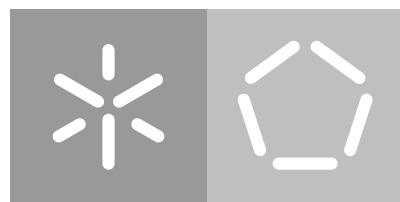


Universidade do Minho
Escola de Engenharia
Departamento de Informática

Alexandre Costa
Etienne Costa
Marcos Silva
Mario Santos
Pedro Costa

Laboratórios de Informática IV
Mentor Me



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Alexandre Costa
Etienne Costa
Marcos Silva
Mario Santos
Pedro Costa

Laboratórios de Informática IV
Mentor Me

Laboratórios de Informática IV
Mestrado Integrado em Engenharia Informática

Trabalho efetuado sob orientação de
Vasco Abelha

RESUMO

Este projeto é referente à Unidade Curricular de Laboratórios de Informática IV, do terceiro ano do Mestrado Integrado em Engenharia Informática da Universidade do Minho, que tem como principal objetivo a realização de uma plataforma de mentoria, com o propósito de minimizar o tempo de espera na obtenção de uma resposta válida bem como a partilha de conteúdos entre os diversos utilizadores, sendo composto por três etapas distintas, de acordo com a metodologia do **Rational Unified Process**.

A primeira destas três etapas (**Fundamentação**) diz respeito à identificação e caracterização geral da aplicação que vai ser desenvolvida. É nesta fase que é feita a fundamentação do seu desenvolvimento e a justificação em termos de modelo de negócio, bem como uma análise da viabilidade do sistema e respectivo plano de desenvolvimento.

A segunda etapa (**Especificação**) corresponde à descrição detalhada do software a desenvolver, utilizando a linguagem de modelação Unified Modeling Language (UML) para construir diversos diagramas como o Diagrama de Use Cases, Diagramas de Sequência, Diagramas de Classes, bem como a respetiva documentação.

A terceira etapa (**Implementação**) diz respeito à execução prática do software, tendo por base o planeamento efetuado nas fases anteriores. Corresponde, assim, à apresentação e caracterização geral da arquitetura do sistema e seus respetivos módulos. O trabalho é distribuído e é seguido o plano desenvolvimento segundo o que foi delineado na primeira fase. As ferramentas previamente acordadas são utilizadas para codificar a aplicação, respeitando os diagramas da segunda fase. A validação da aplicação criada será também incluída nesta última fase.

Área de Aplicação: Desenvolvimento de aplicação web-based , utilizando metodologias da Engenharia de Software.

CONTEÚDO

1	INTRODUÇÃO	1
1.1	Contexto	1
1.2	Apresentação do caso de estudo	1
1.3	Motivação e Objectivos	1
1.4	Estrutura do Relatório	2
2	FUNDAMENTAÇÃO	3
2.1	Definição da entidade de sistema	3
2.2	Justificação, Viabilidade e Utilidade do Sistema	3
3	PLANEAMENTO	5
3.1	Identificação dos recursos necessários	5
3.2	Modelo do sistema a implementar	5
3.3	Definição de medidas de sucesso	5
3.4	Plano de desenvolvimento	6
4	DESENVOLVIMENTO	8
4.1	Levantamento de Requisitos	8
4.2	Requisitos Funcionais	8
4.2.1	Registar Utilizador	8
4.2.2	Autenticar Utilizador	9
4.2.3	Actualizar Dados	9
4.2.4	Criar Sala	9
4.2.5	Publicar Resposta	10
4.2.6	Publicar Pergunta	10
4.2.7	Consultar Ficheiros	10
4.2.8	Adicionar Ficheiros	10
4.2.9	Remover Ficheiros	10
4.2.10	Promover Mentorando	11
4.2.11	Remover Mentorando	11
4.2.12	Validar Resposta	11
4.3	Requisitos Não Funcionais	11
5	MODELO DE DOMÍNIO	12
6	DIAGRAMA DE USE CASES	13
6.1	Especificação de use cases	14
6.1.1	Registrar	14

6.1.2	Autenticar	14
6.1.3	Actualizar Dados	15
6.1.4	Criar Sala	16
6.1.5	Publicar Resposta	17
6.1.6	Publicar Pergunta	17
6.1.7	Consultar Ficheiros	17
6.1.8	Adicionar Ficheiros	18
6.1.9	Remover Ficheiros	18
6.1.10	Promover Mentorando	19
6.1.11	Remover Mentorando	20
6.1.12	Validar Resposta	20
7	ARQUITETURA DA SOLUÇÃO	21
8	ARQUITETURA DA CAMADA DE NEGÓCIOS	23
8.1	Camada de negócios	23
8.2	Dicionário das principais classes	23
8.3	Descrição da arquitetura	23
8.4	Diagrama de classes	24
9	CAMADA DE DADOS	25
9.1	Modelo Lógico	25
9.2	Diagrama do Modelo Lógico	26
10	PROPOSTAS DE INTERFACE	27
11	METODOLOGIA DE IMPLEMENTAÇÃO	30
12	FERRAMENTAS UTILIZADAS NA IMPLEMENTAÇÃO	31
13	DESENVOLVIMENTO DO PROJECTO	32
13.1	Conexão à base de dados	32
14	PRODUTO FINAL	33
15	CONCLUSÃO	38
15.1	Conclusões	38
15.2	Trabalho Futuro	39
A	SUPPORT MATERIAL	41

LISTA DE FIGURAS

Figura 1	Especificação das tarefas	6
Figura 2	Diagrama de Gantt	7
Figura 3	Modelo de domínio.	12
Figura 4	Diagrama de Use cases	13
Figura 5	Use case registar utilizador.	14
Figura 6	Use case autenticar utilizador.	15
Figura 7	Use case actualizar dados.	16
Figura 8	Use case criar sala.	16
Figura 9	Use case publicar resposta.	17
Figura 10	Use case publicar pergunta.	17
Figura 11	Use consultar ficheiros.	18
Figura 12	Use case adicionar ficheiro.	18
Figura 13	Use case remover ficheiro.	19
Figura 14	Use case promover mentorando.	19
Figura 15	Use case remover mentorando.	20
Figura 16	Use case validar resposta.	20
Figura 17	Comunicação entre componentes de um sistema MVC	21
Figura 18	Exibição em cebola da arquitetura limpa	22
Figura 19	Diagrama de classes.	24
Figura 20	Modelo Lógico	26
Figura 21	Homepage	27
Figura 22	Dashboard	28
Figura 23	Room	28
Figura 24	Menu de utilizador	29
Figura 25	Homepage final	33
Figura 26	Homepage mobile	34
Figura 27	Registo de utilizador	34
Figura 28	Janela de aviso	35
Figura 29	Dashboard	35
Figura 30	Sala	36
Figura 31	Definições de membros	36
Figura 32	Upload	37
Figura 33	Resposta por validar	37

Figura 34

Resposta válida

37

LISTA DE TABELAS

Tabela 1	Identidade do Sistema	3
----------	-----------------------	---

1

INTRODUÇÃO

1.1 CONTEXTO

Em qualquer contexto educacional a proximidade entre os mentores e mentorandos influencia profundamente a rapidez e facilidade na aprendizagem, mas muitas das vezes esta proximidade é exclusiva as aulas ou acções presenciais de formação. Devido a isso estima-se que uma percentagem significativa de pessoas recorre a alternativas, como o e-mail, para mitigar lacunas em diversas áreas científicas, não obtendo na maior parte das vezes respostas em tempo útil .

1.2 APRESENTAÇÃO DO CASO DE ESTUDO

O Mentor Me é um software de perguntas e respostas para profissionais e entusiastas em diversas áreas de conhecimento. Foi inicialmente projectado para o contexto académico, mas devido a sua escalabilidade decidiu-se generalizar permitindo um fluxo maior de perguntas e respostas. Este software teve desde sempre os objectivos bem estabelecidos, proporcionando assim uma experiência mais agradável e desafiante aos utilizadores. O grande diferencial entre o Mentor Me e outros softwares semelhantes consiste na possibilidade de integrar várias áreas de conhecimento podendo num curto espaço do tempo consultar conteúdos e efectuar questões obtendo respostas no menor tempo possível.

1.3 MOTIVAÇÃO E OBJECTIVOS

O Mentor Me compromete-se a prolongar essa proximidade entre mentores e mentorandos além da aula presencial, permitindo de forma directa, intuitiva e rápida a colocação de perguntas, disponibilização de respostas, documentos e multimédia em tempo real para todos os utilizadores inscritos numa determinada sala virtual de mentoria. Aos mentores cabe a disponibilização de conteúdos no depósito da sala e a resposta a perguntas colocadas pelos mentorandos, estes que colocam perguntas ou pesquisam perguntas já respondidas.

Um mentorando poderá submeter uma resposta a uma pergunta de outro mentorando, mas esta apenas será considerada válida se for aprovada por um mentor desta sala. A integridade e veracidade da informação são garantidas, o tempo de resposta encurtado e a aprendizagem melhorada.

1.4 ESTRUTURA DO RELATÓRIO

O presente relatório é composto por quinze capítulos:

Numa fase inicial elaboramos sobre a natureza do problema que tentamos solucionar assim como o contexto em que se encontra inserido. No segundo capítulo, Fundamentação, identificamos e caracterizamos a aplicação a desenvolver, bem como a fundamentação do seu desenvolvimento. Posteriormente, fizemos a justificação do projeto em termos de modelo de negócio e analisamos a viabilidade e utilidade do mesmo. Em Planeamento identificamos os recursos necessários e definimos as medidas de sucesso do projeto. Fizemos também um plano de desenvolvimento, com recurso ao Diagrama de Gantt. Em seguida, aborda-se o Levantamento de Requisitos, onde se encontram os requisitos funcionais e não funcionais do sistema e as técnicas utilizadas para os recolher. Em Modelação o sistema é especificado com bastante precisão, estando ilustrado o respetivo modelo de domínio, o diagrama de use cases e a especificação dos mesmos, arquitetura da solução e a arquitetura da camada de negócios (diagrama de classes). No capítulo referente à Camada de Dados, encontra-se o modelo lógico da base de dados, assim como as devidas análises detalhadas. Em Propostas de Interface, encontram-se as mockups para a aplicação web. No capítulo seguinte, Metodologia de Implementação, é exposta a decisão do padrão arquitetural utilizado e a respetiva justificação. Em Ferramentas utilizadas na implementação, apresentamos as ferramentas que recorremos ao longo do desenvolvimento do projeto. No capítulo Desenvolvimento do Projeto explicamos como desenvolvemos algumas das componentes fulcrais do projeto, como por exemplo a conexão à base de dados. No penúltimo capítulo, Produto Final, apresentamos o workflow da utilização da aplicação e demonstramos as funcionalidades implementadas. Numa última fase, Conclusão, é feita uma retrospectiva do todo o trabalho concebido.

2

FUNDAMENTAÇÃO

2.1 DEFINIÇÃO DA ENTIDADE DE SISTEMA

Mentor Me consiste numa aplicação web que permite a usuários cadastrados e de participação activa criarem salas virtuais com o propósito de haver um intercâmbio de conhecimento por intermédio de perguntas e respostas , sendo que o controlo absoluto da sala é de autoria do criador da mesma ou de futuros usuários promovidos.

Ao contrário de aplicações semelhantes o Mentor Me está aberto a todas as áreas de conhecimento, sendo que cada uma das questões só é considerada efectivamente válida caso seja validada por um dos responsáveis da sala.

A utilização da mesma não está sujeita a qualquer tipo de pagamento, mas exige-se um comportamento adequado a todos os utilizadores visto que estão sujeitos a uma suspensão , visando cortar o acesso à respectiva sala bem como os privilégios que tinha na mesma.

Nome	Mentor Me
Requer Pagamento?	Não
Gênero	Q&A
Idiomas	PT - EN
País de origem	Portugal
Lançamento	5 de Julho 2020
Cadastro	Obrigatório
Faixa Etária	13+

Tabela 1: Identidade do Sistema

2.2 JUSTIFICAÇÃO, VIABILIDADE E UTILIDADE DO SISTEMA

O MentorMe vem encurtar distâncias entre mentorados e mentores, criando uma ponte segura de troca de informações validadas relativas a acções de formação de qualquer natureza. Não nos é incomum a presença de ferramentas de partilha de dados em contextos didáticos, mas estas são, na maioria das vezes, unidireccionais. Os mentorandos/alunos/formandos carecem na grande maioria dos casos de um contacto direto, personalizado e organizado com a equipa de mentores, tendo que utilizar ferramentas múltiplas para concluir um simples

contacto de pergunta-resposta, e esta dificuldade extende-se também aos mentores, que ficam impedidos de gerir facilmente as comunicações com as suas turmas. O MentorMe vem solucionar esse mesmo problema, erguendo vias de comunicação bi-direcional facilitada com modelos pergunta-resposta e partilha de ficheiros sempre com a aprovação de quem mais sabe: o mentor.

3

PLANEAMENTO

3.1 IDENTIFICAÇÃO DOS RECURSOS NECESSÁRIOS

Para o desenvolvimento de software é necessário alocar inúmeros recursos até conseguir obter o software pretendido. Relativamente ao Mentor Me foi necessário ter à disposição uma equipa de engenheiros informáticos, sendo que cada um destes contribuiu com o seu tempo de forma a concretizar todas as etapas, desde a especificação até à construção, do sistema de software. Para assegurar a qualidade de todas as fases tivemos o auxílio de um docente universitário que prontificou-se a encaminhar o projecto na melhor direcção. Num contexto empresarial, isto traduz-se na necessidade de um investimento financeiro. No nosso caso, foi possível tirar partido do estatuto de estudante da Universidade do Minho para obter acesso a ferramentas privilegiadas.

3.2 MODELO DO SISTEMA A IMPLEMENTAR

O software desenvolvido teve em consideração a portabilidade total. A aplicação deve ser suportada em qualquer tipo de dispositivo sem comprometer o desempenho ou solidez. Um smartphone, um portátil ou um tablet devem conseguir aceder a todas as funcionalidades sem limitações e de forma exterior ao utilizador. Esta solução foi desenvolvida tendo sempre em vista a melhoria constante da experiência de utilização por parte do utilizador com o sistema, tentando conjugar um design intuitivo com um vasto conjunto de funcionalidades.

3.3 DEFINIÇÃO DE MEDIDAS DE SUCESSO

A métrica de sucesso que, na nossa apreciação, é um indicador significativamente mais informativo que o número de utilizadores ativos, embora este seja também um indicador valioso, é o número de mentorandos promovidos a mentor. Pois com base no crescimento deste valor podemos concluir que há uma participação activa de um determinado número de utilizadores sendo que cumprido um certo objectivo acabam por usufruir de um novo

estatuto e consequentemente de novas funcionalidades. Este crescimento é primordial pois só com o mesmo é que será possível ter um poder de atração superior aos respectivos concorrentes da mesma área.

3.4 PLANO DE DESENVOLVIMENTO

MENTOR ME

MENTOR ME							
* = an automatically calculated cell							
Task Name	Start Date	Day of Month*	End Date	Duration* (Work Days)	Days Complete*	Days Remaining*	Team Member
FASE 1 - ESPECIFICAÇÃO							
Análise dos requisitos	2/13	13	2/15	3	3	0	Pedro Costa
Modelo de domínio	2/15	15	2/16	2	2	0	Etienne Costa
Diagramas de Use Case	2/15	15	2/16	2	1	1	Alexandre Costa
Diagramas de Sequência	2/15	15	2/21	7	3	4	Marcos Silva
Mockup de Interface	2/20	20	2/24	5	1	4	Mario Santos
Diagrama de Classes	2/25	25	2/29	5	0	5	Pedro Costa
Modelo Conceptual da Base de Dados	2/29	29	3/1	2	0	2	Pedro Costa
FASE 2-DESENVOLVIMENTO							
Arquitectura de software	3/2	2	3/4	3	3	0	Etienne Costa
Descrição da modularização	3/5	5	3/6	2	2	0	Alexandre Costa
Implementação da Base de dados	3/7	7	3/18	12	7	5	Mario Santos
Implementação do modelo	3/21	21	4/14	25	10	15	Alexandre e Etienne
Implementação da interface	3/28	28	4/17	21	0	21	Marcos e Mario
Desenvolvimento de teste	/04/2020 00:00:	19	4/21	3	0	3	Todos
Correcção de erros detectados pelos testes	4/20	20	4/23	4	0	4	Todos
Documentação	3/20	20	4/25	37	0	37	Todos

Figura 1: Especificação das tarefas

Com o intuito de preparar e planear o desenvolvimento de todo o projeto, foi desenhado o Diagrama de Gantt acima. Este está dividido em duas partes, Especificação e Desenvolvimento. Para a fase de **especificação** foi dada especial importância aos passos de levantamento e análise de requisitos, diagramas de use case, criação do diagrama de classe e modelação da Base de Dados. Tomamos esta decisão pois achamos que estes modelos são os mais importantes para o sucesso do projeto. Para a fase de **desenvolvimento** optou-se por fazer uma implementação parcial tirando entre 2/3 dias para possíveis identificações e correcções de erros.

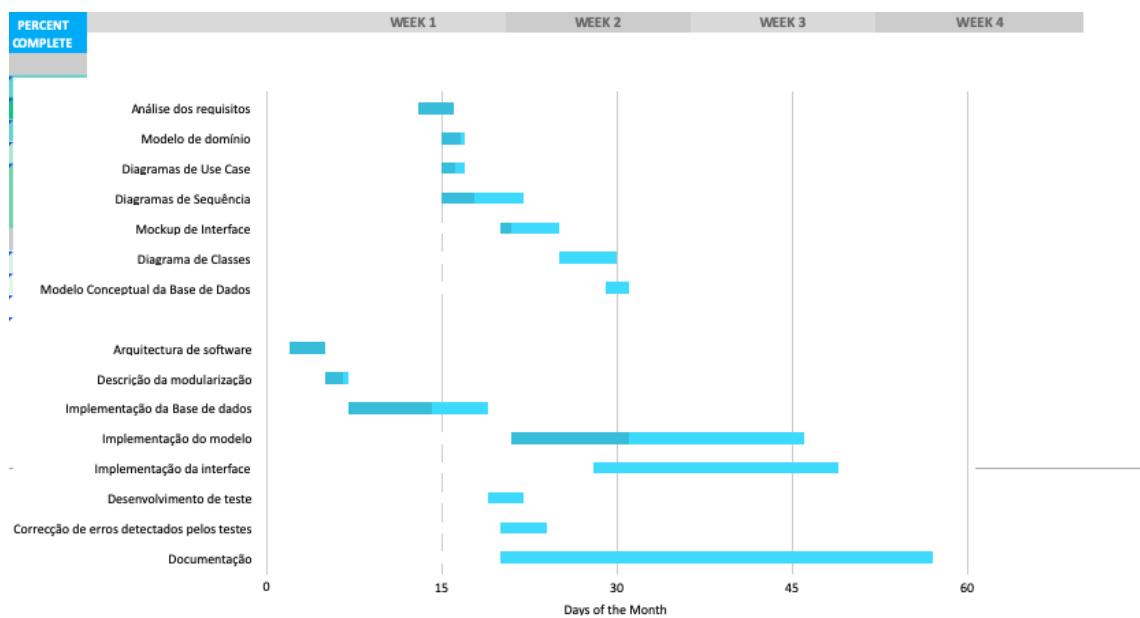


Figura 2: Diagrama de Gantt

4

DESENVOLVIMENTO

4.1 LEVANTAMENTO DE REQUISITOS

De modo a obter as funcionalidades imprescindíveis fez-se um breve inquérito com todas as partes interessadas, sendo que as mesmas procuraram entrar em concesso e de seguida informar a equipa responsável pelo projecto. Sendo assim , passado 3 dias úteis obtivemos a seguinte resposta :

"Pretendemos manter a simplicidade que existe em outras aplicações semelhantes, mas atribuir um toque mais moderno , podendo preservar informação relevante sem que haja demasiada complexidade.

Quanto aos utilizadores, pretendemos que haja uma distinção entre os mesmos e que sejam guardadas as informações básicas, tais como , e-mail , nome , password e o conjunto de salas de acordo com o seu respectivo estatuto.

Em relação as salas gostaríamos de ver presente informações pertinentes , tais como , um nome, uma breve descrição sobre a mesma e ainda ter a possibilidade de disponibilizar ficheiros para facilitar a aprendizagem. Com a intenção de incentivar utilizadores ter um mecanismo para promover podendo assim ter novas funcionalidades de acordo com o desempenho. Sendo o intercâmbio entre utilizadores feito a custa de perguntas e respostas é importante guardar toda informação referente as mesmas bem com a respectiva data, gerando assim uma notificação de modo a perceber o fluxo de informação presente na sala".

4.2 REQUISITOS FUNCIONAIS

4.2.1 *Registar Utilizador*

Definição de requisitos de utilizador

1. O utilizador tem que se registar na aplicação para a poder utilizar.

Especificação de requisitos de sistema

2. Aquando do registo, o sistema deve solicitar os dados necessários ao utilizador, nomeadamente o email, nome de utilizador e password.
3. O sistema não deve permitir o registo de utilizadores com o mesmo email.
4. O sistema deve armazenar os dados do utilizador.

4.2.2 Autenticar Utilizador

Definição de requisitos de utilizador

1. O utilizador deverá introduzir o seu nome de utilizador e a respetiva password para iniciar sessão na aplicação.

Especificação de requisitos de sistema

2. O sistema deve verificar a validade da tentativa de autenticação, verificando se o par username e password inseridos correspondem a algum utilizador existente no sistema. Caso não correspondam, o sistema não deve permitir a autenticação.

4.2.3 Actualizar Dados

Definição de requisitos de utilizador

1. O utilizador deve conseguir editar as suas informações pessoais, nomeadamente o seu email e password.

Especificação de requisitos de sistema

2. O sistema deve armazenar as alterações.

4.2.4 Criar Sala

Definição de requisitos de utilizador

1. O utilizador deverá introduzir o nome, uma breve descrição e um conjunto de tags, sendo este o último opcional.

Especificação de requisitos de sistema

2. Aquando da criação , o sistema deverá validar os dados introduzidos,sendo que em caso de insucesso não deve permitir a criação.
3. O sistema deve armazenar os dados da sala.

4.2.5 *Publicar Resposta*

Definição de requisitos de utilizador

1. O utilizador deve conseguir publicar respostas a uma determinada questão.

Especificação de requisitos de sistema

2. O sistema deve armazenar os dados da resposta.

4.2.6 *Publicar Pergunta*

Definição de requisitos de utilizador

1. O utilizador deve conseguir publicar perguntas numa determinada sala.

Especificação de requisitos de sistema

2. O sistema deve armazenar os dados da questão.

4.2.7 *Consultar Ficheiros*

Definição de requisitos de utilizador

1. O utilizador deve conseguir visualizar todos os ficheiros existentes numa sala.

Especificação de requisitos de sistema

2. O sistema deve apresentar a lista de todos os ficheiros existentes na sala.

4.2.8 *Adicionar Ficheiros*

Definição de requisitos de utilizador

Especificação de requisitos de sistema

4.2.9 *Remover Ficheiros*

Definição de requisitos de utilizador

Especificação de requisitos de sistema

4.2.10 Promover Mentorando

Definição de requisitos de utilizador

Especificação de requisitos de sistema

4.2.11 Remover Mentorando

Definição de requisitos de utilizador

Especificação de requisitos de sistema

4.2.12 Validar Resposta

Definição de requisitos de utilizador

1. O utilizador com o estatuto de mentor deve conseguir validar uma resposta , tornado a mesma **oficial**,i.e, uma das mais correctas no meio de várias.

Especificação de requisitos de sistema

2. O sistema deve permitir ao utilizador validar uma resposta.
3. O sistema deverá guardar os dados.

4.3 REQUISITOS NÃO FUNCIONAIS

- **Requisitos de eficiência :** O sistema deverá processar **n** requisições por um determinado tempo.
- **Requisitos de confiabilidade :** O sistema deverá ter alta disponibilidade, p.exemplo, 99% do tempo.
- **Requisitos de interoperabilidade :** O sistema deverá comunicar-se com a base de dados SQL Server.
- **Requisitos de implementação :** O sistema deverá ser desenvolvido com ferramentas da Microsoft.

5

MODELO DE DOMÍNIO

Tendo por base os requisitos previamente recolhidos e analisados, foi desenvolvido o modelo de domínio que visa recolher e representar de maneira geral os conceitos, comportamentos e algumas interações que estarão incluídas no sistema a desenvolver.

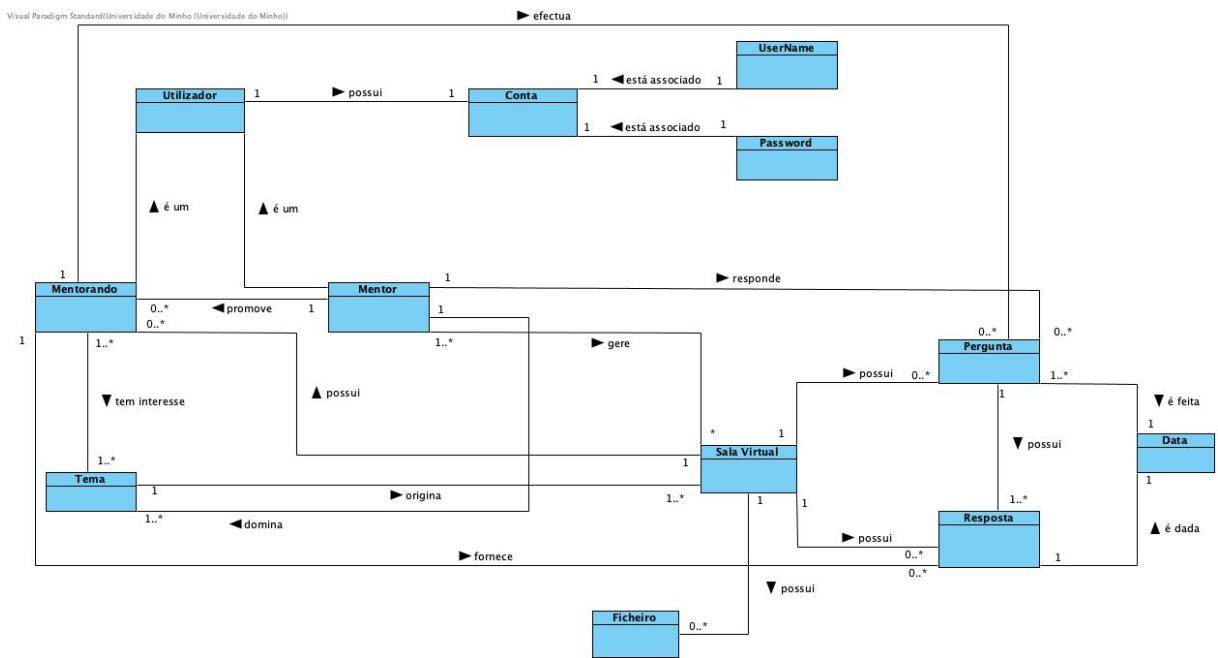


Figura 3: Modelo de domínio.

Numa pequena análise rápida é bastante fácil e intuitivo de perceber que já estão modelados todos os principais objetivos da aplicação nomeadamente a interação Mentor - Mentorando por intermédio de perguntas e respostas.

6

DIAGRAMA DE USE CASES

O actor representado no diagrama como Mentor, comprehende os casos de uso que se enquadram na administração de uma sala. O actor representado como User, abrange a globalidade de todos os outros casos.

Os casos de uso relativos à administração da sala passam por adicionar ficheiros pertinentes às dúvidas colocadas(assim como remover os mesmos), validar respostas realizadas por mentorados e promoção(e demoção) de novos mentores.

O utilizador de baixo estatuto, o mentorado, comprehende os casos intrínsecos em qualquer aplicação como efetuar registo ou logine e alterar os seus dados pessoais, como também as funcionalidades básicas nesta aplicação como criar a sua própria sala, criar as próprias perguntas e responder a outras.

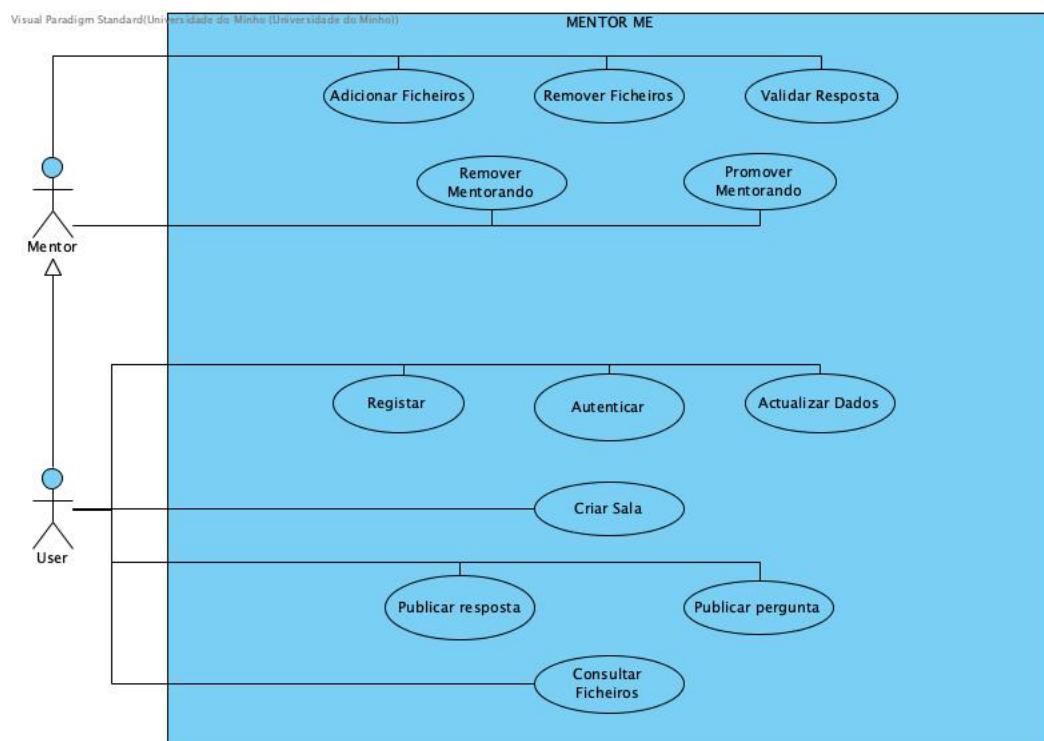


Figura 4: Diagrama de Use cases

6.1 ESPECIFICAÇÃO DE USE CASES

6.1.1 Registar

O use case **registrar** consiste no registo de um utilizador que ainda não se encontra no sistema, sendo o seu registo feito a custa de três parâmetros,tais como, email, nome de utilizador e password. Se ocorrer uma tentativa de inserir um utilizador que já se encontra registado ocorre um erro.

Use Case:	Registrar	
Actor:	User	
Pré condição:		
Pós condição:	Actor registado no sistema	
Cenário Normal	Actor input	System response
	1.Insere dados pessoais	2.Valida dados inseridos 3.Informa que o utilizador foi registado
Excepção [Dados inválidos] (passo 2)		2.1 Informa que os dados são inválidos

Figura 5: Use case registrar utilizador.

6.1.2 Autenticar

Para tirar partido do software o utilizador terá de estar autenticado no sistema, sendo que para isso necessita de indicar o seu nome de utilizador e a respectiva password , que em caso de sucesso é redirecionado para a sua dashboard.

Use Case:	Autenticar	
Actor:	User	
Pré condição:	Actor não está autenticado	
Pós condição:	Actor está autenticado	
Cenário Normal	Actor input	System response
	2. Fornece credenciais	1. Pede credenciais de acesso 3. Valida credenciais 4. Informa que o utilizador está autenticado
Excepção 1 [username inexistente] (passo 3)		3.1 Informa que o username não existe
Excepção 1 [password incorrecta] (passo 3)		3.1 Informa que a password está incorrecta

Figura 6: Use case autenticar utilizador.

6.1.3 Actualizar Dados

O use case **actualizar dados** permite ao utilizador modificar o respectivo email e password, sendo que todos os outros dados manter-se-ão inalterados.

Use Case:	Actualizar dados	
Actor:	Mentor	
Pré condição:	Dados existem no sistema	
Pós condição:	Dados actualizados	
Cenário Normal	Actor input	System response
	1.Seleciona o tipo de dados a actualizar.	2.Apresenta informações dos dados selecionados.
	3.Alterar os campos necessários.	4.Informa que os dados foram actualizados.
Excepção [Dados inválidos] (passo 4)		4.1 Informa que os dados não podem ser inseridos.

Figura 7: Use case actualizar dados.

6.1.4 Criar Sala

O use case **criar sala** é uma das principais funcionalidades de todo o sistema, sendo que para a criação de uma sala é necessário o utilizador estar autenticado podendo de seguida fornecer um nome, uma descrição e um conjunto de tags.

Use Case:	Criar Sala	
Actor:	User	
Pré condição:	Utilizador autenticado no sistema	
Pós condição:	Nova sala presente no sistema	
Cenário Normal	Actor input	System response
	1.Introduz tags e nome da sala	2.Verifica disponibilidade do nome 3.Cria a sala 4.Informa que a sala foi criada
Excepção [Nome indisponível] (passo 2)		2.1 Informa que o nome não se encontra disponível

Figura 8: Use case criar sala.

6.1.5 Publicar Resposta

Este use case permite aos utilizadores responder a questões presentes numa determinada sala.

Use Case:	Publicar resposta	
Actor:	Utilizador	
Pré condição:	Utilizador com acesso à sala	
Pós condição:	Resposta é publicada	
Cenário Normal	Actor input	System response
	1. User clica na questão para responder	2. Sistema carrega janela de resposta
	3. User preenche a resposta	3. Sistema valida a ação 4. Sistema publica a nova resposta

Figura 9: Use case publicar resposta.

6.1.6 Publicar Pergunta

Este use case permite aos utilizadores realizar perguntas a um conjunto de mentores numa determinada sala.

Use Case:	Publicar pergunta	
Actor:	Utilizador	
Pré condição:	Utilizador com acesso à sala na qual a questão é colocada	
Pós condição:	Questão é publicada	
Cenário Normal	Actor input	System response
	1. User clica no botão de nova questão	2. Sistema carrega janela de questão
	3. User preenche a questão	3. Sistema valida a ação 4. Sistema publica a nova questão

Figura 10: Use case publicar pergunta.

6.1.7 Consultar Ficheiros

Permite a um utilizador descarregar ou consultar ficheiros presentes numa determinada sala.

Use Case:	Consultar ficheiros	
Actor:	User	
Pré condição:	Estar autenticado numa sala	
Pós condição:	Ficheiro consultado	
Cenário Normal	Actor input	System response
	1.Pede para consultar listagem de ficheiros.	2.Apresenta listagem de ficheiros.
	3.Insere nome do ficheiro.	4.Apresenta o conteúdo do ficheiro.
Excepção [Ficheiro inexistente] (passo 4)		4.1 Informa que o ficheiro não existe.

Figura 11: Use consultar ficheiros.

6.1.8 Adicionar Ficheiros

Este use case permite a um utilizador com estatuto de mentor, adicionar ficheiros de modo a facilitar o aprendizado de diversas áreas de conhecimento.

Use Case:	Adicionar ficheiros	
Actor:	Utilizador	
Pré condição:	Utilizador estar logado	
Pós condição:	Ficheiro adicionado com sucesso	
Cenário Normal	Actor input	System response
	1.User clica em ficheiros	
		2. Sistema carrega janela de ficheiros
	3. User clica em adicionar ficheiro	3. Sistema abre pop-up
	4. User seleciona ficheiro a carregar	5. Sistema carrega ficheiro

Figura 12: Use case adicionar ficheiro.

6.1.9 Remover Ficheiros

Permite a utilizador com estatuto de mentor, remover ficheiros de uma determinada sala.

Use Case:	Remover Ficheiro	
Actor:	Mentor	
Pré condição:	Utilizador tem privilégios de mentoria no sistema	
Pós condição:	Ficheiro removido do sistema	
Cenário Normal	Actor input	System response
	1.Informa que pretende eliminar um ficheiro	2.Apresenta lista de ficheiros
	3.Insere nome ou id do ficheiro a remover	4.Informa que o ficheiro foi removido
Excepção [Dados inválidos] (passo 4)		4.1 Informa que nome ou id não existe
		4.2 Informa que o ficheiro não foi removido

Figura 13: Use case remover ficheiro.

6.1.10 Promover Mentorando

Este use case permite promover um determinado utilizador para um patamar diferente , fornecendo novas funcionalidades , tais como, validar respostas.

Use Case:	Promover Mentorando	
Actor:	Mentor	
Pré condição:	Utilizador tem privilégios administrativos no sistema	
Pós condição:	Novo mentor identificado no sistema	
Cenário Normal	Actor input	System response
	1.Seleciona que pretende promover um mentor	2.Apresenta lista de utilizadores ativos
	3.Seleciona utilizador a promover	4.Adiciona permissões de mentoria ao utilizador
		5.Notifica novo mentor

Figura 14: Use case promover mentorando.

6.1.11 Remover Mentorando

Este use case permite ao criador de uma respectiva sala remover utilizadores .

Use Case:	Remover Mentorando	
Actor:	Mentor	
Pré condição:	Utilizador tem privilégios administrativos no sistema	
Pós condição:	Mentorando removido da sala	
Cenário Normal	Actor input	System response
	1.Informa que pretende remover mentorando	2.Apresenta a lista de mentornados
	3.Seleciona mentorando a remover	4.Informa que o mentorando foi removido

Figura 15: Use case remover mentorando.

6.1.12 *Validar Resposta*

Este use case permite a utilizadores com estatuto de mentor , validar respostas tornando-as oficiais num conjunto de respostas.

Use Case:	Validar resposta	
Actor:	Mentor	
Pré condição:	Utilizador com permissões de mentor	
Pós condição:	Resposta é validada	
Cenário Normal	Actor input	System response
	1. User seleciona resposta a validar	2. Sistema requer confirmação
	3. User confirma a decisão	3. Sistema valida a ação 4. Sistema exibe resposta validada

Figura 16: Use case validar resposta.

ARQUITETURA DA SOLUÇÃO

Neste projeto trabalhamos unicamente com o serviço ASP.NET Core MVC. Este, tal como o próprio nome indica, baseia-se numa arquitetura MVC dado que permite obter uma separação de conceitos bastante clara. De forma muito simples, os pedidos do utilizador são enviados para um controller, sendo este responsável por trabalhar com a camada de dados para obter o resultado de alguma query ou realizar alguma ação pedida. O controller, por fim, escolhe a view que deve ser mostrada ao utilizador e dá-lhe os dados necessários. De forma ilustrativa, esta arquitetura MVC segue este padrão de comunicação:

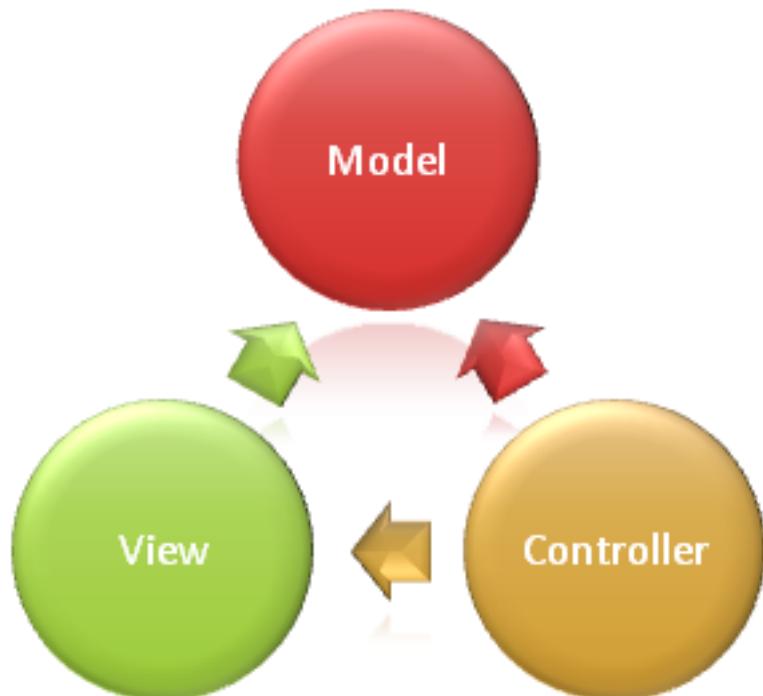


Figura 17: Comunicação entre componentes de um sistema MVC

Esta arquitetura MVC é uma framework construída sobre o ASP.NET Core, um aplicativo web. Este, por sua vez, utiliza uma arquitetura em camadas, sendo esta, especificamente, a arquitetura limpa. Tem, então, três componentes, sendo estas a interface gráfica, a lógica de negócios e o acesso a dados. A arquitetura limpa coloca a lógica de negócios no centro do aplicativo. Nesta arquitetura, em vez de haver uma dependência da lógica de negócios ao acesso a dados, essa dependência é invertida. Pode-se, então, representar esta arquitetura de forma circular, em que as camadas externas dependem do centro.

Clean Architecture Layers (Onion view)

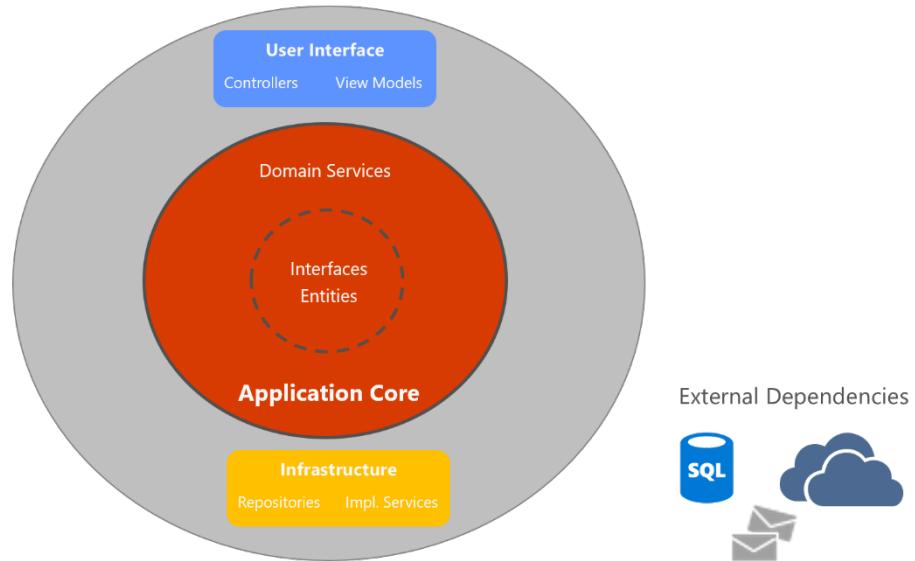


Figura 18: Exibição em cebola da arquitetura limpa

Uma vantagem desta arquitetura é a facilidade de incluir testes unitários no núcleo da aplicação uma vez que não depende de ninguém. O mesmo se aplica à camada de interface, como esta não depende diretamente da infra-estrutura acaba por ser extremamente simples trocar de implementações ou realizar testes de integração à mesma.

8

ARQUITETURA DA CAMADA DE NEGÓCIOS

8.1 CAMADA DE NEGÓCIOS

Como consequência lógica da análise conjunta dos requisitos e do modelo de domínio apresentado, identificamos e caracterizamos as classes e desenvolvemos uma pequena descrição da arquitetura da solução a desenvolver. Esta descrição foi essencial na construção do diagrama de classes representativo da arquitetura da camada de negócios.

8.2 DICIONÁRIO DAS PRINCIPAIS CLASSES

User- Corresponde à representação no sistema dos utilizadores da aplicação e contém a informação pessoal e informação de acesso.

Room- Classe que representa um conjunto de users,tags,files, questions e notifications. Contém também informações mais descritivas , tais como, nome e uma breve descrição.

Tag- É a classe que representa um dos principais parâmetros de pesquisa no sistema.

File- Classe que representa o conteúdo multimédia disponibilizado para utilizadores de uma determinada sala.

Notification- Classe que representa o fluxo de informação entre os utilizadores de uma determinada sala.

Question- Representa uma componente necessária a uma sala, tendo como informação complementar um título, conteúdo, e uma data de realização.

Answer- Representa uma componente necessária a uma questão, tendo como informação complementar um conteúdo e uma data de realização.

8.3 DESCRIÇÃO DA ARQUITETURA

Os utilizadores podem surgir em dois contextos diferentes durante a interação com o sistema , i.e, como mentor após a criação de uma sala ou então como mentorando após entrar em uma sala, tendo por sua vez um conjunto de notificações associadas. As salas por sua vez

têm uma lista de utilizadores, ficheiros, tags e perguntas. As tags presentes representam um elemento essencial para efectuar pesquisas e os ficheiros por sua vez são elementos complementares as perguntas encontradas nas salas. As perguntas possuem um conjunto de respostas associadas podendo perceber aos utilizadores que façam parte da sala aonde foi realizada tal questão.

8.4 DIAGRAMA DE CLASSES

O modelo de classes derivado através da descrição apresentada é o apresentado na figura abaixo. Adicionalmente, às entidades mencionadas foi também criado um facade. A criação deste teve como objetivo, mediar a interação com toda a camada de negócios para que a solução apresentada fosse modular e composicional com um sistema congruente com uma arquitetura de três camadas.

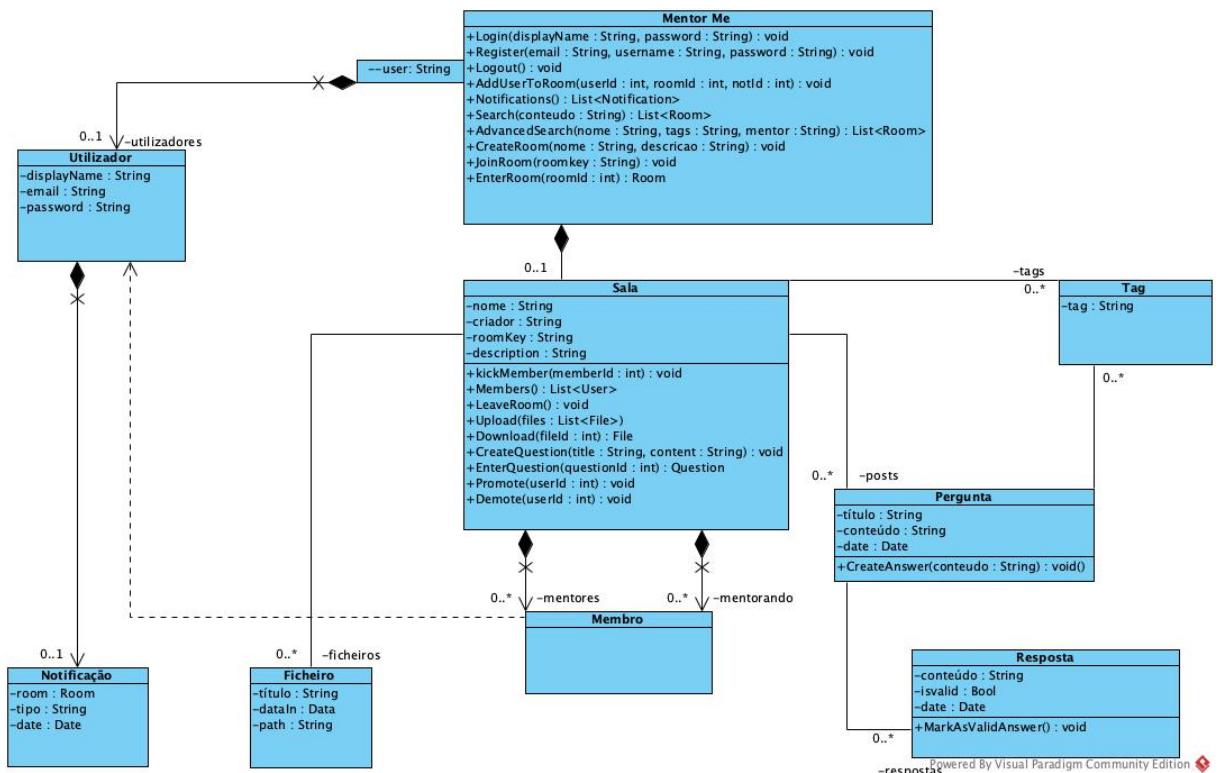


Figura 19: Diagrama de classes.

9

CAMADA DE DADOS

A camada de dados, sendo que suportará a natureza distribuída da solução de software proposta, terá que garantir a consistência, integridade, segurança e acessibilidade aos dados guardados. Por esta razão, decidimos desenvolver uma base de dados num Sistema de Gestão de Base de Dados Relacional.

9.1 MODELO LÓGICO

O modelo lógico apresentado, resulta essencialmente, da estruturação da informação importante a guardar da modelação da camada de negócios apresentada. Para cada uma das principais classes do modelo de classes, mais precisamente, as classes que representam a **Room**,**File**, **User**, **Notification** ,**Tag** ,**Question** e **Answer**, foi associada uma relação. De modo a reduzir a redundância de dados e as possibilidades dos dados se tornarem inconsistentes normalizou-se a base de dados , garantindo um armazenamento consistente e um acesso eficiente aos dados numa base de dados relacional.

9.2 DIAGRAMA DO MODELO LÓGICO

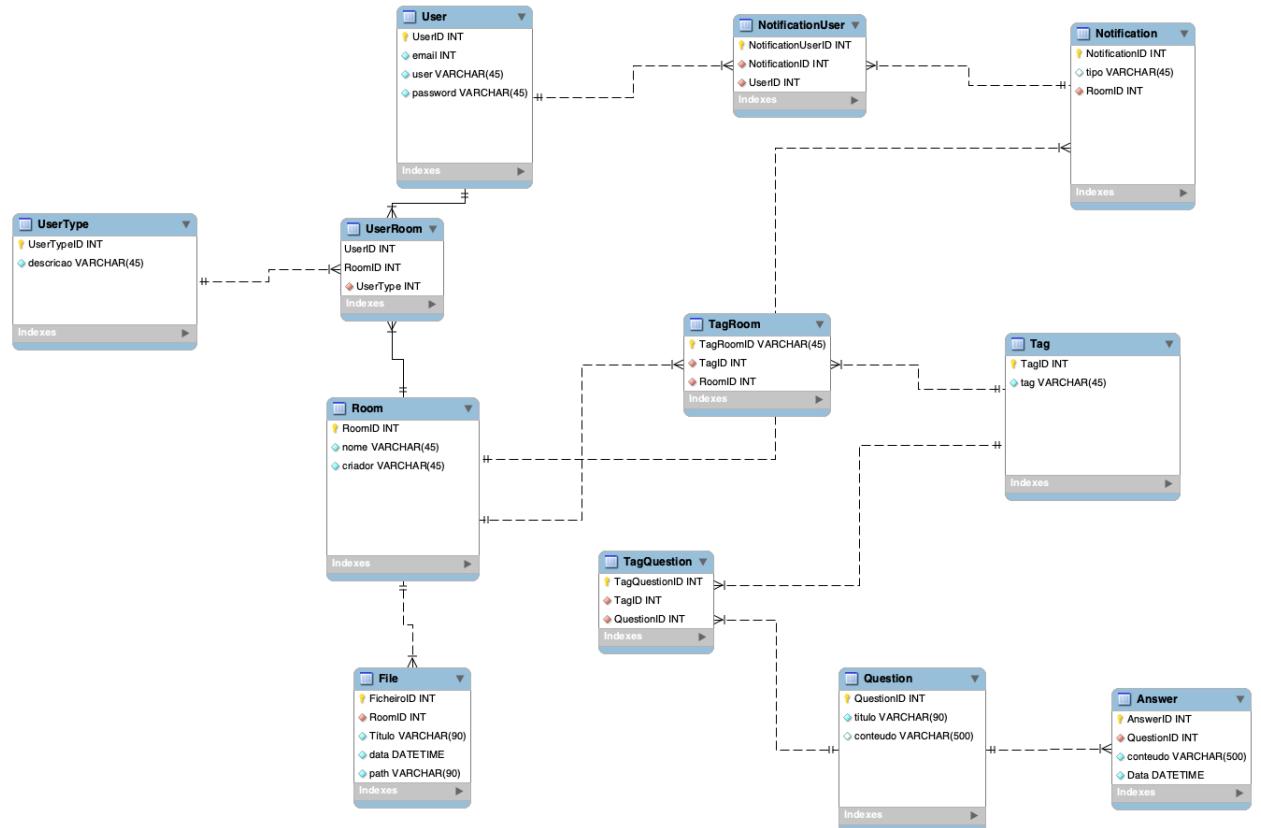


Figura 20: Modelo Lógico

10

PROPOSTAS DE INTERFACE

Parte integrante do planeamento inicial passa por realizar mockups do que considerámos ser a melhor interface de utilizador que permita a melhor expressão e facilite a utilização da aplicação. O fluxo das operações, o uso de linguagem clara e a consistência do design são fundamentais para criar uma interface que o utilizador aprecie e comprehenda facilmente.

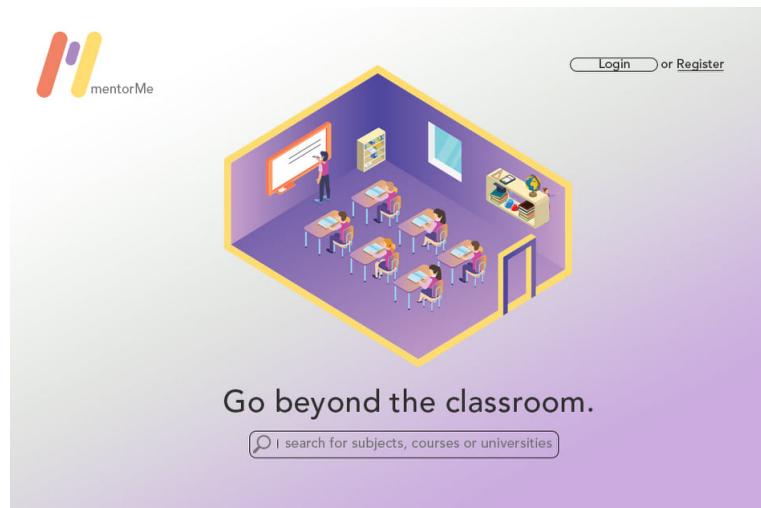


Figura 21: Homepage

A figura acima exibe a página inicial. Destaca-se no centro um elemento ilustrativo seguido de uma barra de pesquisa. No topo, mantendo-se em todas as janelas, está o logótipo da aplicação que referencia sempre a página inicial e os botões de início de sessão e registo de utilizador. Este último elemento metarmofa-se após o login.

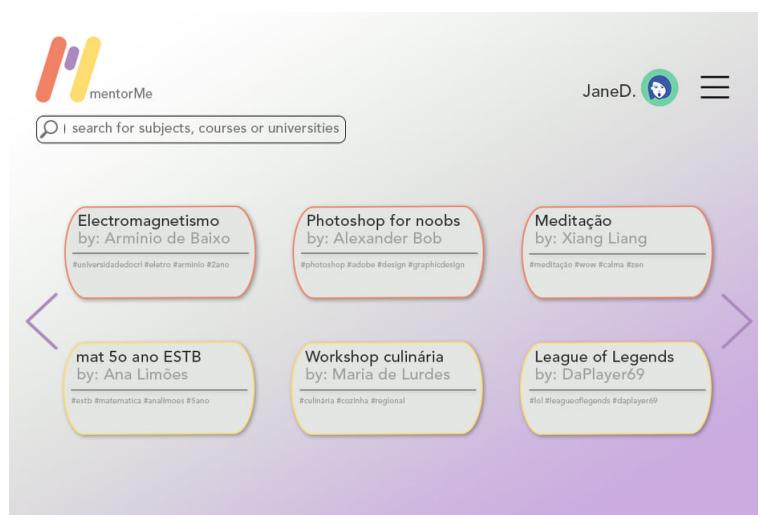


Figura 22: Dashboard

Na figura superior podemos observar o dashboard. Nesta janela são exibidas as salas que o utilizador tem acesso, é possível ver também a barra de pesquisa, o avatar do utilizador e os controladores do carrossel de navegação.

About the Mentor(s):
Alexander Bob
Professor at the Cenas University, North Corea.

About the Room:
Photoshop for noobs
Learn photoshop from scratch with my videos:
youtube.com/abob

Here you can ask me questions about any difficulty you might have, I'll answer them as soon as I can :)

Figura 23: Room

A figura 18 mostra o interior de uma sala. Contém uma breve descrição do mentor e do tema em questão. As dúvidas já colocadas são exibidas, tendo uma hiperligação para as respetivas respostas. À semelhança de outras salas, contém uma barra de pesquisa, neste caso para procurar questões.

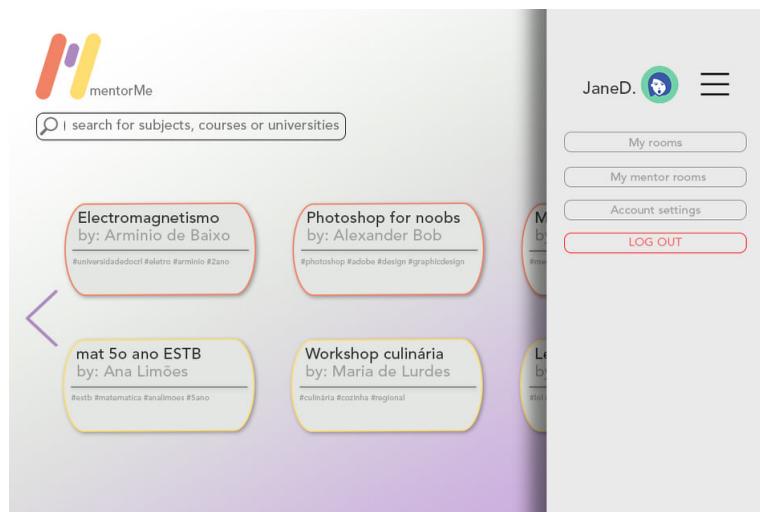


Figura 24: Menu de utilizador

Por fim, nesta maquete apresenta-se o que seria o menu de navegação do utilizador, onde estão presentes ações desde terminar sessão até às definições do próprio.

11

METODOLOGIA DE IMPLEMENTAÇÃO

O padrão arquitetural usado, escolhido devido às necessidades da equipa de obter um conjunto de componentes modulares que facilitam o desenvolvimento em paralelo e asseguram a simples incorporação de testes no decorrer do desenvolvimento, foi o MVC (Model View Controller). Desta forma, o requisito de implementação de uma aplicação multi-plataforma foi extremamente simplificado, dado que a estrutura interna dos dados e procedimentos torna-se independente da interface com o utilizador. Adicionalmente, a componente do modelo foi posteriormente decomposta numa camada de negócios e de dados. Esta decisão foi inteiramente desencadeada pela necessidade de proporcionar a integridade, consistência e disponibilidade dos dados de toda a aplicação de uma forma totalmente independente da tecnologia de base de dados em uso.

12

FERRAMENTAS UTILIZADAS NA IMPLEMENTAÇÃO

Para desenvolver esta aplicação de forma simples e eficiente tiramos partido de muitas frameworks e ferramentas diferentes. Num ponto de vista de desenvolvimento, como já foi mencionado antes, utilizamos a framework ASP.NET Core MVC da Microsoft.

Esta framework está desenhada para tornar extremamente simples e consistente o desenvolvimento de websites dinâmicos, tornando bastante fácil de ter acesso a funcionalidades como upload/download de ficheiros, integração de CSS nas páginas, entre praticamente qualquer outra coisa necessária, isto tudo numa arquitetura com grande nível de abstração.

Num ponto de vista de auxiliar o desenvolvimento em grupo, quisemos tirar partido dum sistema de controlo de versões, tendo optado pelo GitHub. Com vista a reduzir problemas, optamos também por usar todos o mesmo ambiente de desenvolvimento, sendo este o Microsoft Visual Studio.

A nível de base de dados, utilizamos o SQL Server. Para facilitar a comunicação com a base de dados tiramos partido da Entity Framework Core que nos permitiu mapear modelos desenvolvidos no código diretamente para entidades do modelo relacional. Utilizamos o Docker para facilitar a conexão à base de dados entre os vários sistemas operativos que cada um de nós utiliza.

Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web. Visto que é uma parte essencial dos aplicativos da web foram implementadas algumas funcionalidades a custa da mesma, dando destaque a promoção de utilizadores.

13

DESENVOLVIMENTO DO PROJECTO

13.1 CONEXÃO À BASE DE DADOS

Através da framework ASP.NET Core temos alguns ficheiros de configuração que tratam automaticamente de nos ligar à base de dados MentorMe, ou criá-la caso não exista. Com a Entity Framework Core mencionada acima, conseguimos então migrar os modelos definidos nas classes para a base de dados.

Temos também um script de inicialização que foi utilizada para facilitar o teste das diversas funcionalidades implementadas.

14

PRODUTO FINAL

Nas figuras seguintes podemos examinar algumas características da implementação final. Por uma questão de organização, apenas é apresentada uma das janelas em modo mobile.

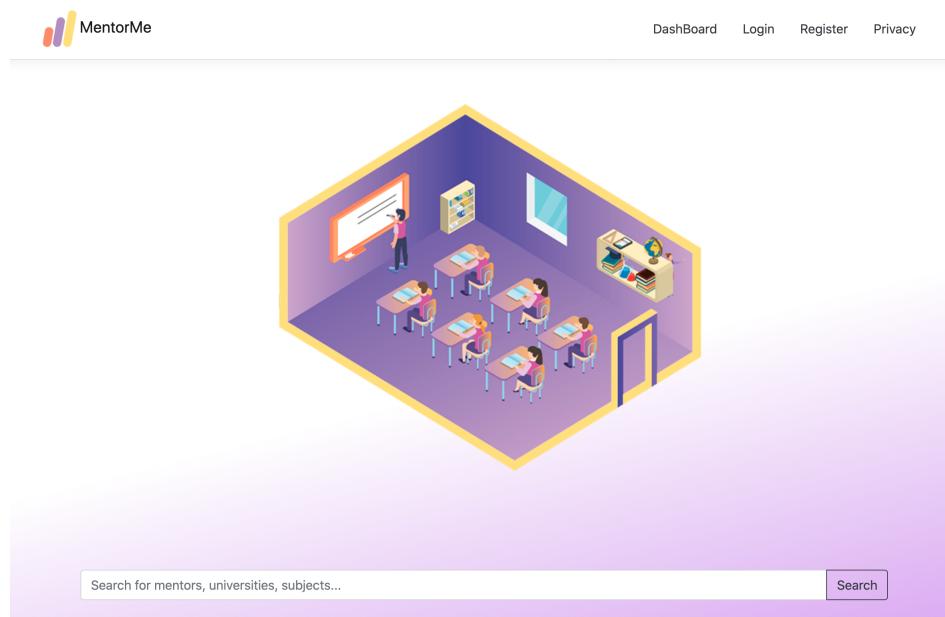


Figura 25: Homepage final

Na página inicial encontram-se os elementos idealizados inicialmente, como a barra de pesquisa e as ligações para navegar pela aplicação.

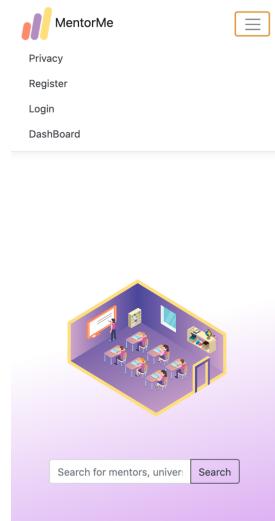


Figura 26: Homepage mobile

Figura 27: Registo de utilizador

O registo de novo utilizador comporta os campos essenciais. Na imagem seguinte encontra-se o pop-up associado a comportamentos de exceção durante o registo.

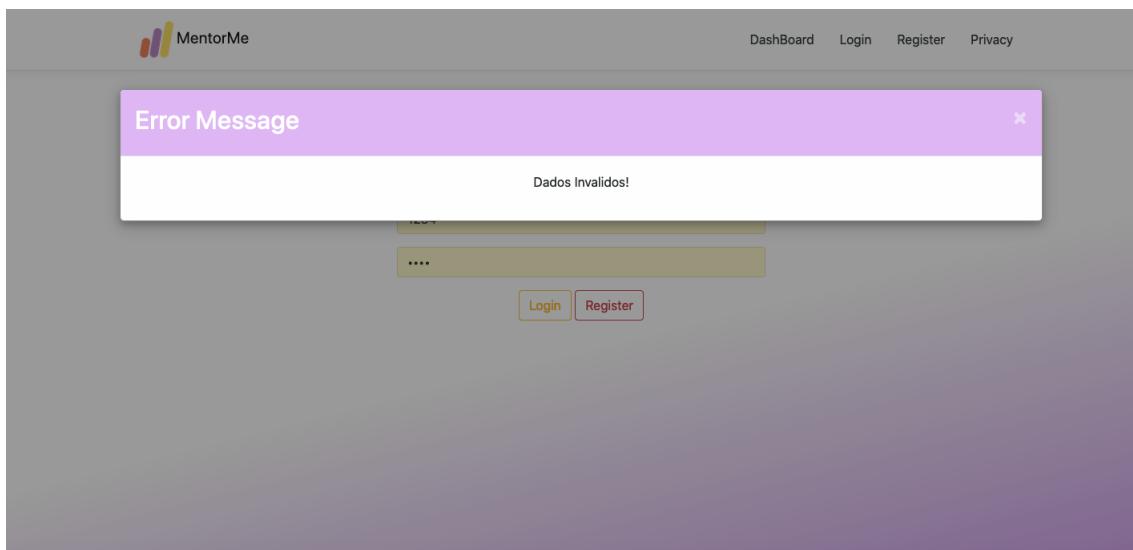


Figura 28: Janela de aviso

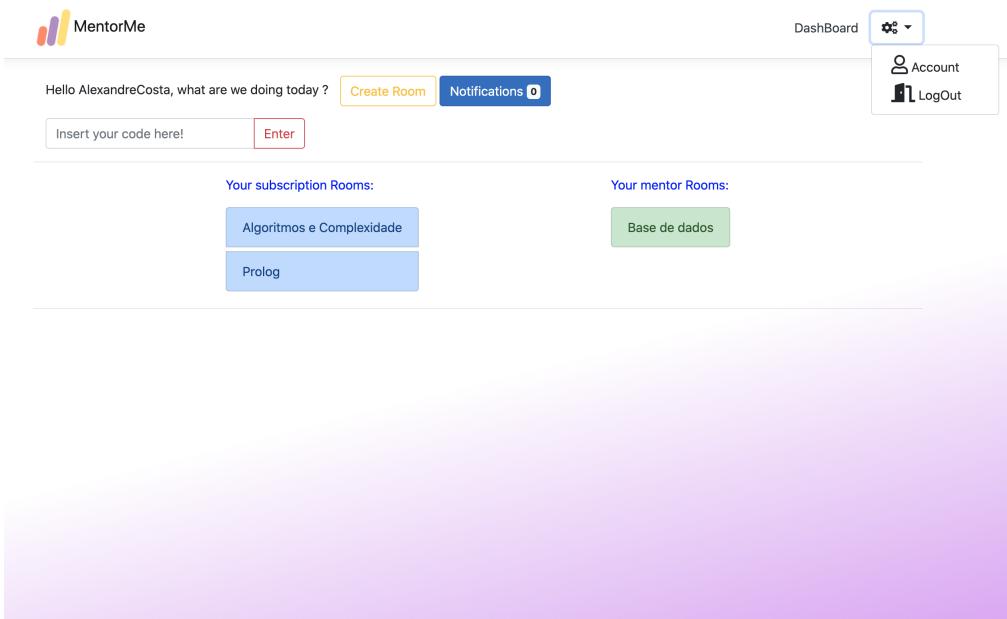


Figura 29: Dashboard

Para aceder ao dashboard é necessário ter a sessão iniciada. Nesta janela é possível criar uma nova sala, aceder às salas cujo acesso já esteja atribuído, consultar as notificações e pesquisar através de um código.

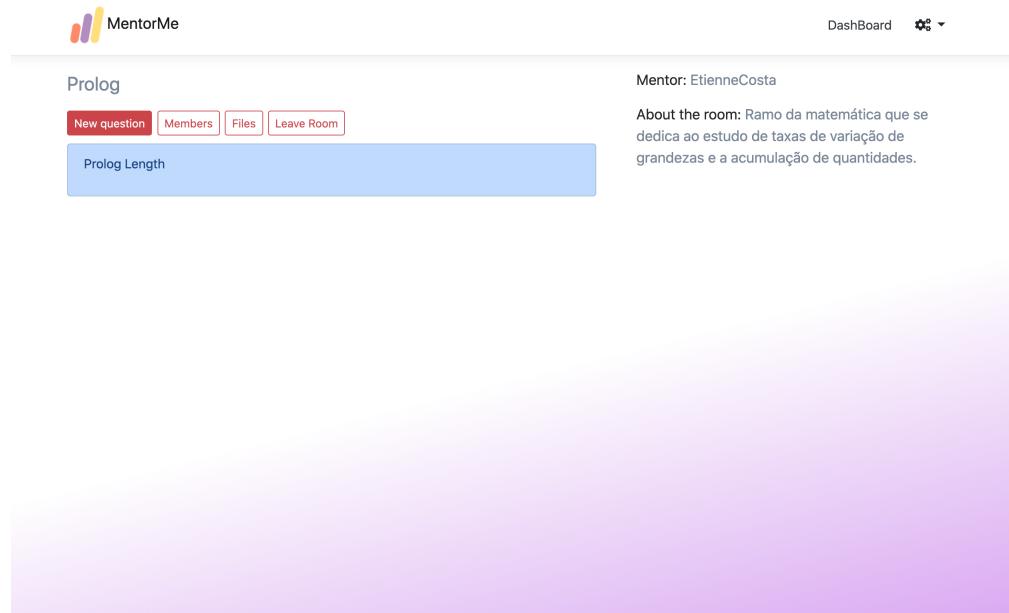


Figura 30: Sala

Já dentro da sala, é possível consultar as perguntas e respostas realizadas, bem como aceder aos ficheiros adicionados.

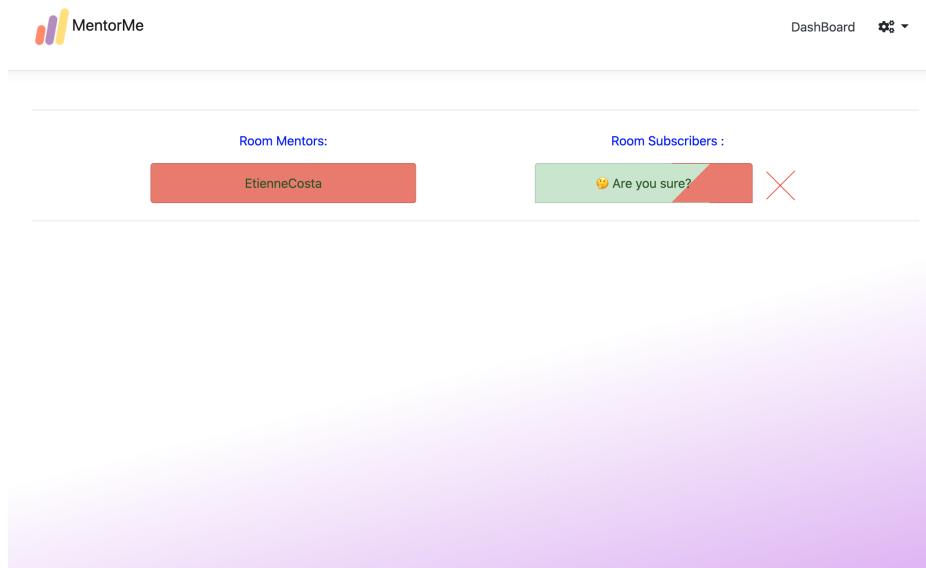


Figura 31: Definições de membros

O mentor da sala pode remover mentorados bem como promovê-los.

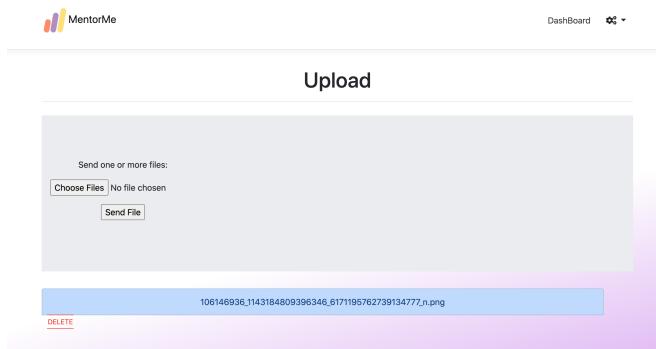


Figura 32: Upload

Seguidamente, na janela de upload de ficheiros é possível ver também os ficheiros já inseridos e a opção de eliminar os mesmos.

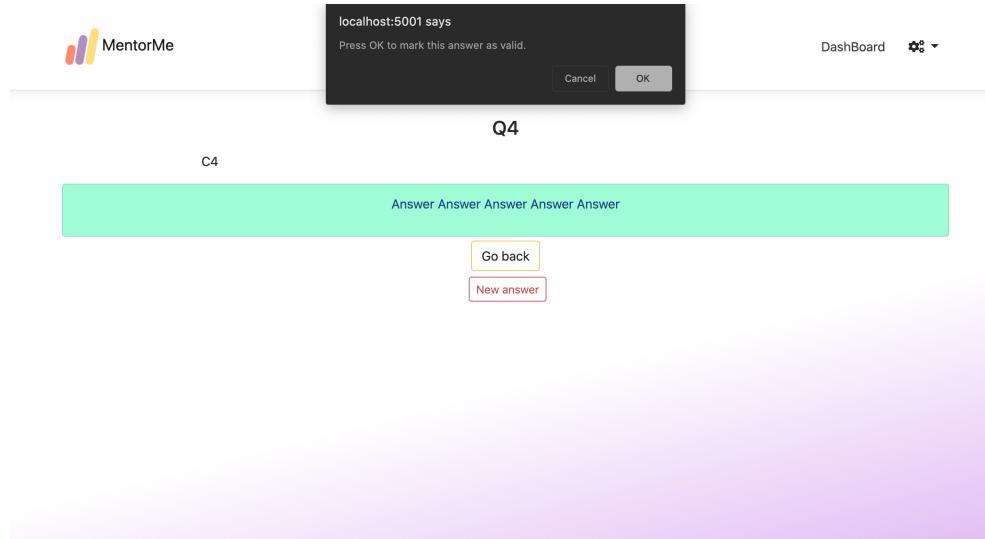


Figura 33: Resposta por validar

Respostas por validar por um mentor da sala encontram-se a verde. Após a validação, estas passam a azul.

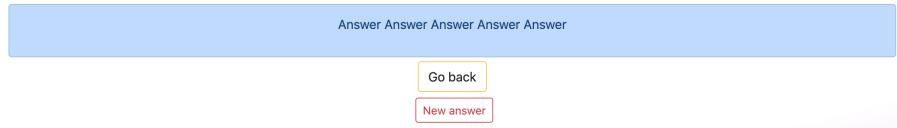


Figura 34: Resposta válida

15

CONCLUSÃO

15.1 CONCLUSÕES

O tempo investido neste planeamento inicial ajudou-nos a perceber em detalhe os pilares fundamentais desta aplicação. Assim, após esta primeira etapa de **fundamentação** do projeto, o grupo passou a ser capaz de:

- enquadrar o projeto no contexto das aplicações móveis/web-based e e saber distinguir-lo das restantes alternativas por força da sua identidade .
- definir claramente os objetivos e as metas que irão ser atingidas com a conceção do projeto .
- justificar a existência da aplicação através do estudo da sua viabilidade e utilidade .
- ter uma ideia da organização geral e concreta do projeto através da sua modelação, bem como dos recursos necessários para a sua conceção.
- planear o desenvolvimento através da divisão das tarefas pelos elementos do grupo ao longo do tempo de vida do projeto .
- definir medidas de sucesso com as quais o grupo irá medir o êxito do resultado final do projeto e sua evolução .

Na segunda etapade de **especificação** procedeu-se à modelação do projecto assente nos princípios ajustados na fase anterior. Elaboraram-se alguns diagramas UML considerados relevantes para a compreensão do sistema a ser desenvolvido, estruturou-se a base de dados sólida e sem redundância de dados e esboçaram-se mockups , idealizando a interface e camadas de apresentação do sistema. Para a terceira e última etapa do projecto , debruçámos sobre a produção do software. Tendo por base todo o trabalho desenvolvido referente à modelação, criou-se uma aplicação Web que considerámos cumprir as diretrizes mais relevantes para os futuros utilizadores.

15.2 TRABALHO FUTURO

Embora os objectivos traçados tenham sido alcançados, o grupo encara com bons olhos a implementação de novas funcionalidades de acordo com a necessidade dos nossos utilizadores mas principalmente focar-se em possíveis gralhas , pois por volta dos anos 60, E. Dijkstra proferiu a seguinte frase, que ficou célebre: "Testing shows the presence, not the absence of bugs ", i.e, o facto do nosso projecto passar nos testes não implica a ausência de falhas ou a impossibilidade de haver melhorias.

BIBLIOGRAFIA

<https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-3.1>

A

SUPPORT MATERIAL
