelagem idêntica, e considerando que essa estrutura é extensa e complexa, foi previamente elaborado um modelo ERD, cuja descrição detalhada se encontra **no Anexo C – Modelo ERD MentHA COG / MentHA CARE**.

A **Figura 3** apresenta o Modelo Entidade-Relação referente ao MentHA CARE e ao MentHA COG, enquanto a **Figura 4** ilustra o Modelo Entidade-Relação do Protocolo MentHA.

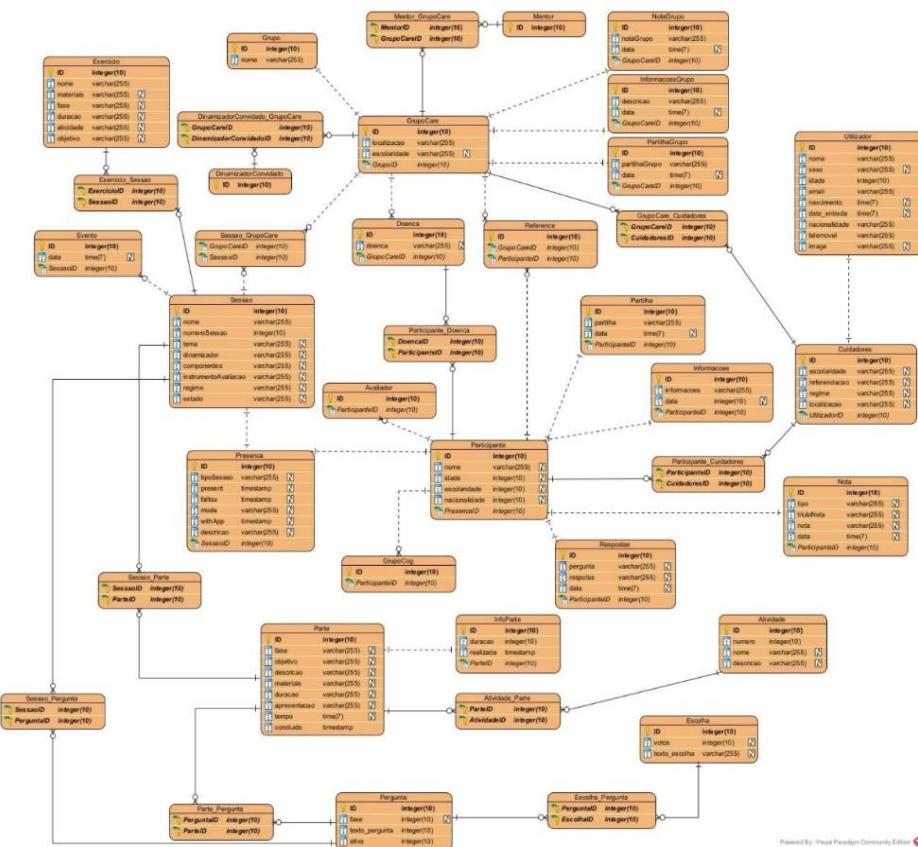


Figura 3 - Modelo E/R Da Aplicação MENTHA CARE/COG

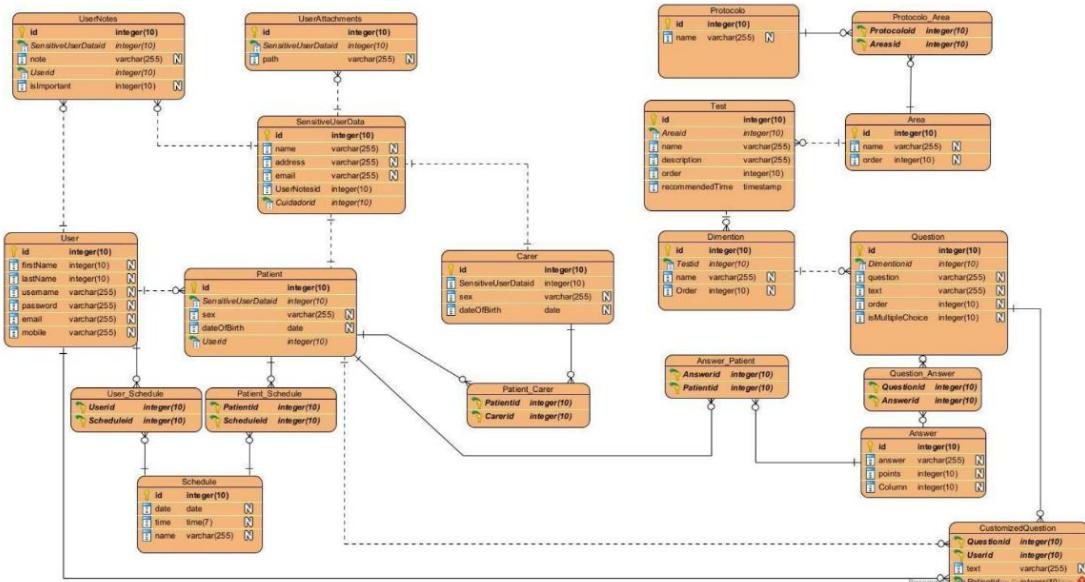


Figura 4 - Modelo E/R Da Aplicação Protocolo MENTHA

3.3 Protótipos de Interface

O MentHA Digital é constituído por três aplicações, MentHA CARE, MentHA COG e o Protocolo MentHA. Segue-se um esquema de cada uma das aplicações que estão representados na **Figura 5**, **Figura 6** e na **Figura 7**.

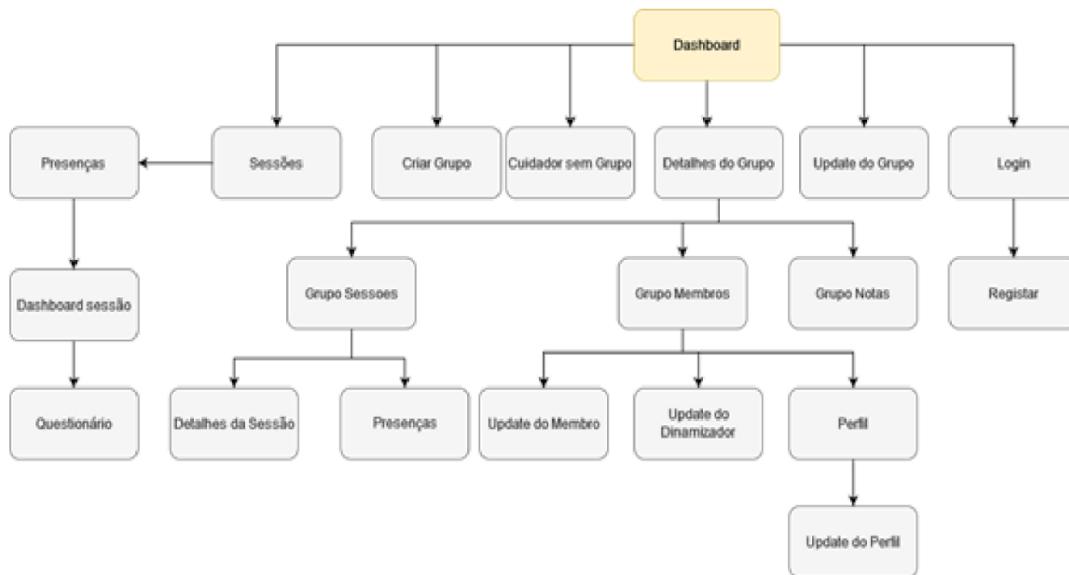


Figura 5 - Mapa Aplicacional do MentHA CARE

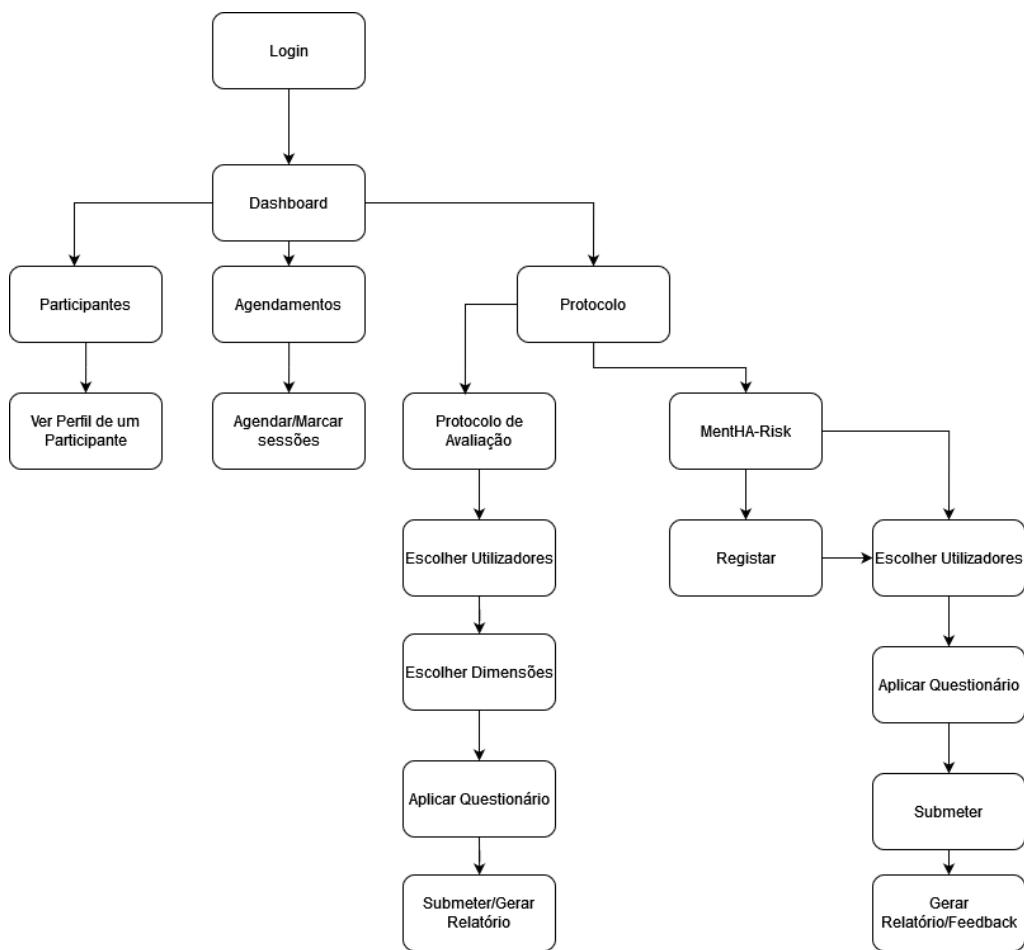


Figura 6 - Mapa aplicacional do Protocolo MentHA

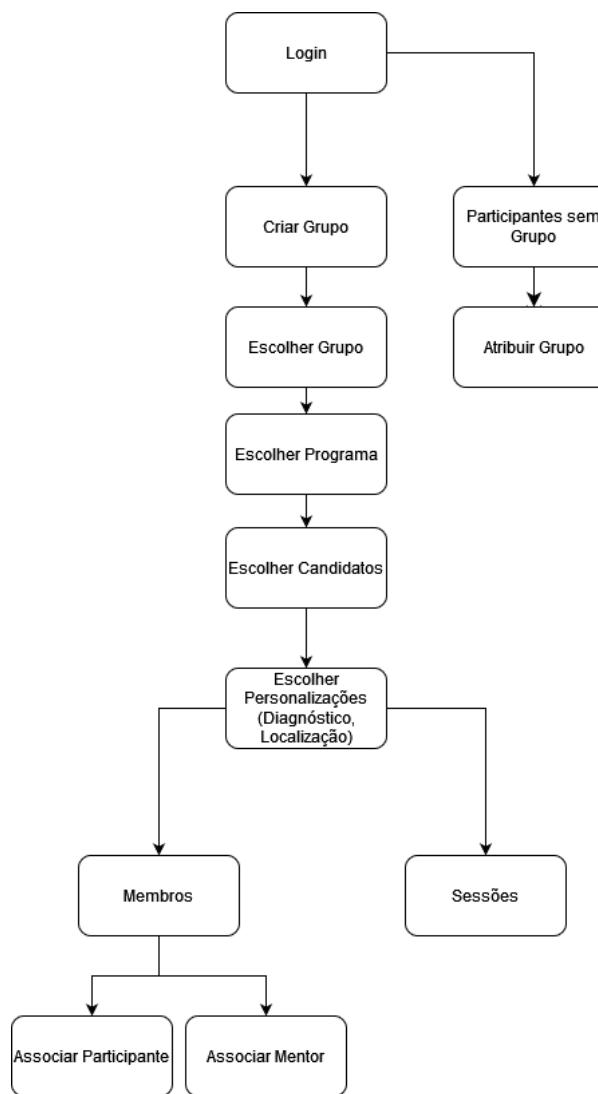


Figura 7 - Mapa aplicacional do MentHA COG

4 Solução Desenvolvida

Com base na análise dos problemas identificados e nos requisitos levantados ao longo do projeto, esta secção tem como objetivo apresentar a solução proposta para o MentHA Digital. Iremos descrever de forma detalhada a abordagem adotada para responder às necessidades dos utilizadores e resolver as falhas encontradas em versões anteriores da aplicação. A solução apresentada visa assegurar a melhoria contínua das funcionalidades já existentes e garantir a estabilidade, usabilidade e eficácia da plataforma.

Ao longo desta secção, abordaremos a estrutura geral da solução, as tecnologias utilizadas, os ambientes de desenvolvimento e produção, bem como a abrangência e os componentes fundamentais do sistema. Adicionalmente, serão apresentadas as interfaces desenvolvidas e recursos complementares que suportam a compreensão da aplicação.

A proposta aqui delineada reflete uma continuidade do trabalho iniciado em projetos anteriores, integrando melhorias essenciais que promovem um suporte mais eficaz para cuidadores e pessoas com défices cognitivos.

4.1 Apresentação

Com base na análise dos problemas identificados e nos requisitos levantados ao longo do projeto, esta secção tem como objetivo apresentar a solução proposta para o MentHA Digital. Iremos descrever de forma detalhada a abordagem adotada para responder às necessidades dos utilizadores e resolver as falhas encontradas em versões anteriores da aplicação. A solução apresentada visa assegurar a melhoria contínua das funcionalidades já existentes e garantir a estabilidade, usabilidade e eficácia da plataforma.

Abaixo, apresentamos uma lista de links úteis para explorar melhor a nossa solução.

MentHA Digital Vídeo: https://youtu.be/MaVV_dgqb_8

Aqui está disponível um vídeo demonstrativo que reúne todas as alterações que implementámos no projeto. Este vídeo apresenta de forma clara e prática a nossa contribuição para o MentHA Digital e evidencia o estado final da plataforma após a nossa intervenção

MentHA COG Vídeo: <https://youtu.be/Yp2jd09S4Rc>

MentHA EVAL Vídeo: <https://youtu.be/QzncV6Ulshw>

MentHA CARE Vídeo: <https://youtu.be/cRAcrlrgil>

Aqui estão apresentados 3 vídeos demonstrativos de cada uma das aplicações. Aqui consegue se ter uma visão geral de como o MentHA Digital se encontra no preciso momento.

Repositório Git: https://github.com/DEISI-ULHT-TFC-2024-25/TFC-DEISI2106-MentHA_Digital

Este repositório Git contém todos os detalhes técnicos da solução desenvolvida, incluindo documentação, código-fonte e outros recursos relevantes. Este repositório contém também um readme com uma explicação geral do sistema e um guia completo de como configurar o sistema localmente.

Atualmente, a versão mais recente e funcional do MentHA Digital encontra-se em produção no domínio:

Solução Funcional em Produção: <http://menthadigital.pt/> (VM provisionada na DigitalOcean)

Esta instância representa o **MVP mais atualizado** da aplicação, com a nova infraestrutura desenhada durante este TFC, baseada em Docker e preparada para escalabilidade, segurança e facilidade de manutenção.

Por outro lado, a versão anterior da aplicação continua disponível no servidor institucional da Universidade Lusófona, no domínio:

Solução Funcional Antiga em Produção: <https://menthadigital.com/>

No entanto, esta versão encontra-se **desatualizada** e não reflete as melhorias estruturais e funcionais implementadas ao longo deste projeto.

Esta secção encontra-se organizada da seguinte forma:

- **4.2 Arquitetura** – Nesta secção é apresentada a arquitetura geral do MentHA Digital, com base numa abordagem já validada por trabalhos anteriores. Aqui destaca-se o nosso papel na inclusão do Docker Compose de forma a criar solução robusta, estável e fácil de manter.
- **4.3 Tecnologias e Ferramentas Utilizadas** – Esta secção apresenta as tecnologias aplicadas no desenvolvimento do MentHA Digital, tanto no Front-End como no Back-End.
- **4.4 Ambientes de Teste e de Produção** – Esta secção descreve os ambientes utilizados para desenvolvimento, testes e produção. São abordadas as dificuldades enfrentadas inicialmente, a migração do SQLite para PostgreSQL, e a introdução do Docker Compose para padronizar o ambiente. Também é detalhada a infraestrutura do servidor de produção da Universidade Lusófona e as melhorias previstas no processo de deploy.
- **4.5 Abrangência** – Esta secção evidencia como o desenvolvimento do projeto incorporou conhecimentos de várias unidades curriculares, como Programação Web, Engenharia de Software, Bases de Dados e Interação Humano-Computador. Destaca-se a aplicação do modelo MVT, a metodologia ágil e a importância de um design centrado no utilizador.
- **4.6 Desafios Enfrentados e Como os Ultrapassamos** – Esta secção descreve as principais dificuldades técnicas e organizacionais enfrentadas ao longo do desenvolvimento do projeto, relacionados com o épico **E2 – Infraestrutura e Repositório**, com especial ênfase nos problemas relacionados ao repositório herdado. O capítulo apresenta as estratégias adotadas para reorganizar o repositório, padronizar os ambientes e garantir maior estabilidade no desenvolvimento.
- **4.7 Componentes** – Nesta secção são descritas as principais componentes do sistema MentHA Digital, com foco nas melhorias realizadas e nas novas funcionalidades desenvolvidas. A intervenção foi dividida entre a correção de erros existentes e o desenvolvimento de soluções para requisitos emergentes, com base na análise do estado atual do sistema. São apresentadas também as modificações implementadas nas aplicações principais e o impacto dessas alterações na usabilidade e estabilidade do sistema.

- **4.8 Interfaces** – Nesta secção são apresentadas as interfaces das aplicações do MentHA Digital, com detalhes organizados e imagens dos ecrãs de cada aplicação.

4.2 Arquitetura

Este TFC segue uma abordagem semelhante a trabalhos realizados em anos anteriores, com soluções que são, na sua maioria, idênticas ou muito próximas das apresentadas em projetos passados, já avaliados e validados.

Como referido anteriormente, este trabalho tem como objetivo dar continuidade ao que já foi desenvolvido, focando-se na melhoria de funcionalidades existentes e na correção de bugs. O foco principal é a melhoria contínua das três aplicações que compõem o MentHA Digital. Na **Figura 8**, é possível observar a arquitetura proposta.

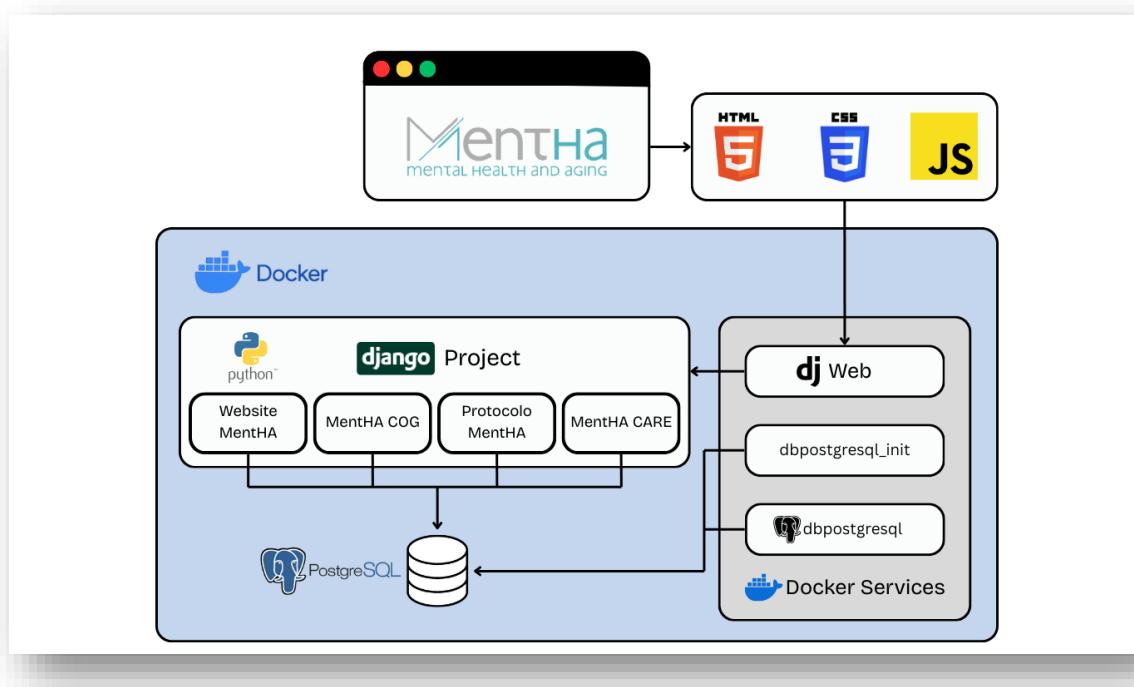


Figura 8 - Arquitetura Proposta para o MentHA Digital

4.3 Tecnologias e Ferramentas Utilizadas

Este trabalho concentra-se na melhoria contínua das três aplicações que compõem o MentHA: Protocolo MentHA, MentHA CARE e MentHA COG. O MentHA consiste numa aplicação web desenvolvida para oferecer suporte aos cuidadores de pacientes que enfrentam patologias ou doenças mentais crónicas.

Tecnologias de *Front-End*:

- **HTML [HTML24]**: Linguagem de marcação utilizada para construir páginas web. É o bloco de construção essencial que define a estrutura e o significado do conteúdo apresentado.

- **CSS [CSS24]:** Cascading Style Sheets (CSS) é a linguagem de estilização que descreve como os conteúdos em HTML devem ser exibidos, permitindo personalizar a aparência das páginas web.
- **JavaScript [JS24]:** Uma linguagem de programação dinâmica, leve e baseada em objetos, conhecida por ser amplamente utilizada para adicionar interatividade a páginas web. Suporta vários estilos, como orientação a objetos, imperativos e declarativos.
- **Bootstrap [BOAP24]:** Framework *open-source* utilizada no desenvolvimento de *Front-End*. Permite criar componentes de interface para sites e aplicações web, proporcionando uma experiência mais confiável e responsiva.

Tecnologias de *Back-End*:

- **Python [PY24]:** Uma linguagem de programação interpretada, de alto nível e orientada a objetos. É conhecida pela sua simplicidade e flexibilidade, sendo ideal tanto para desenvolvimento rápido de aplicações quanto para integração de componentes.
- **Django [DGO24]:** Framework web de alto nível baseada em Python, que facilita o desenvolvimento rápido e promove uma conceção limpa e bem estruturada das aplicações.
- **PostgreSQL [POES24]:** Sistema de gestão de bases de dados relacional *open-source*, com mais de 20 anos de desenvolvimento comunitário. É altamente confiável, escalável e utilizado para armazenar e gerir dados de forma eficiente.
- **Docker [DOCK25]:** Plataforma que permite empacotar aplicações e respetivas dependências em contentores isolados. No projeto MentHA, o Docker é utilizado para garantir que os diferentes componentes da aplicação (como o Django e a base de dados) funcionem de forma consistente em qualquer ambiente, facilitando o desenvolvimento, testes e implantação.
- **Docker Compose [DOCC25]:** Ferramenta que permite definir e gerir múltiplos containers Docker através de um único ficheiro de configuração (`docker-compose.yml`). No contexto do projeto MentHA, o Docker Compose é utilizado para garantir um ambiente de desenvolvimento padronizado, isolado e replicável entre diferentes máquinas e sistemas operativos. Automatiza o processo de inicialização dos serviços essenciais, como a base de dados PostgreSQL e a aplicação Django, contribuindo para a estabilidade e a consistência do ambiente de desenvolvimento.

4.4 Ambientes de Teste e de Produção

4.4.1 Ambiente de Testes

Durante as fases iniciais do desenvolvimento, enfrentámos diversas dificuldades ao tentar configurar o ambiente de desenvolvimento local. A ausência de documentação clara e atualizada, combinada com conflitos de dependências e versões desatualizadas no ficheiro `requirements.txt`, tornou o processo bastante instável e demorado.

Além disso, a base de dados inicialmente utilizada no ambiente de testes era o SQLite. Contudo, devido à crescente complexidade da estrutura de dados e ao facto de a base de dados SQLite se

encontrar bastante desatualizada, foi necessário adotar uma solução mais robusta, optando assim pelo PostgreSQL.

Para resolver os problemas enfrentados e garantir um ambiente de desenvolvimento padronizado e replicável entre os membros da equipa, decidimos integrar o **Docker Compose** no projeto. Esta ferramenta permitiu isolar os serviços da aplicação e garantir que todos os programadores pudessem trabalhar com as mesmas condições, independentemente do sistema operativo ou da configuração da máquina local.

A implementação detalhada do Docker Compose, incluindo a estrutura dos serviços e ficheiros envolvidos, pode ser consultada no **Anexo G – Configuração do Ambiente de Desenvolvimento com Docker e Docker Compose**

4.4.2 Ambiente de Produção

O ambiente de produção da aplicação MenthA Digital foi significativamente evoluído durante esta fase do projeto, adotando práticas modernas de **DevOps** e infraestrutura automatizada, com o objetivo de garantir maior estabilidade, segurança e eficiência no processo de deployment.

A nova infraestrutura de produção foi completamente redesenhada com base em **Docker** e **Docker Compose (compose.prod.yaml)**, permitindo a orquestração automatizada de todos os serviços da aplicação, com isolamento completo dos componentes e maior previsibilidade nos processos de atualização.

O ambiente de produção inclui os seguintes serviços principais:

- **dbpostgresql**: Serviço de base de dados **PostgreSQL**, com volume Docker persistente, garantindo a integridade e conservação de dados entre atualizações.
- **web**: Serviço principal de aplicação Django, executado através de um servidor **Gunicorn (WSGI)**, otimizado para ambientes de produção e preparado para lidar com múltiplos pedidos simultâneos.
- **nginx**: Serviço de **reverse proxy containerizado**, responsável por servir conteúdos estáticos e media, aplicar cabeçalhos de segurança HTTP, realizar compressão de tráfego e encaminhar as requisições de forma segura para o serviço web.

Toda a gestão de variáveis sensíveis e credenciais de produção é realizada através de GitHub Secrets, garantindo que nenhuma informação confidencial (como a SECRET_KEY do Django, credenciais da base de dados ou chaves SSH) se encontra exposta no código-fonte ou no repositório.

A infraestrutura encontra-se ainda totalmente integrada com um **pipeline automatizado de CI/CD** utilizando o **GitHub Actions**, permitindo que, sempre que são introduzidas alterações no repositório principal (push para a branch main), seja executado automaticamente:

1. Um ambiente de testes automatizados isolado, com base de dados temporária;
2. A execução de todos os testes automatizados da aplicação Django;
3. Em caso de sucesso, o **deployment automático** da nova versão para o servidor de produção.

Este processo garante **consistência entre os ambientes de desenvolvimento, testes e produção**, elimina etapas manuais suscetíveis a erro e permite uma atualização rápida e segura da aplicação.

A infraestrutura de produção implementada atualmente apresenta as seguintes características técnicas principais:

- Orquestração via **Docker Compose** (compose.prod.yaml)
- Servidor **Ubuntu** com **Docker v2** atualizado
- Gestão automatizada de deployment com **GitHub Actions**
- Base de dados **PostgreSQL** com volumes persistentes
- Servidor de aplicação **Gunicorn (WSGI)**
- Reverse proxy seguro via **NGINX** com headers de segurança e compressão ativa
- Gestão de segredos segura via **GitHub Secrets**

Apesar dos avanços significativos na infraestrutura da aplicação MentHA Digital, houve uma tarefa que não conseguimos concretizar nesta fase: a ligação da nova infraestrutura de produção ao servidor oficial da Universidade Lusófona. A máquina virtual (VM) atualmente utilizada pela instituição já se encontra em uso, com o NGINX instalado diretamente no sistema e a alojar múltiplos websites institucionais. Esta configuração impede a adoção de uma infraestrutura conteinerizada completa, como a nossa, que inclui o seu próprio serviço NGINX, sem correr o risco de comprometer os serviços existentes. Por falta de tempo e indisponibilidade de recursos técnicos, não foi possível solicitar e configurar uma nova VM dedicada à nossa aplicação.

No entanto, deixámos este processo cuidadosamente documentado e preparado para futuras equipas técnicas. No ficheiro README.md do repositório, incluímos um **guiia completo para replicação, deploy e gestão da infraestrutura**, com instruções passo a passo que facilitam a migração futura da plataforma para o ambiente institucional da Universidade Lusófona, mantendo a automação, segurança e consistência atualmente implementadas.

Neste momento, o **novo MVP da plataforma encontra-se disponível publicamente no domínio <http://menthadigital.pt/>**, associado à VM provisionada na **plataforma DigitalOcean**, que serve como ambiente principal de *staging* e validação de produção. Esta VM possui as seguintes especificações técnicas:

- **1 GB de Memória RAM**
- **25 GB de Disco SSD**
- **Sistema Operativo: Ubuntu 24.04 (LTS) x64**

Em contraste, o **servidor institucional da Universidade Lusófona**, que ainda aloja uma versão anterior da aplicação, possui as seguintes especificações:

- **CPU com frequência mínima de 2 GHz**
- **Pelo menos 500 MB de memória RAM**
- **Armazenamento local ou em nuvem para ficheiros estáticos**
- **Ligação de rede estável e de alta capacidade**

Apesar de tudo a versão desatualizada alojada no servidor da Iusófona ainda pode ser acedido em <https://menthadigital.com/>.

4.5 Abrangência

Conforme destacado na **Secção 4.2**, este projeto foi desenvolvido com base em ferramentas de programação web, sendo fortemente influenciado pela unidade curricular de Programação Web. O conhecimento adquirido na construção de modelos MVT (Model-View-Template) foi fundamental para o desenvolvimento das diversas páginas da aplicação, um processo que teve origem em projetos de TFC anteriores, servindo como alicerce para a presente fase do trabalho.

As disciplinas de Engenharia de Requisitos e Testes e Engenharia de Software tiveram um papel crucial na identificação de requisitos, elaboração de casos de uso e criação de diagramas de suporte. Além disso, contribuíram significativamente para o planeamento e a organização do projeto, uma vez que, nelas, adquirimos o conhecimento sobre a metodologia ágil, que está sendo aplicada na gestão desta iniciativa. A aplicação prática dessa metodologia será detalhada posteriormente, na **Secção 6**.

A modelagem das bases de dados, um componente essencial para a gestão eficiente das informações dos utilizadores, foi estruturada com base nos conhecimentos adquiridos na unidade curricular de Bases de Dados. Por fim, a aplicação dos princípios de Interação Humano-Computador foi determinante para garantir um design intuitivo e uma experiência de usabilidade que atenda plenamente às necessidades dos utilizadores.

4.6 Desafios Enfrentados e Como os Ultrapassamos

O desenvolvimento do MentHA Digital tem sido um desafio singular dentro do contexto de Trabalhos Finais de Curso (TFC). Diferente de projetos tradicionais, em que se inicia do zero com uma definição prévia de requisitos e um desenvolvimento estruturado, encontramos um sistema complexo já em funcionamento. Herdar um projeto de grande dimensão, desenvolvido por vários grupos ao longo de cinco gerações, exigiu adaptação, organização e resiliência.

Até este ponto do desenvolvimento, enfrentámos diversas dificuldades técnicas e organizacionais, que foram sendo superadas com diferentes estratégias. No entanto, o projeto ainda está em andamento e novos desafios poderão surgir à medida que avançamos. Neste capítulo, abordamos as principais dificuldades encontradas, dentro de cada uma dessas dificuldades abordamos os principais desafios encontrados até agora e as soluções que implementámos para os ultrapassar.

4.6.1 Adaptação a um Sistema Existente

4.6.1.1 Complexidade do Código Herdado

Ao ingressarmos no projeto, um dos primeiros desafios foi compreender a estrutura de um código denso e complexo, desenvolvido ao longo dos anos por diferentes equipas. A falta de padrões consistentes de desenvolvimento e a variedade de abordagens utilizadas dificultaram a

nossa curva de aprendizagem. Além disso, cada geração de desenvolvedores adotou metodologias diferentes, o que criou inconsistências no código e dificultou a sua manutenção.

Solução:

- **Analise detalhado do repositório:** Para reduzir a complexidade inicial, realizámos uma análise detalhada do repositório, identificando os principais módulos e a sua interdependência. Esta análise permitiu-nos organizar melhor as diferentes partes do código, separando funcionalidades e documentando padrões recorrentes para facilitar futuras modificações.
- **Criação de documentação:** Conscientes da falta de documentação, iniciámos um esforço contínuo para criar um padrão de documentação.
- **Adoção de boas práticas:** Por fim, adotámos um conjunto de boas práticas e guidelines, documentadas internamente, para garantir uma abordagem mais uniforme ao desenvolvimento. Esta estratégia não só melhorou a organização do código existente, como também tornou mais fácil a implementação de novas funcionalidades.

4.6.1.2 Falta de Documentação

Um dos problemas mais impactantes foi a escassez de documentação. A inexistência de um guia de configuração fez com que gastássemos várias semanas apenas para conseguir executar a aplicação localmente. Muitas configurações estavam desatualizadas ou não eram compatíveis com as versões mais recentes das tecnologias utilizadas, o que exigiu um esforço adicional para tornar o sistema funcional.

Para superar esse desafio, implementámos diversas estratégias para garantir que as informações essenciais fossem organizadas e acessíveis para futuras equipas:

- **Criação de documentação técnica detalhada:** Elaborámos um manual técnico que descreve passo a passo como configurar o ambiente de desenvolvimento, incluindo requisitos de software, dependências e instruções para executar o projeto localmente. Este guia está armazenado no repositório para fácil acesso e atualização.
- **Comentários explicativos no código:** Assim como feito para lidar com a complexidade do código herdado, adicionámos comentários detalhados em trechos críticos, explicando a lógica implementada e a função de cada componente. Isso reduz a curva de aprendizagem para novos programadores e facilita futuras manutenções.

A Secção Anexo E - Documentação deste relatório detalha os desafios específicos encontrados na documentação e como os ultrapassámos.

4.6.2 Manutenção e Melhoria do Projeto

4.6.2.1 Problemas no Re却tório e Configuração Inicial

Ao ingressarmos no projeto, um dos primeiros desafios foi a estrutura do repositório, que continha diversas configurações desatualizadas, arquivos desnecessários e problemas na gestão de dependências. Esses problemas resultaram em dificuldades para configurar o ambiente de desenvolvimento e garantir que o sistema fosse executado corretamente em diferentes máquinas.

Os principais problemas identificados foram:

- Arquivos redundantes e desorganização geral do repositório.
- Dependências desatualizadas no requirements.txt, com algumas bibliotecas obsoletas ou incompatíveis com versões recentes do Django e do PostgreSQL.
- Ficheiros de cache e migrações desnecessárias no repositório, dificultando a gestão das versões.
- Problemas com o ficheiro gitignore, que permitia a inclusão de arquivos que não deveriam ser rastreados, como cache de pacotes, migrações locais e configurações específicas.

Solução:

1. **Limpeza e reestruturação do repositório:** Realizamos uma análise detalhada dos arquivos e removemos itens desnecessários, garantindo que apenas arquivos essenciais estivessem no repositório. Criamos uma estrutura de pastas mais organizada para facilitar a navegação.
2. **Atualização e padronização das dependências:** Atualizamos o requirements.txt, substituindo bibliotecas obsoletas e testando a compatibilidade com as versões mais recentes do Django e PostgreSQL. Além disso, implementamos um requirements-dev.txt para diferenciar dependências de desenvolvimento e produção.
3. **Correção do gitignore:** Adicionamos um arquivo gitignore adequado para projetos Django, evitando que arquivos de cache, chaves de configuração e migrações locais fossem rastreados. Esse problema é abordado em mais detalhe na **Anexo F - Importância do gitignore** deste relatório.

4.6.2.2 Configuração do Ambiente de Desenvolvimento

A ausência de um guia de configuração adequado tornou a inicialização do ambiente uma tarefa extremamente complicada. A única documentação disponível era um README.md vago, com versões desatualizadas de pacotes e sem informações sobre a configuração do ambiente virtual, base de dados e dependências necessárias. Esse problema resultou em semanas de tentativas e erros até que conseguíssemos executar o projeto corretamente.

Solução:

1. **Elaboração de uma documentação detalhada:** Criamos um manual técnico que cobre todas as etapas de configuração do ambiente de desenvolvimento, desde a instalação do Python e criação de um ambiente virtual até a execução local do projeto.
2. **Padronização do ambiente de desenvolvimento:** Adoção de práticas de **DevOps** como **Infrastructure as Code (IaC)** através de Docker e Docker Compose [DOCK25] permitiu a padronização e automação da configuração do ambiente, reduzindo erros humanos e acelerando a integração de novos membros na equipa. A explicação detalhada desta abordagem está presente no **Anexo G – Configuração do Ambiente de Desenvolvimento com Docker e Docker Compose** deste relatório.

4.6.2.3 Inconsistências entre Repositório e Servidor de Produção

Um dos problemas mais críticos que encontramos foi a discrepância entre o código presente no repositório e a versão em produção. Descobrimos que alterações haviam sido feitas diretamente no servidor, sem registo no controle de versão, o que causava inconsistências e dificultava a manutenção do sistema. Esta tarefa foi bastante difícil devido ao facto de nenhuma das versões do código ter sido desenvolvida por nós, ou seja, foi das nossas primeiras vezes a olhar para o código e tivemos de fazer uma análise completa das várias componentes e diferenças para chegar ao melhor resultado.

Solução:

1. **Extração da versão em produção e comparação com o repositório:** Fizemos um levantamento das diferenças entre os dois códigos, analisando quais alterações eram relevantes e quais eram irrelevantes ou problemáticas.
2. **Realização de um merge detalhado:** Fizemos um merge manualmente das diferenças após um processo extensivo de revisão, garantindo que a versão final refletisse tanto os ajustes em produção quanto as melhores práticas de desenvolvimento.
3. **Implementação de um workflow adequado para Git:** Para evitar problemas futuros, estruturamos um fluxo de trabalho padronizado para controle de versão, inspirado em metodologias empresariais. Esse fluxo é descrito em detalhe no **Anexo D - Workflow de Desenvolvimento** deste relatório.

Com essas medidas, conseguimos tornar o repositório mais confiável e evitar problemas de sincronização entre o código local e o ambiente de produção. Isso garantiu maior segurança e previsibilidade no desenvolvimento e na implantação do sistema.

4.6.3 Tecnologia e Infraestrutura

4.6.3.1 Trabalho com Estruturas de Dados Grandes e Complexas

Um dos principais desafios que enfrentamos no projeto foi lidar com uma estrutura de dados extensa e interdependente. Muitas partes do sistema dependem umas das outras, tornando difícil realizar modificações sem afetar funcionalidades relacionadas. Essa interdependência aumentou significativamente a complexidade do desenvolvimento e a necessidade de testes rigorosos antes de qualquer alteração.

Os principais problemas encontrados foram:

- Dependências ocultas entre diferentes partes do sistema, tornando difícil prever o impacto de alterações.
- Dificuldade em entender a estrutura de dados existente sem sobrecarregar-se com informações desnecessárias.
- Falta de documentação clara sobre a relação entre os módulos do sistema.

Solução:

1. **Foco seletivo e abstração:** Para evitar sobrecarregar-nos com toda a complexidade do sistema de uma só vez, adotamos uma abordagem baseada na abstração. Concentrámos-nos apenas nas partes da estrutura de dados relevantes para os componentes

específicos em que estávamos a trabalhar, garantindo um entendimento gradual do sistema.

2. **Testes modulares:** Implementamos testes modulares para validar alterações em partes isoladas do código antes de realizar integrações mais amplas. Como complemento aos testes modulares, integrarmos esses testes nas pipelines de CI, seguindo a prática DevOps de **validação contínua**, garantindo que alterações em estruturas de dados críticas sejam verificadas automaticamente antes de chegarem ao ambiente de produção.

4.6.3.2 Adaptação a Novas Tecnologias

Durante o desenvolvimento do MentHA Digital, tivemos de utilizar tecnologias com as quais não estávamos familiarizados, o que representou um desafio adicional. Ferramentas como Bootstrap e Ajax já estavam parcialmente implementadas no sistema, e a necessidade de compreender o código existente antes de fazer ajustes dificultou a adaptação.

Os principais problemas enfrentados foram:

- Necessidade de aprender tecnologias ao mesmo tempo em que trabalhávamos em funcionalidades críticas.
- Código legado implementado com abordagens variadas, dificultando a compreensão de como certos componentes funcionavam.
- Dificuldade em depurar erros devido ao desconhecimento das ferramentas utilizadas.

Solução:

1. **Análise do código legado:** Antes de modificar funcionalidades existentes, realizámos uma análise detalhada do código legado para compreender como as tecnologias estavam a ser aplicadas.
2. **Testes incrementais:** Implementamos mudanças de forma incremental e validamos cada alteração antes de integrá-la no sistema principal.

4.6.3.3 Desenvolvimento com Ambiente de Produção e Desenvolvimento Separados

Trabalhar num ambiente que possui tanto uma versão de desenvolvimento como uma versão de produção trouxe desafios significativos. Alterações que funcionavam corretamente no ambiente de desenvolvimento, muitas vezes, apresentavam problemas quando implementadas em produção. Além disso, diferenças de configuração entre os dois ambientes dificultavam a deteção e resolução de erros.

Os principais desafios enfrentados foram:

- Diferenças de configuração entre os ambientes, causando erros inesperados em produção.
- Risco de interrupção do sistema ao realizar atualizações na produção.

- Para lidar com as diferenças entre os ambientes, estudamos práticas DevOps como o **princípio da paridade entre desenvolvimento e produção (dev/prod parity)**, que reforça a importância de manter os ambientes o mais semelhantes possível, usando ferramentas como Docker.

4.6.4 Colaborar com Diferentes Áreas

O nosso projeto está fortemente ligado à área da saúde, o que representa um desafio, pois somos alunos de engenharia informática e não dominamos os aspectos técnicos desse domínio. Desenvolver uma solução baseada em conceitos que não conhecemos na totalidade exige uma pesquisa aprofundada para garantir que atendemos às exigências propostas.

Os principais desafios enfrentados foram:

- Necessidade de compreender conceitos da área da saúde sem formação específica na área.
- Dificuldade em traduzir a linguagem técnica dos especialistas em saúde para algo que a máquina consiga interpretar corretamente.

Solução:

- **Análise detalhada do modelo e das necessidades do utilizador:** Sempre que realizamos alterações a um componente, analisamos todas as classes do modelo associadas a ele. Além disso, procuramos compreender o objetivo do utilizador final, pesquisando, por exemplo, porque determinados dados médicos são coletados e qual a sua utilidade.

4.7 Componentes

O nosso trabalho ao contrário de muitos projetos académicos que começam do zero, insere-se num sistema já desenvolvido e em funcionamento, com várias versões e gerações de equipas que contribuíram para o seu crescimento ao longo do tempo. A natureza herdada deste projeto exigiu da nossa parte não só a compreensão profunda das componentes existentes, como também a sua manutenção, correção de falhas e desenvolvimento de novas funcionalidades.

Cada uma destas componentes encontra-se descrita em detalhe no **Anexo A – Aplicações MentHA Digital**, onde são explicados os seus objetivos, estrutura funcional e impacto esperado nos utilizadores finais.

Com base na análise feita ao estado atual do sistema e às necessidades identificadas durante a fase de levantamento de requisitos, foi possível constatar que, para além das tarefas inicialmente planeadas, tivemos de lidar com uma série de desafios técnicos e organizacionais. Estes obstáculos, amplamente abordados na **Secção 4.6**, obrigaram-nos a realizar tarefas que ultrapassam a simples manutenção, como por exemplo a reestruturação de repositórios, uniformização de ambientes, análise do código existente e resolução de inconsistências entre versões. Estas tarefas estão relacionadas com o épico **E2 – Infraestrutura e Re却itório**, que foca na organização e depuração do re却itório e na estabilidade do ambiente de desenvolvimento.

Posto isto, a nossa intervenção nas componentes do projeto passou a englobar dois eixos principais:

- Correção de erros existentes, como bugs funcionais e problemas de compatibilidade entre versões;
- Desenvolvimento de novas funcionalidades, visando responder a requisitos técnicos e operacionais que surgiram durante o uso real da plataforma pelas equipas parceiras.

As alterações realizadas ao nível de cada aplicação serão descritas de forma detalhada na secção seguinte, onde explicamos quais os componentes específicos modificados, as tecnologias envolvidas e o impacto direto de cada melhoria na usabilidade e estabilidade do sistema.

Na **Secção 5.1**, mais á frente estão explicados os **testes realizados e os resultados obtidos de cada uma destas alterações**.

4.7.1 Exportação de Avaliações Risk em Ficheiro CSV

Requisito associado: E1FPTU8 – Exportação de Dados Estatísticos em CSV

Foi implementada uma nova funcionalidade no módulo MentHA EVAL da aplicação Protocolo MentHA, que permite a exportação de todas as avaliações do questionário MentHA-Risk realizadas pelos participantes registados na base de dados. Esta funcionalidade foi desenvolvida com o objetivo de facilitar a análise posterior dos dados recolhidos, nomeadamente para efeitos de estudo, monitorização clínica ou melhoria contínua do protocolo de avaliação.

A funcionalidade encontra-se disponível através do endpoint “*/protocolo/exportar_risk/*”, o qual está protegido por autenticação. Quando acedido, este endpoint invoca a view **export_risk_to_csv**, que gera automaticamente um ficheiro CSV contendo, por linha, os dados relevantes de uma instância de avaliação Risk associada a um participante.

- O ficheiro gerado é denominado relatorio_risk.csv e inclui as seguintes colunas:
- Dados identificativos do participante (nome, instituição associada);
- Data e hora da avaliação;
- Variáveis clínicas e biométricas obtidas durante o preenchimento do questionário Risk (ex.: idade, sexo, IMC, pressão arterial, níveis de colesterol, hemoglobina glicada, etc...);
- Informação complementar como o risco estimado de enfarte e os identificadores internos atribuídos ao participante;

Para garantir a coerência dos dados exportados, foram implementadas validações que substituem valores não preenchidos por 'null', de forma a evitar ruído estatístico nas análises futuras. Adicionalmente, participantes sem avaliações associadas à avaliação Risk são devidamente tratados com o preenchimento automático das colunas com a indicação '**risk nao associado**', permitindo assim manter a uniformidade da estrutura do ficheiro CSV. Na **Figura 9** pode se observar a parte inicial do código desta view.

```

@login_required(login_url='login')
def export_risk_to_csv(request):
    response = HttpResponse(content_type='text/csv; charset=utf-8-sig')
    response['Content-Disposition'] = 'attachment; filename="relatorio_risk.csv"'

    writer = csv.writer(response)
    writer.writerow([
        'Participante', 'Instituição', 'Data Questionario', 'Hora Questionario', 'Sexo', 'Idade', 'Peso', 'Altura', 'IMC', 'Pressão Arterial',
        'Colesterol Total', 'Colesterol HDL', 'Colesterol Não HDL', 'Hemoglobina Glicilada',
        'EAG HBA1', 'IFCC HBA1', 'NGSP HBA1', 'Horas Jejum', 'Doença Cognitiva', 'Pré-Diabético',
        'Pergunta Cardiovascular', 'Fumador', 'Diabetes', 'Anos de Diabetes', 'AVC', 'Enfarre',
        'Doença nos Rins', 'Doença nas Pernas', 'Hipercolesterol', 'TG', 'LDL', 'CHOL-HDL', 'Batimentos', 'Pressão Arterial Diastólica',
        'Risco de Enfarre', 'Pat ID', 'Pat ID V2'
    ])
    #removemos campos comentários e recomendações pois para a analise de dados não são necessários

    for participante in Participante.objects.all():
        instituicao = participante.referenciacao if participante.referenciacao else 'Instituição não Associada' # Caso não tenha instituição
        for partes in ParteDoUtilizador.objects.filter(participante=participante, part__name='MentHA-Risk').order_by('-data', '-time'):
            # partes = ParteDoUtilizador.objects.filter(participante=participante, part__name='MentHA-Risk').order_by('-data', '-time')

            if partes:
                try:
                    risk = partes.risk

                    if risk.altura != 0:
                        altura = risk.altura
                    else:
                        altura = 'null'

                    if risk.horas_jejum != 0:
                        horas_jejum = risk.horas_jejum
                    else:
                        horas_jejum = 'null'

                    if risk.pressao_arterial_diastolica != 0:
                        pressao_arterial_diastolica = risk.pressao_arterial_diastolica
                    else:
                        pressao_arterial_diastolica = 'null'
                except:
                    pass
    
```

Figura 9 - View Exportação Risk

4.7.1.1 Considerações Técnicas

A exportação é realizada utilizando a biblioteca csv do Python, sendo o ficheiro gerado em tempo real e enviado como resposta HTTP com `content_type='text/csv; charset=utf-8-sig'`, o que garante compatibilidade com editores como o Excel e previne problemas de codificação de caracteres. A estrutura da função prevê ainda:

- Ordenação das avaliações por data e hora (do mais recente para o mais antigo);
- Tratamento de exceções caso uma avaliação Risk não esteja corretamente associada ao utilizador;
- Adaptação futura em caso de alteração da estrutura do modelo Risk ou dos seus campos. Esta funcionalidade representa um passo importante na consolidação de ferramentas de análise integradas ao sistema MentHA, permitindo maior controlo e acessibilidade sobre os dados clínicos recolhidos ao longo do tempo.

Os testes de validação desta nova componente estão presentes na **Secção Teste de Validação de Alterações na Criação de Grupos – MentHA COG**

4.7.2 Criação de Grupos de Sessão com Filtros Multivariados (Reformulação da View)

Requisito associado: E1FCOGU12 – Reformulação de View de Criação de Grupo

Foi realizada uma reformulação estrutural da view responsável pela criação de grupos de sessão no módulo MentHA DIGITAL, mais especificamente no contexto do programa MentHA-COG. Esta reformulação surgiu da necessidade de resolver problemas críticos relacionados com o armazenamento incorreto de dados e a inexistência de ligações adequadas entre entidades fundamentais como grupos, participantes, secções e exercícios.

Esta parte da aplicação revelou-se crítica, uma vez que todo o funcionamento do protocolo MentHA-COG depende da correta criação e organização dos grupos. Dentro de cada grupo encontram-se encapsulados os participantes, as secções e os exercícios a realizar, o que torna esta view central para a integridade dos dados e o correto funcionamento das sessões.

Problemas Identificados

Durante a fase de testes e validação, foram identificadas diversas falhas importantes:

- A view permitia a submissão do formulário mesmo sem o preenchimento dos campos obrigatórios, levando a situações em que um grupo aparentemente era criado, mas não era persistido na base de dados.
- As ligações entre os participantes e os grupos não estavam a ser corretamente estabelecidas, o que comprometia todo o fluxo do protocolo MentHA-COG.

Na **Figura 10**, está apresentada uma pequena parte desta view.

```
@login_required(login_url='diario:login')
@check_user_able_to_see_page('Todos')
def new_group(request):
    # Obtém os grupos associados ao utilizador
    grupos, sg, is_participante, is_cuidador = get_grupos(request.user)

    # Verifica se há uma próxima sessão agendada
    tem_proxima, datas = get_proxima_sessao(grupos)

    # Variáveis auxiliares
    flag = None
    referenciaicao = None

    # Inicializa o formulário de criação de grupo
    formGrupo = GrupoForm(request.POST or None)

    # Obtém o papel do utilizador logado (dinamizador, mentor ou administrador)
    dinamizador = DinamizadorConvidado.objects.filter(user=request.user).first()
    mentor = Mentor.objects.filter(user=request.user).first()
    administrador = Administrador.objects.filter(user=request.user).first()

    # Define a referência com base no papel do utilizador
    if dinamizador:
        referenciaicao = dinamizador.reference
    elif mentor:
        referenciaicao = mentor.reference
    elif administrador:
        referenciaicao = administrador.reference

    # Filtragem de cuidadores (para o programa CARE)
    cuidadores = Cuidador.objects.all()
    filtrados_care = cuidadores.filter(grupo=None)

    conjunto_doenças = set()
    for cuidador in cuidadores:
        conjunto_doenças.update(cuidador.doenças_object)
```

Figura 10 - View Criação de Grupo

Solução Implementada

A solução passou por uma reformulação total da lógica da view e da interface correspondente, com ênfase em três grandes áreas:

- **Validação Preventiva no Frontend:** Foi desenvolvido um conjunto de scripts em JavaScript que impede o carregamento da interface de seleção de participantes caso o nome do grupo e o programa (CARE ou COG) não estejam simultaneamente

preenchidos. Só após esta dupla validação é que é revelada ao utilizador a tabela com os potenciais candidatos ao grupo, bem como o botão “Criar Grupo”.

- **Renderização Condicional de Filtros:** A interface agora apresenta apenas os filtros relevantes consoante o programa selecionado. Estes filtros encontram-se separados por classes (.filtro_care e .filtro_cog) e são ativados dinamicamente por JavaScript sempre que o utilizador altera a seleção do programa.
- **Reestruturação da View Backend:** A view foi reorganizada para garantir que todos os dados são corretamente validados e ligados antes de serem guardados. Foram introduzidas verificações adicionais no backend para prevenir a criação de grupos inválidos, e as associações entre grupo, participantes e exercícios foram revistas para garantir a integridade referencial.

A **Figura 11**, apresenta um excerto do código JavaScript responsável por validar os campos obrigatórios antes de permitir a interação com a tabela de participantes.

```

function verificaNome() {
    const nome = document.getElementById('nome').value;
    if (nome.length > 0) {
        nomePreenchido = true; // Nome preenchido
    } else {
        nomePreenchido = false; // Nome não preenchido
    }
    atualiza_candidatos()
}

function verificaPrograma() {
    const programa = document.getElementById('programa').value;
    if (programa.length > 0) {
        programaSelecionado = true; // Programa selecionado
    } else {
        programaSelecionado = false; // Nenhum programa selecionado
    }
    atualiza_candidatos()
}

function atualiza_candidatos() {
    form_data = new FormData();
    form_data.append("programa", document.getElementById('programa').value);
    form_data.append("nome", document.getElementById('nome').value);

    if (programaSelecionado && nomePreenchido) {
        // Se ambos os campos estiverem preenchidos
        if (document.getElementById('programa').value == "CARE"){
            form_data.append("localizacao", document.getElementById('Localizações_care').value);
            form_data.append("diagnóstico", document.getElementById('Diagnósticos_care').value);
            form_data.append("escolaridade", document.getElementById('Escolaridades_care').value);
            form_data.append("referenciacao", document.getElementById('Referências_care').value);

            document.querySelector('.filtro_care').style.display = "block";
            document.querySelector('.filtro_cog').style.display = "none";
        }
        else if (document.getElementById('programa').value == "COG"){
            form_data.append("localizacao", document.getElementById('Localizações_cog').value);
            form_data.append("diagnóstico", document.getElementById('Diagnósticos_cog').value);
            form_data.append("escolaridade", document.getElementById('Escolaridades_cog').value);
            form_data.append("referenciacao", document.getElementById('Referências_cog').value);

            document.querySelector('.filtro_care').style.display = "none";
            document.querySelector('.filtro_cog').style.display = "block";
        }
    }
}

```

Figura 11 - Validação Condicional da Criação de Grupo via JavaScript

4.7.2.1 Considerações Técnicas

Esta reformulação representou um desafio técnico e exigiu uma análise profunda da lógica de negócio da aplicação MentHA-COG. Entre os pontos técnicos mais relevantes, destacam-se:

- **Prevenção de submissão inválida:** A submissão do formulário está agora condicionada à presença de dados essenciais, evitando registo incompletos.

- **Feedback visual e UX melhorado:** O sistema apenas revela a interface de seleção de participantes quando os campos obrigatórios estão corretamente preenchidos, melhorando a experiência de utilização.
- **Uso da API Fetch para atualização dinâmica:** O carregamento dos candidatos é feito através da função `atualiza_candidatos()`, que utiliza `fetch()` para enviar os dados ao servidor e atualizar dinamicamente a tabela sem recarregamento da página.

Esta alteração veio resolver um problema estrutural importante, contribuindo significativamente para a estabilidade e fiabilidade do módulo MentHA COG. Ao garantir que a criação de grupos só ocorre quando todos os critérios são corretamente preenchidos, assegura-se a consistência dos dados e a continuidade lógica das sessões. Os testes de validação desta alteração estão presentes na **Secção Teste de Validação de Alterações na Criação de Grupos – MentHA COG**

4.7.3 Criação de Protótipo para Remodelação Visual do Website MentHA Digital

Requisito associado: E1FMDU3 – Criação de Protótipo em Figma como sugestão de melhoria visual do website MentHA Digital

Durante a fase de testes exploratórios descrita mais à frente na **Secção 5.1.2**, foi possível identificar um conjunto significativo de falhas relacionadas com a interface atual do website oficial do projeto MentHA. Através de uma análise detalhada da experiência do utilizador e do comportamento da interface em diferentes dispositivos, concluiu-se que o website apresentava vários problemas críticos que comprometiam a usabilidade, acessibilidade e imagem institucional do projeto.

Na **Figura 12, Figura 13 e Figura 14**, estão demonstradas algumas partes do web site atual.

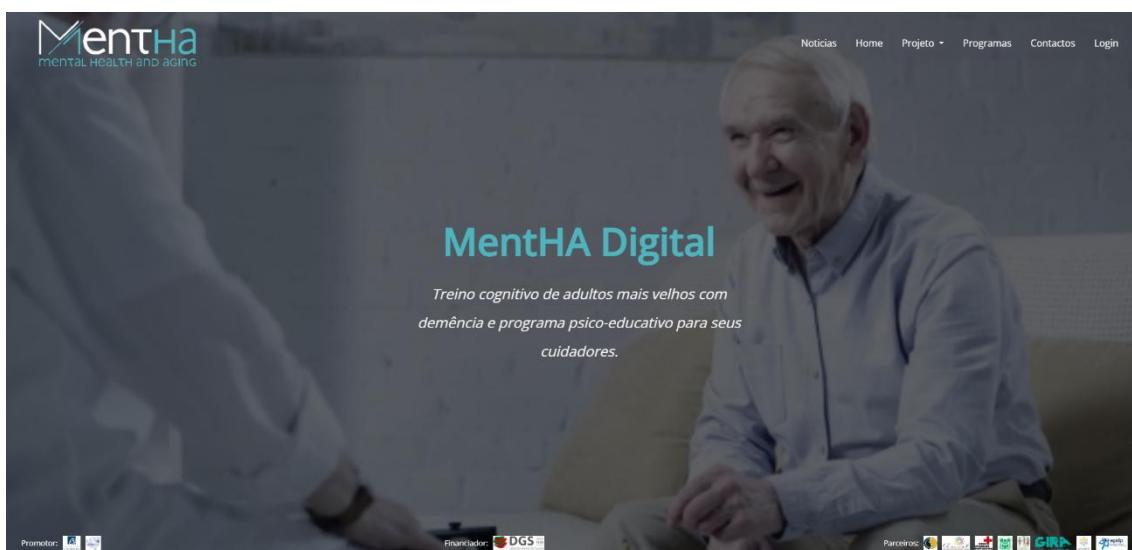


Figura 12 - Website MentHA Digital 1



Figura 13 - Website MentHA Digital 2

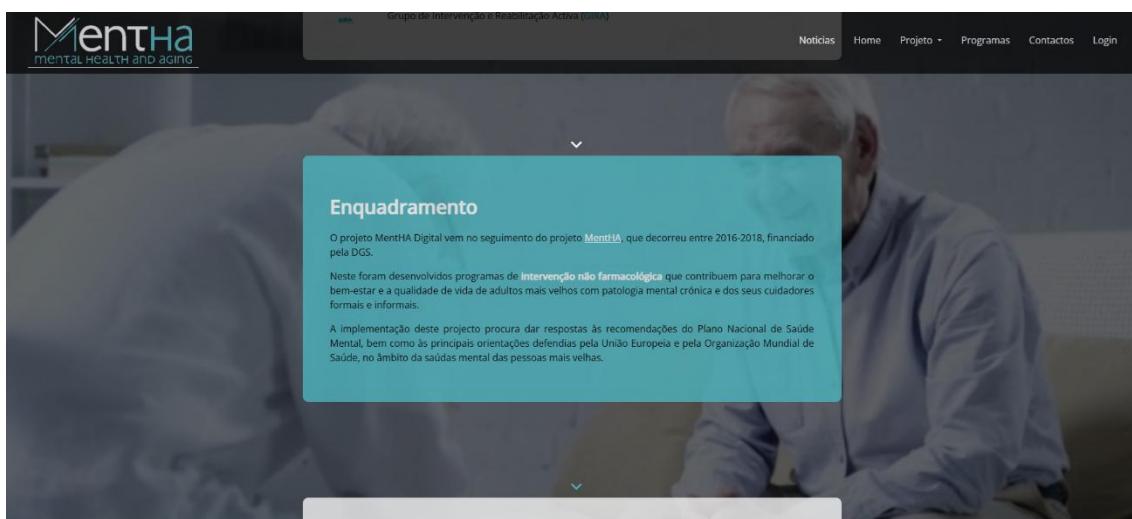


Figura 14 - Website MentHA Digital

Tendo em conta a importância estratégica do website enquanto ponto de contacto com utilizadores finais, instituições parceiras, investigadores e potenciais financiadores, foi tomada a decisão de desenvolver um **protótipo de reformulação visual** com o objetivo de corrigir estas lacunas.

O novo protótipo foi criado na plataforma **Figma**, com base em princípios modernos de design e usabilidade, e respeitando integralmente a identidade visual do projeto MentHA. Este trabalho visou não só melhorar a apresentação e navegação do site, como também garantir maior acessibilidade e coerência visual em todos os dispositivos.

Este protótipo já se encontra concluído e será a base para o desenvolvimento da nova versão do website, que será abordado posteriormente no requisito **E1FMDU4 – Alteração da Interface Visual do Website com base no Protótipo**, atualmente em fase pendente de implementação.

Na **Figura 15**, **Figura 16** e **Figura 17**, estão demonstradas algumas partes do protótipo.



Figura 15 - Protótipo Website MentHA 1

A screenshot of the second prototype of the MentHA website, featuring a teal header with the text 'Descubra nossas aplicações: Melhorando a saúde mental e o bem-estar'. To the right of the header is a text block about the project's goal to improve the quality of life of elderly adults with cognitive deficits and chronic mental diseases. Below the header are three mobile device screenshots displaying different applications: 'MentHA COG: Programa de Estimulação Cognitiva', 'MentHA EVAL: Avaliação Neuropsicológica', and 'MentHA CARE: Programa de Apoio Psicoeducativo para Cuidadores'. Each application section includes a brief description and a 'Ver mais' link.

Figura 16 - Protótipo Website MentHA 2



Expectativas do projeto MentHA Digital

Esta plataforma permitirá que profissionais de saúde realizem **intervenções psicosociais em modo misto** - presencial e remoto - por forma a ultrapassar os efeitos do agravamento do estado de saúde, da alteração de contexto geográfico, do distanciamento social e/ou de situação de isolamento.

Disponibilização da plataforma

Através da DGS, pretende-se que a plataforma aplicacional MentHA Digital seja disponibilizada a instituições e técnicos da área, em todo o país, potencializando a viabilidade deste projeto em grande escala.

Figura 17 - Protótipo Website MentHA 3

Estas figuras são as novas propostas para as partes referidas em cima nesta Secção que se podem ver nas figuras **Figura 12**, **Figura 13** e **Figura 14**.

4.7.4 Desenvolvimento da Nova Home Page do Website MentHA Digital

Requisito associado: E1FMDU4 – Alteração da Interface Visual do Website com base no Protótipo Figma

Após a conclusão e validação do protótipo visual na plataforma Figma (conforme descrito na **Secção 4.7.3**), avançou-se para a fase de desenvolvimento efetivo da nova interface, começando pela remodelação da **Home Page** do website MentHA Digital.

O processo de implementação seguiu as seguintes etapas:

- **Análise funcional do protótipo:** Mapeamento detalhado dos elementos visuais e funcionais do protótipo Figma, identificando componentes, secções, hierarquia de conteúdos e interações a transpor para o código.
- **Adaptação do template Django existente:** Reformulação do template principal do projeto (mentha/templates) para incorporar a nova estrutura visual, mantendo compatibilidade com o restante sistema e com os controladores de backoffice existentes.
- **Implementação responsiva:** Garantia de que a nova interface é totalmente **responsive**, com adaptação automática a múltiplos dispositivos e tamanhos de ecrã (desktop, tablet e mobile).

- **Testes de compatibilidade:** Validação do correto funcionamento da nova interface em diferentes navegadores, sistemas operativos e dispositivos, assegurando a máxima compatibilidade e acessibilidade.

Na **Figura 18** e **Figura 19** são apresentados alguns exemplos visuais da nova **Home Page** já desenvolvida e integrada no sistema real.

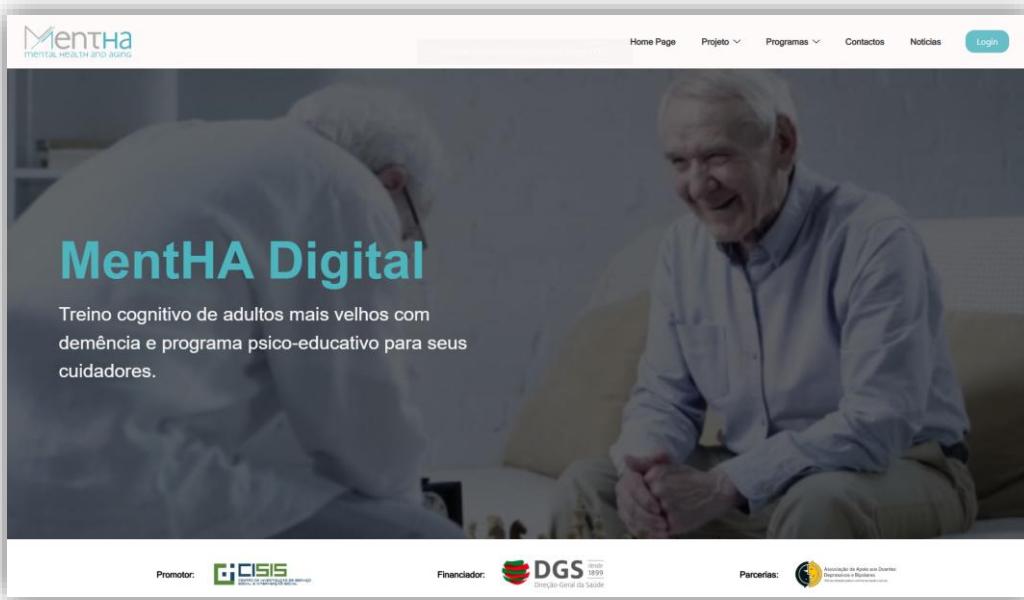


Figura 18 - Home Page Screen 1



Figura 19 - Home Page Screen 2

4.8 Interfaces

As figuras dos ecrãs das aplicações do MentHA estão organizadas e podem ser consultadas com mais profundidade no **Anexo A – Aplicações MentHA Digital** deste relatório, onde a explicação de cada aplicação é fornecida separadamente, incluindo os prints de todos os ecrãs de cada aplicação.

Neste anexo podem encontrar:

- **A.1 MentHA COG – Programa de Estimulação Cognitiva para Adultos Mais Velhos:** Explicação da Aplicação MentHA COG e as suas respetivas prints dos ecrãs.
- **A.2 Protocolo MentHA – Avaliação Neuropsicológica para Validação de Treino Cognitivo:** Explicação da Aplicação Protocolo MentHA e as suas respetivas prints dos ecrãs.
- **A.3 MentHA CARE – Programa de Apoio Psicoeducativo para Cuidadores:** Explicação da Aplicação MentHA CARE e as suas respetivas prints dos ecrãs.

5 Testes e Validação

Este capítulo é fundamental para demonstrar o funcionamento prático da solução desenvolvida, bem como para validar se esta cumpre os objetivos inicialmente propostos. Ao longo do desenvolvimento do nosso Trabalho Final de Curso, enfrentámos diversos desafios de natureza estrutural que afetavam a operacionalidade e a fiabilidade do sistema. Assim, grande parte do nosso foco inicial centrou-se na resolução desses problemas fundamentais, de forma a garantir uma base sólida e funcional sobre a qual futuras melhorias pudessem ser implementadas.

Após estas correções estruturais, foi possível avançar com testes direcionados à validação funcional, à entrada e normalização de dados, bem como à experiência do utilizador. Complementarmente, foram também realizados testes unitários e testes de integração com o ambiente local, utilizando Docker Compose, de forma a garantir a replicabilidade e estabilidade do sistema.

Importa referir que, apesar de não termos realizado diretamente sessões de testes com instituições parceiras durante este ano letivo, os testes conduzidos no ano anterior (com parceiros como GIRA, ADEB, ASMAL e Crinabel) foram fundamentais para o levantamento de requisitos e para a deteção dos problemas que abordámos. O nosso trabalho atual parte, assim, dos resultados desses testes, procurando responder com soluções concretas às necessidades anteriormente identificadas.

5.1 Testes de Funcionalidades

Os testes de funcionalidades realizados incidiram sobre alterações e melhorias concretas nas aplicações MentHA COG e MentHA EVAL, bem como na instalação e execução do sistema em ambientes distintos. Estes testes visaram garantir que os problemas identificados foram efetivamente resolvidos e que a solução é operacional e robusta.

5.1.1 Teste de Instalação e Execução em Ambiente Local com Docker Compose

Após diversas melhorias estruturais na organização do projeto, a adoção do **Docker Compose** tornou-se uma peça-chave na padronização e portabilidade da aplicação. Para validar a robustez da nova estrutura e a eficácia da documentação produzida, foi realizado um teste prático de instalação e execução do sistema numa máquina distinta da de desenvolvimento, sem qualquer configuração prévia.

Objetivo:

Verificar se o projeto pode ser instalado e executado corretamente noutra máquina, utilizando apenas o guia disponibilizado no README do repositório.

Metodologia:

- Utilização de uma máquina nova, sem ambiente de desenvolvimento previamente instalado.
- Clonagem do repositório.
- Seguir passo a passo as instruções do README.md, incluindo:
- Acesso ao sistema via navegador e validação da interface e funcionalidades principais.

5.1.2 Teste de Validação de Alterações na Criação de Grupos – MentHA COG

Foi identificada uma falha crítica na **view de criação de grupos** no módulo **MentHA COG**: a submissão de formulários com campos obrigatórios em branco era aceite, resultando na criação de grupos inválidos.

Objetivo:

Evitar a submissão de grupos incompletos e garantir que apenas grupos válidos são criados, melhorando a experiência do utilizador e a integridade dos dados.

Intervenção:

- Reestruturação da lógica da interface com **JavaScript**, implementando uma lógica condicional:
 - Apenas quando os campos “Nome do grupo” e “Programa” estão preenchidos, é exibida a caixa para inserção de elementos e o botão de submissão.

Metodologia de Teste:

- Tentativas de submissão com campos em branco;
- Teste da sequência correta de preenchimento;
- Criação de grupos com diferentes combinações válidas e inválidas.

5.1.3 Teste de Exportação de Avaliações – MentHA Eval

No módulo **MentHA EVAL**, foi adicionada uma nova **view de exportação para CSV** contendo dados de avaliações feitas a pacientes, incluindo:

- Nome do paciente;
- Associação;
- Data da avaliação;
- Dados preenchidos nos formulários.

Objetivo:

Permitir a análise detalhada dos dados de avaliação pelas associações parceiras, incluindo a identificação de inconformidades, com possibilidade de aplicação de técnicas de data science.

Intervenção:

- Criação de uma nova rota e view;
- Implementação de lógica para exibir pacientes com e sem avaliações;
- Exportação direta dos dados para ficheiro .csv.

Metodologia de Teste:

- Realização de **testes unitários** para verificar:
 - A inclusão de todos os campos esperados;

- A presença de pacientes sem avaliações;
- Validação manual do ficheiro .csv gerado;
- Teste em ambiente real com dados reais das instituições.

5.2 Teste de CI/CD e Deployment com GitHub Actions

Com a integração completa da infraestrutura de Continuous Integration / Continuous Deployment (CI/CD) através do GitHub Actions, tornou-se essencial validar se o pipeline automatizado é fiável e garante deployments consistentes para o ambiente de staging, alojado num droplet da DigitalOcean. Este teste teve como objetivo garantir que qualquer alteração no código da aplicação – seja a nível visual, estrutural ou funcional – desencadeia corretamente todo o processo de build, teste e deployment sem intervenção manual.

Objetivo

Validar que:

- A pipeline GitHub Actions realiza corretamente os testes e o deploy sempre que há um push na branch main;
- Alterações feitas em ficheiros estáticos (ex: CSS, JS) ou HTML são corretamente refletidas no ambiente de staging após o deploy automatizado;
- O servidor executa corretamente os comandos de rebuild e reload dos serviços Docker com a nova versão da aplicação.

Metodologia

- Modificação simples no projeto (ex: alteração no style.css ou no ficheiro base.html);
- Commit e push para a branch main do repositório remoto no GitHub;
- Monitorização da execução do pipeline deploy.yml no separador “Actions” do GitHub;
- Acompanhamento dos passos do workflow:
 - Execução de testes em container PostgreSQL temporário;
 - Upload dos ficheiros via scp para o servidor remoto;
 - Execução de comandos remotos via SSH: shutdown dos containers antigos, rebuild e restart da infraestrutura via docker-compose.prod.yaml;
 - Execução de collectstatic para atualização dos ficheiros estáticos;
- Validação visual da aplicação no navegador, garantindo que as alterações estão refletidas na versão em produção (staging).

5.3 Testes Unitários

Os testes unitários, juntamente com o uso intensivo das ferramentas de debug do Django, foram essenciais para garantir o funcionamento correto das funcionalidades desenvolvidas. Estes testes permitiram validar a lógica dos modelos e das views, assegurando que os dados são guardados corretamente na base de dados e que os resultados apresentados ao utilizador correspondem ao esperado.

Muitos dos erros estruturais detetados inicialmente foram resolvidos com base em testes iterativos realizados por nós próprios, durante o desenvolvimento.

5.4 Testes da Interface do Utilizador (UI/UX)

Foram realizados testes manuais em várias aplicações do ecossistema MentHA com o objetivo de verificar se todos os componentes da interface eram exibidos corretamente e apresentavam o comportamento esperado. Estes testes desempenharam um papel importante no levantamento de requisitos, uma vez que uma das maiores tarefas ao longo do ano consistiu precisamente na identificação e correção de problemas em componentes já existentes, especialmente aqueles que impactavam diretamente a experiência do utilizador.

Um dos testes mais relevantes incidiu sobre o website oficial do projeto MentHA, cuja interface se revelou particularmente problemática. Durante a análise foram detetados diversos erros e inconsistências, tais como:

- Layout visual completamente desatualizado, com aparência pouco moderna e desorganizada;
- Estrutura da informação pouco clara, o que dificultava a navegação e a compreensão dos conteúdos;
- Vários elementos desalinhados, com má adaptação a ecrãs de diferentes tamanhos;
- Falta de responsividade e ausência de princípios básicos de acessibilidade;
- Experiência de navegação confusa e pouco intuitiva, especialmente para públicos mais sensíveis, como pessoas idosas ou com dificuldades cognitivas.

Estes problemas revelaram-se particularmente significativos tendo em conta o público-alvo da plataforma. O website era, por natureza, um ponto de contacto essencial com instituições parceiras, cuidadores, pacientes e potenciais financiadores ou entidades interessadas. Assim, tornou-se evidente que uma apresentação cuidada, apelativa e acessível era essencial para transmitir confiança e profissionalismo.

Face a este cenário, foi decidido avançar com um processo de reformulação visual completa do website MentHA. Foi então criado um protótipo de uma nova interface, mais moderna, funcional e acessível, alinhado com boas práticas de design e usabilidade. Este novo design foi um dos focos principais das fases finais do projeto, garantindo uma experiência mais coerente e positiva para todos os utilizadores.

Estes testes revelaram-se, portanto, fundamentais não só para identificar falhas críticas, mas também para fundamentar a necessidade de uma intervenção profunda no design e na estrutura de navegação do sistema.

5.5 Teste de Entrada de Dados

Os testes de entrada de dados revelaram-se fundamentais para o desenvolvimento e estabilidade das funcionalidades descritas na **Secção 4.7**, sobretudo no que diz respeito ao **Protocolo MentHA**, que continua a ser uma das áreas mais sensíveis e críticas da aplicação.

Durante os testes realizados, foram identificados diversos problemas estruturais relacionados com os formulários de avaliação, nomeadamente:

- **Campos com valores pré-preenchidos** que apareciam estar ali apenas como exemplo, mas que, na prática, podiam ser interpretados como dados válidos, comprometendo a fiabilidade da informação.
- **Dados não normalizados**, com diferentes formatos, unidades ou nomenclaturas, dificultando o tratamento posterior da informação e a sua utilização noutras funcionalidades da aplicação.
- **Falta de obrigatoriedade** em campos considerados críticos, o que pode resultar em avaliações incompletas ou inconsistentes.

Estes problemas, apesar de aparecerem ser apenas erros pontuais na interface, têm implicações diretas e profundas no funcionamento geral do sistema, uma vez que a entrada de dados alimenta praticamente todas as outras áreas da aplicação, desde os relatórios até às análises e exportações de dados. A presença de dados mal estruturados pode gerar erros em cascata, afetando desde funcionalidades básicas até a interpretação final dos resultados por parte de cuidadores e profissionais de saúde.

Com base nestes testes, tornou-se claro que uma das prioridades – seja ainda nesta fase de desenvolvimento ou por parte de futuras equipas que deem continuidade ao projeto – deve ser a **normalização e reorganização completa dos formulários do Protocolo MentHA**. Esta tarefa deverá envolver:

- A redefinição das obrigatoriedades por campo;
- A eliminação de valores de exemplo;
- A uniformização das estruturas de dados (ex.: datas, medidas, classificações);

A melhoria destes aspetos será determinante para garantir que o sistema seja robusto, fiável e escalável, tornando-se uma ferramenta segura e eficaz para os profissionais que a utilizam no terreno.

5.6 Testes de Utilizador

Os testes realizados no ano letivo anterior envolveram diretamente utilizadores finais de várias instituições parceiras, permitindo o levantamento de dados valiosos. Os resultados desses testes incluíram:

- GIRA: 14 utilizadores registados, dos quais 12 realizaram o MentHA Risk.
- ADEB: 8 utentes, com 5 avaliações realizadas.
- ASMAL: mais de 40 utilizadores, a maioria com avaliações completas.
- Crinabel: 43 utilizadores com dados registados.

Estes testes foram fundamentais para identificar problemas reais de usabilidade, consistência de dados e estrutura do sistema. A nossa equipa, neste ano letivo, trabalhou principalmente na resolução dos problemas estruturais identificados durante essas sessões. Embora não tenhamos conduzido novos testes com parceiros, as melhorias que realizámos estão diretamente alinhadas com os requisitos emergentes desses testes anteriores.

6 Método e Planeamento

6.1 Planeamento inicial

O desenvolvimento do projeto MentHA segue uma abordagem Agile, escolhida pela sua capacidade de adaptação contínua e entrega incremental de valor ao longo do tempo. Esta metodologia é especialmente adequada para projetos complexos e dinâmicos como o MentHA, pois baseia-se em ciclos iterativos e incrementais, onde cada iteração resulta em entregáveis funcionais, promovendo feedback constante e ajustes frequentes.

A flexibilidade proporcionada pelo Agile permite incorporar novas necessidades ou alterações de requisitos sem comprometer a qualidade do produto final. Além disso, a divisão do trabalho em pequenos incrementos facilita a mitigação de riscos e uma gestão eficiente do tempo, visto que a equipa não está dedicada exclusivamente ao projeto. Este modelo de organização oferece uma visão clara e imediata das tarefas pendentes, permitindo que a equipa otimize o tempo disponível para o desenvolvimento.

Utilizamos o Trello como ferramenta principal de gestão de tarefas, estruturando o trabalho em épicos, features e user stories. Esta organização garante uma visualização clara do progresso, facilita a priorização de atividades e promove.

Na **Figura 20** está presente uma imagem do Trello do nosso projeto.

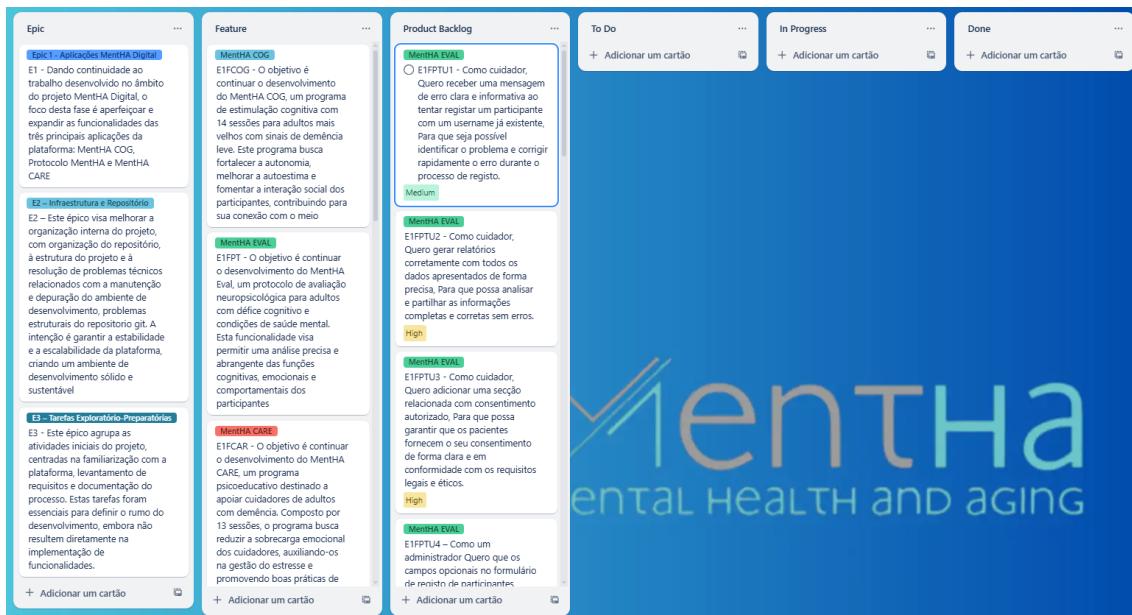


Figura 20 - Imagem do Trello: Product Backlog e o Quadro Kanban

No planeamento e organização das tarefas utilizámos a ferramenta **Trello**, adotando uma estrutura baseada em **épicos, features e user stories**, conforme detalhado na **Secção 3.1.3** deste

relatório. Essa organização trouxe maior clareza e rastreabilidade às atividades realizadas ao longo do projeto, permitindo a monitorização contínua do progresso.

A estrutura atual do Trello está segmentada em três grandes **épicos**, cada um com foco em diferentes dimensões do projeto:

- **E1 – Aplicações MentHA Digital:** dedicado ao desenvolvimento e melhoria das aplicações MentHA COG, MentHA Eval (Protocolo MentHA), MentHA CARE e do website MentHA Digital.
- **E2 – Infraestrutura e Repositório:** foca na organização e depuração do repositório e na estabilidade do ambiente de desenvolvimento.
- **E3 – Tarefas Exploratório-Preparatórias:** inclui tarefas de levantamento de requisitos, familiarização com a plataforma e elaboração do relatório.

Cada **épico** contém várias **features**, que por sua vez agrupam as **user stories** relacionadas a funcionalidades específicas ou tarefas de suporte ao desenvolvimento. As **features** preparatórias, como “Levantamento de Requisitos”, “Familiarização com a Plataforma” e “Relatório”, não resultam diretamente na implementação de componentes funcionais, mas foram fundamentais para garantir a integração da equipa e a organização do projeto.

Ao final de cada sprint, criamos uma coluna específica no Trello identificada com o número do sprint correspondente, onde são alocadas todas as user stories e tarefas concluídas. Este processo não apenas facilita a revisão e documentação das entregas, mas também proporciona uma visão clara e detalhada da evolução do projeto ao longo do tempo. Essa abordagem permite uma gestão mais eficiente, uma vez que a equipa consegue monitorizar facilmente o histórico das atividades realizadas, identificar padrões de produtividade e ajustar prioridades para os próximos sprints, assegurando um fluxo contínuo de melhorias e entregas.

Esta estrutura robusta de planeamento e organização contribuiu de forma decisiva para manter o projeto alinhado aos seus objetivos, mesmo diante de desafios e necessidades de adaptação ao longo do desenvolvimento.

Na **Figura 21** , **Figura 22** e na **Figura 23** está apresentada uma visão geral das colunas de Sprints do nosso Trello.

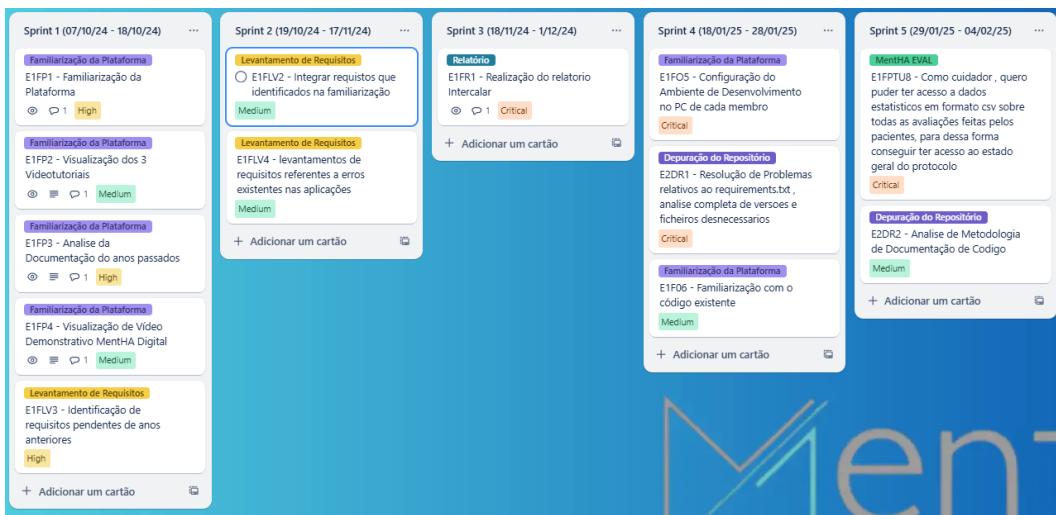


Figura 21 - Imagem Trello Sprints Realizados 1

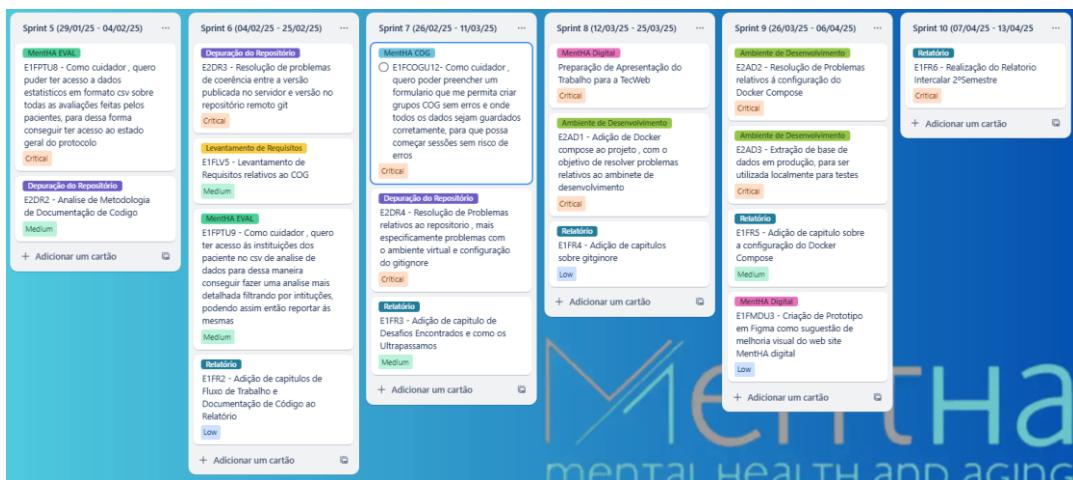


Figura 22 - Imagem Trello Sprints Realizados 2

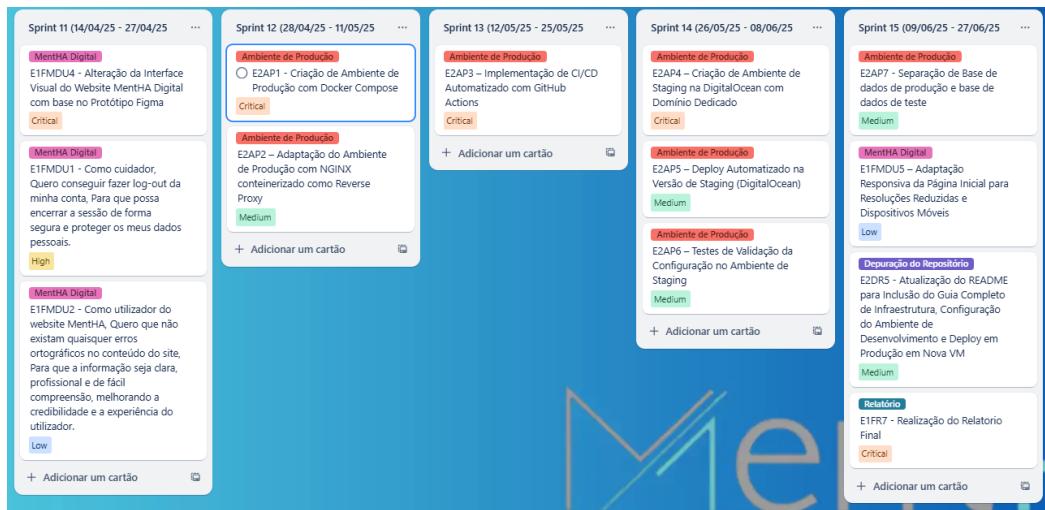


Figura 23 - Imagem Trello Sprints Realizados 3

6.1.1 Sprints Realizados

Sprint 1 - Familiarização da Plataforma

Data: (07/10/24 - 18/10/24)

Objetivo: Familiarização com a plataforma, entendendo a estrutura existente e documentando os requisitos pendentes. Este sprint visou criar uma base sólida para o trabalho subsequente.

Tarefas Realizadas:

- E1FP2 – Visualização dos 3 vídeo verstutoriais.
- E1FP3 – Análise da documentação de anos anteriores.
- E1FP4 – Visualização de vídeo demonstrativo MenthA Digital.
- E1FLV3 – Identificação de requisitos pendentes de anos anteriores.

Sprint 2 - Levantamento de Requisitos Inicial

Data: (19/10/24 - 17/11/24)

Objetivo: Integrar os requisitos identificados durante a familiarização e levantar os erros existentes nas aplicações. Este sprint concentrou-se na melhoria contínua e na estabilização do sistema.

Tarefas Realizadas:

- E1FLV2 – Integração dos requisitos identificados na familiarização.
- E1FLV4 – Levantamento de requisitos referentes a erros existentes nas aplicações.

Sprint 3 - Realização do Relatório

Data: (18/11/24 - 01/12/24)

Objetivo: Foco na elaboração do relatório intercalar, consolidando as descobertas e os avanços realizados nos sprints anteriores.

Tarefas Realizadas:

- E1FR – Elaboração do relatório intercalar.

Sprint 4 - Configuração de Ambiente Tecnológico

Data: (18/01/25 – 28/01/25)

Objetivo: Configurar o ambiente de desenvolvimento local para todos os membros da equipa, resolver problemas técnicos relacionados com dependências da aplicação e aprofundar o conhecimento sobre o código existente. Este sprint foi fundamental para garantir a capacidade de desenvolvimento autónomo e colaborativo.

Tarefas Realizadas:

- E1FO5 – Configuração do Ambiente de Desenvolvimento no PC de cada membro.

- E1DR1 – Resolução de Problemas relativos ao *requirements.txt*, análise completa de versões e ficheiros desnecessários.
- E1FO6 – Familiarização com o código existente.

Sprint 5 - Extração de Dados em CSV e Padronização da Documentação

Data: (29/01/25 - 04/02/25)

Objetivo: Início do desenvolvimento de novas funcionalidades relacionadas com a análise de dados no formato CSV, e definição de boas práticas para a documentação de código. Este sprint visou reforçar a clareza, organização e rastreabilidade das funcionalidades desenvolvidas.

Tarefas Realizadas:

- E1FPTU8 – Como cuidador, quero poder ter acesso a dados estatísticos em formato CSV sobre todas as avaliações feitas pelos pacientes, para dessa forma conseguir ter acesso ao estado geral do protocolo.
- E1DR2 – Análise de Metodologia de Documentação de Código.

Sprint 6 - Integração Técnica, Requisitos do MentHA COG e Detalhamento da Análise CSV

Data: (04/02/25 - 25/02/25)

Objetivo: Este sprint concentrou-se na resolução de problemas de integração entre o servidor e o repositório remoto, no levantamento de novos requisitos da aplicação MentHA COG e na continuação da funcionalidade de exportação de dados. Além disso, iniciou-se a documentação técnica no relatório.

Tarefas Realizadas:

- E1DR3 – Resolução de problemas de coerência entre a versão publicada no servidor e a versão no repositório remoto Git.
- E1FLV5 – Levantamento de Requisitos relativos ao COG.
- E1FPTU9 – Como cuidador, quero ter acesso às instituições dos pacientes no CSV de análise de dados para dessa maneira conseguir fazer uma análise mais detalhada filtrando por instituições, podendo assim reportar às mesmas.
- E1FR2 – Adição de capítulos de Fluxo de Trabalho e Documentação de Código ao Relatório.

Sprint 7 - Criação de novas funcionalidades no MentHA COG, Ambiente Virtual e Documentação de Desafios

Data: (26/02/25 - 11/03/25)

Objetivo: Foco na criação de novas funcionalidades no MentHA COG, garantindo a fiabilidade dos formulários de criação de grupos. Paralelamente, foram resolvidos problemas técnicos relacionados com o repositório e iniciada a documentação dos desafios enfrentados.

Tarefas Realizadas:

- E1FCOGU12 – Como cuidador, quero poder preencher um formulário que me permita criar grupos COG sem erros e onde todos os dados sejam guardados corretamente, para que possa começar sessões sem risco de erros.
- E1DR4 – Resolução de problemas relativos ao repositório, mais especificamente problemas com o ambiente virtual e configuração do .gitignore.
- E1FR3 – Adição de capítulo de Desafios Encontrados e Como os Ultrapassámos.

Sprint 8 - Apresentação TecWeb e Otimização do Ambiente com Docker Compose

Data: (12/03/25 - 25/03/25)

Objetivo: Preparação da apresentação do trabalho para a unidade curricular de Tecnologias Web e melhoria do ambiente de desenvolvimento através da introdução do Docker Compose.

Tarefas Realizadas:

- E1FMDU3 – Preparação de Apresentação do Trabalho para a TecWeb.
- E1DR5 – Adição de Docker Compose ao projeto, com o objetivo de resolver problemas relativos ao ambiente de desenvolvimento.
- E1FR4 – Adição de capítulos sobre .gitignore.

Sprint 9 - Testes Locais com Docker Compose e Início da Prototipagem Visual

Data: (26/03/25 - 06/04/25)

Objetivo: Consolidação da utilização do Docker Compose e preparação de um ambiente de testes local com dados reais. Também se iniciou a sugestão de melhorias visuais através de protótipos em Figma.

Tarefas Realizadas:

- E1DR6 – Resolução de problemas relativos à configuração do Docker Compose.
- E1DR7 – Extração de base de dados em produção, para ser utilizada localmente para testes.
- E1FR5 – Adição de capítulo sobre a configuração do Docker Compose.
- E1FMDU4 – Criação de protótipo em Figma como sugestão de melhoria visual do website MentHA Digital.

Sprint 10 - Relatório Intercalar e Consolidação dos Sprints do 2.º Semestre

Data: (07/04/25 - 13/04/25)

Objetivo: Elaboração do segundo relatório intercalar, consolidando o trabalho realizado durante os sprints anteriores no segundo semestre.

Tarefas Realizadas:

- E1FR6 – Realização do Relatório Intercalar 2.º Semestre.

Sprint 11 – Melhorias Visuais e Gestão de Sessões

Data: (14/04/25 – 28/04/25)

Objetivo: Realizar a atualização da interface visual do website MentHA Digital com base no protótipo em Figma, e implementar uma funcionalidade que permita aos cuidadores visualizar os constituintes de cada sessão ao criar um grupo, facilitando a gestão e acompanhamento dos participantes.

Tarefas Realizadas:

- E1FMDU4 - Alteração da Interface Visual do Website MentHA Digital com base no Protótipo Figma
- E1FMDU1 - Como cuidador, quero conseguir fazer log-out da minha conta, para que possa encerrar a sessão de forma segura e proteger os meus dados pessoais.
- E1FMDU2 - Como utilizador do website MentHA, quero que não existam quaisquer erros ortográficos no conteúdo do site, Para que a informação seja clara, profissional e de fácil compreensão, melhorando a credibilidade e a experiência do utilizador.

Sprint 12 – Criação da Infraestrutura Base de Produção

Objetivo: Implementar a infraestrutura inicial de produção da aplicação MentHA Digital, incluindo a orquestração de serviços essenciais através do Docker Compose e a configuração do NGINX como reverse proxy para servir a aplicação com maior segurança e desempenho.

Tarefas Realizadas:

- E2AP1 – Criação de Ambiente de Produção com Docker Compose, incluindo base de dados, aplicação Django e volumes persistentes.
- E2AP2 – Adaptação do Ambiente de Produção com NGINX conteinerizado como Reverse Proxy, assegurando uma gestão eficiente do tráfego e a aplicação de boas práticas de segurança.

Sprint 13 – Automatização do Processo de Deploy com CI/CD

Objetivo: Garantir a automação do ciclo de integração e deployment contínuo (CI/CD), permitindo que o código atualizado na branch principal seja automaticamente testado e implementado em produção, reduzindo erros humanos e acelerando entregas.

Tarefas Realizadas:

- E2AP3 – Implementação de CI/CD Automatizado com GitHub Actions, com testes e deploy contínuo para o ambiente de produção.

Sprint 14 – Criação e Validação do Ambiente de Staging

Objetivo: Implementar um ambiente de staging funcional e espelhado da produção para testar novas versões da aplicação com segurança, utilizando uma instância da DigitalOcean com domínio próprio e deploy automatizado.

Tarefas Realizadas:

- **E2AP4** – Criação de Ambiente de Staging na DigitalOcean com Domínio Dedicado.
- **E2AP5** – Configuração de Deploy Automatizado na Versão de Staging via GitHub Actions.
- **E2AP6** – Testes de Validação da Configuração no Ambiente de Staging, assegurando a sua prontidão para deploys reais.

Sprint 15 – Finalização Técnica e Documental do Projeto

Objetivo: Encerrar o desenvolvimento com tarefas de melhoria da interface, separação de ambientes críticos, documentação técnica e elaboração do relatório final do projeto, assegurando a completude e manutenibilidade futura da plataforma MentHA Digital.

Tarefas Realizadas:

- **E2AP7** – Separação de Base de Dados de Produção e de Testes, garantindo segurança e integridade dos dados.
- **E1FMDU5** – Adaptação Responsiva da Página Inicial para Resoluções Reduzidas e Dispositivos Móveis, otimizando a experiência do utilizador em ecrãs pequenos.
- **E2DR5** – Atualização do README com Guia Completo de Infraestrutura, Configuração do Ambiente de Desenvolvimento e Deploy em Produção numa Nova VM.
- **E1FR7** – Realização do Relatório Final, documentando o progresso do projeto e as decisões técnicas implementadas.

Em seguida na **Figura 24** apresentamos o Diagrama de Gantt que desenvolvemos, que descreve melhor a cronologia das entregas e a previsão dos próximos sprints do projeto.

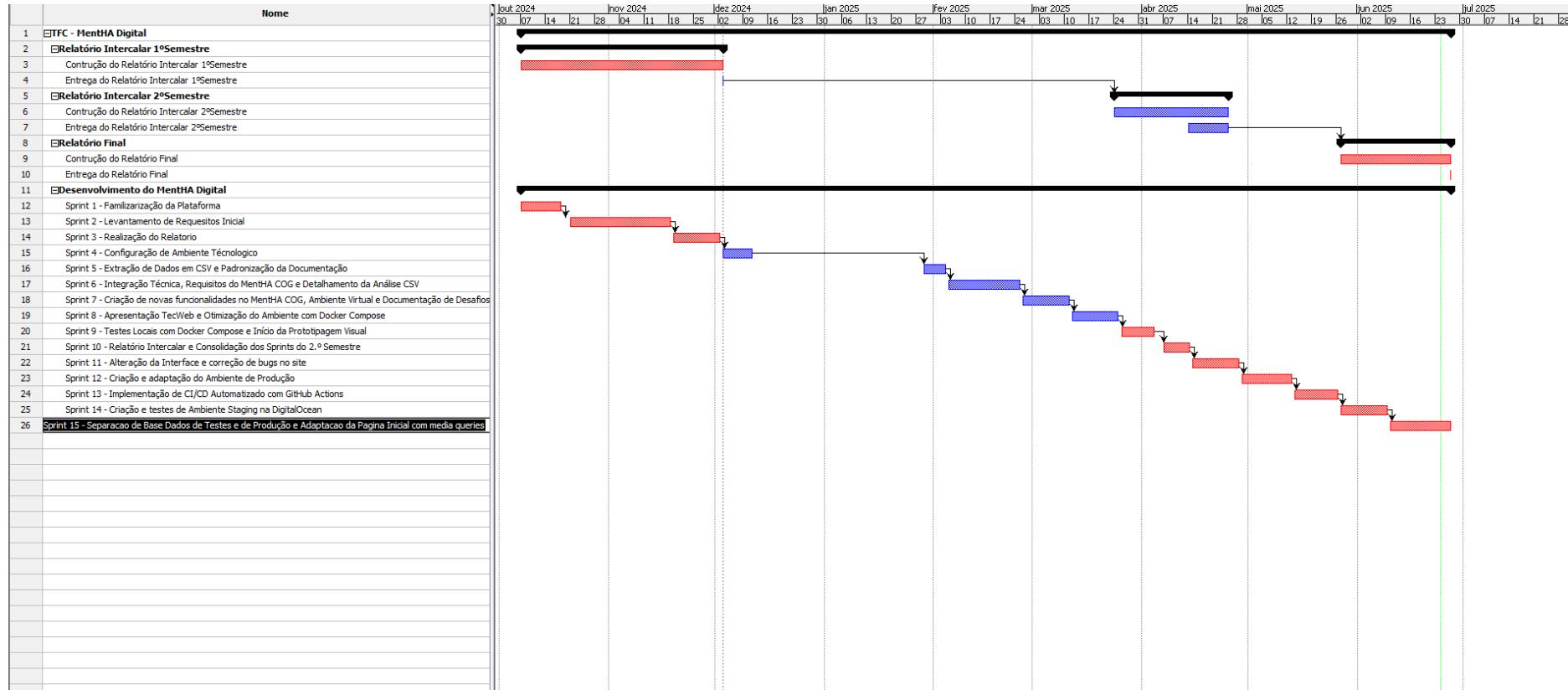


Figura 24 - Diagrama de Gantt Final

6.2 Análise Crítica ao Planeamento

O planeamento inicial do projeto foi definido com base numa visão clara e orientada para a melhoria contínua das aplicações do MentHA, em particular da aplicação MentHA COG. Com a adoção da metodologia Agile, previam-se ciclos curtos de desenvolvimento, iterativos e incrementais, com espaço para adaptações consoante o feedback e os desafios encontrados. No entanto, a execução prática revelou-se significativamente mais complexa do que o inicialmente previsto, sendo necessário proceder a vários ajustamentos estratégicos ao longo do projeto.

6.2.1 Planeamento Inicial e Previsão de Sprints

Tal como descrito na **Secção 1.2.1**, o objetivo inicial passava pela correção de erros críticos e pela melhoria contínua das aplicações MentHA, com especial incidência no MentHA COG, considerado o componente menos desenvolvido e funcional do sistema. Estavam identificados alguns problemas como falhas na atribuição de grupos, ausência de visualização de conteúdos e impossibilidade de logout, os quais comprometiam a usabilidade, segurança e eficácia das aplicações.

Posto isto, e segundo os princípios do Agile, foi traçado um plano com os seguintes sprints previstos:

- **Sprint 4:** Configuração do ambiente de desenvolvimento nos dispositivos da equipa, com garantia de comunicação entre o servidor e as máquinas locais.
- **Sprint 5:** Correção de erros técnicos nas aplicações com o objetivo de promover uma familiarização prática com a arquitetura existente.
- **Sprint 6:** Levantamento de novos requisitos funcionais, com apoio dos orientadores, e definição de prioridades para funcionalidades futuras, centrando-se no MentHA COG.

Os sprints também estão representados no Diagrama de Gantt que inicialmente tínhamos construído, no primeiro semestre, como consta na **Figura 25**.

6.2.2 Desvios ao Planeamento

Contudo, durante as primeiras semanas de trabalho, percebemos que a realidade era bem diferente do que pensávamos, como já explicámos também na **Secção 1.2.2**. O projeto estava muito desorganizado a nível técnico. As aplicações tinham sido feitas por várias equipas diferentes, em momentos diferentes, e sem uma estrutura comum. Isto causava vários problemas nomeadamente:

- A impossibilidade de executar corretamente as aplicações em ambiente local;
- Dependências obsoletas e configurações desatualizadas;
- Divergência entre o código em produção e o presente no repositório;
- Ausência de documentação clara sobre o processo de setup e deploy;
- Complexidade técnica que dificultava a introdução de novas funcionalidades.

Com tudo isto, percebemos que não podíamos seguir o plano inicial. Em vez de começar logo a desenvolver ou corrigir funcionalidades, tivemos de parar e focar-nos em organizar a base técnica do projeto. Sem isso, qualquer outra tarefa ia ser muito mais difícil ou até impossível de fazer.

6.2.3 Adaptação da Estratégia e Reposicionamento das Prioridades

A nossa adaptação a este novo cenário foi suportada pela metodologia Agile, cuja flexibilidade se revelou fundamental. Através de uma abordagem iterativa, conseguimos reavaliar e reajustar continuamente as prioridades, mantendo um fluxo de trabalho funcional apesar dos obstáculos inesperados.

A prioridade passou, assim, a ser a criação de uma base técnica sólida, garantindo que qualquer programador futuro pudesse:

- Clonar o repositório sem erros;
- Instalar e executar localmente a aplicação de forma eficiente;
- Trabalhar num ambiente coeso e reproduzível.

Estas tarefas, embora invisíveis ao utilizador final, foram essenciais para a viabilidade do projeto a médio e longo prazo. A documentação de processos, a atualização de dependências e a organização do código passaram a ser o centro do nosso plano de ação.

6.2.4 Fase de Retoma Funcional e Modernização Visual

Com a maioria dos problemas estruturais resolvidos, como referido na **Secção 1.2.3**, o projeto entrou numa fase mais estável, o que permitiu retomar o foco funcional, conforme inicialmente previsto. Uma das prioridades nesta fase foi a análise crítica da interface da aplicação web MentHA Digital, que revelou um design visual desatualizado e pouco apelativo.

Como resposta, foi incluído no plano de trabalho o objetivo de **modernização da interface**, tendo por base os protótipos desenvolvidos no Figma. Esta atualização visou não apenas a estética, mas também a **melhoria da usabilidade e acessibilidade** da aplicação. Paralelamente, foram corrigidos erros visuais e ortográficos e adicionadas funcionalidades de gestão de sessão, como o **log-out seguro** dos utilizadores.

Estas melhorias foram realizadas durante a **Sprint 11**, com foco específico no refinamento visual e na experiência do utilizador, sendo a primeira sprint da nova fase de avanços funcionais da plataforma.

6.2.5 Fase Final – Consolidação Técnica e Avaliação do Planeamento

A fase final do projeto representou a consolidação das principais melhorias estruturais, com foco na estabilidade, documentação e preparação da plataforma para o futuro. Esta etapa permitiu fechar o ciclo de desenvolvimento com uma infraestrutura de produção robusta, automatização do processo de deploy, criação de um ambiente de staging funcional e separação clara entre ambientes de teste e produção.

Embora o planeamento inicial tenha sido ajustado ao longo do tempo, a flexibilidade da abordagem Agile permitiu reavaliar prioridades e responder eficazmente aos desafios encontrados. O esforço investido na documentação técnica, nomeadamente no README, e na melhoria da interface web para dispositivos móveis, garantiu que a plataforma se tornasse não só mais funcional, mas também mais acessível e sustentável.

A elaboração do relatório final e a consolidação das boas práticas técnicas encerram o projeto com um saldo positivo, permitindo que equipas futuras possam dar continuidade ao trabalho de forma clara e segura. Esta fase demonstrou que a capacidade de adaptação é essencial num contexto real de desenvolvimento, e que um planeamento bem gerido deve sempre prever espaço para reformulações estratégicas.

6.2.6 Avaliação Geral e Aprendizagem

Apesar das alterações ao plano e da necessidade de reordenar prioridades ao longo do projeto, consideramos que a gestão foi eficaz e sempre alinhada com os princípios da metodologia **Agile**. A capacidade da equipa para se adaptar rapidamente aos obstáculos, reavaliar objetivos e reorganizar tarefas foi essencial para manter o projeto em progresso constante.

Desde o início, estruturámos o desenvolvimento com base num conjunto inicial de **sprints** definidos no primeiro semestre, como representado no diagrama de Gantt inicial **Figura 25**. No entanto, à medida que o trabalho avançou e surgiram diversas dificuldades técnicas, muitas delas resultantes da natureza herdada do projeto e da complexidade da infraestrutura existente, tornou-se necessário proceder a várias reestruturações significativas do planeamento. Estes reajustes refletem-se no diagrama de Gantt da **Figura 26** elaborado no segundo semestre, onde se observa claramente o impacto dos imprevistos enfrentados e as alterações de prioridade que fomos implementando.

Para além dos desafios técnicos encontrados, é importante salientar que a nossa disponibilidade para o projeto não foi em regime de full-time, devido a aulas, outros projetos e compromissos pessoais, o que exigiu uma gestão cuidada dos recursos de tempo e uma maior flexibilidade na alocação de tarefas. Esta realidade obrigou-nos a fazer adaptações frequentes aos sprints inicialmente previstos, reorganizando o backlog de tarefas em função dos problemas mais críticos e urgentes que surgiram em cada fase.

Apesar deste cenário dinâmico e muitas vezes imprevisível, consideramos que a experiência contribuiu de forma significativa para o nosso crescimento enquanto equipa, reforçando várias lições fundamentais:

- A **documentação técnica** e a organização clara do repositório são pilares essenciais para a continuidade de qualquer projeto;
- A construção de uma **base técnica sólida** é indispensável antes de avançar com desenvolvimentos funcionais relevantes;
- A **flexibilidade metodológica**, característica das metodologias **Agile**, é vital em ambientes reais, complexos e sujeitos a alterações constantes de contexto.

Mesmo perante um cenário inicial caótico, com problemas estruturais e técnicos profundos, conseguimos reorientar o trabalho de forma pragmática. Este sucesso foi possível, em grande

parte, graças à abordagem iterativa, incremental e adaptativa que aplicámos ao longo de todo o projeto.

Infelizmente, a única tarefa que não conseguimos concretizar foi a ligação da infraestrutura de produção ao servidor oficial da Universidade Lusófona. A VM atualmente utilizada pela instituição já possui um serviço de **NGINX** instalado diretamente e aloja diversos websites institucionais, o que impossibilita a utilização direta da nossa infraestrutura conteinerizada com **NGINX** sem comprometer os serviços já existentes. Por falta de tempo, não foi possível solicitar e configurar uma nova VM dedicada para suportar esta integração.

Contudo, deixámos esta tarefa preparada para futuras equipas, através de um guia completo e detalhado no ficheiro **README**, onde está descrito todo o processo de setup, deployment e gestão da infraestrutura. Com este contributo, acreditamos que facilitámos substancialmente a continuidade e manutenção da plataforma **MentHA Digital** a médio e longo prazo.

Em suma, apesar de o percurso não ter seguido exatamente o trajeto inicialmente idealizado, os objetivos centrais desta fase do projeto foram concretizados com sucesso. O resultado final reflete um trabalho técnico robusto, sustentado em decisões fundamentadas, planeamento adaptativo, forte aprendizagem prática e uma constante comunicação com os orientadores, aproximando a nossa experiência das dinâmicas reais de desenvolvimento de software em contexto profissional.

Esta experiência aproximou-nos assim de um cenário real de engenharia de software, onde a capacidade de adaptação, resiliência e aprendizagem contínua são tão ou mais determinantes do que o planeamento inicial.

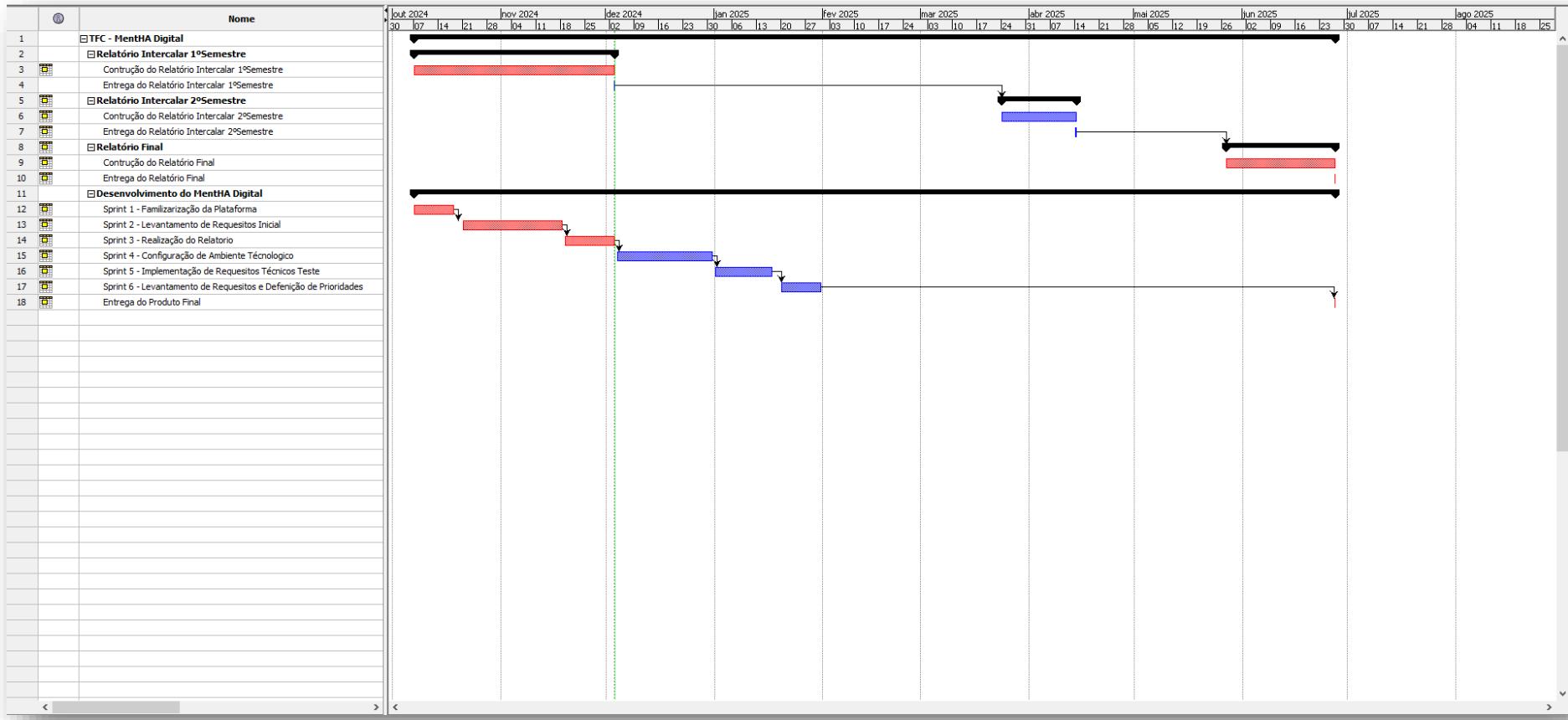


Figura 25 - Diagrama de Gantt Intercalar 1ºSemeste

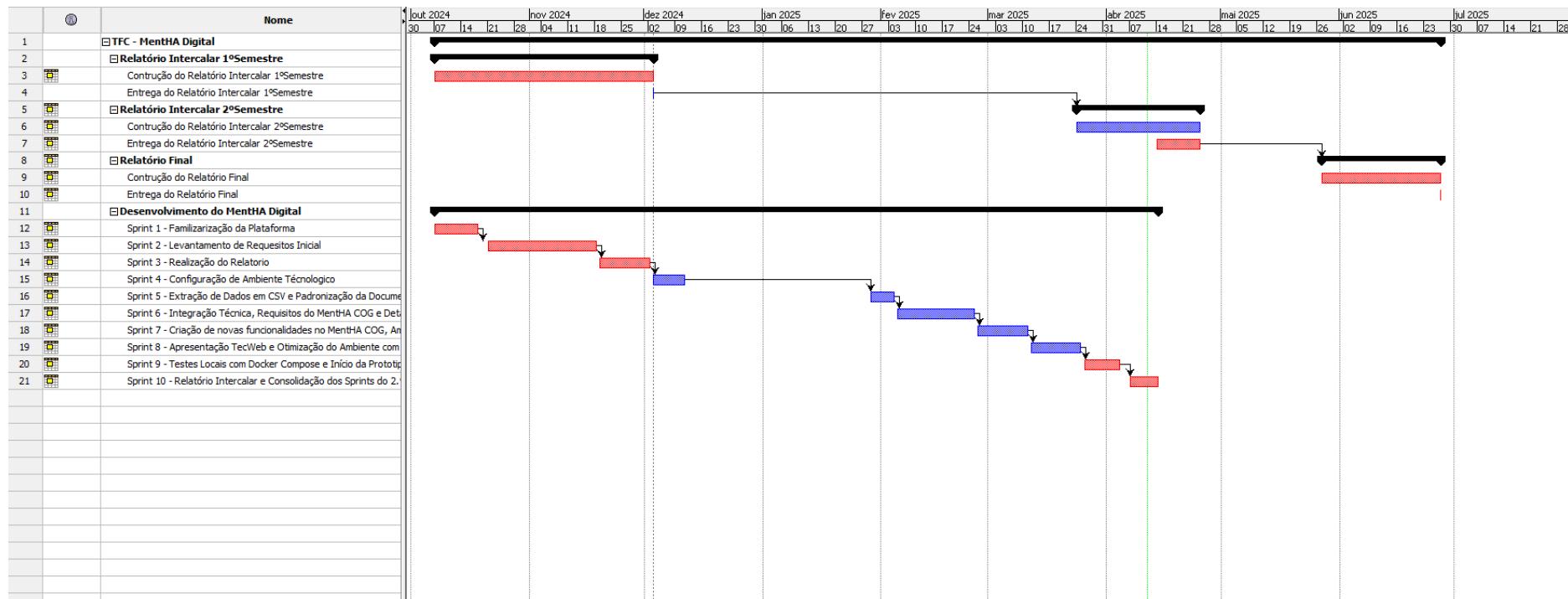


Figura 26 - Diagrama de Gantt Intercalar 2ºSemestre

7 Resultados

Após a execução de múltiplos testes funcionais, unitários, estruturais e visuais descritos na **Secção 5**, esta Secção apresenta os resultados obtidos, detalhando os outputs e efeitos práticos das intervenções realizadas. Estes resultados confirmam o impacto das correções efetuadas, bem como a eficácia das funcionalidades desenvolvidas, com evidências fotográficas e validações em diferentes ambientes.

Sempre que possível, procurou-se documentar os resultados de forma objetiva e transparente, com base em evidências visuais e análises de comportamento do sistema. Todos os testes descritos nesta secção têm como objetivo comprovar que os requisitos definidos previamente foram cumpridos ou devidamente justificados em caso de não cumprimento.

As imagens dos testes, os ficheiros de exportação gerados e outros elementos de suporte encontram-se incluídos em anexo e são também referenciados diretamente neste capítulo. Para maior detalhe sobre a metodologia e contexto de cada teste, consulta a **Secção Testes e Validação**.

7.1 Resultados dos Testes

7.1.1 Instalação e Execução do Ambiente de Desenvolvimento com Docker Compose

Este teste teve como principal objetivo validar a robustez e a portabilidade da nova estrutura do projeto, garantindo que a aplicação pode ser instalada e executada corretamente noutra máquina, sem qualquer configuração prévia por parte do utilizador. Esta validação é fundamental para assegurar que a infraestrutura desenvolvida é efetivamente replicável e pronta para ambientes diversos — uma condição essencial para a escalabilidade do sistema.

Tal como descrito na **Secção 5.1.1 - Teste de Instalação e Execução em Ambiente Local com Docker Compose**, o processo foi realizado numa máquina limpa, sem ambiente de desenvolvimento previamente configurado. Seguindo apenas as instruções contidas no ficheiro README.md, foi possível clonar o repositório, executar os comandos de inicialização com Docker Compose e aceder ao sistema via navegador, sem qualquer erro relacionado com dependências, serviços ou configuração.

O sucesso deste teste confirma que a organização e documentação do projeto foram eficazmente estruturadas, permitindo a replicação por parte de outras equipas técnicas ou instituições, com um mínimo de intervenção. Este resultado reforça a fiabilidade da solução como base para futuros desenvolvimentos ou transferências tecnológicas.

As **Figura 27**, **Figura 28** e **Figura 29** ilustram o ambiente onde o teste foi realizado, confirmando o funcionamento esperado da aplicação após a instalação numa máquina nova.

The screenshot shows a code editor interface with the following details:

- EXPLORER:** Shows files and folders for the 'MENTHA_DIGITAL' project, including 'settings.py', 'compose.yaml', and various Python files like '_init_.py', 'admin.py', 'apps.py', etc.
- CODE:** The 'compose.yaml' file is open, defining services for 'mentha_web' (using 'dbpostgres:12.9' as a dependency) and 'dbpostgresql_init'. It includes healthcheck logic and volumes.
- TERMINAL:** A terminal window displays logs from the 'django_web' container. The logs show Django starting at port 8000, handling multiple requests for URLs like '/aplicacoes/' and '/login/'. It also shows PostgreSQL activity and Django's internal system checks.
- STATUS BAR:** Shows the author 'martim-lusofona', the date '2 weeks ago', line 16, column 18, and encoding 'UTF-8'.

Figura 27 - Projeto a correr numa máquina nova

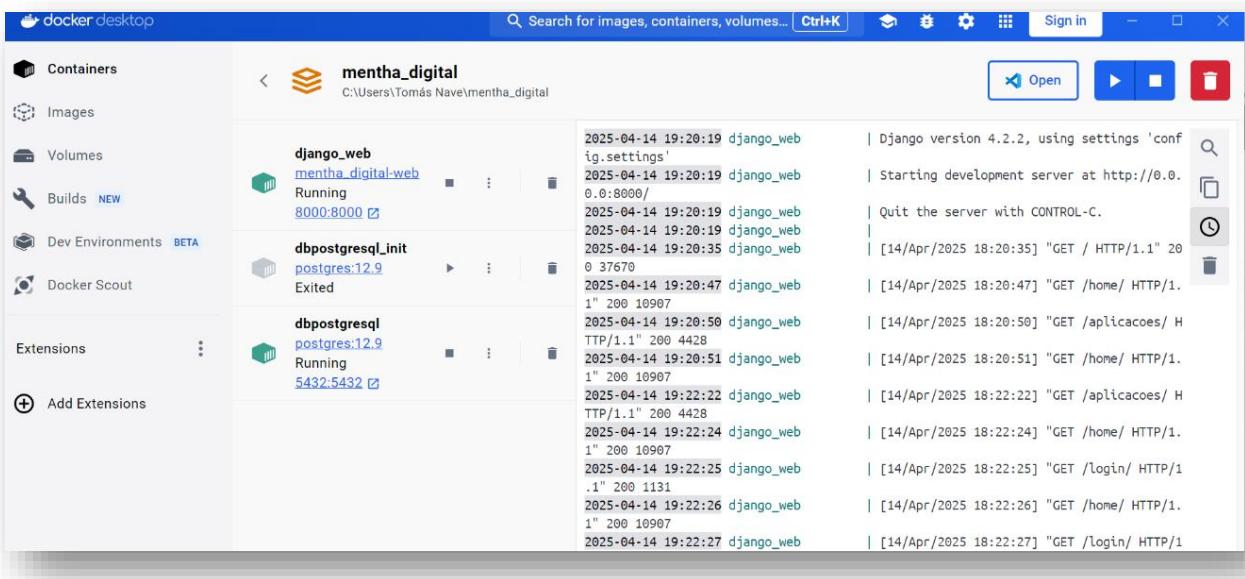


Figura 28 - Docker compose máquina nova



Figura 29 - Web site a correr máquina nova

7.1.2 Validação da Criação de Grupos – MentHA COG

Este teste teve como objetivo validar a correção de uma falha crítica previamente identificada na interface do módulo **MentHA COG**, mais especificamente na view de criação de grupos. Conforme descrito na **Secção 5.1.2 Teste de Validação de Alterações na Criação de Grupos – MentHA COG**, a aplicação permitia anteriormente a submissão de formulários com campos obrigatórios por preencher, o que resultava na criação de grupos inválidos e comprometia a integridade da base de dados.

Os resultados demonstraram que a nova lógica implementada impediu com sucesso a submissão de dados incompletos. O comportamento da interface está agora em conformidade com os requisitos funcionais definidos, garantindo maior fiabilidade na gestão de grupos. Esta melhoria contribui diretamente para a qualidade dos dados e para uma experiência de utilização mais fluida e coerente.

A **Figura 30** e **Figura 31** ilustram o estado da interface antes e depois da intervenção. A primeira evidencia a ausência de validações, permitindo a criação de grupos com dados incompletos, enquanto a segunda confirma a aplicação bem-sucedida da nova lógica condicional, alinhando o comportamento da interface com os requisitos definidos para o módulo **MentHA COG**.

Criar Novo Grupo

Nome: _____ Programa: _____

MentHA-COG

Diagnósticos disponíveis Localizações disponíveis Escolaridades disponíveis Referenciações disponíveis GDS disponíveis

Nome	Diagnóstico	Localização	Escolaridade	Referenciação	GDS
Manuel Felisberto Guerreiro	N/A	Almancil	1-4	ASMAL	1 <input type="checkbox"/>
Maria Odete Martins Madeira	N/A	Almancil	1-4	ASMAL	1 <input type="checkbox"/>
João Manuel Rosa Chapim	Esquizofrenia	Faro	5-10	ASMAL	1 <input type="checkbox"/>
Elisabete Teresa De Araújo Costa dos Santos Saldanha	N/A	N/A	1-4	ADEB	1 <input type="checkbox"/>
Graça Maria	N/A	N/A	1-4	ADEB	1 <input type="checkbox"/>
Otávio José Vargues da Silva	N/A	Faro	5-10	ASMAL	1 <input type="checkbox"/>
Ligia Martinez Romero Cereceda	N/A	Miraflor	1-4	Cruz Vermelha	1 <input type="checkbox"/>
José Luís Iria Rofino	Esquizofrenia	Faro	5-10	ASMAL	1 <input type="checkbox"/>
Tito Emanuel Aguas de Almeida	N/A	N/A	1-4	ADEB	1 <input type="checkbox"/>
Maria José de Jesus Gonçalves	N/A	Almancil	1-4	ASMAL	1 <input type="checkbox"/>
Mário Nicolau Pereira Caetano	N/A	Almancil	11+	ASMAL	1 <input type="checkbox"/>

Figura 30 - Criar Grupo COG Pré Alterações

Criar Novo Grupo

Nome: _____ Programa: _____

MentHA-COG

X Cancelar

Figura 31 - Criar Grupo COG Pós Alterações

7.1.3 Exportação de Avaliações – MentHA EVAL

Este teste teve como objetivo validar a nova funcionalidade de exportação de dados de avaliações no módulo **MentHA EVAL**, permitindo às instituições parceiras aceder a ficheiros CSV com a informação relevante sobre os seus pacientes. Tal como descrito na **Secção 5.1.3 Teste de Exportação de Avaliações – MentHA Eval**, a funcionalidade foi concebida para disponibilizar dados de forma clara, estruturada e pronta a ser analisada com ferramentas externas, nomeadamente em contextos de auditoria, monitorização clínica ou aplicação de técnicas de data science.

A funcionalidade foi validada com sucesso, tendo produzido ficheiros corretos e completos. Foram gerados dados relativos a pacientes com e sem avaliações, assegurando a consistência da exportação. Todos os campos relevantes — incluindo nome do paciente, associação, data e respostas — foram incluídos de forma organizada no ficheiro .csv, o que demonstra que a lógica de extração e formatação foi corretamente implementada.

Esta solução representa uma mais-valia concreta para a plataforma MentHA, uma vez que promove a transparência no tratamento dos dados e facilita a comunicação com os parceiros institucionais, que agora podem consultar e analisar a informação recolhida sem depender da interface gráfica.

As **Figura 32** e **Figura 33** mostram a nova view de exportação implementada na aplicação, enquanto as **Figura 34** e **Figura 35** apresentam os ficheiros CSV gerados com dados reais, demonstrando que os resultados obtidos correspondem exatamente ao comportamento esperado.

Nome	Próximo Agendamento
Abel Djassi Tavares Alvarenga	Sem agendamentos
Ada Barros	Sem agendamentos
Alexandra Cristina Curado Pereira	Sem agendamentos
Alexandra do Carmo Farinha Teste	Sem agendamentos
Alexandra Isabel Pacheco Vilhena	Sem agendamentos
Alexis Francisco Montes Cangala	Sem agendamentos
Amadeus Rafael Azevedo	Sem agendamentos
Amélia Rosa Rodrigues	Sem agendamentos
Ana Carina Simões Ferreira	Sem agendamentos
Ana Cristina Bandeira do Nascimento Alves Fróes	Sem agendamentos
Ana Isabel Dias	Sem agendamentos

Figura 32 - View Exportação CSV Parte 1

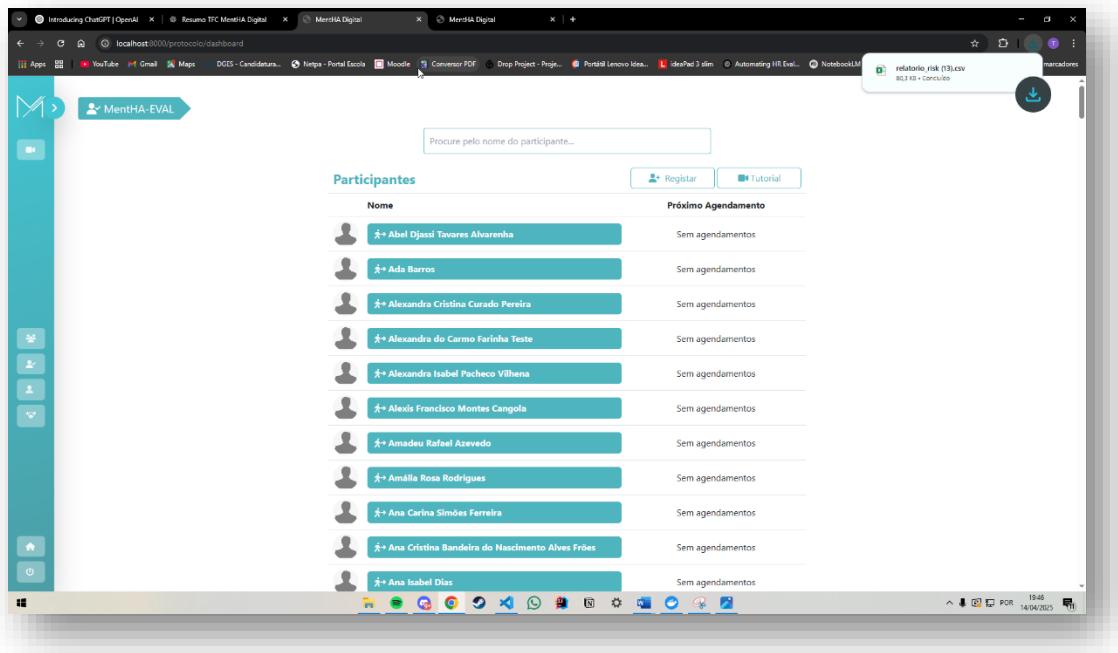


Figura 33 - View Exportação CSV Parte 2

◆Manuel Nicolau Freitas de Barros Teixeira Homem	GIRA	2024-05-28	10:24:00	Masculino	65	97	
◆Maria Manuela Santos António	ASMAL	2024-03-14	10:35:00	Feminino	56	76	
◆Maria Manuela Santos António	ASMAL	2023-07-14	15:56:00	Feminino	56	76	
◆Mário Rafael Rodrigues Silva	ASMAL	2024-03-14	16:07:00	Masculino	45	94	
◆Maria Deolinda Dias Soares	ASMAL	2024-03-14	12:40:00	Feminino	59	34	
◆Manuel Eduardo Lança Santos	ASMAL	2024-03-14	15:25:00	Masculino	54	78	
◆Olga Maria Marques Silva	GIRA	2024-05-28	11:44:00	Feminino	64	60	
◆Sara Rosa Lino Mendes	GIRA	2024-05-28	11:37:00	Feminino	48	92	
◆Paula Cristina Silva Palma	ASMAL	2024-03-14	16:09:00	Feminino	59	67	
◆Nelson de Andrade Silva	GIRA	2024-05-28	11:32:00	Masculino	45	78	
◆Sandrina Correia Candeias Pereira	ASMAL	2024-03-14	15:03:00	Feminino	58	89	
◆Miguel Alexandre de Meneses Lopes Cardoso	GIRA	2024-05-28	11:30:00	Masculino	53	6	
◆Pedro Manuel Rodrigues Barros Leal	GIRA	2024-05-28	11:55:00	Masculino	44	64	
◆Pedro Manuel Rodrigues Barros Leal	GIRA	2023-10-14	13:46:00	Masculino	44	65	
◆Sónia Isabel Rosa Do Nascimento	ASMAL	2024-03-13	16:43:00	Feminino	37	70	
◆Marcelino José Ribeiro Frederico	ASMAL	2024-03-13	12:47:00	Masculino	63	68	
◆José Maria Correia de Melo	ASMAL	2024-03-13	11:29:00	Masculino	71	75	
◆Rui Miguel Mendes Nunes	GIRA	2024-05-28	12:19:00	Masculino	51	0	
◆Carlos Manuel Marques de Agonia	GIRA	2024-05-28	10:42:00	Masculino	64	90	
◆João Carlos Ramos Gomes	GIRA	2024-05-28	11:15:00	Masculino	49	106	
◆Germana Maria de Jesus Flores	GIRA	2024-05-28	11:47:00	Feminino	67	83	
◆Abel Djassi Tavares Alvarenga	GIRA	risk não associado					
◆Abel Djassi Tavares Alvarenga	GIRA	2024-06-18	18:30:00	Masculino	39	75	

Figura 34 - CSV Protocolo MentHA 1

Figura 35 - CSV Protocolo MentHA 2

7.1.4 CI/CD com GitHub Actions

Este teste teve como objetivo validar o correto funcionamento da pipeline de **Continuous Integration / Continuous Deployment (CI/CD)** implementada com **GitHub Actions**, assegurando que qualquer alteração no repositório principal resulta num processo automatizado de build, testes e deploy para o servidor de staging. Tal como descrito na **Secção 5.2 – Teste de CI/CD e Deployment com GitHub Actions**, este mecanismo é fundamental para garantir que a aplicação permanece atualizada, estável e facilmente gerível, reduzindo a necessidade de intervenção manual nos ciclos de entrega.

O comportamento da aplicação em ambiente de produção refletiu imediatamente as alterações submetidas, validando a viabilidade do processo automatizado. Este resultado demonstra que a pipeline está corretamente configurada e funcional, sendo um pilar importante para a estabilidade contínua do sistema.

As figuras incluídas documentam todas as fases deste teste: a modificação local do código, a alteração visível no site em ambiente de desenvolvimento, a execução da pipeline no GitHub Actions e, por fim, a confirmação da alteração refletida no ambiente de staging.

7.1.5 Testes UI/UX

Com o objetivo de melhorar a experiência de utilização e assegurar a qualidade visual de todas as interfaces, foram realizados testes manuais de UI/UX nas principais aplicações do ecossistema MentHA: **MentHA COG**, **MentHA EVAL** e **MentHA CARE**, bem como no [website institucional](#). Estes testes centraram-se na análise de responsividade, legibilidade, coerência visual e facilidade de navegação, tendo sido essenciais para identificar pontos críticos e fundamentar intervenções futuras.

Tal como descrito na **Secção 5.4 – Testes da Interface do Utilizador (UI/UX)**, foram detetadas diversas falhas recorrentes nos módulos aplicacionais, nomeadamente:

- Elementos desalinhados ou sobrepostos;
 - Má adaptação a diferentes tamanhos de ecrã;

- Textos pouco legíveis em determinadas resoluções;
- Inconsistências no layout e comportamento interativo.

Apesar de terem sido aplicadas algumas correções pontuais ao longo do desenvolvimento, os testes confirmaram que ainda persistem falhas visuais e de usabilidade nas três aplicações mencionadas. Estes resultados evidenciam a necessidade de uma futura revisão visual e funcional mais aprofundada para garantir uma experiência de utilização mais coesa, especialmente considerando o público-alvo da plataforma.

Face aos problemas particularmente críticos identificados no **website MentHA**, foi tomada a decisão de avançar com a reestruturação completa da homepage e navegação principal. Esta intervenção teve como base um protótipo desenvolvido em Figma, que respeita boas práticas de design inclusivo, modernidade estética e clareza estrutural.

A nova interface foi implementada com sucesso, resultando numa versão significativamente mais funcional, legível e compatível com diferentes dispositivos. O feedback interno preliminar foi bastante positivo, evidenciando uma melhoria clara na percepção visual e na experiência geral de navegação no site.

As evidências visuais deste processo podem ser consultadas na **4.7.3 - Criação de Protótipo para Remodelação Visual do Website MentHA Digital**, onde são apresentadas imagens comparativas da homepage antes da intervenção e os protótipos desenvolvidos em Figma. Já na **Secção 4.7.4 - Desenvolvimento da Nova Home Page do Website MentHA Digital** é possível visualizar o resultado da implementação, com capturas da nova versão em funcionamento.

7.1.6 Entrada de Dados no Protocolo MentHA

Tal como descrito na **Secção 5.5 Teste de Entrada de Dados**, foram realizados testes específicos à estrutura e comportamento dos formulários de avaliação do Protocolo MentHA, uma das áreas mais críticas da aplicação. Estes testes permitiram identificar várias fragilidades na forma como os dados são inseridos e tratados, com impacto direto na fiabilidade da informação recolhida.

Os resultados demonstraram que os formulários atuais ainda apresentam problemas estruturais relevantes. Foram detetados campos com valores de exemplo que podem induzir em erro os utilizadores, ausência de validações obrigatórias em campos críticos e falta de normalização nos dados recolhidos — nomeadamente em unidades de medida, formatos e nomenclaturas. Estas falhas não só comprometem a qualidade dos dados, como afetam funcionalidades associadas, como a geração de relatórios, análises estatísticas e exportações.

Face a estas constatações, foi elaborado um plano de melhoria que inclui:

- A redefinição das obrigatoriedades por campo;
- A eliminação de valores de exemplo nos inputs;
- A uniformização dos formatos de dados utilizados.

A entrada de dados foi assim identificada como uma prioridade estrutural para o futuro do projeto, sendo essencial para assegurar a escalabilidade e robustez do sistema em contexto real de utilização.

7.2 Cumprimento de requisitos

Nesta secção, apresentamos uma análise detalhada do grau de cumprimento de todos os requisitos definidos na **Secção 3.1.1** ao longo do desenvolvimento do projeto. Cada requisito é acompanhado de uma indicação do seu estado atual: **Concluído** ou **pendente**. Nos casos em que o requisito não foi totalmente cumprido, incluímos uma justificação no final da Secção clara para esse resultado.

Para facilitar a leitura e organização, os requisitos foram agrupados em tabelas distintas, consoante a área funcional ou aplicação a que dizem respeito. Assim, temos:

- **Tabela 9 - Lista de Requisitos Referentes ao MentHA Digital**
- **Tabela 10 - Lista de Requisitos Referentes ao Protocolo MentHA**
- **Tabela 11 - Lista de Requisitos Referentes ao MentHA COG**
- **Tabela 12 - Lista de Requisitos Referentes à Depuração do Repositório**
- **Tabela 13 - Lista de Requisitos Referentes ao Ambiente de Desenvolvimento**
- **Tabela 14 - Lista de Requisitos Referentes ao Ambiente de Produção**

Estas tabelas permitem ter uma visão clara e segmentada do estado de evolução de cada parte do projeto, facilitando tanto a avaliação do trabalho realizado como a identificação de pontos de melhoria futura.

Tabela 9 - Lista de Requisitos Referentes ao MentHA Digital

ID do Requisito	Descrição do Requisito	Importância/Esfórcio	Estado do Requisito
E1FMDU1 – Erro a fazer log-out	Na aplicação do MentHA, quando se inicia sessão não se consegue fazer Log- out.	Este requisito é de importância "Must have" e requer o trabalho de 1 pessoa e a duração prevista de 3 horas.	Concluído
E1FMDU2 – Corrigir erros ortográficos na web site MentHA	No website MentHA, deve ser garantido que não existam erros ortográficos em qualquer conteúdo exibido nas páginas do site.	Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 4 horas	Concluído

Tabela 9 - Lista de Requisitos Referentes ao MentHA Digital

E1FMDU3 - Criação de Protótipo em Figma como sugestão de melhoria visual do web site MentHA digital	A equipa deve desenvolver um protótipo de interface no Figma, com o objetivo de propor melhorias visuais e de usabilidade para o website MentHA Digital, respeitando a identidade visual do projeto e as necessidades dos utilizadores.	Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.	Concluído
E1FMDU4 – Alteração da Interface Visual do Website MentHA Digital com base no Protótipo	A equipa deve implementar alterações na interface visual do website MentHA Digital, com base no protótipo desenvolvido no Figma.	Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.	Concluído
E1FMDU5 – Adaptação Responsiva da Página Inicial para Resoluções Reduzidas e Dispositivos Móveis	A equipa deve adaptar a página inicial do website MentHA Digital para garantir uma apresentação visual adequada em dispositivos com resoluções reduzidas, como portáteis com ecrãs pequenos, tablets e telemóveis	Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 5 horas.	Concluído

Tabela 10 - Lista de Requisitos Referentes ao Protocolo MentHA

ID do Requisito	Descrição do Requisito	Importância/Esfoco	Estado do Requisito

Tabela 10 - Lista de Requisitos Referentes ao Protocolo MentHA

E1FPTU1 – Registar com o mesmo nome de utilizador deve dar um erro com alguma informação	Na aplicação do MentHA EVAL, ao registar um Utilizador, se registamos um Participante com o mesmo Username de outro Participante, este dá apenas um prompt JavaScript.	Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração prevista de 3 horas.	Pendente
E1FPTU2 – Relatório Gerado corretamente no MentHA EVAL	Na aplicação do MentHA EVAL, o relatório deve ser gerado corretamente com os dados todos corretos.	Este requisito é de importância “Must have” e requer o trabalho de 1 pessoa e a duração prevista de 3 horas.	Parcial
E1FPTU3 – No Protocolo MentHA adicionar uma parte relacionada a um consentimento autorizado	No Protocolo MentHA deve-se realizar uma parte relacionada a um consentimento autorizado para os pacientes.	Este requisito é de importância “Should Have” e exige o trabalho de 2 pessoas e a duração prevista de 4 horas.	Pendente
E1FPTU4 – Os campos opcionais no formulário de registo de participantes devem estar vazios por padrão e usar labels	No Protocolo MentHA, os campos opcionais não devem ter valores predefinidos, devendo ser apresentados como campos vazios e devidamente identificados com labels.	Este requisito é de importância “Must Have” e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.	Pendente
E1FPTU5 – No Protocolo MentHA, adicionar um aviso visual na avaliação MentHA Risk de dados em falta	No Protocolo MentHA, deve ser exibido um aviso visual claro na lista de avaliações MentHA Risk indicando que o teste foi submetido com campos por preencher e está incompleto.	Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.	Pendente

Tabela 10 - Lista de Requisitos Referentes ao Protocolo MentHA

E1FPTU6 – Corrigir erro ao gerar relatório de avaliação no Protocolo MentHA	No Protocolo MentHA, ao tentar gerar o relatório de uma avaliação, ocorre um erro que resulta na exibição de uma página não disponível.	Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 6 horas	Pendente
E1FPTU7 – No Protocolo MentHA, corrigir a barra de progresso do questionário sociodemográfico no MentHA Risk	No Protocolo MentHA, deve ser corrigida a barra de progresso do questionário sociodemográfico no MentHA Risk para garantir que ela funcione corretamente e acompanhe o progresso do utilizador durante a avaliação	Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.	Pendente
E1FPTU8 – Exportação de Dados Estatísticos em CSV	O sistema deve permitir aos utilizadores com perfil de cuidador aceder e exportar, em formato CSV, dados estatísticos referentes a todas as avaliações realizadas pelos pacientes, no âmbito do protocolo MentHA	Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 2 horas	Concluído
E1FPTU9 – Inclusão da Informação da Instituição no CSV de Análise de Dados	O sistema deve incluir, no ficheiro CSV de análise de dados exportado pelos cuidadores, a informação relativa à instituição a que cada paciente pertence, de forma a permitir uma análise segmentada por instituição	Este requisito é de importância "Should Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas	Pendente

Tabela 11 - Lista de Requisitos Referentes ao MentHA COG

ID do Requisito	Descrição do Requisito	Importância/Esfórcio	Estado do Requisito
E1FCOGU1 - Visualização do Grupo COG	No MentHA COG, ao criar um Grupo não se consegue visualizar os Constituintes de cada sessão	Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e duração prevista de 3 horas.	Pendente
E1FCOGU2 – MentHA COG, Materiais da sessão não são mostrados com sucesso	No MentHA COG, os materiais da sessão não aparecem de todo, mas já se consegue mostrar os materiais dentro de outra página	Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração de 8 horas	Pendente
E1FCOGU3 – MentHA COG, barra de progresso não funciona	Na aplicação MentHA COG, a barra de progresso no sociodemográfico não funciona corretamente.	Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração de 8 horas.	Pendente
E1FCOGU4 – Caderno de cotação MentHA COG não finalizado	Na aplicação, O caderno de estímulos deve ser implementado (não está na aplicação)	Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração prevista de 10 horas	Pendente
E1FCOGU5 – Criar avaliações no MentHA COG	Na aplicação criar uma opção que permitirá o Orientador das sessões criar e submeter avaliações do MentHA COG	Este requisito é de importância “Should have” e requer o trabalho de 1 pessoa e a duração prevista de 8 horas	Pendente

Tabela 11 - Lista de Requisitos Referentes ao MentHA COG

E1FCOGU6 – Feedback ao guardar no Diário de Bordo	Na aplicação, o Dinamizador deve receber algum tipo de feedback para saber se as alterações foram realizadas com sucesso ou não	Este requisito é de importância "Should have" e requer o trabalho de 1 pessoa e a duração prevista de 9 horas	Pendente
E1FCOGU7 – Submeter avaliações do MentHA COG com sucesso	As avaliações do COG têm de ser submetidas com sucesso para estar totalmente funcional	Este requisito é de importância "Should Have" e exige o trabalho de 2 pessoas e a duração prevista de 10 horas	Pendente
E1FCOGU8 – No Protocolo MentHA COG, corrigir o enquadramento de imagens e botões nos exercícios das Sessões	No Protocolo MentHA COG, deve ser corrigido o enquadramento das imagens e botões no quadro de exercícios das Sessões, garantindo que todos os elementos visuais e interativos estejam devidamente alinhados e funcionando corretamente	Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoa, com a duração prevista de 10 horas	Pendente
E1FCOGU9 – No Protocolo MentHA COG, corrigir erro de código ao atribuir grupo a participantes sem grupo	No Protocolo MentHA COG, ao clicar no botão de "Participantes sem grupo" e tentar atribuir um grupo a um participante, o sistema deve funcionar sem gerar erro de código.	Este requisito é de importância "Should Have" e exige o trabalho de 2 pessoas, com a duração prevista de 4 horas	Pendente
E1FCOGU10 – Corrigir erro de Pesquisar pessoa pelo nome. Erro na barra de pesquisa	No Protocolo MentHA COG, deve ser corrigida a funcionalidade de pesquisa de participantes, de forma que, ao pesquisar pelo nome de uma pessoa, os resultados sejam corretamente exibidos	Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas	Pendente

Tabela 11 - Lista de Requisitos Referentes ao MentHA COG

E1FCOGU11 – No Protocolo MentHA COG, corrigir o quadro de exercícios da Sessão 4	No Protocolo MentHA COG, deve ser corrigido o quadro de exercícios da Sessão 4 garantindo que o exercício esteja correto.	Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 2 horas	Pendente
E1FCOGU12 - Reformulação de View de Criação de Grupo	Disponibilização de um formulário intuitivo e funcional que permita aos utilizadores com perfil de cuidador criar grupos COG.	Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas e a duração prevista de 10 horas	Concluído

Tabela 12 - Lista de Requisitos Referentes à Depuração do Repositório

ID do Requisito	Descrição do Requisito	Importância/Esfórcio	Estado do Requisito
E1DR1 - Resolução de Problemas relativos ao requirements.txt, análise completa de versões e ficheiros desnecessários	A equipa deve realizar uma análise completa ao ficheiro requirements.txt, com o objetivo de identificar e resolver problemas relacionados com versões incompatíveis, dependências obsoletas e ficheiros desnecessários no projeto, garantindo a estabilidade e portabilidade do	Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas e a duração prevista de 3 horas	Concluído

Tabela 12 - Lista de Requisitos Referentes à Depuração do Repositório

	ambiente de desenvolvimento		
E1DR2 - Analise de Metodologia de Documentação de Código	A equipa deve analisar e definir uma metodologia eficaz para a documentação do código e dos processos do projeto, com base numa pesquisa sobre abordagens tradicionais e ferramentas modernas,	Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 5 horas.	Concluído
E1DR3 - Resolução de problemas de coerência entre a versão publicada no servidor e versão no repositório remoto git	A equipa deve identificar e resolver quaisquer problemas de incoerência entre a versão do código publicada no servidor de produção e a versão armazenada no repositório remoto Git, garantindo que ambas as versões estejam alinhadas e atualizadas	Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 5 horas	Concluído

Tabela 12 - Lista de Requisitos Referentes à Depuração do Repositório

E1DR4 - Resolução de Problemas relativos ao repositório, mais especificamente problemas com o ambiente virtual e configuração do gitignore	A equipa deve identificar e resolver problemas relacionados com o repositório, especificamente em relação à configuração do ambiente virtual e ao ficheiro .gitignore, garantindo que dependências desnecessárias não sejam incluídas no	Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 6 horas.	Concluído
E2DR5 – Atualização do README para Inclusão do Guia Completo de Infraestrutura, Configuração do Ambiente de Desenvolvimento e Deploy em Produção em Nova VM	A equipa deve atualizar o ficheiro README do projeto para incluir documentação detalhada sobre a infraestrutura geral da aplicação MentHA Digital.	Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 4 horas.	Concluído

Tabela 13 - Lista de Requisitos Referentes ao Ambiente de Desenvolvimento

ID do Requisito	Descrição do Requisito	Importância/Esfórm	Estado do Requisito
E2AD1 - Adição de Docker compose ao projeto, com o objetivo de resolver problemas relativos ao	A equipa deve integrar o Docker Compose no projeto, com o objetivo de resolver problemas relacionados com a configuração e consistência do ambiente de	Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 3 horas	Concluído

Tabela 13 - Lista de Requisitos Referentes ao Ambiente de Desenvolvimento

ambiente de desenvolvimento	desenvolvimento, garantindo que o ambiente seja facilmente replicável e isolado, independentemente do sistema operativo utilizado pelos membros da equipa		
E2AD2 - Resolução de Problemas relativos á configuração do Docker Compose	A equipa deve identificar e resolver problemas relacionados com a configuração do Docker Compose, garantindo que todos os serviços definidos sejam corretamente configurados, funcionando de forma integrada e sem conflitos no ambiente de desenvolvimento.	Este requisito é de importância "Must Have" e exige o trabalho de 2 pessoas, com a duração prevista de 4 horas	Concluído
E2AD3 - Extração de base de dados em produção, para ser utilizada localmente para testes	A equipa deve desenvolver um processo seguro e eficiente para extrair a base de dados em produção e permitir que seja utilizada localmente,	Este requisito é de importância "Must Have" e exige o trabalho de 1 pessoa, com a duração prevista de 2 horas	Concluído

Tabela 14 - Lista de Requisitos Referentes ao Ambiente de Produção

ID do Requisito	Descrição do Requisito	Importância/Esfoco	Estado do Requisito

Tabela 14 - Lista de Requisitos Referentes ao Ambiente de Produção

E2AP1 – Criação de Ambiente de Produção com Docker Compose	A equipa deve configurar um ambiente de produção para o projeto MentHA Digital utilizando Docker Compose, garantindo isolamento, portabilidade e consistência entre ambientes. Esta configuração deve incluir todos os serviços essenciais (base de dados, aplicação Django, volumes persistentes, etc.), preparados para execução em contexto real.	Este requisito é de importância “Must Have” e exige o trabalho de 2 pessoas, com a duração prevista de 15 horas	Concluído
E2AP2 – Adaptação do Ambiente de Produção com NGINX conteinerizado como Reverse Proxy	A equipa deve adaptar o ambiente de produção para utilizar o NGINX em container como reverse proxy. Esta configuração permite gerir o tráfego HTTP, servir ficheiros estáticos e aplicar cabeçalhos de segurança, melhorando a performance e a robustez da infraestrutura.	Este requisito é de importância “Must Have” e exige o trabalho de 2 pessoas, com a duração prevista de 4 horas.	Concluído
E2AP3 – Implementação de CI/CD Automatizado com GitHub Actions	A equipa deve desenvolver um pipeline automatizado de integração e deployment contínuo (CI/CD) com GitHub Actions, garantindo que qualquer	Este requisito é de importância “Must Have” e exige o trabalho de 2 pessoas, com a duração prevista de 15 horas.	Concluído

Tabela 14 - Lista de Requisitos Referentes ao Ambiente de Produção

	alteração na branch principal (<code>main</code>) desencadeie automaticamente testes e deploy para o ambiente de produção.		
E2AP4 – Criação de Ambiente de Staging na DigitalOcean com Domínio Dedicado	A equipa deve configurar uma instância de staging na plataforma DigitalOcean, simulando o ambiente de produção. Este ambiente deve utilizar o mesmo domínio e configuração do futuro servidor oficial, permitindo testes e validações antes do deploy definitivo.	Este requisito é de importância “Should Have” e exige o trabalho de 2 pessoas, com a duração prevista de 5 horas.	Concluído
E2AP5 – Deploy Automatizado na Versão de Staging (DigitalOcean)	A equipa deve configurar o processo de deploy automatizado para a instância de staging, garantindo que o código seja copiado, reconstruído e iniciado de forma automática via GitHub Actions, simulando o funcionamento da produção real.	Este requisito é de importância “Should Have” e exige o trabalho de 2 pessoas, com a duração prevista de 10 horas.	Concluído

Tabela 14 - Lista de Requisitos Referentes ao Ambiente de Produção

E2AP6 – Testes de Validação da Configuração no Ambiente de Staging	A equipa deve realizar testes funcionais e técnicos no ambiente de staging para validar a infraestrutura, comportamento da aplicação, configuração de rede e disponibilidade do domínio, assegurando que o ambiente está pronto para deploy final.	Este requisito é de importância “Must Have” e exige o trabalho de 2 pessoas, com a duração prevista de 3 horas	Concluído
E2AP7 – Separação de Base de Dados de Produção e Base de Dados de Teste	A equipa deve garantir a separação clara entre a base de dados de produção e a de testes, assegurando que os dados reais nunca sejam utilizados ou modificados em ambientes de desenvolvimento ou staging.	Este requisito é de importância “Must Have” e exige o trabalho de 1 pessoa, com a duração prevista de 2 horas.	Concluído

7.2.1 Considerações Finais da Secção

Importa referir que os requisitos que permanecem **pendentes** estão, na sua maioria, associados a componentes e funcionalidades das aplicações que ainda requerem correções técnicas ou reestruturação. A definição destes requisitos teve como principal objetivo levantar o maior número possível de problemas e áreas de melhoria, não sendo considerados **critérios finais de conclusão** do projeto.

Por essa razão, o não cumprimento integral de alguns requisitos reflete essencialmente **limitações de tempo e priorização de tarefas essenciais** ao funcionamento geral da solução. Ainda assim, este conjunto de requisitos pendentes constitui uma **base sólida e bem documentada** para equipas futuras, oferecendo uma orientação clara sobre os aspetos que necessitam de correção ou otimização nos desenvolvimentos seguintes do projeto.

8 Conclusão

Neste capítulo final, apresentamos uma análise abrangente do trabalho desenvolvido ao longo do projeto MenthA Digital. Começamos por avaliar o grau de concretização do plano inicial e as principais diferenças entre a solução planeada e a solução realmente implementada. Em seguida, refletimos sobre as maiores dificuldades enfrentadas durante o desenvolvimento e sobre a forma como evoluímos técnica e pessoalmente ao longo do processo. Por fim, propomos um conjunto de sugestões para trabalhos futuros, identificando áreas prioritárias de melhoria e oportunidades de valorização da plataforma, com o objetivo de assegurar a continuidade sustentada do projeto e aumentar o seu impacto a médio e longo prazo.

8.1 Conclusão

8.1.1 Grau de concretização do plano

Em termos de grau de concretização do plano inicialmente delineado, consideramos que os resultados obtidos foram bastante positivos. A principal razão para esse sucesso deveu-se à adoção de uma metodologia ágil de desenvolvimento, que se revelou fundamental para lidar com as constantes alterações de foco e objetivos que o projeto enfrentou ao longo do tempo — conforme explicado na **Secção 1.2** deste relatório.

A estrutura iterativa e flexível do método ágil permitiu-nos adaptar rapidamente a nossa organização, prioridades e fluxos de trabalho. Essa adaptabilidade foi essencial para garantir a conclusão das funcionalidades consideradas críticas em cada fase do projeto. Conforme detalhado na **Secção 7.2.1**, desde o início assumimos como objetivo principal a identificação e resolução dos principais problemas da aplicação, mesmo que tal não implicasse uma cobertura total de todos os requisitos inicialmente levantados.

Consequentemente, o não cumprimento de alguns requisitos não representa falhas de execução, mas sim decisões estratégicas de priorização, baseadas na relevância, complexidade e impacto das tarefas em aberto.

Importa ainda destacar uma limitação concreta relacionada com a infraestrutura: não foi possível realizar a migração da aplicação para o servidor institucional da Universidade Lusófona. Esta decisão foi condicionada por fatores já descritos na **Secção 4.4.2 - Ambiente de Produção**.

Ainda assim, esta limitação não comprometeu os objetivos essenciais do projeto. O MVP encontra-se atualmente em funcionamento num servidor **DigitalOcean**, com uma infraestrutura moderna, escalável e devidamente automatizada. Esta versão demonstra na prática a viabilidade da arquitetura proposta e permite validar a aplicação em ambiente real de produção. Adicionalmente, desenvolvemos e deixámos disponível um guia técnico detalhado, incluído no repositório do projeto, que permite a migração futura para o ambiente institucional com relativa facilidade.

Em suma, a maioria dos objetivos delineados foi atingida com sucesso. A estrutura modular e documentada da solução garante que, mesmo nos aspectos não totalmente concretizados, o

trabalho realizado representa um avanço significativo e oferece continuidade sustentável para futuras fases de desenvolvimento.

8.1.2 Diferenças entre a Solução Proposta Inicialmente e a Solução Desenvolvida

Neste subcapítulo fazemos um apanhado geral das principais diferenças entre a solução inicialmente prevista e a solução que efetivamente foi desenvolvida. Estas diferenças foram identificadas ao longo do processo de adaptação e execução do projeto, e estão descritas em maior detalhe na **Secção 4**, com especial destaque para a **Secção 4.7 - Componentes**, onde cada uma das componentes é analisada em profundidade.

Consideramos que a nossa contribuição para o projeto MentHA Digital foi bastante positiva para a sua continuidade, mesmo que o número de erros resolvidos tenha sido inferior ao inicialmente previsto. O foco foi ajustado para garantir intervenções de elevada qualidade em diferentes áreas da aplicação, tendo sempre em conta o impacto a médio e longo prazo.

No **frontend**, a principal melhoria consistiu na modernização da homepage do website institucional. Esta reformulação oferece agora uma base visual sólida e um layout consistente que pode ser estendido a outras páginas, com o objetivo de tornar a plataforma mais atrativa para utilizadores, parceiros e potenciais entidades colaboradoras.

Na **camada backend**, embora a intervenção tenha sido mais limitada, conseguimos realizar alterações importantes. No módulo MentHA COG, foi ajustada uma das funções centrais da aplicação, e no Protocolo MentHA foi integrada uma nova funcionalidade que abre portas para futuras análises baseadas em dados. Esta evolução permitirá, no futuro, implementar dashboards estatísticos e aplicar técnicas de Data Science com maior facilidade.

Por fim, a área onde a nossa intervenção mais se destacou foi a da **infraestrutura**. Com o objetivo de mitigar os problemas recorrentes associados ao desenvolvimento e à publicação da aplicação, desenhámos uma infraestrutura completa para ambientes de desenvolvimento e produção, baseada em **Docker** e **Docker Compose**. Esta solução personalizada foi construída com princípios de **DevOps**, garantindo robustez, consistência e facilidade de replicação. Acreditamos que esta foi a nossa contribuição mais relevante e estruturante, pois cria as bases para que futuras equipas possam focar-se no código da aplicação com confiança, beneficiando de um processo de desenvolvimento, testes e deploy muito mais estável e automatizado.

8.1.3 Maiores Dificuldades na Realização do TFC

A realização do projeto MentHA Digital apresentou um conjunto significativo de desafios, tanto de natureza técnica como organizacional. Ao contrário de um projeto tradicional, onde se parte de uma base nova com requisitos bem definidos, encontrámo-nos perante um sistema já em funcionamento, desenvolvido ao longo de vários anos por diferentes equipas, com estilos de programação variados, lógica de negócio densa e documentação muito limitada. Este cenário tornou a curva de aprendizagem mais acentuada e obrigou a uma adaptação constante.

O principal desafio foi, sem dúvida, a **imersão numa arquitetura complexa** e herdada, o que exigiu uma análise aprofundada do código existente, identificação de dependências críticas, execução de testes constantes e tomada de decisões seguras para não comprometer funcionalidades já em produção. Esta complexidade revelou-se, no entanto, uma das maiores fontes de aprendizagem, aproximando-nos da realidade que enfrentaremos no mercado de trabalho.

Adicionalmente, enfrentámos **diversos problemas técnicos e não técnicos**, muitos dos quais detalhados no **Secção 4.6 - Desafios Enfrentados e Como os Ultrapassamos** com uma descrição estruturada dos principais obstáculos encontrados e das soluções implementadas para os superar. Destacamos, entre outros:

- A dificuldade de **configurar o ambiente de desenvolvimento**, devido à inexistência de documentação clara;
- A **desorganização e inconsistência do repositório**, com bibliotecas obsoletas e versões divergentes entre produção e desenvolvimento;
- A necessidade de **adotar e compreender novas tecnologias**, como Bootstrap, Ajax e outras, que estavam integradas no sistema sem explicação formal;
- Problemas de **sincronização entre ambientes de produção e desenvolvimento**, que obrigaram à introdução de boas práticas DevOps e à criação de uma infraestrutura conteinerizada;
- A exigência de colaborar com áreas fora da nossa zona de conforto, como a **saúde**, interpretando corretamente os requisitos e necessidades clínicas, mesmo sem formação direta nessa área.

Estas dificuldades obrigaram-nos a desenvolver não apenas competências técnicas mais avançadas, mas também a reforçar capacidades como a resiliência, a colaboração multidisciplinar e a comunicação com stakeholders de diferentes áreas. Ao ultrapassá-las, conseguimos criar uma base sólida e moderna para o futuro do projeto, contribuindo de forma estrutural para a sua continuidade e crescimento.

8.1.4 Evolução do Trabalho e Conhecimentos ao Longo do TFC

O desenvolvimento do projeto MentHA Digital representou uma etapa central e profundamente formativa no nosso percurso académico, proporcionando uma experiência prática num contexto real de intervenção em saúde digital. Trabalhar numa aplicação com impacto direto na vida de adultos mais velhos, com défices cognitivos, demência ligeira ou outras condições neurológicas, permitiu-nos aplicar os nossos conhecimentos técnicos ao serviço de uma causa com elevado valor social e clínico.

Este tipo de trabalho exigiu uma abordagem progressiva e cuidadosa, sempre com foco na estabilidade do sistema, qualidade do código e integridade dos dados. Este cenário, mais próximo de um contexto profissional real do que de um projeto criado de raiz, obrigou-nos a desenvolver competências de leitura e interpretação de código pré-existente, bem como a atuar de forma responsável em sistemas interdependentes e em produção.

Do ponto de vista técnico, tivemos oportunidade de consolidar e aprofundar competências em **Python com Django, manipulação de bases de dados relacionais (PostgreSQL)**,

desenvolvimento e organização de **interfaces web**, **controlo de versões com Git**, e **boas práticas de segurança**. A metodologia de trabalho ágil adotada ao longo do projeto permitiu-nos também vivenciar práticas reais de gestão de tarefas, ciclos iterativos de desenvolvimento e validação contínua de objetivos.

Destacamos especialmente a oportunidade de trabalhar de forma aprofundada na área de **DevOps e infraestrutura**, um domínio cada vez mais essencial no desenvolvimento de software moderno, mas que raramente é explorado com este grau de profundidade em projetos académicos. Tivemos o privilégio de desenhar e implementar uma infraestrutura robusta e modular, baseada em **Docker e Docker Compose**, que simula de forma realista ambientes de desenvolvimento e produção utilizados em contextos empresariais. Ao longo deste processo, entrámos em contacto com conceitos essenciais de **CI/CD (Integração e Entrega Contínuas)**, automação de pipelines e organização de sistemas escaláveis. Esta experiência revelou-se extremamente educativa, mostrando-nos que criar um software funcional vai muito além da escrita de código: implica considerar toda uma base técnica de suporte, infraestrutura, monitorização, segurança e manutenção. Esta aprendizagem prática em DevOps não só reforçou a nossa compreensão sobre o ciclo de vida completo de uma aplicação, como também nos preparou melhor para os desafios reais do mercado de trabalho, onde estas competências são cada vez mais valorizadas.

Para além do crescimento técnico, o projeto proporcionou também uma evolução significativa ao nível das **soft skills**. Desenvolvemos capacidades de **resolução de problemas, comunicação eficaz, trabalho colaborativo, autonomia e gestão de tempo**. Estas competências foram cruciais para enfrentar os desafios diários do projeto, desde a leitura de código até à tomada de decisões técnicas e organização do trabalho em equipa.

8.1.5 Conclusão Final

Em conclusão, o trabalho desenvolvido ao longo deste TFC não só atingiu os objetivos principais delineados inicialmente, como também lançou as bases para uma evolução contínua e sustentada da plataforma MentHA Digital. Apesar das limitações encontradas e dos obstáculos técnicos enfrentados, acreditamos que conseguimos entregar um contributo sólido e estruturante, tanto ao nível técnico como organizacional. A documentação criada, as práticas implementadas e as soluções desenvolvidas posicionam agora o projeto numa nova fase, mais madura e pronta para escalar. Com esta fundação, futuras equipas terão condições mais favoráveis para inovar, melhorar e expandir o sistema, mantendo sempre o foco no impacto real e na relevância social da aplicação.

8.2 Trabalhos Futuros

Com base no trabalho desenvolvido e descrito ao longo deste relatório, consideramos que o projeto MentHA Digital está longe de estar finalizado, mas acreditamos que a base criada nesta fase representa uma fundação sólida e moderna para que futuras equipas possam continuar o seu desenvolvimento com confiança. A nossa intervenção permitiu organizar, estabilizar e

documentar diversas camadas críticas do sistema — desde a interface e o backend até à infraestrutura de produção — oferecendo agora uma arquitetura mais clara, testável e escalável. Abaixo descrevemos os principais passos que deverão orientar os próximos ciclos de trabalho, divididos entre prioridades técnicas, etapas de continuidade e oportunidades de valorização.

8.2.1 Prioridades Técnicas Essenciais

Autenticação e Sessão de Utilizador

Um dos primeiros aspetos a ser abordado pelas equipas futuras deve ser a resolução definitiva dos problemas relacionados com a autenticação entre as diferentes partes da plataforma. Atualmente, o sistema apresenta falhas de persistência de sessão, especialmente quando um utilizador alterna entre o website (frontend) e as aplicações internas como o MentHA Care ou o MentHA Cog. Estes problemas, além de causarem frustração aos utilizadores, colocam também em risco a integridade da experiência de navegação e a consistência da informação gerida. A implementação de um sistema de autenticação centralizada, com gestão de sessão estável e partilhada entre componentes, deverá ser uma prioridade absoluta para garantir a fluidez e segurança no acesso à plataforma.

Refatoração do Protocolo MentHA

O Protocolo MentHA, por ser a base de entrada dos dados clínicos e cognitivos na plataforma, deve ser alvo de uma revisão profunda antes de qualquer melhoria nas aplicações que dele dependem. As próximas fases devem começar por uma refatoração completa dos formulários de avaliação, garantindo que todos os campos obrigatórios estão corretamente definidos, os tipos de dados validados e os modelos de dados ajustados às exigências funcionais reais. Esta revisão é essencial para assegurar que a informação recolhida no protocolo seja fidedigna, coerente e bem integrada nas funcionalidades do MentHA Care e do MentHA Cog, permitindo o correto funcionamento de relatórios, avaliações e fluxos clínicos.

Migração para o Servidor Institucional

Tal como descrito na **Secção 4.4.2**, um dos objetivos que não conseguimos concretizar foi a migração da nova infraestrutura para o servidor institucional da Universidade Lusófona. O servidor atual da faculdade está ocupado com outras aplicações e utiliza uma instalação do NGINX diretamente no sistema operativo, o que impede a utilização de uma infraestrutura totalmente containerizada como a que desenvolvemos. No entanto, deixámos este processo devidamente documentado no ficheiro README.md, incluindo instruções completas para configurar a infraestrutura em qualquer máquina com Docker. Assim que se obtenha uma nova VM dedicada, a migração poderá ser feita com relativa facilidade, bastando seguir o nosso guia técnico. Esta transição para o servidor institucional permitirá consolidar a plataforma no seu ambiente oficial e garantir maior fiabilidade a longo prazo.

8.2.2 Etapas de Continuidade

Novo Levantamento de Requisitos

Com o encerramento desta fase do projeto, torna-se essencial que as futuras equipas realizem um novo levantamento de requisitos com base nos pontos pendentes já identificados na **Secção 7.2 - Cumprimento de requisitos** deste relatório. Muitos dos requisitos que não foram totalmente implementados nesta fase foram devidamente analisados, classificados e documentados como “pendentes”, o que representa uma base sólida para priorização e planeamento futuro. A partir dessa análise, será possível organizar um novo roadmap de desenvolvimento que reflete as reais necessidades do projeto, garantindo que os esforços se concentram em áreas críticas que continuam por resolver.

Melhorias no MentHA COG

A aplicação MentHA COG representa, atualmente, a parte mais frágil e inacabada de todo o ecossistema MentHA Digital. A sua interface apresenta falhas de usabilidade, a navegação é pouco fluida e existem várias funcionalidades incompletas ou com comportamentos erráticos. Além disso, os requisitos pendentes identificados para esta aplicação incluem erros funcionais, dados mal processados e inconsistências nos fluxos de avaliação. Por esse motivo, recomendamos que a próxima equipa concentre uma parte significativa do seu esforço na reestruturação e melhoria do MentHA COG, com o objetivo de aproximar da maturidade funcional já atingida por outras partes do sistema.

Modernização Progressiva do Frontend

Durante esta fase, apenas a página inicial do website institucional foi modernizada, servindo como base visual para uma futura renovação mais abrangente. Recomendamos que as próximas equipas deem continuidade a esta modernização progressiva do frontend, atualizando outras páginas relevantes de acordo com os padrões estéticos e técnicos estabelecidos na homepage. Esta abordagem, além de garantir uma imagem mais profissional da plataforma, permite também que cada equipa deixe uma marca visual clara e coesa no projeto, sem comprometer a consistência global do design.

8.2.3 Sugestões de Valorização e Inovação

Continuidade da Documentação

A documentação foi um dos pontos mais críticos que enfrentámos no início do nosso trabalho. Por esse motivo, dedicámos um esforço significativo à criação e melhoria de documentação técnica, incluindo um README.md detalhado, comentários explicativos no código, e anexos técnicos no relatório. No entanto, esta documentação precisa de ser continuamente atualizada para acompanhar as mudanças no projeto. Reforçamos a importância de manter este nível de detalhe e rigor nos registos técnicos, pois isso é essencial para garantir a continuidade do projeto e evitar o tipo de problemas que nos afetaram inicialmente. A ausência de documentação, como se demonstrou, compromete drasticamente a capacidade de evolução da plataforma.

Criação de Dashboards com Estatísticas

A funcionalidade de exportação de dados desenvolvida no MentHA Eval representa uma oportunidade clara para a evolução analítica do projeto. Recomendamos que as futuras equipas explorem a criação de dashboards interativos, que permitam visualizar estatísticas e tendências com base nos dados recolhidos. Esses dashboards podem oferecer insights valiosos para investigadores, profissionais de saúde ou entidades parceiras, tornando o MentHA Digital não apenas uma ferramenta de avaliação, mas também uma plataforma de suporte à decisão e análise. Além disso, esta vertente pode ser aproveitada para aplicar e aprofundar conteúdos abordados em unidades curriculares como Data Science ou Business Intelligence, ampliando o valor educativo e inovador do projeto.

Bibliografia

- [DEISI24] DEISI, Regulamento de Trabalho Final de Curso, out. 2024.
- [DEISI24b] DEISI, www.deisi.ulusofona.pt, out. 2024.
- [ULHT24] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt, acedido em out. 2024.
- [ADEB24] Associação de Apoio aos Doentes Depressivos e Bipolares, <https://www.adeb.pt/>; acedido em nov. 2024.
- [ASMAL24] Associação de Saúde Mental do Algarve; <https://www.asmal.org.pt/>; acedido em nov.2024.
- [CVP24] Cruz Vermelha Portuguesa; <https://www.cruzvermelha.pt/>; acedido em nov.2024.
- [ELO24] Elo Social; <https://www.elosocial.org/>; acedido em nov.2024.
- [FAME24] FamiliarMente; <https://familiarmente.pt/>; acedido em nov.2024.
- [GIRA24] Grupo Intervenção e Reabilitação Ativa; <https://www.gira.org.pt/>; acedido em nov.2024.
- [CRL24] CRINABEL – Lar residencial e Centro de Atividades Ocupacionais; <https://www.crinabel.pt/>; acedido em nov.2024.
- [APDP24] Associação Protetora dos diabéticos de Portugal; <https://apdp.pt/> ; acedido em nov.2024.
- [PPI90] Programa de Psicoestimulação Integral (Boada y Tárraga, 1990) ; <https://www.fundacioace.com/es/proyectos/psicología-medica/programa-de-psicoestimulacion-integrada-ppi.html>; acedido em nov.2024
- [COWE24] CogWeb, [Cogweb - Treino cognitivo online](#); acedido em nov. 2024.
- [HUME24] HumanaMente, [Humanamente - Há muito para fazer com e pela pessoa com demência \(humana-mente.pt\)](#); acedido em nov. 2024.
- [SILI24] Sioslife, [Sioslife](#); acedido em nov. 2024.
- [COFI24] CogniFit, [Treinamento cerebral | Com a confiança dos médicos - CogniFit](#); acedido em nov. 2024.
- [HTML24] MDNwebdocs, <https://developer.mozilla.org/en-US/docs/Web/HTML>; acedido em nov. 2024.
- [CSS24] MDNwebdocs, <https://developer.mozilla.org/en-US/docs/Web/CSS>; acedido em nov. 2024.
- [JS24] MDNwebdocs, [JavaScript | MDN \(mozilla.org\)](#); acedido em nov. 2024.
- [BOAP24] Bootstrap, [Bootstrap · The most popular HTML, CSS, and JS library in the world. \(getbootstrap.com\)](#); acedido em nov. 2024.
- [PY24] Python, [What's New In Python 3.12 — Python 3.12.0 documentation](#); acedido em nov.2024.
- [DGO24] Django, [The web framework for perfectionists with deadlines | Django \(djangoproject.com\)](#); acedido em nov. 2024.
- [SQLI24] SQLite, [SQLite Documentation](#); acedido em nov. 2024.

- [DOCC25] Docker Compose <https://docs.docker.com/compose/>; acedido em março 2025
- [DOCK25] Docker <https://docs.docker.com/desktop/>; acedido em março 2025
- [POES24] PostgreSQL, <https://www.postgresql.org/docs/>; acedido em nov.2024.
- [TFC21a] Afonso Rodrigues, “Plataforma MentHA”, relatório de Trabalho Final de Curso DEISI152 desenvolvido no ano letivo de 2020/2021,
https://teses.pythonanywhere.com/media/teses/TFC_21_DEISI152.pdf, acedido em nov.2024
- [TFC22a] Nuno Rasteiro, “Aplicação Web do Protocolo MentHA”, relatório de Trabalho Final de Curso DEISI233 desenvolvido no ano letivo de 2021/2022,
https://teses.pythonanywhere.com/media/teses/TFC_22_DEISI233.pdf, acedido em nov.2024
- [TFC22b] Inês Rocha, João Santos “Aplicação MentHA CARE”, relatório de Trabalho Final de Curso DEISI235 desenvolvido no ano letivo de 2021/2022,
https://informatica.ulusofona.pt/wpcontent/uploads/sites/57/2022/10/TFC_22_DEISI235.pdf, acedido em nov.2024
- [TFC23a] Tiago Silva, Rodrigo Eira, “Aplicação MentHA CARE”, relatório de Trabalho Final de Curso DEISI 295 desenvolvido no ano letivo de 2022/2023,
https://teses.pythonanywhere.com/media/teses/TFC_23_DEISI295.pdf, acedido em nov.2024
- [TFC23b] Bernardo Roque, Alexandre Godinho, “Aplicação Web do Protocolo MentHA”, , relatório de Trabalho Final de Curso DEISI354, desenvolvido no ano letivo de 2022/2023,
https://teses.pythonanywhere.com/media/teses/TFC_23_DEISI354.pdf, acedido em nov.2024
- [TFC24a] Miguel Cardoso, António Ventura “Plataforma MentHA”, relatório de Trabalho Final de Curso DEISI71, desenvolvido no ano letivo de 2023/2024,
https://teses.pythonanywhere.com/media/teses/DEISI71_RELATORIO_FINAL.pdf, acedido em nov.2024
- [SFHX25] Sphinx, <https://www.sphinx-doc.org/en/master/>; acedido em fevereiro de 2025.
- [MTFY25] Mintlify, <https://mintlify.com/>; acedido em fevereiro de 2025.
- [QODO25] QodoAI, <https://www.qodo.ai/>; acedido em fevereiro de 2025.
- [CGPT25] ChatGPT, <https://chatgpt.com/>; acedido em fevereiro de 2025.

Anexo A – Aplicações MentHA Digital

A.1 MentHA COG – Programa de Estimulação Cognitiva para Adultos Mais Velhos

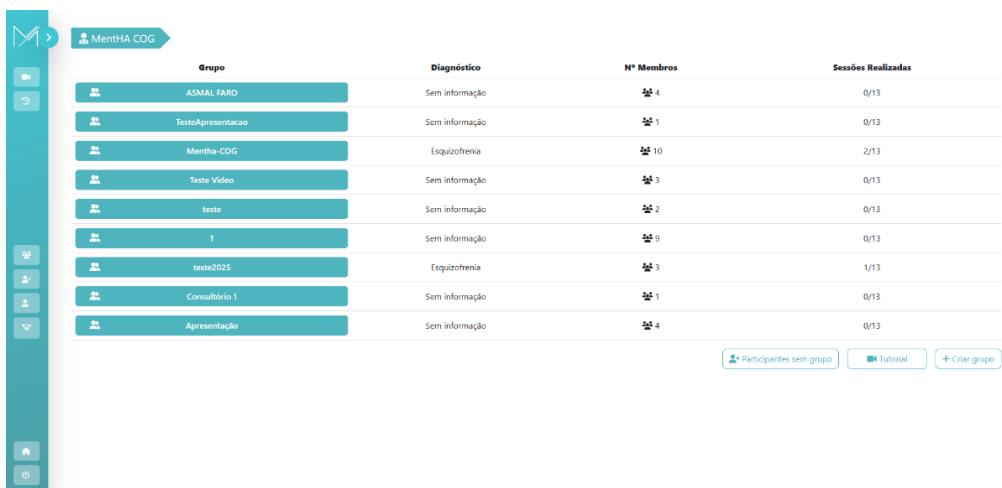
O MentHA COG é um programa de estimulação cognitiva, projetado para mitigar o declínio cognitivo em adultos mais velhos que apresentam sinais de demência ou comprometimento neuropsicológico. Este programa, desenvolvido no contexto de intervenções não farmacológicas, consiste em 14 sessões realizadas duas vezes por semana. Cada sessão, com duração de 90 minutos, está estruturada em três fases: **Social**, para acolhimento dos participantes; **Intervenção**, com foco nas atividades cognitivas e sociais; e **Encerramento**, para reflexão e finalização.

Baseado em evidências científicas e adaptado do Programa de Psicoestimulação Integral (Boada y Tárraga, 1990) [PPI90], o MentHA COG tem como principais objetivos:

- Evitar a desconexão da pessoa do seu meio, potenciando as relações sociais
- Proporcionar segurança funcional
- Aumentar a autonomia e diminuir a dependência
- Melhorar a imagem pessoal e autoestima
- Minimizar o stress e evitar reações anómalas (ex. agressividade, agitação, etc.)

Além disso, o programa integra o treino de funções cognitivas específicas, utilizando atividades adaptativas que ajustam a dificuldade com base no desempenho individual. Com uma abordagem global, o MentHA COG combina a estimulação cognitiva e social, promovendo um impacto positivo nos domínios neurocognitivo, emocional, funcional e comportamental.

Na **Figura 36**, na **Figura 37**, na **Figura 38**, na **Figura 39** e na **Figura 40** pode-se observar as principais características do MentHA COG.



The screenshot shows the main interface of the MentHA COG application. On the left, there is a vertical sidebar with icons for home, user profile, and other navigation options. The main content area is titled "MentHA COG" and displays a table of groups. The columns in the table are "Grupo", "Diagnóstico", "Nº Membros", and "Sessões Realizadas". The data rows are as follows:

Grupo	Diagnóstico	Nº Membros	Sessões Realizadas
ASMAL FARO	Sem informação	4	0/13
TesteApresentacao	Sem informação	1	0/13
Mentha-COG	Esquizofrenia	10	2/13
Teste Video	Sem informação	3	0/13
teste	Sem informação	2	0/13
1	Sem informação	9	0/13
teste2025	Esquizofrenia	3	1/13
Consultório 1	Sem informação	1	0/13
Apresentação	Sem informação	4	0/13

At the bottom right of the table, there are three buttons: "Participantes sem grupo", "Tutorial", and "+ Criar grupo".

Figura 36 - Página principal do MentHA COG

Figura 37 - Página de Grupo do MentHA COG

Figura 38 - Página de Sessões do Grupo MentHA COG

Figura 39 - Página de Sessão do MentHA COG



Figura 40 - Página de uma Atividade de uma Sessão do MentHA COG

A.2 Protocolo MentHA – Avaliação Neuropsicológica para Validação de Treino Cognitivo

O Protocolo MentHA é uma ferramenta de avaliação neuropsicológica desenvolvida para monitorizar e validar os efeitos do programa de Estimulação Cognitiva MentHA-COG. Este protocolo tem como objetivo principal facilitar o acesso remoto à avaliação neuropsicológica de adultos mais velhos com défice cognitivo ou patologias mentais prévias, contribuindo significativamente para a promoção do bem-estar e da qualidade de vida desses indivíduos, independentemente do contexto em que se encontram.

A avaliação é baseada em testes breves amplamente reconhecidos na prática clínica e validados para a população, garantindo elevada precisão técnica na identificação de funções cognitivas. As provas utilizadas estão organizadas em quatro domínios principais:

- **Neuro cognitivo:** atenção, memória, linguagem, função executiva, cognição social e habilidades preceptivo-motoras.
- **Emocional:** ansiedade e depressão.
- **Funcional:** avaliação das atividades básicas e instrumentais da vida diária.
- **Comportamental:** análise da sintomatologia psicopatológica.

O protocolo está dividido em duas etapas principais:

1. **Caracterização dos Participantes:** Focada na recolha de informações sociodemográficas, avaliação psicológica e funcional, bem como critérios para inclusão no programa, como idade superior a 50 anos, diagnóstico neuropsiquiátrico, ou presença de queixas cognitivas ou demência. Esta etapa assegura que os participantes atendem aos requisitos necessários para integrar o programa.
2. **Avaliação Pré e Pós-Intervenção:** Envolve a análise de variáveis como cognição, humor, funcionalidade instrumental e estágio de deterioração cognitiva, permitindo medir o impacto da intervenção no indivíduo.

O Protocolo MentHA (MentHA EVAL) tem como principais objetivos:

- Proporcionar acesso remoto a um protocolo de avaliação neuropsicológica para diagnóstico preciso e personalizado.
- Apoiar a recolha de dados clínicos, permitindo intervenções mais eficazes com base em avaliações detalhadas.

Na **Figura 41**, na **Figura 42**, na **Figura 43** e na **Figura 44** pode-se observar as principais características do MentHA COG.

The screenshot shows the participant profile for 'Tomás Nave'. Basic details include Name: Tomás Nave, Date of Birth: Out. 9, 2024 (0 anos), Sex: Masculino, Nationality: Portugal, and Contact: Teléfono: 987654321, Email: example@gmail.com, Escolaridade: 11+, Localização: Lisboa. Referrals section shows 'Sem Informação' (No information). The 'Avaliações' (Assessments) section lists five completed sessions: Rastreio, Quantidão Sociodemográfico, 2 meses, Teste Livre, and MentHA-Risk, along with their dates, times, progress (9%), and options to generate reports or delete them. A date range selector at the bottom allows setting from 21/11/2024 to 20/46.

Figura 41 - Página de perfil de participantes do Protocolo MentHA

The screenshot shows the 'MentHA-Risk' assessment page. It includes a summary of completed questionnaires on 31/10/2024, such as 'Tem Doenças Cognitivas?' (Não), 'Fumador Ativo' (Não), 'Diabetes' (Sim), 'Acidente Vascular Cerebral (AVC)' (Não), and 'Doença das Rins' (Não). Below this are sections for 'Dados Pessoais' (Height, Weight, Blood Pressure) and 'Comorbidades' (Hypertension, Diabetes, etc.). A note indicates 'Se tem diabète, há quantos anos' (If you have diabetes, how many years).

Figura 42 - Página de Avaliação MentHA-Risk do MentHA COG

The screenshot shows the 'Visão geral do Protocolo MentHA' (General view of the MentHA protocol). It displays a grid of assessments across three time points: 'Rastreio' (18/10/2024), 'Questionário Sociodemográfico' (18/10/2024), and '2 meses' (18/10/2024). The grid includes rows for 'Questionário Sociodemográfico (Cuidador)', 'Sobrecarga do Cuidador - QSCI', 'Rede de Apoio Social - Lubben', 'Estratégias de Coping - brief COPE', 'Consciência, humor e comportamento', 'Funcionalidade - ADL/ IADL', 'Palissiatomologia - BSI', 'Cognição - ACE-R', 'Cognição - MMSE', 'Consciência, humor e comportamento', 'Estado de Ánimo - PANAS', 'Funcionalidade - AND', 'Personalidade - NEO-FFI 20', 'Ansiedade e Depressão - HADS', 'Cognição - Áreas Complementares', and 'Estádio de Deterioração - GDS'. Colored boxes indicate the status of each assessment: red for pending, green for completed, and light blue for partially completed.

Figura 43 - Página de Visão Geral do Protocolo MentHA



Figura 44 - Página de Questionário de uma Avaliação do Protocolo MentHA

A.3 MentHA CARE – Programa de Apoio Psicoeducativo para Cuidadores

O MentHA CARE é um programa psicoeducativo inovador, desenhado para apoiar cuidadores informais e formais na gestão dos desafios associados ao cuidado de pessoas com comprometimentos neuropsicológicos. Este programa combina duas componentes fundamentais: suporte educativo, focado na aquisição de competências e boas práticas, e suporte emocional, promovendo o bem-estar psicológico dos cuidadores.

A intervenção é composta por 13 sessões psicoeducativas realizadas duas vezes por semana, com a duração de 90 minutos cada. As sessões seguem uma estrutura em três fases:

1. **Fase Inicial:** Receção e enquadramento dos participantes.
2. **Fase de Desenvolvimento:** Realização de atividades práticas e dinâmicas baseadas em estratégias propostas por profissionais da área do envelhecimento e da saúde mental.
3. **Fase Final:** Reflexão conjunta e síntese dos resultados, proporcionando um espaço de partilha de experiências e aprendizagens entre os participantes.

O principal objetivo do programa é capacitar os cuidadores com conhecimentos e estratégias eficazes, não apenas por meio da orientação de profissionais, mas também pela troca de vivências com outros cuidadores em contextos semelhantes. Além disso, visa reduzir os níveis de ansiedade e depressão associados ao impacto emocional do cuidado, enquanto incentiva a adoção de práticas que melhorem a qualidade da assistência prestada.

O MentHA CARE foi concebido para operar de forma paralela e complementar ao programa de estimulação cognitiva MentHA COG, possibilitando uma intervenção integrada tanto para os receptores de cuidados quanto para os cuidadores. Dessa forma, promove-se um suporte abrangente que abrange aspectos emocionais, educacionais e funcionais, com impacto direto no bem-estar e na qualidade de vida de todos os envolvidos.

O MentHA CARE tem como principais objetivos:

- Educar os cuidadores sobre perturbações neuro cognitivas prevalentes, como demência, esquizofrenia e perturbação bipolar.
- Explorar estratégias de comunicação e gestão de comportamentos desafiantes.
- Promover boas práticas de cuidado e a gestão emocional dos cuidadores.
- Incentivar o autocuidado e a valorização do papel do cuidador.

Na **Figura 45**, na **Figura 46**, na **Figura 47**, na **Figura 48**, na **Figura 49** pode-se observar as principais características do MentHA CARE.

The screenshot shows the main dashboard of the MentHA CARE application. On the left is a vertical sidebar with icons for home, user management, and search. The main area has a header 'MentHA CARE' with a back arrow. Below it is a table with columns: 'Grupo', 'Diagnóstico', 'Nº Membros', and 'Sessões Realizadas'. The table lists various groups, all of which have 'Sem informação' in the 'Diagnóstico' column. The 'Nº Membros' column shows counts ranging from 0 to 4. The 'Sessões Realizadas' column shows counts from 0/13 to 1/13. At the bottom right are buttons for 'Cuidadores sem grupo', 'Tutorial', and 'Criar grupo'.

Grupo	Diagnóstico	Nº Membros	Sessões Realizadas
Grupo António	Sem informação	1	0/13
Grupo Teste	Sem informação	0	0/13
Mentha-CARE Test Group	Sem informação	0	0/13
Mentha-CARE Test Group	Sem informação	0	0/13
Tó	Sem informação	0	0/13
dasdas	Sem informação	1	0/13
GrupoTesteApresentação	Sem informação	4	0/13
Grupo Teste	Sem informação	3	0/13
Lar	Sem informação	0	0/13
Grupo Filtragem	Sem informação	2	0/13
asmal	Sem informação	3	0/13
Tomás Nave	Sem informação	4	1/13

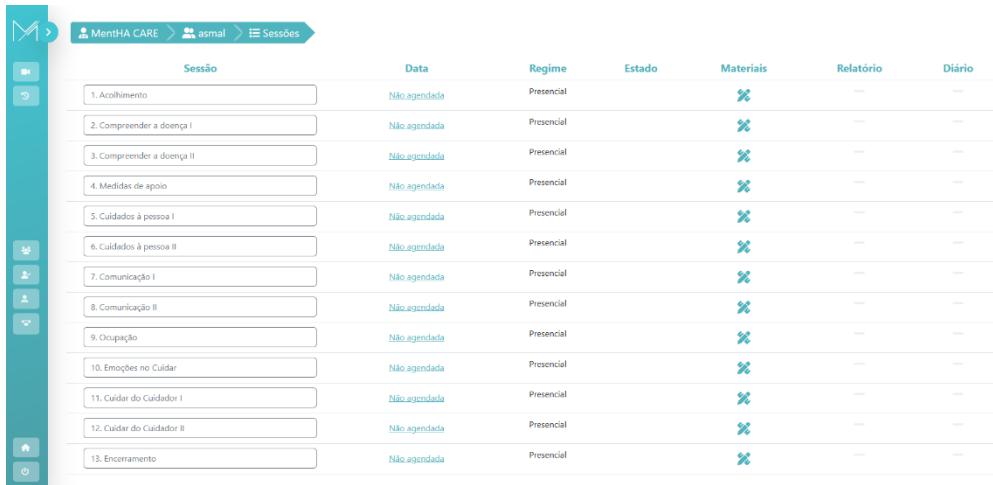
Figura 45 - Página principal do MentHA CARE

The screenshot shows the 'Membros do Grupo: asmal' section. It displays a table with columns: Nome, E-mail, Telefone, 0 meses, 2 meses, 6 meses, and Sessões (Atendidas / Realizadas). Two members are listed: 'Antonio Ventu' (aaa2341@gmail.com) with 0/0 sessions and 'Carolina' (carolina@asmal.pt) with 1/0 sessions. Below this is the 'Dinamizadores do Grupo: asmal' section, showing a table with columns: Nome, E-mail, and Telefone. One dinamizer, 'Almira' (almira@asmal.pt), is listed with the phone number 123456789. At the bottom are buttons for 'Associar Cuidador' and 'Associar Dinamizador'.

Nome	E-mail	Telefone	0 meses	2 meses	6 meses	Sessões (Atendidas / Realizadas)
Antonio Ventu	aaa2341@gmail.com	987644321	—	—	—	0/0
Carolina	carolina@asmal.pt	987854321	—	—	—	1/0

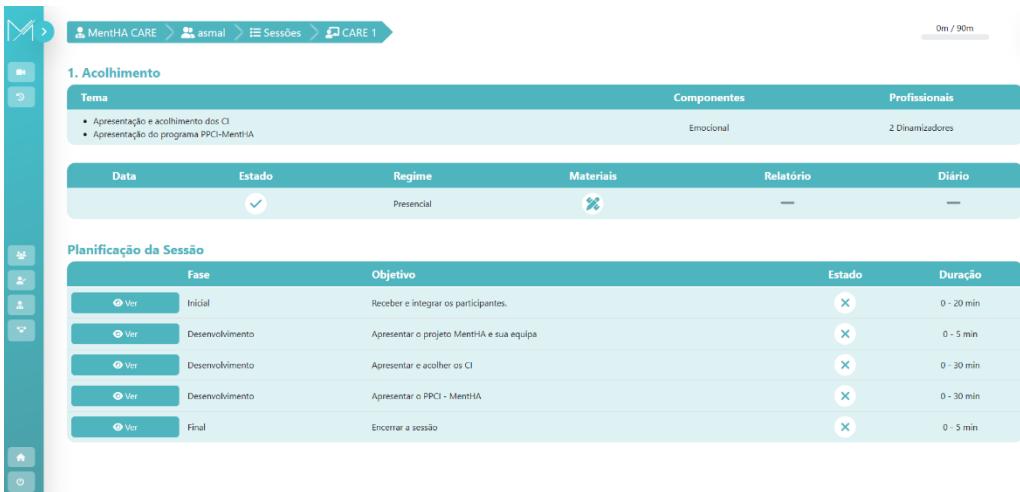
Nome	E-mail	Telefone
Almira	almira@asmal.pt	123456789

Figura 46 - Página do Grupo do MentHA CARE



Sessão	Data	Regime	Estado	Materiais	Relatório	Diário
1. Acolhimento	Não agendada	Presencial		X	—	—
2. Compreender a doença I	Não agendada	Presencial		X	—	—
3. Compreender a doença II	Não agendada	Presencial		X	—	—
4. Medidas de apoio	Não agendada	Presencial		X	—	—
5. Cuidados à pessoa I	Não agendada	Presencial		X	—	—
6. Cuidados à pessoa II	Não agendada	Presencial		X	—	—
7. Comunicação I	Não agendada	Presencial		X	—	—
8. Comunicação II	Não agendada	Presencial		X	—	—
9. Ocupação	Não agendada	Presencial		X	—	—
10. Emoções no Cuidar	Não agendada	Presencial		X	—	—
11. Cuidar do Cuidador I	Não agendada	Presencial		X	—	—
12. Cuidar do Cuidador II	Não agendada	Presencial		X	—	—
13. Encerramento	Não agendada	Presencial		X	—	—

Figura 47 - Página de Sessões do Grupo MentHA CARE



1. Acolhimento

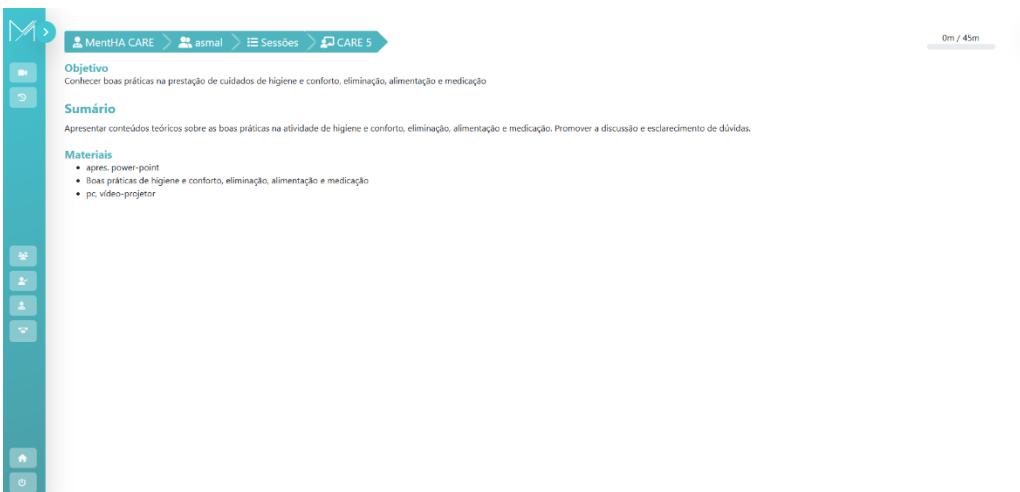
Tema	Componentes	Profissionais
• Apresentação e acolhimento dos CI • Apresentação do programa PPCI-MentHA	Emocional	2 Dinamizadores

Data	Estado	Regime	Materiais	Relatório	Diário
✓		Presencial	X	—	—

Planificação da Sessão

Fase	Objetivo	Estado	Duração
Ver	Inicial Receber e integrar os participantes.	X	0 - 20 min
Ver	Desenvolvimento Apresentar o projeto MentHA e sua equipa	X	0 - 5 min
Ver	Desenvolvimento Apresentar e acolher os CI	X	0 - 30 min
Ver	Desenvolvimento Apresentar o PPCI - MentHA	X	0 - 30 min
Ver	Final Encerrar a sessão	X	0 - 5 min

Figura 48 - Página de Sessão do MentHA CARE



Objetivo
Conhecer boas práticas na prestação de cuidados de higiene e conforto, eliminação, alimentação e medicação

Sumário
Apresentar conteúdos teóricos sobre as boas práticas na atividade de higiene e conforto, eliminação, alimentação e medicação. Promover a discussão e esclarecimento de dúvidas.

Materiais

- apes; power-point
- Boas práticas de higiene e conforto, eliminação, alimentação e medicação
- pc; vídeo-projetor

Figura 49 - Página de uma Atividade de uma Sessão do MentHA CARE

Anexo B – Casos de Uso

Os Casos de Uso são diagramas que descrevem, justificam e organizam todas as tarefas possíveis de realizar na aplicação, com base nas interações dos utilizadores. Estes diagramas permitem compreender as ações que cada entidade pode executar na aplicação MentHA e o objetivo de cada uma.

No âmbito deste projeto, foram identificadas as seguintes entidades:

- **Aplicação:** Refere-se à aplicação como um todo.
- **Relatório:** Documento emitido após a sessão.
- **Dinamizador:** É o utilizador responsável pela sessão.
- **Cuidador:** É o utilizador que irá usufruir das sessões.
- **Administrador:** É o utilizador responsável pela plataforma
- **Sessão:** Momento de reunião com os cuidadores

Os Casos de Uso que iremos apresentar foram realizados pelos colegas Miguel Cardoso e António Ventura, que desenvolveram o TFC no ano passado [TFC24a].

Na **Figura 50** apresenta-se os Casos de Uso para o Dinamizador, um dos atores da Aplicação que no caso é o utilizador responsável para guiar a Sessão.

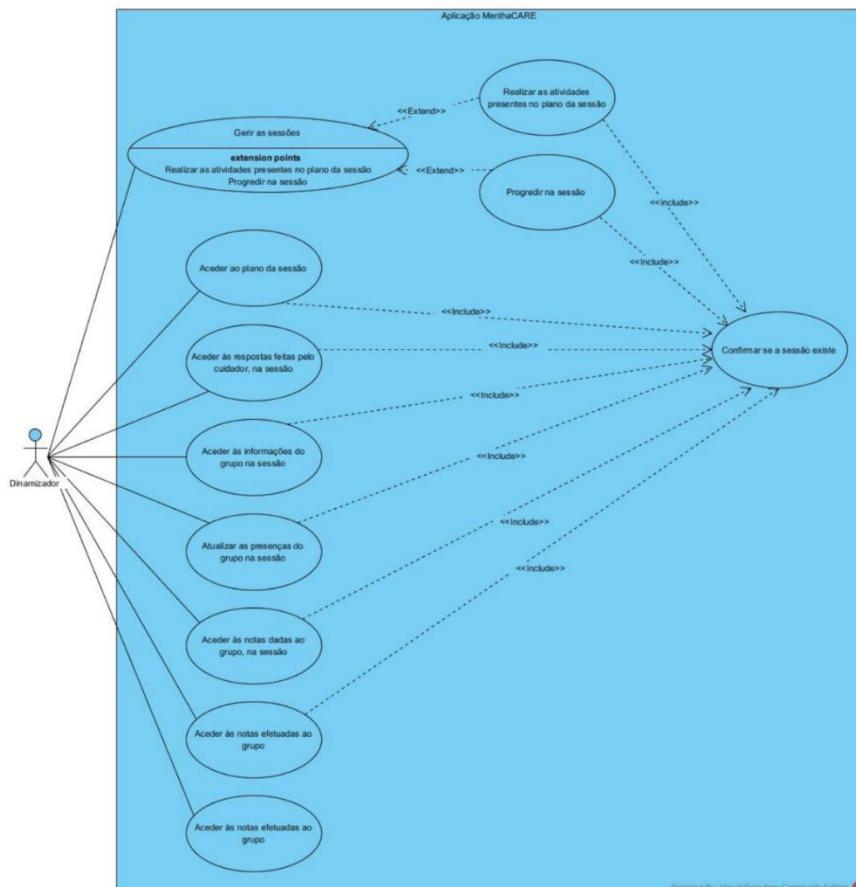


Figura 50 - Casos de Uso do Dinamizador MentHA CARE

Na **Figura 51** vê-se as funcionalidades que o Administrador consegue fazer na aplicação MentHA CARE.

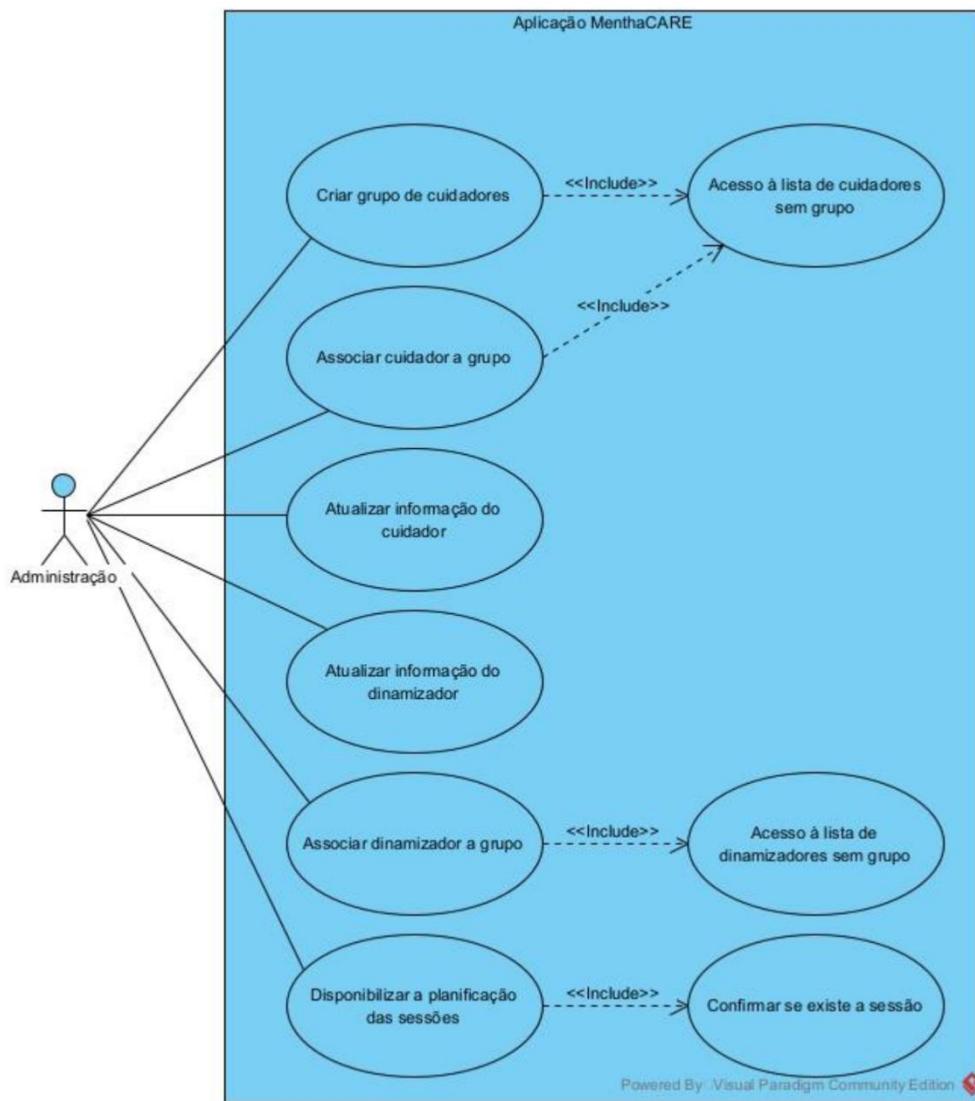


Figura 51 - Casos de Uso do Administrador no MentHA CARE

Para o Protocolo MentHA, demonstramos um Caso de Uso referente ao Avaliador na **Figura 52**.

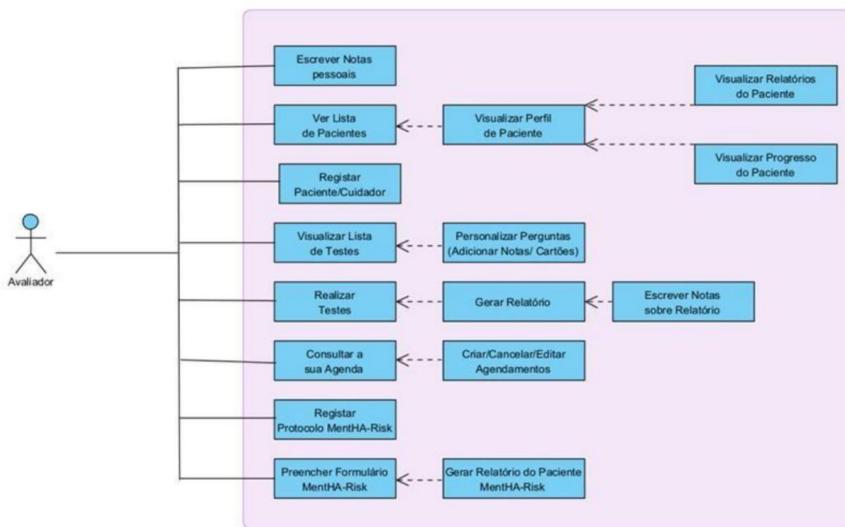


Figura 52 - Casos de Uso relativos ao Protocolo MentHA

Anexo C – Modelo ERD MentHA COG / MentHA CARE

Um **Diagrama de Entidade-Relacionamento (ERD)** é uma representação visual que descreve a estrutura lógica de uma base de dados, evidenciando as entidades, seus atributos e os relacionamentos entre elas. Este tipo de diagrama é amplamente utilizado no design e na documentação de sistemas de informação. Foi construído este modelo para as aplicações MentHA COG e MentHA CARE devido ao seu tamanho e complexidade.

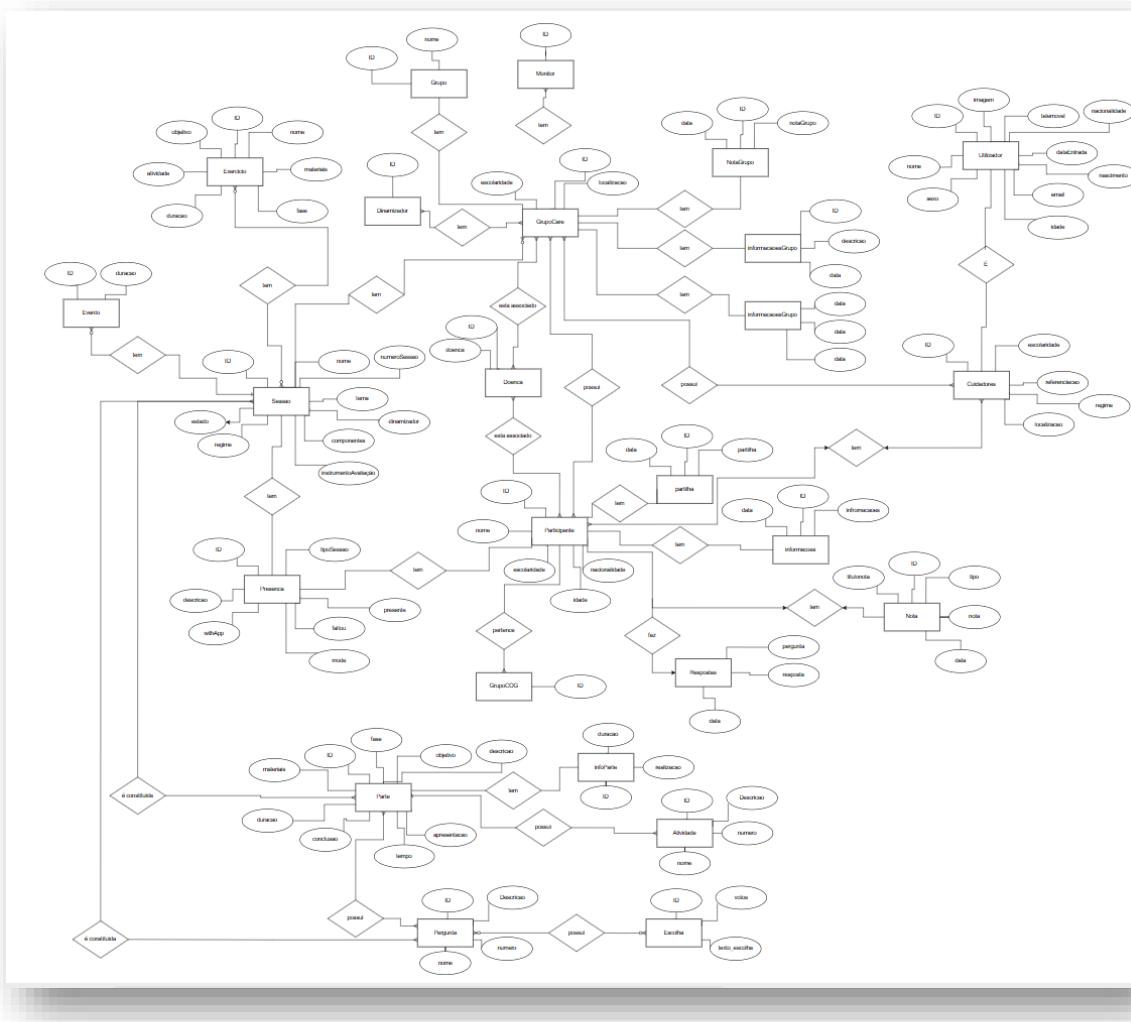


Figura 53 - Diagrama ERD MentHA CARE/COG

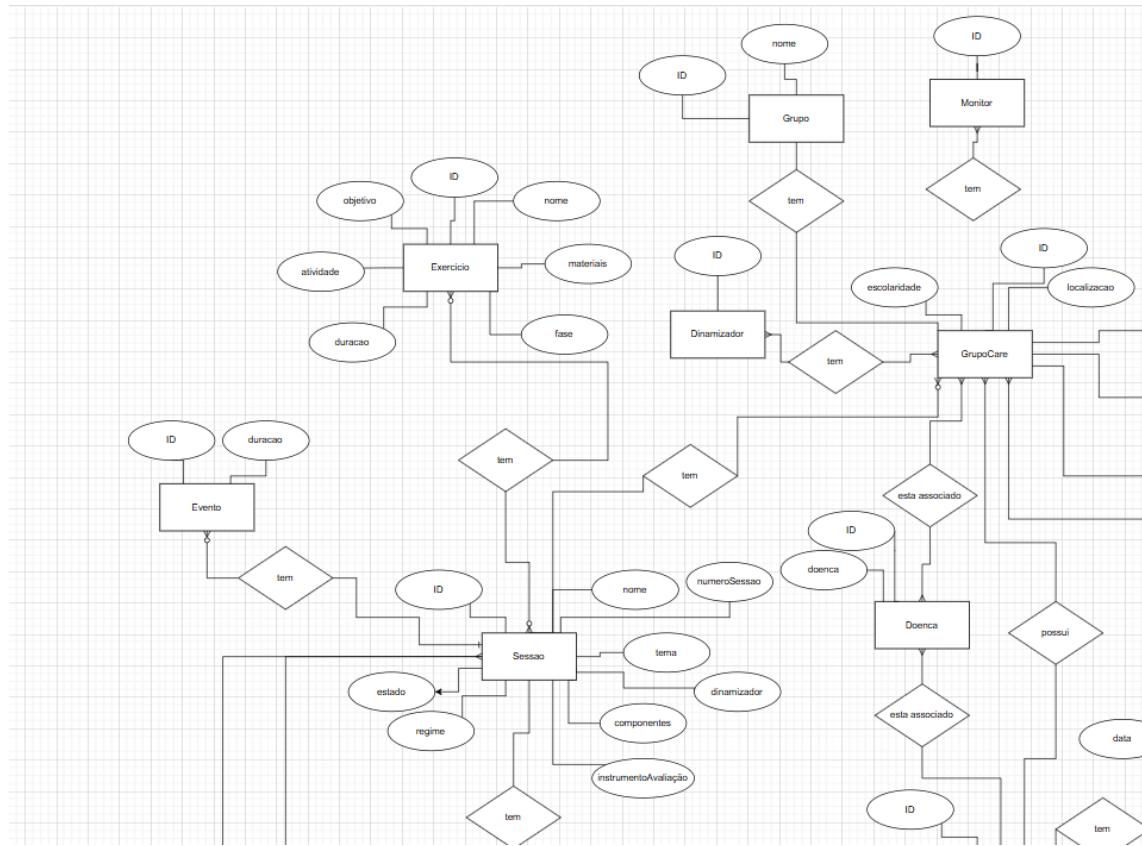


Figura 54 – Diagrama ERD parte 1

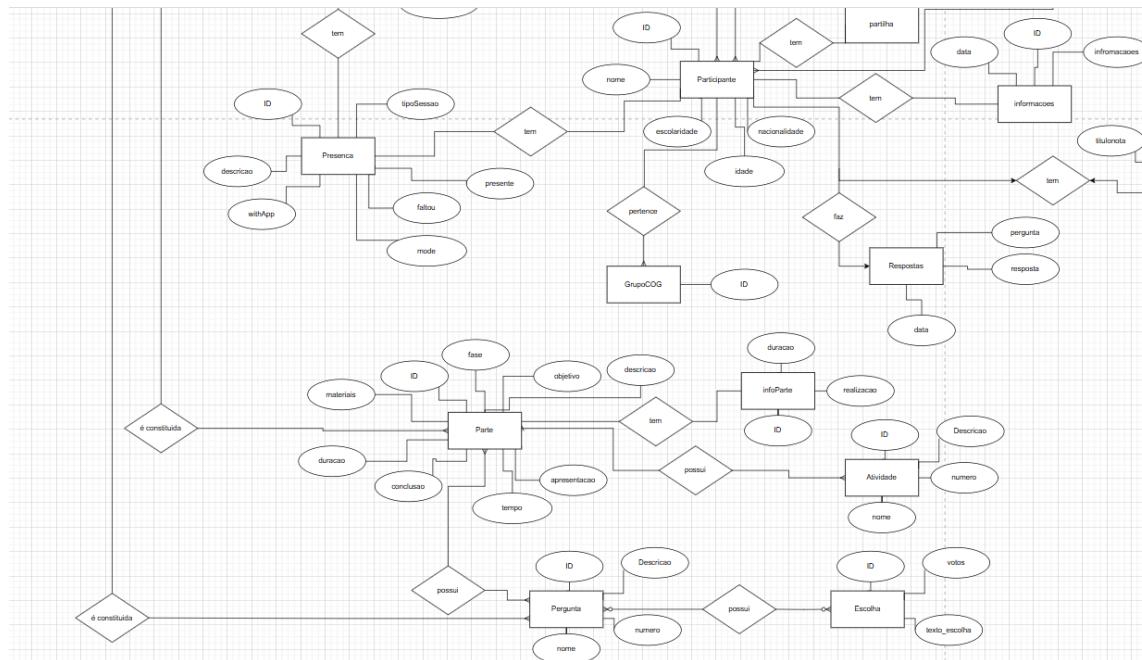


Figura 55 - Diagrama ERD parte 2

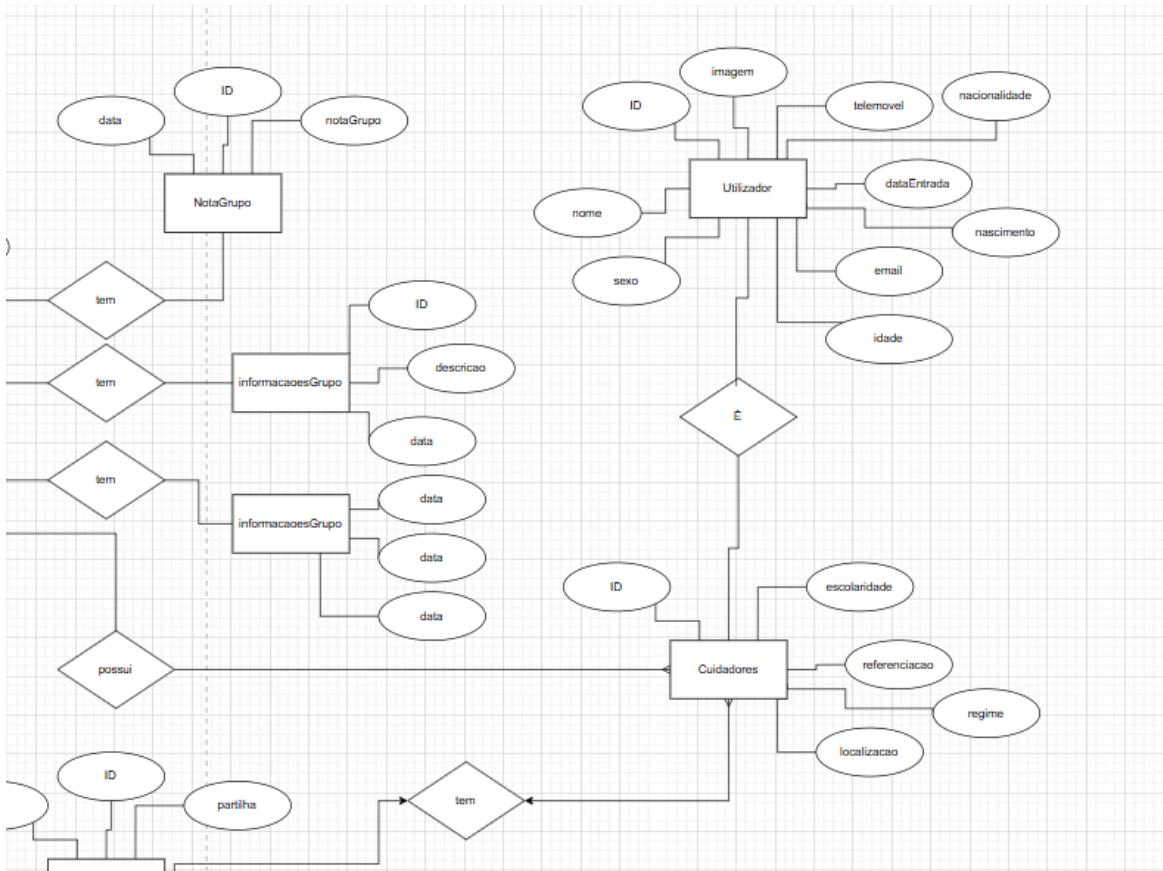


Figura 56 - Diagrama ERD parte 3

Anexo D - Workflow de Desenvolvimento

Para garantir um processo de desenvolvimento estruturado e seguro, adotámos um workflow baseado em ramificações no Git. Decidimos utilizar esta abordagem para podermos realizar alterações no código sem correr o risco de introduzir erros graves na base de código. Como este não é um projeto criado do zero por nós, mas sim um sistema que já tem vários anos de desenvolvimento, com um tamanho e estrutura consideráveis, a utilização deste workflow ajuda-nos a prevenir problemas ao modificar componentes já existentes.

Além disso, este método de trabalho está alinhado com a nossa organização e planeamento. Realizamos reuniões semanais para avaliar o progresso do desenvolvimento e validar as alterações antes de serem integradas na versão principal do projeto. Dessa forma, conseguimos garantir que apenas código revisto e aprovado chega à branch principal (main), evitando instabilidades no sistema e permitindo-nos focar no desenvolvimento sem grandes preocupações com erros críticos.

D.1 Estrutura do Workflow

A estrutura do nosso workflow segue três níveis principais de branches:

1. Branch Principal (main)

A branch main contém a versão mais estável e atualizada do projeto, pronta para ser enviada para o servidor. Qualquer alteração que chegue a esta branch passou previamente por revisões e validações.

2. Branch de Desenvolvimento (dev)

A branch dev é derivada da main e serve como o espaço principal de desenvolvimento. Todas as alterações feitas pelos membros da equipa são primeiro integradas aqui. Após implementarmos novas funcionalidades ou correções, apresentamos as modificações ao nosso orientador. Apenas após a validação é que realizamos o merge da dev para a main, garantindo que as mudanças foram analisadas antes da publicação no servidor.

3. Branches Pessoais

Para permitir que cada membro da equipa trabalhe em simultâneo sem conflitos, cada um de nós (eu e o André) cria uma branch pessoal clonada da dev. Nestas branches, implementamos as nossas funcionalidades ou correções de forma isolada. Assim que a implementação está concluída, submetemos um **pull request** para a dev, onde as alterações passam por revisão e possíveis ajustes antes de serem integradas.

Este fluxo de trabalho permite-nos um desenvolvimento mais seguro e organizado, minimizando riscos ao modificar um código com grande estrutura e garantindo que cada alteração seja avaliada antes de ser disponibilizada na versão final do projeto.

D.2 Fluxo de Trabalho

O processo que seguimos para desenvolver no projeto é o seguinte:

1. Clonar o repositório para o computador:

Antes de começarmos a trabalhar, precisamos de obter uma cópia local do projeto através do comando:

```
C:\Users\Tomás Nave>git clone https://github.com/MentHA-ULHT/mentha_digital
Cloning into 'mentha_digital'...
remote: Enumerating objects: 14226, done.
remote: Counting objects: 100% (90/90), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 14226 (delta 49), reused 44 (delta 23), pack-reused 14136 (from 1)
Receiving objects: 100% (14226/14226), 54.42 MiB | 2.04 MiB/s, done.
Resolving deltas: 100% (6892/6892), done.
Updating files: 100% (6306/6306), done.

C:\Users\Tomás Nave>cd mentha_digital

C:\Users\Tomás Nave\mentha_digital>code .
```

Figura 57 - WorkFlow 1

2. Criar a branch dev:

Caso a branch de desenvolvimento (dev) ainda não exista, criamo-la a partir da main:

```
● PS C:\Users\Tomás Nave\mentha_digital> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
● PS C:\Users\Tomás Nave\mentha_digital> git pull origin main
From https://github.com/MentHA-ULHT/mentha_digital
 * branch      main       -> FETCH_HEAD
Already up to date.
● PS C:\Users\Tomás Nave\mentha_digital> git checkout -b dev1
Switched to a new branch 'dev1'.
● PS C:\Users\Tomás Nave\mentha_digital> git pull origin dev
From https://github.com/MentHA-ULHT/mentha_digital
 * branch      dev        -> FETCH_HEAD
Updating f9e6d50..61dc59c
Fast-forward
 config/settings.py          |  2 ++
 .../Risco_Cardiovascular_LucioAba6bb6734898.docx | Bin 0 -> 83493 bytes
 .../Risco_Cardiovascular_aaaa371dffca08.docx    | Bin 0 -> 83506 bytes
 .../Risco_Cardiovascular_aaaa48ea1b6fe4.docx    | Bin 0 -> 83508 bytes
 .../Risco_Cardiovascular_aaaa4f380e957f.docx    | Bin 0 -> 83512 bytes
 .../Risco_Cardiovascular_aaaa82b258adf4.docx    | Bin 0 -> 83513 bytes
 .../Risco_Cardiovascular_aaaaa4db768c31.docx    | Bin 0 -> 83484 bytes
 .../Risco_Cardiovascular_aaaaac7da5deb88.docx   | Bin 0 -> 83505 bytes
 .../Risco_Cardiovascular_aaaaae925023aa4.docx   | Bin 0 -> 83507 bytes
 .../Risco_Cardiovascular_aaaaaf33ed509c5.docx   | Bin 0 -> 83506 bytes
 .../Risco_Cardiovascular_aaaaaf52b9b90dc.docx   | Bin 0 -> 83513 bytes
```

Figura 58 - WorkFlow 2

- Git checkout -b dev1 (cria o branch apenas no ficheiro local do projeto)
- Git pull origin dev1 (Para mandar o branch para o repositório remoto)

3. Aceder a branch dev:

Sempre que iniciamos um novo ciclo de desenvolvimento, certificamo-nos de que temos a versão mais recente da dev localmente:

```
● PS C:\Users\Tomás Nave\mentha_digital> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
● PS C:\Users\Tomás Nave\mentha_digital> git checkout dev1
Switched to branch 'dev1'
Your branch is up to date with 'origin/dev1'.
● PS C:\Users\Tomás Nave\mentha_digital> git pull origin dev1
From https://github.com/MentHA-ULHT/mentha_digital
 * branch           dev1      -> FETCH_HEAD
   Already up to date.
○ PS C:\Users\Tomás Nave\mentha_digital> []
```

Figura 59 - WorkFlow 3

- Git pull origin dev1 (Para obter a versão que está no repositório remoto)

4. Criar a branch pessoal:

Cada desenvolvedor cria a sua própria branch a partir da dev para desenvolver funcionalidades de forma isolada:

```
● PS C:\Users\Tomás Nave\mentha_digital> git checkout -b bTomas1
Switched to a new branch 'bTomas1'
● PS C:\Users\Tomás Nave\mentha_digital> git push origin bTomas1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'bTomas1' on GitHub by visiting:
remote:     https://github.com/MentHA-ULHT/mentha_digital/pull/new/bTomas1
remote:
To https://github.com/MentHA-ULHT/mentha_digital
 * [new branch]      bTomas1 -> bTomas1
○ PS C:\Users\Tomás Nave\mentha_digital> []
```

Figura 60 - WorkFlow 4

5. Fazer as alterações e commits:

Depois de realizar as alterações necessárias no código, adicionamos os ficheiros ao Git, criamos um commit e enviamos as mudanças para o repositório remoto:

```
● PS C:\Users\Tomás Nave\mentha_digital> git add .
● PS C:\Users\Tomás Nave\mentha_digital> git commit -m "Descrição de Alteração"
[bTomas1 a93f6a3] Descrição de Alteração
 1 file changed, 8 insertions(+), 8 deletions(-)
● PS C:\Users\Tomás Nave\mentha_digital> git push origin bTomas1
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 539 bytes | 539.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/MentHA-ULHT/mentha_digital
 61dc59c..a93f6a3  bTomas1 -> bTomas1
○ PS C:\Users\Tomás Nave\mentha_digital> []
```

Figura 61 - WorkFlow 5

6. Criar um pull request da branch pessoal para dev:

A partir da interface do GitHub/GitLab, realizamos um pull request para integrar as alterações na dev.

Se houver conflitos, resolvemo-los antes de concluir a fusão.

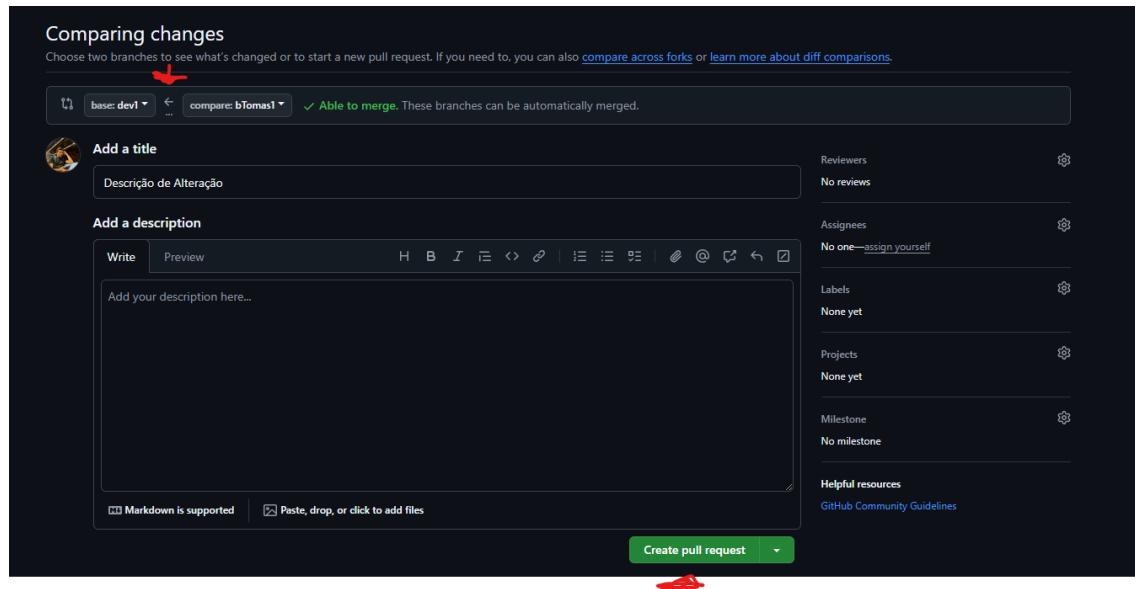


Figura 62 - WorkFlow 6

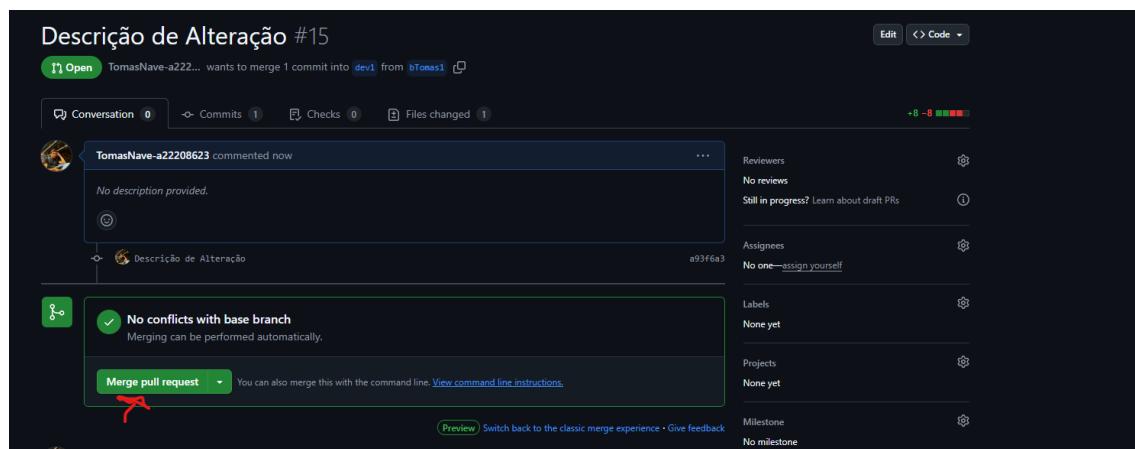


Figura 63 - WorkFlow 7

7. Criar um pull request da dev para main (dev <- Main):

Após a revisão e validação das alterações, submetemos um pull request da dev para a main.

Este processo ocorre geralmente após a reunião semanal, garantindo que o código foi analisado antes de ser integrado na versão final do projeto.

Anexo E - Documentação

A documentação é um dos pilares fundamentais para a continuidade e evolução de um projeto de software. No nosso caso, ao iniciarmos o trabalho neste projeto, rapidamente percebemos a dificuldade de compreender e modificar o código devido à ausência de documentação estruturada. Como este sistema já está em desenvolvimento há vários anos e possui uma base de código extensa, a falta de registos claros sobre o funcionamento das funcionalidades e do fluxo de trabalho tornou o processo de habituação mais demorado e propenso a erros.

E.1 A importância da documentação

Sem uma documentação bem definida, qualquer novo programador que assuma o projeto terá de gastar um tempo considerável para compreender as regras de negócio, a estrutura do código e as melhores práticas adotadas no desenvolvimento. Se este problema persistir, a manutenção e evolução do projeto tornar-se-ão cada vez mais complicadas, podendo levar a retrabalho, perda de informações essenciais e dificuldades no controle do repositório.

O nosso principal objetivo com a padronização da documentação é garantir que o projeto possa continuar a ser desenvolvido ao longo dos próximos anos sem que novos membros da equipa precisem de investir um tempo excessivo na sua compreensão. Para isso, identificámos algumas áreas críticas que precisam de estar bem documentadas:

- **Processo de obtenção do código através do repositório**
- **Workflow correto a ser seguido para evitar problemas durante o desenvolvimento**
- **Documentação de cada funcionalidade diretamente no código**
- **Boas práticas de desenvolvimento dentro do projeto**

Sem um bom controle do repositório e sem documentação adequada, os programadores futuros poderão enfrentar dificuldades significativas, como perda de trabalho, dificuldades em testar novas funcionalidades e até a necessidade de refazer código que já poderia estar implementado. Além disso, a ausência de documentação pode fazer com que tarefas que normalmente demorariam poucas horas acabem por consumir três vezes mais tempo do que o necessário.

E.2 Pesquisa sobre padrões de documentação

Sabendo da importância da documentação, fizemos uma pesquisa sobre diferentes abordagens e ferramentas disponíveis para documentar código e processos dentro do projeto. Durante essa análise, avaliamos tanto métodos tradicionais quanto soluções mais recentes que utilizam inteligência artificial (IA) para automatizar parte do trabalho.

E.2.1 Geradores automáticos de documentação

Essas ferramentas permitem a criação de documentação automática a partir do código, gerando informações detalhadas sobre módulos e classes.

- **Sphinx [SFHX25]:** Muito utilizado em projetos Python, permite estruturar documentação extensa e gerar documentos formatados.

Os pontos positivos e negativos da utilização deste tipo de ferramentas no nosso projeto são:

- **Pontos positivos:** Automação eficiente, documentação sempre atualizada conforme o código evolui.
- **Pontos negativos:** Exigem uma estrutura bem definida desde o início do projeto, algo que o nosso sistema não possui, além de não cobrirem a necessidade de documentar regras de negócio específicas.

E.2.2 Ferramentas de Documentação com IA Integrada

Com os avanços recentes, surgiram ferramentas que utilizam IA para auxiliar na documentação automática do código. Avaliamos algumas opções:

- **QodoAi [QODO25]:** Analisa o código e gera documentação automática com explicações detalhadas. Sugere melhorias e pode gerar testes automáticos.
- **Mintlify [MTFY25]:** Focado em gerar documentação de código baseada em padrões já existentes no projeto. Ideal para projetos grandes.
- **ChatGPT (OpenAI) [CGPT25]:** Pode ser usado para gerar explicações personalizadas do código e até reescrever docstrings com melhor estruturação.

Os pontos positivos e negativos da utilização deste tipo de ferramentas no nosso projeto são:

- **Pontos positivos:** Reduzem o tempo gasto com documentação, sugerem melhorias automaticamente e podem ser integradas ao fluxo de trabalho.
- **Pontos negativos:** Algumas dessas ferramentas não geram docstrings no formato esperado pelo nosso projeto, tornando necessário um trabalho adicional para ajustar as descrições. Além disso, algumas exigem estruturação prévia do código para funcionarem corretamente.

Após avaliar todas as opções, percebemos que muitas delas, apesar de eficientes, não se encaixavam completamente nas nossas necessidades. As ferramentas automáticas de documentação podem ser úteis, mas exigem uma padronização que o nosso código não possui atualmente. Já as soluções baseadas em IA são promissoras, mas nem sempre geram docstrings no formato que estabelecemos para o projeto, o que poderia resultar em retrabalho.

Optámos, portanto, por um modelo de documentação padronizado e simples, garantindo que todas as partes do sistema que desenvolvemos ou alteramos estejam bem descritas para os programadores futuros, sem depender de ferramentas que poderiam introduzir inconsistências no formato utilizado.

E.3 Definição do padrão de documentação adotado

Após a nossa análise, decidimos criar um padrão de documentação que fosse simples, eficiente e fácil de manter. Optámos por documentar cada funcionalidade seguindo um modelo padronizado, que será utilizado para todas as views que desenvolvermos. A **Figura 64**, demonstra o template desenvolvido.

```

#-----#
#           =====( Nome da View/Função )=====
#
# Ultima Alteração: [Data]
#
# Descrição:
#   [Descrição do que a função faz, incluindo o seu propósito e principais funcionalidades]
#
# Input: [Parâmetros de entrada]
# Output: [Resultados esperados]
#
# Módulo: [Módulo onde a função está inserida]
# Relacionado com: [Arquivos ou templates relacionados]
#
# Observações:
#   - [Regras específicas que devem ser seguidas ao modificar a função]
#   - [Dependências e possíveis impactos de alterações futuras]
#
#-----#

```

Figura 64 - Template de Documentação

E.3.1 Explicação do modelo de documentação

Este modelo foi criado com o objetivo de fornecer as informações essenciais sobre cada funcionalidade, sem sobrecarregar a equipa com um excesso de detalhes desnecessários. Aqui estão os principais componentes do template e o seu propósito:

1. **Nome da View/Função:** Define claramente qual parte do código está a ser documentada.
2. **Última Alteração:** Data da última modificação, para facilitar a rastreabilidade das mudanças.
3. **Descrição:** Explica de forma clara o que a função faz e qual a sua utilidade no sistema.
4. **Input e Output:** Informa quais são os dados de entrada e o que a função retorna.
5. **Módulo:** Indica a que parte do sistema esta funcionalidade pertence.
6. **Relacionamento com templates:** Caso a função dependa de templates ou outros arquivos, estes são mencionados.
7. **Observações:** Instruções importantes para evitar problemas ao modificar a funcionalidade, como dependências de outros componentes ou regras de negócio específicas.

E.3.2 Exemplo aplicado

```

#-----#
#           =====( View export_risk_to_csv )=====
#
# Ultima Alteração: 2025-02-06
#
# Descrição:
#   Essa função é usada para gerar um csv e exporta-lo. Este csv contem por linha:
#   - participante
#   - e os dados relativos ao questionário Risk mais recente feito pelo participante.
#
# Input: request
# Output: CSV com o nome "relatorio_risk.csv" é baixado automaticamente
#
# Módulo: `Protocolo MentHA (MentHA EVAL)`
# Relacionado com: `templates/.....`
#
# Observações:
#   - Sempre que forem feitas alterações aos tipos de campos ,
#     mudanças de campos obrigatórios, restrições dos campos do questionário Risk ,
#     consequentemente também na Classe Risk, tem de ser alteradas as verificações nesta função,
#     se não os dados exportados para o csv vão estar incoerentes e errados para futuras possíveis análises
#
#-----#

```

Figura 65 - Exemplo de Documentação

E.4 Documentação de Instalação e Execução

Para evitar que o próximo grupo enfrente os mesmos desafios que encontramos nas semanas iniciais, onde passámos bastante tempo tentando resolver problemas de configuração antes de conseguir executar o projeto, elaborámos uma documentação detalhada de instalação e execução.

O objetivo é garantir que o processo de configuração do ambiente de desenvolvimento e execução seja claro, eficiente e acessível a qualquer programador. Esta documentação oferece um guia passo a passo, desde a obtenção do código no repositório até a execução do sistema localmente, proporcionando uma experiência de configuração mais ágil e sem complicações.



Figura 66 - README Guia de Instalação e Execução

O README do projeto foi estruturado para fornecer todas as informações necessárias para iniciar o desenvolvimento e execução do sistema, abordando os seguintes tópicos:

- **Requisitos de Software:** Detalhamento das ferramentas e versões necessárias para a execução do projeto.
- **Arquitetura Geral do Projeto:** Explicação sobre a estrutura do projeto e das aplicações que o compõem.
- **Base de Dados:** Informações sobre o tipo de base de dados utilizada e o processo de inicialização com dados pré-carregados.
- **Serviços Docker Compose:** Descrição de como o Docker Compose é utilizado para orquestrar os containers e garantir a correta execução da aplicação.
- **Guia completo de Configuração do Projeto Localmente com Docker Compose:** Passos detalhados para configurar e iniciar o projeto localmente, incluindo a configuração de variáveis de ambiente e a inicialização dos serviços.
- **Guia de como fazer Deploy no Servidor (Produção):** Passos para fazer o deploy da aplicação em um servidor de produção.
- **Acesso Online (Versão Produção):** Instruções para aceder a versão online da aplicação após o deploy.
- **Workflow recomendado:** Melhor prática para o uso do controle de versão, incluindo o fluxo de trabalho do Git.
- **Documentação recomendada:** Diretrizes sobre o estilo de documentação a ser seguido durante o desenvolvimento.
- **Observações Importantes:** Recomendações adicionais sobre a manutenção do projeto, como atualização de dependências e ajustes necessários em caso de alterações significativas no projeto.

Este documento pode ser encontrado no repositório do projeto.

E.5 Conclusão Final da Documentação

Ao implementar este padrão de documentação, garantimos que todas as partes do sistema que desenvolvemos ou alteramos estarão bem descritas para os programadores que futuramente assumirem este projeto.

Se este projeto continuar a ser trabalhado sem documentação adequada, os futuros programadores enfrentarão dificuldades cada vez maiores para compreender a estrutura do sistema, podendo gerar atrasos no desenvolvimento, retrabalho desnecessário e até perda de código valioso. Por isso, acreditamos que a adoção deste modelo de documentação ajudará a tornar o desenvolvimento mais eficiente e a garantir a continuidade do projeto de forma sustentável.

Anexo F - Importância do .gitignore

O arquivo `.gitignore` é uma ferramenta essencial para a gestão de projetos rastreados no Git. A sua principal função é evitar que arquivos desnecessários ou problemáticos sejam adicionados ao repositório, reduzindo conflitos e garantindo um histórico de versões mais limpo e organizado.

F.1 Problemas Encontrados no Projeto

O nosso projeto já está no seu quinto ano de desenvolvimento e passou por diversas equipes ao longo do tempo. Infelizmente, algumas dessas equipes não configuraram corretamente o `.gitignore`, o que resultou em uma série de dificuldades que impactaram diretamente a nossa produtividade e a integridade do código.

Desde o início do nosso trabalho, enfrentamos problemas recorrentes ao realizar merges no repositório. Arquivos relacionados à cache da base de dados e modelos do Django eram frequentemente adicionados, tornando os merges de branches um processo problemático, com constantes conflitos e necessidade de correções manuais.

Outro problema crítico foi a versionação de arquivos específicos de configuração local, como bancos de dados SQLite (`db.sqlite3`) e diretórios de cache (`__pycache__/`). Como esses arquivos variam de máquina para máquina, a sua presença no repositório gerou inconsistências entre os diferentes ambientes de desenvolvimento, levando a bugs difíceis de diagnosticar.

F.2 Problemas Comuns em Projetos Django com `.gitignore`

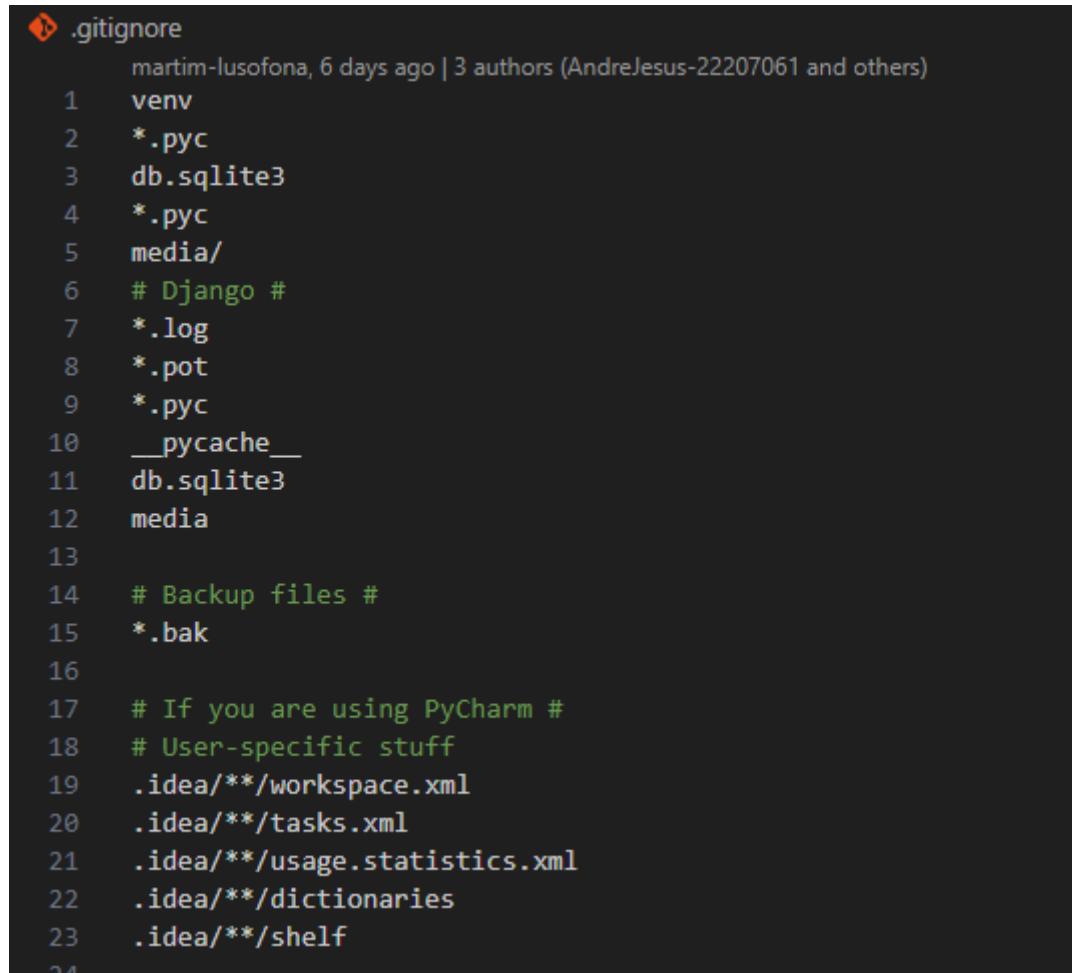
Projetos Django têm requisitos específicos de `.gitignore`, pois geram muitos arquivos temporários e caches. Quando esse ficheiro não está corretamente configurado, problemas comuns incluem:

- **Inclusão de `db.sqlite3`:** Se o projeto utiliza SQLite como banco de dados de desenvolvimento, o arquivo `db.sqlite3` pode acabar por ser commitado, causando inconsistências quando diferentes desenvolvedores trabalham no mesmo repositório.
- **Pastas `__pycache__/` e arquivos `*.pyc`:** Esses arquivos são gerados automaticamente pelo Python e podem ser fonte de conflitos desnecessários.
- **Arquivos de configuração sensíveis:** Arquivos como `.env`, que contêm variáveis de ambiente com credenciais ou chaves secretas, podem acabar accidentalmente no repositório.
- **Arquivos gerados pelo Django:** Diretórios como `staticfiles/` e `media/` podem crescer rapidamente, tornando o repositório desnecessariamente grande.

F.3 Melhor Prática: Utilizar Presets de .gitignore

Para evitar estes problemas, é recomendável utilizar arquivos .gitignore pré-definidos para cada tipo de projeto. O GitHub, por exemplo, disponibiliza um modelo específico para Django.

A **Figura 67**, apresenta uma parte do nosso ficheiro .gitignore.



```
↳ .gitignore
martim-lusofona, 6 days ago | 3 authors (AndreJesus-22207061 and others)
1   venv
2   *.pyc
3   db.sqlite3
4   *.pyc
5   media/
6   # Django #
7   *.log
8   *.pot
9   *.pyc
10  __pycache__
11  db.sqlite3
12  media
13
14  # Backup files #
15  *.bak
16
17  # If you are using PyCharm #
18  # User-specific stuff
19  .idea/**/workspace.xml
20  .idea/**/tasks.xml
21  .idea/**/usage.statistics.xml
22  .idea/**/dictionaries
23  .idea/**/shelf
24
```

Figura 67 - Ficheiro .gitignore

Anexo G – Configuração do Ambiente de Desenvolvimento com Docker e Docker Compose

G.1 Introdução

Durante as fases iniciais do desenvolvimento, deparámo-nos com sérias dificuldades na configuração do ambiente local. A ausência de um guia claro e atualizado tornou a tarefa de colocar o projeto a funcionar localmente num processo frustrante e demorado. O único recurso disponível era um ficheiro README.md desatualizado, que não mencionava aspectos essenciais como a configuração do ambiente virtual, a instalação de dependências ou a inicialização da base de dados.

Esta falta de padronização resultou em inúmeros erros, causados por incompatibilidades entre versões de pacotes, conflitos de dependências e falhas na ligação à base de dados. Um dos principais obstáculos prendeu-se com o ficheiro requirements.txt, que incluía versões desatualizadas de bibliotecas, algumas das quais já obsoletas ou sequer utilizadas no projeto. Isso obrigou-nos a realizar uma análise manual e exaustiva para identificar e remover dependências redundantes ou desnecessárias.

Após várias semanas de tentativa e erro, tornou-se evidente a necessidade de adotar uma abordagem mais robusta e consistente. Foi nesse contexto que optámos por integrar o Docker Compose, como forma de garantir um ambiente de desenvolvimento isolado, replicável e independente da configuração da máquina de cada programador.

G.2 O que é o Docker Compose?

O **Docker Compose** é uma ferramenta que permite definir e gerir múltiplos containers Docker para uma aplicação de forma simples, através de um único ficheiro de configuração (docker-compose.yml). Em vez de iniciar manualmente cada serviço, o Docker Compose permite orquestrar a construção, configuração e execução simultânea de serviços como servidores de aplicação, bases de dados, cache, entre outros.

A sua principal função é facilitar a criação de ambientes consistentes em diferentes máquinas de desenvolvimento, evitando os tradicionais problemas de “funciona na minha máquina”.

G.3 Estrutura e Configuração do Docker Compose no Projeto

No contexto do nosso projeto, um website com três aplicações integradas desenvolvidas em Django, o Docker Compose é responsável por:

1. **Iniciar e configurar a base de dados PostgreSQL**, com persistência de dados.
2. **Executar automaticamente um script de inicialização da base de dados (dump_file.sql)** com dados previamente definidos.
3. **Construir a aplicação Django**, instalar dependências e lançar o servidor de desenvolvimento.
4. **Gerir as dependências entre os serviços**, garantindo que o servidor Django apenas arranca após a base de dados estar funcional.

Serviços definidos:

- **dbpostgresql:** Serviço responsável por executar o container oficial do PostgreSQL, utilizando as variáveis de ambiente definidas no ficheiro .env. Os dados persistem entre reinícios graças ao volume postgres_data.
- **dbpostgresql_init:** Um container temporário usado exclusivamente para importar um ficheiro SQL (dump_file.sql) na base de dados. Este serviço depende do dbpostgresql e apenas é executado após a base de dados estar operacional.
- **web:** O serviço principal do projeto, que constrói a imagem Docker da aplicação Django a partir do Dockerfile. Antes de arrancar o servidor, executa as migrações e cria os modelos necessários nas apps diario, mentha e protocolo.

G.4 Estrutura de Ficheiros Envoltos

Abaixo apresentamos os ficheiros principais que integram a nossa configuração com Docker Compose:

1. docker-compose.yml

Este ficheiro orquestra os três serviços principais (dbpostgresql, dbpostgresql_init e web), define as redes e volumes necessários, e garante que os serviços são lançados na ordem correta.

```

compose.yaml
You, yesterday | 2 authors (martim-lusofona and one other)
1 services:
2   dbpostgresql:
3     image: postgres:12.9
4     container_name: dbpostgresql
5     env_file:
6       - .env
7     ports:
8       - 5432:5432
9     volumes:
10      - postgres_data:/var/lib/postgresql/data
11     networks:
12       - mentha_network
13     healthcheck:
14       test: ["CMD-SHELL", "pg_isready -U $POSTGRES_USER -d $POSTGRES_DB"]
15       interval: 5s
16       timeout: 5s
17       retries: 10
18
19
20   dbpostgresql_init:
21     image: postgres:12.9          You, last week + Adição de ficheiros relativos ao docker
22     container_name: dbpostgresql_init
23     command: psql -h dbpostgresql -d mentha -U leda -f dump_file.sql
24     volumes:
25       - ./dump_file.sql:/dump_file.sql
26     environment:
27       PGPASSWORD: [REDACTED]
28     env_file:
29       - .env
30     networks:
31       - mentha_network
32     depends_on:
33       dbpostgresql:
34         condition: service_started
35
36

```

Figura 68 - Docker compose.yml Parte 1

```

web:
  build: .
  container_name: django_web
  ports:
    - "8000:8000"
  command: >
    bash -c "python manage.py migrate contenttypes
    && python manage.py makemigrations admin auth sessions
    && python manage.py migrate admin
    && python manage.py migrate auth
    && python manage.py migrate sessions
    && python manage.py makemigrations diario mentha protocolo
    && python manage.py migrate diario
    && python manage.py migrate mentha
    && python manage.py migrate protocolo
    && python manage.py runserver 0.0.0.0:8000"
  volumes:
    - ./app
  env_file:
    - .env
  networks:
    - mentha_network
  depends_on:
    dbpostgresql:
      condition: service_healthy
    dbpostgresql_init:
      condition: service_started

networks:
  mentha_network:
    driver: bridge

volumes:
  postgres_data:

```

Figura 69 - Docker compose.yml Parte 2

2. Dockerfile

O Dockerfile define a imagem do serviço web, instalando as dependências via pip e copiando o código-fonte para dentro do container. Inclui ainda os comandos para executar todas as migrações necessárias e iniciar o servidor Django.

```

Dockerfile
You, last week | 1 author (You)
1 FROM python:3.10
2 RUN mkdir /app
3 WORKDIR /app
4 ENV PYTHONUNWRITEBYTECODE=1
5 ENV PYTHONUNBUFFERED=1
6 RUN pip install --upgrade pip
7 COPY requirements.txt /app/
8 RUN pip install --no-cache-dir -r requirements.txt
9 COPY . /app/
10 EXPOSE 8000
11
12 CMD python manage.py migrate contenttypes \
13   && python manage.py makemigrations admin auth sessions \
14   && python manage.py migrate admin \
15   && python manage.py migrate auth \
16   && python manage.py migrate sessions \
17   && python manage.py makemigrations diario mentha protocolo \
18   && python manage.py migrate diario \
19   && python manage.py migrate mentha \
20   && python manage.py migrate protocolo \
21   && python manage.py runserver 0.0.0.0:8000
You, last week • Adição de fich

```

Figura 70 - Dockerfile

3. env

Este ficheiro contém as variáveis de ambiente partilhadas entre os serviços, como o utilizador e palavra-passe da base de dados. O uso deste ficheiro permite manter os dados sensíveis fora do código-fonte.

4. dump_file.sql

O ficheiro SQL utilizado pelo serviço dbpostgresql_init contém os dados base que queremos importar para a base de dados PostgreSQL ao inicializar o ambiente pela primeira vez.

G.5 Conclusão

A adoção do Docker com Docker Compose trouxe uma melhoria significativa na estabilidade e replicabilidade do ambiente de desenvolvimento. Ao resolver problemas relacionados à falta de padronização, ele também reduziu drasticamente o tempo necessário para configurar o projeto em novas máquinas. Com essa abordagem, todos os desenvolvedores trabalham no mesmo ambiente, independentemente do sistema operacional ou das configurações locais, o que resulta em maior produtividade e colaboração entre a equipe.

Essa solução assegura que o próximo grupo que trabalhar no projeto não enfrentará os mesmos desafios que tivemos no início do desenvolvimento. Graças ao Docker, qualquer pessoa que precise transferir o projeto poderá começar a trabalhar imediatamente, sem enfrentar problemas, independentemente do sistema operacional utilizado. Isso garante uma transição mais fluida e uma continuidade mais eficiente no desenvolvimento, tornando o processo mais ágil e sem contratemplos relacionados ao ambiente.