



Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2020/2021

NewsBook

Be connected to the world.

Ana Filipa Pereira (A89589),

Carolina Santejo (A89500),

Raquel Costa (A89464),

Sara Marques (A89477)

Março, 2021

Data de Recepção	
Responsável	
Avaliação	
Observações	

NewsBook

Be connected to the world.

Ana Filipa Pereira (A89589),

Carolina Santejo (A89500),

Raquel Costa (A89464),

Sara Marques (A89477)

Março, 2021

Resumo

Este relatório retrata não só as várias etapas de todo o processo de desenvolvimento da aplicação “NewsBook”, como também as várias decisões tomadas ao longo deste. O objetivo deste projeto é criar um software de monitorização de eventos, que seja capaz de notificar os seus utilizadores. Para tal, a nossa equipa da *TakingCtrl* pretende desenvolver uma plataforma onde seja possível consultar notícias de várias regiões e ainda receber artigos das mais variadas áreas de interesse.

O trabalho é realizado em três etapas distintas de acordo com a metodologia RUP, e que estão detalhadamente descritas ao longo deste relatório.

A primeira etapa, a Fundamentação, consiste na caracterização da aplicação a ser desenvolvida. Foi feita a apresentação do plano de desenvolvimento, a análise da viabilidade do sistema bem como o estudo das medidas para a satisfação dos objetivos.

A segunda etapa, a Especificação, consiste em materializar as ideias obtidas na primeira fase, havendo assim, uma descrição detalhada do software que irá ser desenvolvido. Recorrendo à linguagem de modelação UML, são feitos vários diagramas, tais como, os diagramas de Use Cases, de packages, de sequência e de classe. Nesta etapa, é feito, também, um planeamento da base de dados a implementar.

A terceira e última etapa, a Implementação, consiste no desenvolvimento do software bem como a implementação das funcionalidades fundamentais, tendo em conta o planeamento das etapas anteriores. É nesta fase que se faz a validação da aplicação e se apresenta o produto final.

Áreas de aplicação: Engenharia de Software, Aplicações Web, App de notícias

Palavras-Chave: UML, RUP, diagramas, planeamento, base de dados, notícias, sistema, Gantt

Índice

Resumo	i
Índice	ii
Índice de Figuras	v
Índice de Tabelas	vii
1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	2
1.3. Motivação e Objectivos	3
1.4. Estrutura do Relatório	3
2. Fundamentação	4
2.1. Justificação do sistema	4
2.2. Utilidade do sistema	5
2.3. Estabelecimento da identidade do projeto	6
3. Planeamento	7
3.1. Identificação dos recursos necessários	7
3.2. Maqueta do sistema	8
3.3. Definição de um conjunto de medidas de sucesso	9
3.4. Plano de desenvolvimento	9
4. Levantamento e Análise de Requisitos	11
4.1. Requisitos Funcionais	12
4.2. Requisitos Não-Funcionais	16
5. Modelo de Domínio	17
6. Diagrama de Use Cases	19
7. Especificação de Use Cases	21
7.1. Registo	21
7.2. <i>Login</i>	22
7.3. <i>Logout</i>	22
7.4. Escolher tipo de Feed	22
7.5. Filtrar publicações por palavra-chave	23
7.6. Filtrar notícias por país	23
7.7. Filtrar notícias por categoria	24
7.8. Ver secção de comentários	24

7.9. Comentar publicações	24
7.10. Eliminar Comentários	25
7.11. Receber alerta de notificações	25
7.12. Aceder ao painel de notificações	25
7.13. Limpar notificações	26
7.14. Guardar uma notícia	26
7.15. Remover uma notícia guardada	26
7.16. Visitar Perfil de Jornalista	27
7.17. Seguir jornalista	27
7.18. Deixar de seguir jornalista	27
7.19. Ver Perfil	28
7.20. Ver notícias guardadas	28
7.21. Configuração Perfil	29
7.22. Publicar artigos	30
7.23. Consultar <i>Feed</i>	30
8. Construção da API da Lógica de Negócio	31
8.1. Métodos da API	31
9. Arquitetura da Aplicação	33
10. Diagrama de Componentes	34
11. Arquitetura da Camada de Negócio	35
11.1. Dicionário das Principais Classes	35
11.2. Diagramas de Classes	36
12. Diagrama de Packages	38
13. Diagramas de Sequência	39
13.1. Validar os dados fornecidos no login pelo utilizador	39
13.2. Validar se um dado <i>userName</i> e email correspondem a algum utilizador já existente	40
13.3. Adicionar novo artigo de opinião de um jornalista ao sistema	40
13.4. Adicionar novo comentário de um utilizador a uma publicação	41
13.5. Adicionar nova notificação à lista de notificações de um utilizador	41
13.6. Adicionar novo utilizador (normal ou jornalista) ao sistema	42
13.7. Configurar atributos de um utilizador normal	42
13.8. Configurar atributos de um jornalista	43
13.9. Obter uma lista de notícias que contém uma dada palavra chave	43
13.10. Adicionar um seguidor à lista de seguidores de um jornalista	44
13.11. Obter uma lista de todos os artigos de um dado jornalista	44
13.12. Obter uma lista de notícias que correspondem a uma das categorias da lista dada para um certo país	45
14. Máquina de Estado	46
15. Arquitetura da Camada de Dados	47

15.1. Caracterização das Entidades	47
15.2. Caracterização dos relacionamentos entre entidades	48
15.3. Diagrama do Modelo Lógico	49
15.4. Estimativa de Crescimento Anual	50
16. Caracterização do Sistema de Interface	54
17. Implementação	62
18. Metodologia de implementação	63
19. Ferramentas utilizadas na implementação	64
20. Desenvolvimento do projeto	65
20.1. Conexão à base de dados:	65
20.2. Pesquisa de notícias:	65
21. Produto Final	66
21.1. Página inicial (Sem sessão iniciada):	66
21.2. Página inicial (Com sessão iniciada):	67
21.3. Perfil do utilizador:	68
21.4. Perfil do jornalista:	69
21.5. Configurações do perfil do utilizador:	70
21.6. Notificações:	71
22. Conclusões e trabalhos futuros	75
Referências	76
Lista de Siglas e Acrónimos	77

Índice de Figuras

Figura 1 - Logotipo da aplicação	6
Figura 2 - Logotipo da <i>startup</i>	6
Figura 3 - Maqueta do Sistema	8
Figura 4 - Especificação de Tarefas	10
Figura 5 - Diagrama de Gantt	10
Figura 6 - Modelo de Domínio	17
Figura 7 - Diagrama de <i>Use Cases</i>	20
Figura 8 - Arquitetura da Aplicação	33
Figura 9 - Diagrama de Componentes	34
Figura 10 - Diagrama de Classes do Subsistema GestPublicacoes	36
Figura 11 - Diagrama de Classes do Subsistema GestUtilizadores	37
Figura 12 - Diagrama de packages	38
Figura 13 - Método <i>validaLogin</i>	39
Figura 14 - Método <i>validaNovoUtilizador</i>	40
Figura 15 - Método <i>addNewArtigo</i>	40
Figura 16 – Método <i>addNewComentario</i>	41
Figura 17 - Método <i>addNewNotificacao</i>	41
Figura 18 - Método <i>addNovoUtilizador</i>	42
Figura 19 - Método <i>configuraPerfil</i>	42
Figura 20 - Método <i>configuraJornalistaPerfil</i>	43
Figura 21 - Método <i>filterNoticiasPalavraChave</i>	43
Figura 22 – Método <i>followJornalista</i>	44
Figura 23 - Método <i>getArtigosJornalista</i>	44
Figura 24 - Método <i>getNoticiasTipo</i>	45
Figura 25 - Modelo máquina de estado	46
Figura 26 - Diagrama do Modelo Lógico	49
Figura 27 - Interface Página de <i>Login</i>	54
Figura 28 - Interface página de registo	55
Figura 29 - Interface <i>HomePage</i>	55
Figura 30 - Interface <i>HomePage</i>	56
Figura 31 - Interface de procura de notícias por palavra-chave	56

Figura 32 - Interface de secção de comentários	57
Figura 33 - Interface de Utilizador a visualizar perfil de jornalista	57
Figura 34 - Interface perfil de utilizador	58
Figura 35 - Interface configuração de perfil de utilizador	58
Figura 36 - Interface <i>HomePage</i> com notificações	59
Figura 37 - Interface configuração de perfil de jornalista	59
Figura 38 - Interface configuração de perfil de jornalista	60
Figura 39 - Interface perfil de jornalista	60
Figura 40 - Interface de adição de um novo artigo de opinião	61
Figura 41 - Login	66
Figura 42 - Registo	67
Figura 43 - <i>HomePage</i> (topo)	68
Figura 44 - <i>HomePage</i> (final da página)	68
Figura 45 - Perfil de utilizador	69
Figura 46 - Perfil de Jornalista	70
Figura 47 - Configurações de perfil	70
Figura 48 - Confirmação de alterações	71
Figura 49 - Notificações	72
Figura 50 - Limpar notificações	72
Figura 51 - Interface artigos (parte 1)	73
Figura 52 - Interface artigos (parte 2)	73
Figura 53 - Interface artigos (parte 3)	74

Índice de Tabelas

Tabela 1 - Identidade do projeto

6

1. Introdução

1.1. Contextualização

Se há algo que podemos afirmar, é que vivemos em plena era digital e que o mundo se desenvolve nas mais diversas áreas a uma rapidez alucinante. Acontecimentos que há meio século atrás, eram lentos e entediantes, atualmente ocorrem quase instantaneamente. Um bom exemplo disto é o acesso a notícias. Hoje em dia, seja com o acesso globalizado à internet, ou através de canais televisivos ou de meios mais convencionais como a rádio, permitem-nos uma contextualização atualizada com o panorama nacional ou mundial em que estamos inseridos, o que promove o desenvolvimento de pensamento crítico em relação a várias temáticas, sejam elas de cariz individual ou da construção social em que vivemos.

Além disto, algo que também é importante é o debate de ideias, de forma a estimular o pensamento crítico de cada um de nós. Enquanto seres humanos temos tendência a formar uma opinião acerca de um dado tema. No entanto, uma das coisas que varia de indivíduo para indivíduo é a maneira de pensar, e torna-se essencial que consigamos entender a opinião dos outros sobre um dado assunto, seja ela semelhante ou diferente da nossa. Desta forma, nós podemos evoluir como seres pensantes e chegar a novas conclusões que podem beneficiar a sociedade.

No entanto, é de realçar que atualmente, um enorme número de pessoas utiliza cada vez mais a web para ler notícias, assinalando assim uma clara transição do papel para o digital. Nos últimos anos tem-se verificado que uma grande parte da população utiliza as redes sociais para se informarem, descartando assim os sites mais tradicionais de notícias, algo que pode parecer vantajoso, mas que na verdade pode ser considerado algo perigoso em muitos dos casos. Segundo um estudo da *Pew Research Center*, um em cada 5 americanos obtém notícias relacionadas com política através das redes sociais e que em 2015 63% dos utilizadores do Twitter e do Facebook utilizam as respetivas redes sociais como fonte primária de informação. Um número que tem vindo a crescer com expressividade nos últimos anos.

Além disso, é necessário ter em consideração que maior parte destas redes sociais utilizam algoritmos que encorajam a adição dos utilizadores às suas plataformas, e, portanto, através destes mesmos, os utilizadores apenas têm acesso a informação muitas das vezes repetitiva e personalizada às opiniões do mesmo, de forma a interessá-los e a captar a sua atenção. Com isto surge um dilema grave na nossa sociedade, uma vez que pessoas sujeitas

ao mesmo tipo de informação e aos mesmos pontos de vista em relação aos mais variados assuntos, não conseguem ter uma visão alargada, nem pensar “fora da caixa”. Estarão as redes sociais a condicionar o nosso conhecimento?

É de notar que com tanta informação que é disponibilizada na web, advém uma problemática pertinente: distinguir o que é real das *fake news*. Este problema surge essencialmente, pelo facto de que qualquer pessoa pode publicar um artigo na internet sem que este seja avaliado ou validado. Constatase, portanto, que a população, apesar de ter cada vez mais facilidade de acesso e disponibilidade de várias fontes de informação, corre o risco de estar mais “desinformada”. “The greatest enemy of knowledge is not ignorance, it is the illusion of knowledge.” – Stephen King.

Assim sendo, a *NewsBook* pretende oferecer aos seus utilizadores uma plataforma de notícias fidedigna, onde terão acesso às notícias das mais diversas categorias. Além de poderem partilhar com outros utilizadores a sua opinião em relação às mesmas, possibilitando-se assim uma plataforma de comunicação com o intuito de expandir os pontos de vista de cada um.

1.2. Apresentação do Caso de Estudo

A *NewsBook* é uma plataforma de publicação de notícias criada pela empresa *startup* de software *TakingCtrl*. Este software é criado de forma a poder ser utilizado por indivíduos que pretendem estar a par dos tópicos mais falados no momento e das notícias mais recentes relativas aos mais diversos temas. É de realçar que a *Newsbook*, lida apenas com notícias cujas fontes são fidedignas, de forma a garantir a sua credibilidade e a combater a propagação da desinformação. Portanto, consideramos que é essencial e fundamental, a variedade e a qualidade de informação que é disponibilizada ao público.

A *TakingCtrl*, traçou desde o início vários objetivos bastante concretos. Em primeiro lugar, a equipa teve em mente proporcionar uma experiência agradável e personalizada aos seus utilizadores, na medida em que se revela uma fusão entre a tradicional app de notícias e uma rede social. Por outro lado, a *NewsBook* dá a conhecer o trabalho de vários jornalistas, fornecendo-lhes uma plataforma onde podem partilhar os seus artigos.

Todas as decisões tomadas durante o planeamento do projeto foram cuidadosamente analisadas e validadas pela *TakingCtrl*, sempre com o desejo de inovação e o interesse do público-alvo em mente.

1.3. Motivação e Objectivos

A principal motivação do projeto é facilitar o acesso a notícias e artigos de várias fontes através de uma única plataforma. Deste modo, os utilizadores terão acesso a notícias sobre vários temas relacionados com os seus interesses de uma forma mais eficiente e organizada. Para além disto, a aplicação permitirá aos utilizadores contactar com diversas opiniões e pontos de vista diferentes, quer seja através de artigos de opinião escritos por jornalistas ou através dos comentários deixados por outros utilizadores.

Visto que todas estas funcionalidades estarão contidas numa única aplicação, o utilizador não terá mais de se preocupar com a consulta de múltiplos sites quer para encontrar uma notícia específica ou para efeitos de fact-checking. A aplicação NewsBook permitirá singularmente que qualquer seu utilizador se consiga manter atualizado sobre qualquer tema que afete o seu país ou o mundo em tempo real – advindo daí o slogan da aplicação “Be connected to the world”.

1.4. Estrutura do Relatório

Nesta secção são referidos os vários capítulos e seus subcapítulos que constituem este relatório.

No capítulo 1, Introdução, é feita uma contextualização do problema, a apresentação da proposta de solução, os seus objetivos e os motivos que levaram á sua escolha.

No capítulo 2, Fundamentação, é feita a fundamentação da aplicação que estamos a desenvolver. É também feita uma justificação do sistema em questão, e uma análise da sua viabilidade e utilidade.

No capítulo 3, Planeamento, são identificados os recursos necessários, é representada a maquete da aplicação e são definidas as medidas de sucesso do projeto. Todo planeamento do trabalho foi feito utilizando um diagrama de “Gantt”.

2. Fundamentação

2.1. Justificação do sistema

O investimento na aplicação *NewsBook* surgiu devido à procura de uma alternativa às redes sociais *Facebook* e *Twitter*, uma vez que todos testemunhamos relatos de familiares, colegas de trabalho ou amigos que afirmam muitas das vezes apenas ter uma conta numa dessas redes sociais para consultar notícias de vários jornais ou os assuntos mais falados do momento.

Devido aos algoritmos presentes nesse tipo de aplicação, as notícias são personalizadas de acordo com o utilizador (seja pela sua localização ou preferências do mesmo) e ainda são apresentadas considerando a sua relevância e popularidade entre outros utilizadores. Sendo que o mais importante é que ambas conseguem acompanhar eventos em tempo real, ou próximo de real, daí esta preferência pelas redes sociais como alternativa aos sites tradicionais de notícias. Além disso, é também possível comentar e debater opiniões com outras pessoas. Consideramos que esta funcionalidade é algo que também agrada ao público-alvo e que não está presente nos sites mais tradicionais.

Vários estudos de diversos países e continentes comprovam que cada vez mais as pessoas utilizam as redes sociais para se informarem. Mas outros estudos revelam, que as pessoas que dependem das redes sociais para se informarem são na verdade as menos informadas e as que estão mais expostas a teorias de conspiração e “fake news”, despoletando a tão falada “desinformação”.

A resposta da nossa *startup* visa evitar e tentar travar as desvantagens resultantes da obtenção de notícias e conhecimento a partir das redes sociais. Daí nós decidimos criar uma aplicação que consiga fundir as funcionalidades presentes nas redes sociais, que mais agradam o público-alvo, com os típicos sites de notícias regionais e nacionais, que consistem em fontes fidedignas de informação.

2.2. Utilidade do sistema

A solução de software contruída pela nossa equipa pretende garantir que todos os utilizadores tenham acesso a informação fiável, às notícias e aos tópicos mais relevantes em todo o mundo, país ou região no momento. Para tal, o utilizador terá sempre acesso às notícias mais recentes e populares em primeiro lugar, podendo filtrá-las consoante o país ou o tema. De modo a tornar a aplicação mais pessoal e personalizável, o utilizador poderá escolher os seus temas preferidos de modo a ser notificado sempre que uma notícia do seu interesse seja publicada. Além disso, como estamos perante uma ferramenta digital, é prioridade que o público consiga tirar partido disso, portanto, a adição de uma secção “media” é fulcral. Com este propósito, inclui-se excertos de Telejornais, para que em poucos minutos, aqueles utilizadores que não conseguem dedicar tanto tempo do seu dia a dia como gostariam, para assistir o Telejornal, consigam atualizar-se. Também incorporamos “podcasts” de vários temas de modo que o utilizador possa obter informação sobre tópicos atuais, enquanto está impossibilitado de interagir com a aplicação, como por exemplo, enquanto estiver a dirigir um automóvel, podendo assim tirar partido da funcionalidade “Bluetooth”, e ouvir os podcasts mais recentes que debatem temas atuais.

Como também é importante expandir o público-alvo, temos também em consideração não só o leitor, mas também o redator, portanto permitimos que certos jornalistas após comunicarem com a *TakingCtrl*, publiquem os seus artigos de opinião em seu nome, e partilhem os seus contactos, para que outros possam entrar em contacto com os mesmos.

Assim sendo, ambicionamos com os vários aspetos mencionados maximizar a interação do utilizador com a aplicação, em diferentes contextos. De modo que a sua vida diária não seja um impedimento para o uso da mesma, e tentando sempre proporcionar informação diversificada e fidedigna. Contornando assim a desinformação gerada nas redes sociais e a passividade integrante dos sites tradicionais de notícias.

2.3. Estabelecimento da identidade do projeto



Figura 1 - Logotipo da aplicação



Figura 2 - Logotipo da startup

Nome	<i>NewsBook</i>
Categoria	Aplicação Web de notícias
Idioma da interface	Inglês
Idioma dos conteúdos	Português, Inglês. (ainda a reconsiderar)
Nome da empresa	<i>TakingCtrl</i>

Tabela 1 - Identidade do projeto

A *NewsBook* é uma aplicação Web de notícias capaz de, não só, monitorar eventos em tempo real, isto é, apresentar as notícias mais recentes e enviar notificações a avisar quando forem publicadas, como também filtrar os conteúdos e apresentá-los de acordo com as preferências do utilizador.

Para além disso, permite também que utilizadores especiais (aos quais lhes vamos chamar jornalistas) publiquem artigos de opinião. Um jornalista pode ser seguido por qualquer utilizador que deseje receber notificação quando seja publicado um seu novo artigo.

3. Planeamento

3.1. Identificação dos recursos necessários

Para a elaboração de todo o projeto foi necessário recorrer a diversos recursos que ajudaram na realização do mesmo.

Antes de mais, o principal recurso utilizado foi a própria equipa da *TakingCtrl* que é responsável por fundamentar, planejar e implementar todas as fases do projeto.

Inicialmente, para a escolha do tema foi necessário garantir que seria possível recolher todos os dados necessários à aplicação, por isso teve de ser feita uma pesquisa das API's disponíveis. Sendo assim, a principal API utilizada foi a *newsAPI* que procura as notícias mais recentes dado um país e uma categoria. Como complemento, para ter acesso a podcasts foi também acedida a API do *Spotify*.

Para garantir sucesso da aplicação foram contactadas diversas pessoas (futuros utilizadores), que dadas as diversas opiniões contribuíram para nos certificarmos de que todas as funcionalidades da aplicação fossem convenientes ao maior número de usuários. Por outro lado, para aceder às estatísticas referidas ao longo do relatório foram feitas pesquisas de diversos estudos realizados.

A plataforma *asana* teve um papel fundamental não só na organização e planeamento de todas atividades (diagrama de "Gantt"), como também na divisão de tarefas pelos elementos da equipa.

A criação de logótipos e maquetes foi feita com recurso às ferramentas *Mockflow*, *Adobe Photoshop*, *TailorBrands* e *Brands.io*.

Por fim, as múltiplas ferramentas da Microsoft tais como o Visual Studio, o Office, SQL Server, etc. contribuíram na implementação do código da aplicação, assim como na realização do relatório (Word) e nas apresentações das fases do projeto (PowerPoint).

3.2. Maqueta do sistema

Como é visível no seguinte esquema, o funcionamento do sistema depende da interação entre o *back-end* (que contém a camada de negócios e de dados) e o *front-end* (que permite a interação entre o utilizador e a aplicação). O *back-end* contém toda a informação necessária ao sistema, e é responsável pela gestão da aplicação e das suas funcionalidades. Por sua vez, o *front-end* exibe todas as informações ao utilizador através da interface, permitindo que este interaja com a aplicação e usufrua das suas funcionalidades.

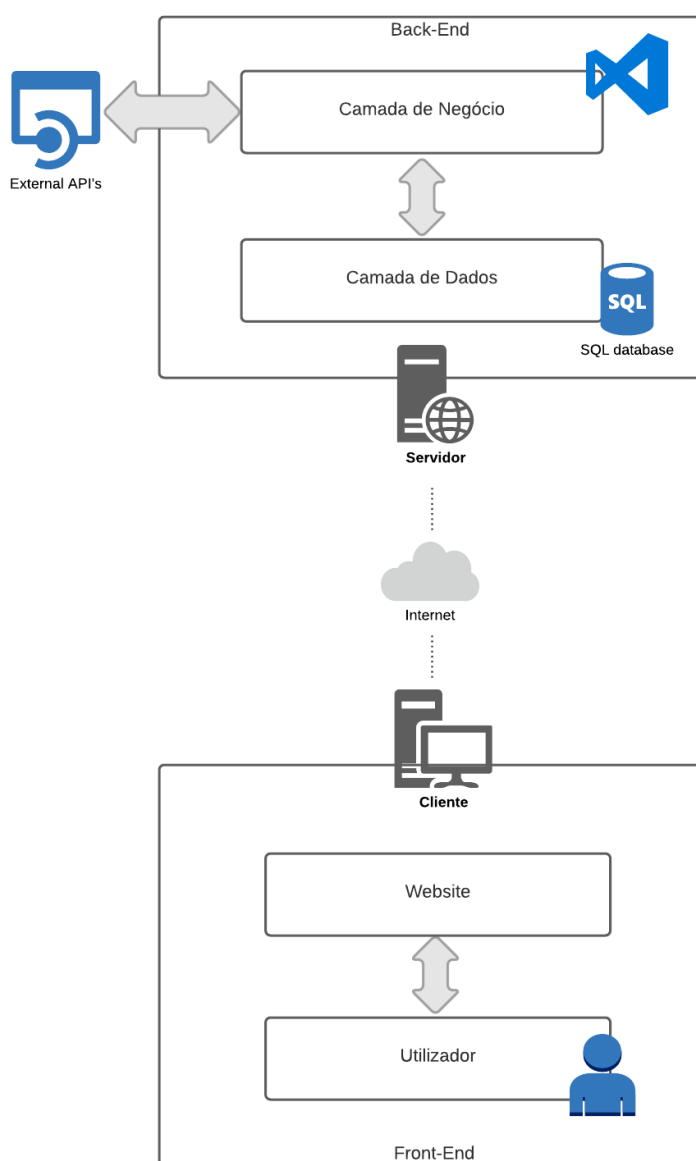


Figura 3 - Maqueta do Sistema

3.3. Definição de um conjunto de medidas de sucesso

De forma a avaliar o êxito do produto final deste projeto, foram definidas as seguintes de medidas de sucesso:

- Em primeiro lugar, é necessário garantir a publicação dos conteúdos o mais perto do tempo real possível. Uma vez que esta é a principal função do projeto, a sua implementação será um dos principais fatores de sucesso.
- Garantir a flexibilidade do sistema. Por exemplo, a existência de diversidade de idiomas implica a globalização da aplicação, o que leva a que seja utilizada por mais pessoas.
- A aplicação ser um “instrumento pessoal”, isto é, apresentar conteúdos específicos de acordo com as preferências de cada pessoa.
- Elaboração de uma interface simples e intuitiva para fornecer ao cliente uma utilização agradável e pouco confusa.
- Por último, é fulcral assegurar o rápido e adequado funcionamento de todas funcionalidades implementadas evitando a existência de bugs no sistema.

3.4. Plano de desenvolvimento

De modo a preparar o processo de desenvolvimento do projeto e a coordenar a equipa, foi necessário traçar um plano, e desenhar o Diagrama de *Gantt*. Primeiro, houve uma especificação de tarefas de modo a percebermos quem ficaria responsável e qual o tempo que poderíamos disponibilizar para cada tarefa, tendo em conta que é fundamental o cumprimento de prazos.

Tal como referido, este projeto está dividido essencialmente em 3 fases. A fundamentação, sendo que esta já se encontra cumprida, a especificação e, finalmente, a implementação.

Em relação à segunda fase, que consideramos de extrema importância, é onde é realizado o levantamento de requisitos, a modelação de diagramas UML, tais como, o de *Use Cases*, de *Classes*, de *Packages* e entre outros. Além disso, ainda é feita a modelação da Base de Dados. Na nossa visão estes passos ditam o sucesso do nosso projeto, daí investirmos bastante tempo neles.

Na terceira fase, é onde iremos implementar a nossa aplicação. Aqui damos uma posição de destaque à fase de testes, e ainda às tarefas correspondentes à correção de eventuais erros e de ajustes finais que sejam necessários aplicar. Para tal é necessário dispendir algum tempo, para nos certificarmos que o produto final corresponde às nossas expectativas, e à do público-alvo.

Nome da tarefa	Responsável	Data de conclusão	Status	
▼ Fundamentação - Fase 1				
✓ Troca de ideias acerca do tema do trabalho 1		8 – 10 mar	Em dia	
✓ Criação do ficheiro excel para comparação de ideias		11 mar	Em dia	
✓ Preenchimento da folha excel		11 – 13 mar	Em dia	
✓ Decisão final do tema do projeto		14 mar	Em dia	
✓ Criação do logotipo da aplicação		14 mar	Em dia	
✓ Criação da maquete		14 mar	Em dia	
✓ Delegação das tarefas		15 mar	Em dia	
✓ Redação do relatório relativo à fase 1	EC Equipa Cyberry	16 – 21 mar	Em dia	
Adicionar tarefa...				
▼ Especificação - Fase 2				
✓ Levantamento de requisitos	ES Engenheiros(as) de software	28 mar – 4 abr	Em dia	
✓ Análise dos requisitos levantados	PR Programador(as)	30 mar – 6 abr	Em dia	
✓ Criação do modelo de domínio	ES Engenheiros(as) de software	7 – 9 abr	Em dia	
✓ Criação do diagrama de Use Cases	ES Engenheiros(as) de software	10 – 12 abr	Em dia	
✓ Especificação dos Use Cases	PR Programador(as)	13 – 16 abr	Em dia	
✓ Criação do diagrama de classes	PR Programador(as)	17 – 21 abr	Em dia	
✓ Criação dos diagramas de packages	ES Engenheiros(as) de software	20 – 22 abr	Em dia	
✓ Criação dos diagramas de sequência	PR Programador(as)	21 – 26 abr	Em dia	
✓ Modelação da base de dados (modelo conceptual e lógico)	ES Engenheiros(as) de software	26 – 29 abr	Em dia	
✓ Redação do relatório relativo à fase 2	EC Equipa Cyberry	1 – 10 Maio	Em dia	
Adicionar tarefa...				
▼ Construção - Fase 3				
✓ Implementação e povoamento da base de dados	PR Programador(as)	12 – 15 Maio	Em dia	
✓ Criação do modelo máquina de estado	ES Engenheiros(as) de software	13 – 15 Maio	Em dia	
✓ Implementação do modelo	PR Programador(as)	16 – 22 Maio	Em dia	
✓ Planeamento da User Interface	ES Engenheiros(as) de software	13 – 18 Maio	Em dia	
✓ Desenvolvimento da User Interface	PR Programador(as)	19 – 23 Maio	Em dia	
✓ Fase de testes	DE Debugger(s)	24 – 30 Maio	Em dia	
✓ Documentação do programa/aplicação	PR Programador(as)	16 – 31 Maio	Em dia	
✓ Redação do relatório relativo à fase 3	EC Equipa Cyberry	2 – 7 Jun	Em dia	
✓ Correção de bugs e ajustes	DE Debugger(s)	24 – 30 Maio	Em dia	
✓ Finalização da aplicação	EC Equipa Cyberry	30 Maio – 1 jun	Em dia	
Adicionar tarefa...				

Figura 4 - Especificação de Tarefas

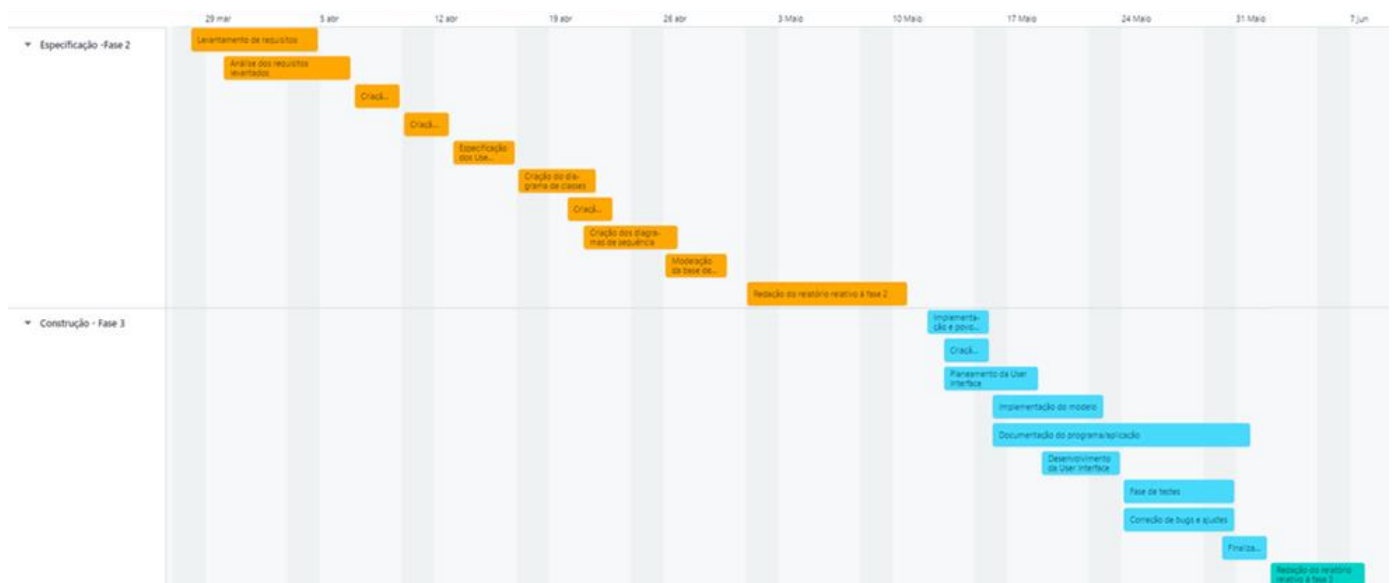


Figura 5 - Diagrama de Gantt

4. Levantamento e Análise de Requisitos

A primeira tarefa a desenvolver na Fase de Especificação, isto é, na segunda fase do projeto, consistiu em realizar um levantamento dos requisitos necessários para a implementação da aplicação visionada, e, posteriormente, uma análise dos mesmos.

Para tal, foi fundamental recorrer-se a diversas metodologias, de forma a não só recolher o máximo de informação possível sobre as especificações e funcionalidades que a aplicação deverá ter, como também a obter opiniões variadas, com o intuito de abranger possíveis casos que o programa deverá implementar.

Primeiramente, foram feitas várias entrevistas a pessoas conhecidas que revelam usar as redes sociais como fonte principal de informação, tais como o *Twitter*, *Facebook* e *Reddit*. Como o objetivo é criar uma aplicação que consiga combater a desinformação que circula neste tipo de redes, é fulcral aliciar esses mesmos utilizadores a optarem pela aplicação “NewsBook”. Para tal, foi necessário realizar uma série de questões para descortinar as funcionalidades que o público-alvo gostaria de encontrar ao deparar-se com um site tradicional de notícias, e que atualmente não encontra.

Após as entrevistas, elaborou-se uma pesquisa aprofundada sobre as várias plataformas de jornais públicos tipicamente utilizadas, em contraste com as redes sociais, analisando as funcionalidades e critérios de cada uma. Assim sendo, foi possível distinguir e destacar as principais características de cada tipo de plataforma, para que o resultado final seja uma conjunção dos melhores atributos levantados com base nas entrevistas realizadas anteriormente. Permitindo assim, obter um produto final que é do agrado do público-alvo.

4.1. Requisitos Funcionais

1) Registo na aplicação

➤ **Requisitos de Utilizador:**

- É necessário efetuar registo na app obrigatoriamente para a utilizar;

➤ **Requisitos de Sistema:**

- O sistema deverá disponibilizar ao utilizador a opção de se registar na plataforma como jornalista;
- O sistema deverá solicitar *email*, *username* e *password* ao utilizador;
- Além disso deverá solicitar outras informações pessoais, tais como: país onde vive, cidade, categorias favoritas, primeiro e último nome, e ainda uma foto de perfil caso o utilizador queira.
- Não é permitido o registo de utilizadores com o mesmo email;
- Caso o email já exista, deverá enviar um aviso ao utilizador;

2) Autenticação na aplicação

➤ **Requisitos de Utilizador:**

- O utilizador deverá introduzir o seu *username* e *password*, de forma a poder aceder à aplicação.

➤ **Requisitos de Sistema:**

- Validar os dados introduzidos pelo utilizador, de acordo com as informações armazenadas na base de dados.
- Caso os dados não sejam válidos, deverá comunicar uma mensagem de erro.

3) Configuração de Perfil

➤ **Requisitos de Utilizador:**

- Deverá conseguir configurar ou editar as suas informações pessoais, isto é, todos os dados que forneceu inicialmente quando realizou o registo na aplicação.
- Caso o utilizador seja um jornalista, terá então ainda a possibilidade de acrescentar novas informações ao seu perfil na aplicação, tais como uma pequena biografia, e ainda o seu contacto de *LinkedIn*.

➤ **Requisitos de Sistema:**

- O sistema deverá armazenar todas as alterações feitas na sua base de dados.

4) Acesso ao Perfil

➤ Requisitos de Utilizador:

- O utilizador deverá conseguir visualizar uma página com todos os conteúdos relativos a si.

➤ Requisitos de Sistema:

- O sistema deverá apresentar as informações pessoais do utilizador: *username*, primeiro e último nome, foto de perfil, categorias preferidas, país e cidade onde vive.
- Apresentar os jornalistas que o utilizador segue
- Apresentar as notícias guardadas pelo mesmo

5) Visualizar perfis de jornalistas

➤ Requisitos de Utilizador:

- O utilizador poderá consultar todos os artigos escritos pelo jornalista em questão

➤ Requisitos de Sistema:

- Deve apresentar as seguintes informações referentes ao jornalista: *username*, primeiro e último nome, foto de perfil, categorias preferidas, biografia, contacto, país e cidade onde vive.
- E, ainda, todos os artigos escritos e publicados pelo jornalista

6) Seguir jornalistas

➤ Requisitos de Utilizador:

- O utilizador deverá poder seguir um jornalista de forma a receber notificações assim que o mesmo publica um artigo

➤ Requisitos de Sistema:

- Deverá armazenar na sua base de dados o utilizador como seguidor do jornalista em questão.

7) Feed de notícias personalizadas

➤ Requisitos de Utilizador:

- O utilizador deverá poder observar todas as notícias referentes ao país onde vive e de acordo com as categorias que selecionou como preferidas.

➤ Requisitos de Sistema:

- Deverá conseguir dispor as notícias de acordo com as preferências do utilizador e por ordem cronológica.

8) Feed de *Trending News*

- **Requisitos de Utilizador:**
 - O utilizador deverá poder visualizar todas as notícias mais faladas no momento em todo o mundo
- **Requisitos de Sistema:**
 - O sistema deverá apresentar as notícias de acordo a popularidade das mesmas

9) Filtrar notícias no feed *Trending News*

- **Requisitos de Utilizador:**
 - O utilizador deverá poder seleccionar uma categoria e um país dentro daqueles apresentados, de forma a filtrar as notícias.
- **Requisitos de Sistema:**
 - O sistema deverá dispor ao utilizador as opções de categoria e de país possíveis.
 - O sistema deverá apresentar as notícias conforme os filtros aplicados pelo utilizador

10) Feed com conteúdo vídeo ou áudio (Extra)

- **Requisitos de Utilizador:**
 - Se possível o utilizador deverá poder reproduzir as publicações.
- **Requisitos de Sistema:**
 - Se possível o sistema deverá apresentar os últimos vídeos dos principais canais de notícias disponíveis.

11) Feed de exposição de opinião

- **Requisitos de Utilizador:**
 - O utilizador deverá poder ler todos os artigos publicados pelos vários jornalistas integrantes da aplicação.
- **Requisitos de Sistema:**
 - O sistema deverá apresentar todos os artigos escritos pelos jornalistas.
 - Além disso, deve também indicar ao utilizador se já segue ou não cada jornalista apresentado no feed

12) Secção de comentários na publicação

➤ Requisitos de Utilizador:

- O utilizador deverá conseguir visualizar comentários escritos por outros utilizadores.
- Poderá ainda comentar a publicação em questão.

➤ Requisitos de Sistema:

- O sistema deverá apresentar todos os comentários feitos na publicação pelos vários utilizadores.
- Deverá apresentar um campo de texto onde o utilizador pode ou não utilizar para comentar a publicação.

13) Guardar notícias

➤ Requisitos de Utilizador:

- O utilizador deve poder guardar uma notícia para depois ler mais tarde ou voltar a reler.

➤ Requisitos de Sistema:

- O sistema deverá armazenar na sua base de dados a notícia guardada pelo utilizador.

14) Notificações

➤ Requisitos de Utilizador:

- O utilizador deverá conseguir receber uma notificação assim que um jornalista que ele segue publica um artigo.
- Deverá receber uma notificação quando uma notícia de uma das suas categorias preferidas for publicada
- Deverá conseguir aceder a um painel para verificar todas as notificações recebidas
- Deverá ainda conseguir limpar as notificações quando já não lhe interessam.

➤ Requisitos de Sistema:

- O sistema deverá conseguir enviar alertas de notificação aos utilizadores quando um jornalista que estes seguem publica um artigo.
- Deverá ainda enviar notificações quando novas notícias são publicadas conforme as categorias favoritas de cada utilizador.
- O sistema deverá conseguir armazenar qualquer tipo de alteração feita pelo utilizador ao painel de notificações.

15) Publicação de Artigos

➤ Requisitos de Utilizador (Jornalista):

- Um jornalista deverá ser capaz de escrever um artigo com título, descrição, corpo e ainda uma *thumbnail* caso assim deseje.

➤ Requisitos de Sistema:

- O sistema deverá armazenar na sua base de dados o novo artigo publicado.
- O sistema deverá apresentar ao jornalista uma página própria onde ele pode escrever o seu artigo e fazer as configurações necessárias antes de publicar.

16) Listagem e Validação de Pedidos (Extra)

➤ Requisitos de Utilizador (Administrador):

- Pedir lista de jornalistas à espera de validação
- Validar pedidos

➤ Requisitos de Sistema:

- O sistema deve fornecer a listagem de jornalistas à espera de validação
- Armazenar alterações feitas na BD.

4.2. Requisitos Não-Funcionais

- 1) A aplicação deverá ter uma interface intuitiva de forma que o utilizador possa facilmente navegar entre as funcionalidades da aplicação.
- 2) A aplicação deve estar disponível em qualquer *browser*.
- 3) A aplicação deverá estar disponível durante 7 dias por semana e 24 horas por dia.
- 4) A aplicação deverá ter o menor tempo de resposta possível, de forma a satisfazer os pedidos dos utilizadores o mais rapidamente possível.

5. Modelo de Domínio

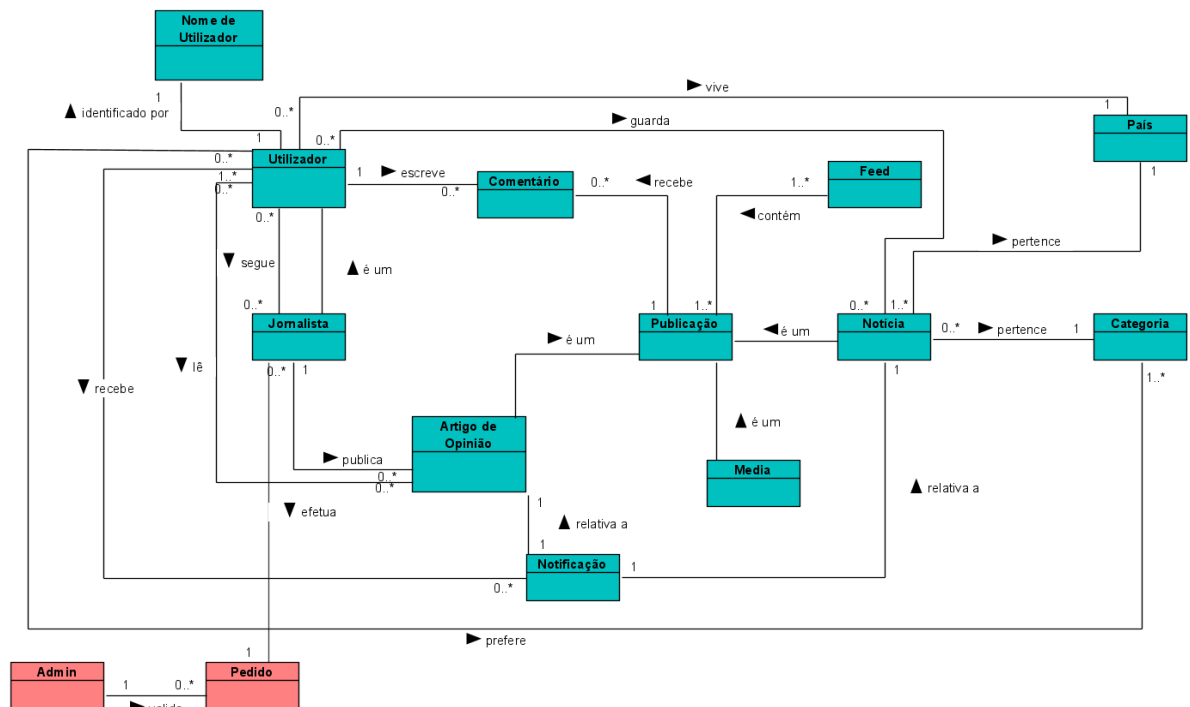


Figura 6 - Modelo de Domínio

Após o levantamento e análise dos requisitos, desenvolveu-se o modelo de domínio cujo objetivo é representar as entidades mais relevantes do sistema, bem como os seus relacionamentos.

Através do diagrama representado, podemos tirar algumas conclusões, tais como:

- Qualquer jornalista é também um utilizador
- Um utilizador é identificado por um *username*
- Um utilizador pode comentar zero ou mais publicações
- Um utilizador possui um determinado país associado
- Um jornalista publica zero ou mais artigos de opinião
- Um utilizador segue zero ou mais jornalistas
- Uma publicação pode ser um artigo de opinião, uma notícia ou um *media*
- Uma notícia tem uma categoria associada
- Um utilizador possui categorias de notícias preferidas

- Um utilizador recebe uma notificação relativa a um artigo publicado de um jornalista que segue ou relativa a uma notícia de uma categoria que ele prefere
- Um *feed* contém uma ou mais publicações

Além disto, o grupo achou que seria interessante adicionar um administrador do sistema, que recebe e valida pedidos de utilizadores que querem ser jornalistas. No entanto, estas funcionalidades são consideradas extra pelo que só implementadas caso haja disponibilidade para tal.

6. Diagrama de Use Cases

Na figura abaixo está representado o nosso diagrama de casos de uso que ilustra de forma simplificada todas as funcionalidades do nosso sistema.

É de notar que, para o caso do ator Admin e das suas funções, representados a vermelho no diagrama, foram ideias que o grupo achou que seriam interessantes de implementar no futuro, de modo a tornar a aplicação mais segura.

Com a aplicação NewsBook irão interagir 3 tipos de atores: Utilizador, Jornalista e Utilizador não registado. Quando um utilizador interage pela primeira vez com o sistema, é identificado como um utilizador não registado, onde a sua única funcionalidade permitida é o seu registo, uma vez que cada individuo só deverá registar conta apenas uma vez. Por outro lado, após ser feito o registo, esse usuário passa a ser, ou um utilizador normal ou um jornalista. Como se pode verificar no diagrama, este ultimo estende todas as funções do utilizador, sendo que lhe são permitidas outras adicionais.

Consequentemente, é possível também observar a existência do conetor include que representa funcionalidades que dependem da realização de outras previamente para poderem ser executadas. Por outro lado, temos o use case 'Configurar opções de jornalista' que é uma extensão da função 'Configuração de Perfil' e significa que o primeiro aplica as mesmas ações do segundo e outras adicionais.

20

7. Especificação de Use Cases

7.1. Registo

Ator: Utilizador Não Registrado

Cenário: O utilizador abre a aplicação pela primeira vez e seleciona a opção de registo caso não tenha conta

Pré-Condição: Utilizador selecionou a opção “sign up”

Pós-Condição: Um novo utilizador é registado no sistema

Fluxo Normal:

1. Sistema solicita email, password, primeiro e último nome, *username*, fotografia (opcional), pais, cidade e categorias preferidas.
2. Utilizador preenche os dados de acesso e as suas informações pessoais.
3. Utilizador identifica-se como “não jornalista”.
4. Sistema verifica se já existe um utilizador registado com o email e *username* introduzidos.
5. Sistema guarda o novo utilizador na sua base de dados.
6. Sistema informa que registo ocorreu com sucesso.

Fluxo de Exceção [Já existe utilizador com o email introduzido] (Passo 4):

- 4.1 Sistema informa que o registo falhou devido ao email.

Fluxo de Exceção [Já existe utilizador com o username introduzido] (Passo 4):

- 4.1 Sistema informa que o registo falhou devido ao username.

Fluxo de Exceção [Utilizador identifica-se como jornalista] (Passo 3):

- 3.1 Utilizador aguarda aprovação do administrador do sistema.

7.2. Login

Ator: Utilizador

Cenário: Utilizador introduz os seus dados de autenticação na página inicial para ter acesso

Pré-Condição: O utilizador não está autenticado

Pós-Condição: O utilizador está autenticado

Fluxo Normal:

1. Utilizador introduz o seu nome de utilizador e password.
2. Sistema valida as credenciais.
3. Sistema redireciona utilizador para a página principal ("homepage").

Fluxo de Exceção [Credenciais inválidas] (Passo 2):

- 2.1 Sistema informa que os dados introduzidos estão incorretos

7.3. Logout

Ator: Utilizador

Cenário: O utilizador sai da sua conta e volta à página inicial

Pré-Condição: O utilizador já está autenticado

Pós-Condição: O utilizador deixa de estar autenticado

Fluxo Normal:

1. Utilizador seleciona o botão de "logout".
2. Sistema redireciona utilizador para a página inicial.

7.4. Escolher tipo de Feed

Ator: Utilizador

Cenário: Seleção do feed que quer visualizar de acordo com as opções dadas

Pré-Condição: Utilizador tem de estar autenticado e na *homepage* da app

Pós-Condição: Utilizador visualiza tipo de feed escolhido

Fluxo Normal:

1. Utilizador seleciona a opção de feed que quer ver.
2. Sistema disponibiliza o feed escolhido.

7.5. Filtrar publicações por palavra-chave

Ator: Utilizador

Cenário: Utilizador pesquisa notícias que contenham no título ou na descrição uma certa *key word*, e, de seguida, visualiza as mesmas

Pré-Condição: Utilizador tem de estar autenticado

Pós-Condição: São mostradas as notícias que contém a palavra-chave inserida

Fluxo Normal:

1. Utilizador insere na *search bar* uma palavra-chave.
2. Sistema procura todas as notícias que contém essa palavra-chave no título ou na descrição.
3. Sistema filtra as notícias encontradas.

Fluxo de Exceção [Palavra-chave não foi encontrada] (Passo 2):

- 2.1 Sistema informa que não encontrou notícias com a palavra-chave inserida.

7.6. Filtrar notícias por país

Ator: Utilizador

Cenário: O utilizador escolhe ver notícias apenas de um país, ou notícias globais

Pré-Condição: O utilizador já está autenticado

Pós-Condição: Utilizador vê a lista de notícias relativas ao país selecionado

Fluxo Normal:

1. Utilizador seleciona o país do qual deseja ver notícias.
2. Sistema apresenta a lista atualizada de notícias do país selecionado ao utilizador.

Fluxo de Exceção 1 [Não existem notícias] (Passo 2):

- 2.1 Sistema informa que não existe nenhuma notícia dentro da seleção realizada pelo do utilizador.

7.7. Filtrar notícias por categoria

Ator: Utilizador

Cenário: O utilizador escolhe ver notícias sobre dado tema (categoria)

Pré-Condição: O utilizador já está autenticado

Pós-Condição: Utilizador vê a lista de notícias que se inserem na categoria selecionada

Fluxo Normal:

1. Utilizador seleciona a categoria sobre as quais deseja ver notícias.
2. Sistema apresenta a lista de notícias da categoria selecionada ao utilizador.

Fluxo de Exceção 1 [Não existem notícias] (Passo 2):

2.1 Sistema informa que não existe nenhuma notícia dentro das especificações do utilizador

7.8. Ver secção de comentários

Ator: Utilizador

Cenário: Utilizador abre a secção de comentários de uma publicação

Pré-Condição: O utilizador está autenticado

Pós-Condição: Sistema apresenta secção de comentários

Fluxo Normal:

1. Utilizador solicita a visualização da secção de comentários de uma publicação.
2. Sistema apresenta interface da secção de comentários da publicação escolhida.

7.9. Comentar publicações

Ator: Utilizador

Cenário: Utilizador faz um comentário dentro da secção

Pré-Condição: O utilizador está autenticado

Pós-Condição: O utilizador comentou uma publicação

Fluxo Normal:

1. <<include>> Ver secção de comentários.
2. Utilizador solicita a adição de um novo comentário.
3. Utilizador escreve o comentário.
4. Utilizador solicita o envio do comentário.
5. Sistema apresenta o novo comentário na caixa de comentários da interface da publicação.

Fluxo de Exceção [Utilizador cancela envio do comentário] (Passo 4):

- 4.1 Sistema informa que o comentário não foi adicionado.
- 4.2 Sistema apresenta secção de comentários.

7.10. Eliminar Comentários

Ator: Utilizador

Cenário: Utilizador quer apagar comentário feito numa notícia

Pré-Condição: Utilizador tem de estar autenticado e ter feito um comentário

Pós-Condição: Comentário feito deixa de existir na secção de comentários

Fluxo Normal:

1. <<include>> Ver secção de comentários.
2. Utilizador seleciona opção de apagar comentário.
3. Sistema elimina comentário da base de dados do sistema.
4. Sistema atualiza secção de comentários.

7.11. Receber alerta de notificações

Ator: Utilizador <- Sistema

Cenário: Sistema envia uma notificação ao utilizador, e utilizador recebe um alerta

Pré-Condição: O utilizador está autenticado

Pós-Condição: O utilizador recebe uma notificação

Fluxo Normal:

1. Sistema recebe uma informação que tem de enviar ao utilizador.
2. Sistema informa ao utilizador o número de notificações que tem por ler.

7.12. Aceder ao painel de notificações

Ator: Utilizador

Cenário: Utilizador abre painel de notificações

Pré-Condição: O utilizador está autenticado

Pós-Condição: Sistema apresenta lista de notificações

Fluxo Normal:

1. Utilizador pede para aceder ao painel de notificações.
2. Sistema apresenta lista de todas as notificações recebidas.

Fluxo de Exceção [Utilizador não tem notificações pendentes] (Passo 2):

- 2.1 Sistema informa o utilizador que não existem notificações pendentes.

7.13. Limpar notificações

Ator: Utilizador

Cenário: Utilizador apaga as notificações recebidas

Pré-Condição: O utilizador está autenticado

Pós-Condição: Sistema elimina todas as notificações do utilizador

Fluxo Normal:

1. <<include>> Aceder ao painel de notificações.
2. Utilizador solicita a remoção de todas as notificações.
3. Sistema retira as notificações desse utilizador da Base de Dados.

7.14. Guardar uma notícia

Ator: Utilizador

Cenário: Utilizador seleciona opção de guardar uma notícia para ler mais tarde.

Pré-Condição: Utilizador está autenticado

Pós-Condição: Notícia é adicionada à lista de notícias guardadas do utilizador

Fluxo Normal:

1. Utilizador indica que quer guardar uma determinada notícia.
2. Sistema adiciona a notícia à lista de notícias guardadas do utilizador na BD.

7.15. Remover uma notícia guardada

Ator: Utilizador

Cenário: Utilizador seleciona botão para eliminar a notícia guardada no seu perfil

Pré-Condição: Utilizador autenticado e notícia a retirar deve ter sido guardada previamente

Pós-Condição: Notícia é retirada da lista de notícias guardadas do utilizador

Fluxo Normal:

1. Utilizador solicita a remoção de uma determinada notícia guardada.
2. Sistema retira a notícia da lista de notícias guardadas do utilizador da base de dados.

7.16. Visitar Perfil de Jornalista

Ator: Utilizador

Cenário: Utilizador está no feed “Opinion Articles” e clica no nome de um jornalista para ver o seu perfil

Pré-Condição: Utilizador está autenticado e está no feed “Opinion Articles”

Pós-Condição: Utilizador visualiza perfil do jornalista pretendido

Fluxo Normal:

1. Utilizador seleciona o nome do jornalista.
2. Sistema redireciona o utilizador para o perfil do jornalista.

7.17. Seguir jornalista

Ator: Utilizador

Cenário: O utilizador escolhe seguir um certo jornalista

Pré-Condição: O utilizador já está autenticado e ainda não segue o jornalista em questão

Pós-Condição: Utilizador passa a receber notificações quando o jornalista publica artigos

Fluxo Normal:

1. «*include*» Visitar Perfil de Jornalista.
2. Utilizador seleciona a opção “follow”.
3. Sistema adiciona o jornalista à lista de jornalistas que o utilizador segue.

7.18. Deixar de seguir jornalista

Ator: Utilizador

Cenário: O utilizador escolhe deixar de seguir um certo jornalista

Pré-Condição: O utilizador já está autenticado e já segue o jornalista em questão

Pós-Condição: Utilizador deixa de receber notificações quando o jornalista publica artigos

Fluxo Normal:

1. «*include*» Visitar Perfil de Jornalista.
2. Utilizador seleciona a opção “unfollow”.
3. Sistema remove o jornalista da lista de jornalistas que o utilizador segue.

7.19. Ver Perfil

Ator: Utilizador

Cenário: Utilizador visualiza todas as suas informações na página do seu perfil

Pré-Condição: Utilizador tem de estar autenticado e deve estar na *homepage* do sistema

Pós-Condição: Sistema apresenta a página do perfil do utilizador

Fluxo Normal:

1. Utilizador seleciona opção de visualizar perfil.
2. Sistema apresenta a página com as informações do utilizador. (Nome, fotografia, *username*, país e cidade onde vive, jornalistas que segue, notícias guardadas e categorias preferidas).

7.20. Ver notícias guardadas

Ator: Utilizador

Cenário: Utilizador seleciona opção no perfil para visualizar as notícias que guardou.

Pré-Condição: Utilizador tem de estar autenticado

Pós-Condição: Utilizador visualiza as notícias guardadas

Fluxo Normal:

1. «*include*» Ver perfil.
2. Utilizador seleciona opção de consultar notícias guardadas.
3. Sistema consulta as notícias guardadas pelo utilizador na BD.
4. Sistema apresenta notícias guardadas pelo utilizador por ordem cronológica.

Fluxo de Exceção 1: [Utilizador não tem notícias guardadas] (passo 3)

- 3.1 Sistema indica ao utilizador que ainda não tem nenhuma notícia guardada.

7.21. Configuração Perfil

Ator: Utilizador / Jornalista

Cenário: Utilizador pretende atualizar as suas informações

Pré-Condição: Utilizador tem de estar autenticado

Pós-Condição: Perfil do utilizador é atualizado

Fluxo Normal:

1. «*include*» Ver perfil.
2. Utilizador seleciona opção de configurar perfil.
3. Sistema apresenta campos que o utilizador pode atualizar.
4. Utilizador preenche campos que quer modificar.
5. Caso seja jornalista, preenche os campos próprios de jornalista («*extend*» - **Configurar opções de jornalista**)
6. Utilizador escolhe a opção de guardar alterações.
7. Sistema solicita password atual ao utilizador.
8. Utilizador introduz password.
9. Sistema valida password.
10. Sistema atualiza a BD com as novas informações do utilizador.
11. Sistema reencaminha utilizador para o seu perfil atualizado.

Fluxo Alternativo 1: [Utilizador introduz password errada] (passo 9)

- 9.1 Sistema avisa utilizador que a password está incorreta.
- 9.2 Utilizador valida aviso.
- 9.3 Passo 7.

Fluxo de Exceção 1: [Utilizador seleciona opção cancelar] (passo 6)

- 6.1 Sistema cancela operações.
- 6.2 Sistema reencaminha utilizador para a sua página de perfil.

7.22. Publicar artigos

Ator: Jornalista

Cenário: Jornalista no seu perfil seleciona botão que permite publicar e configurar o seu artigo de acordo com os campos solicitados pelo sistema

Pré-Condição: O jornalista está na página de perfil e está autenticado

Pós-Condição: O jornalista publicou um artigo

Fluxo Normal:

1. Jornalista solicita a adição de um novo artigo.
2. Sistema aceita solicitação.
3. Sistema apresenta interface de edição do novo artigo.
4. Jornalista escreve o artigo de opinião.
5. Jornalista envia o novo artigo de opinião para o sistema.

Fluxo de Exceção [Jornalista cancela edição do artigo] (Passo 5):

- 5.1 Sistema informa que o artigo não foi publicado.
- 5.2 Sistema apresenta página de perfil.

7.23. Consultar *Feed*

Ator: Utilizador

Cenário: Utilizador deseja ver *feed* de notícias

Pré-Condição: Utilizador tem de estar autenticado e noutra página sem ser a homepage

Pós-Condição: Sistema disponibiliza notícias por ordem cronológica

Fluxo Normal:

1. Utilizador seleciona ícone da aplicação.
2. Sistema apresenta *feed* de notícias principal (*Trending News*).

8. Construção da API da Lógica de Negócio

A partir da análise dos Use Cases determinados e da especificação de cada um, foi possível levantar várias responsabilidades que o sistema deverá ter. De forma a simplificar e a aprimorar a organização da arquitetura da aplicação, foi decidido dividir as responsabilidades por dois subsistemas: **GestUtilizadores** e **GestPublicacoes**. O **GestUtilizadores** está encarregue de todos os métodos e responsabilidades relacionados com o Utilizador, e o **GestPublicacoes** é encarregue de todos aqueles associados às Publicações.

8.1. Métodos da API

A partir das responsabilidades levantadas foi possível elaborar os seguintes métodos que irão integrar a API da Lógica de Negócio da aplicação.

- **registarUtilizador**(username : String, password : String, email : String, nome : String, pais : String, city : String, urlFoto : String, cateys : List<String>, jornalista : Boolean) -> **GestUtilizadores**
- **login**(username : String, password : String) -> **GestUtilizadores**
- **logout**() -> **GestUtilizadores**
- **getHomePageInfo**(tipoFeed : Int) : Map<Int,<Boolean,Publicacao>> (ir buscar noticias guardadas) -> **GestPublicacoes**
- **getHomePageSearch**(palavraChave : String) : Map<Int,<Boolean,Publicacao>> (ir buscar noticias guardadas) -> **GestPublicacoes**
- **getSeccaoComentarios**(idPub : Int) : Map<Int,Comentario> -> **GestPublicacoes**
- **comentaPublicacao**(idPub : Int, conteudo : String) : Map<Int,Comentario> -> **GestPublicacoes**
- **eliminaComentario**(IdPub : Int) : Map<Int,Comentario> -> **GestPublicacoes**
- **getNotificacoes**() : List<Notificacao> -> **GestUtilizadores**

- **clearNotificacoes()** - > **GestUtilizadores**
- **addRemoveNoticiaGuardada**(addRem : Boolean, idNoticia : Int)
- **getPerfillInfo()** : Utilizador -> **GestUtilizadores**
- **followJornalista**(seguir : Boolean, codJorn : Int) -> **GestUtilizadores**
- **configuraPerfil**(jornalista : Boolean, urlFoto : String, name : String, newPass : String, pais : String, city : String, categs : List<String>, bio : String, linkedIn : String) -> **GestUtilizadores**
- **publicaArtigo**(idArtigo : String, titulo : String, descricao : String, imagem : String, conteudo : String) -> **GestPublicacoes**

9. Arquitetura da Aplicação

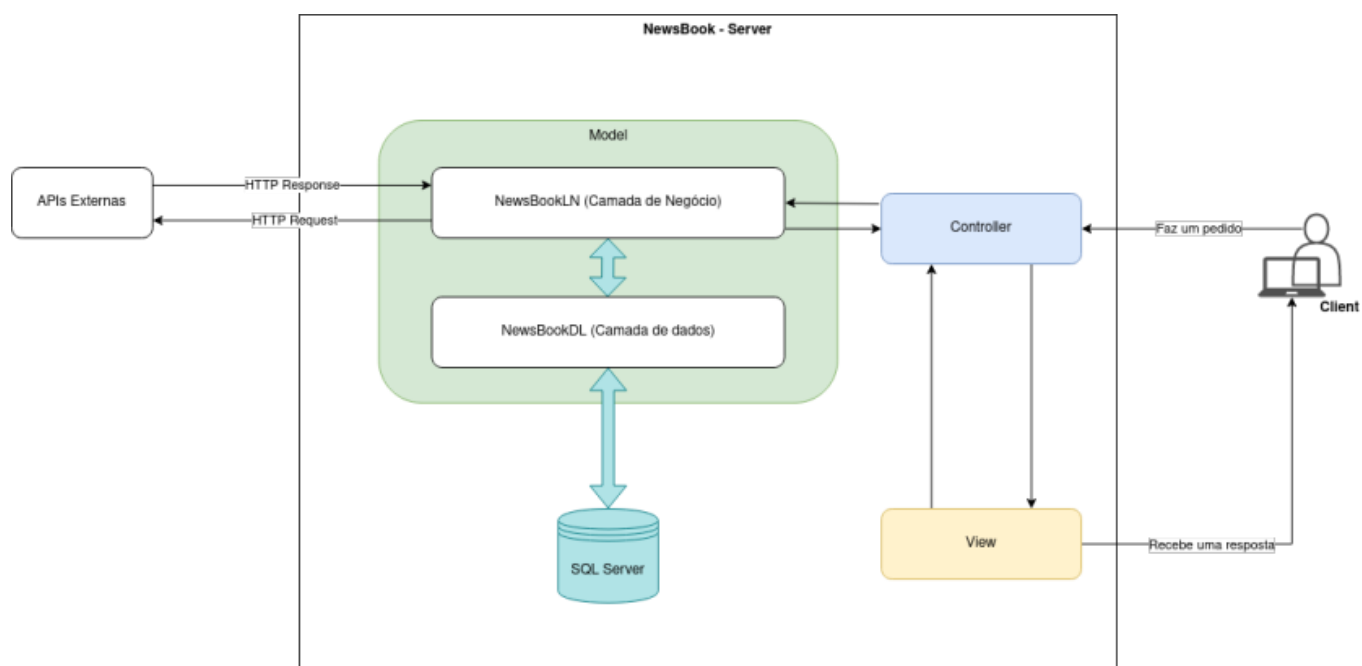


Figura 8 - Arquitetura da Aplicação

10. Diagrama de Componentes

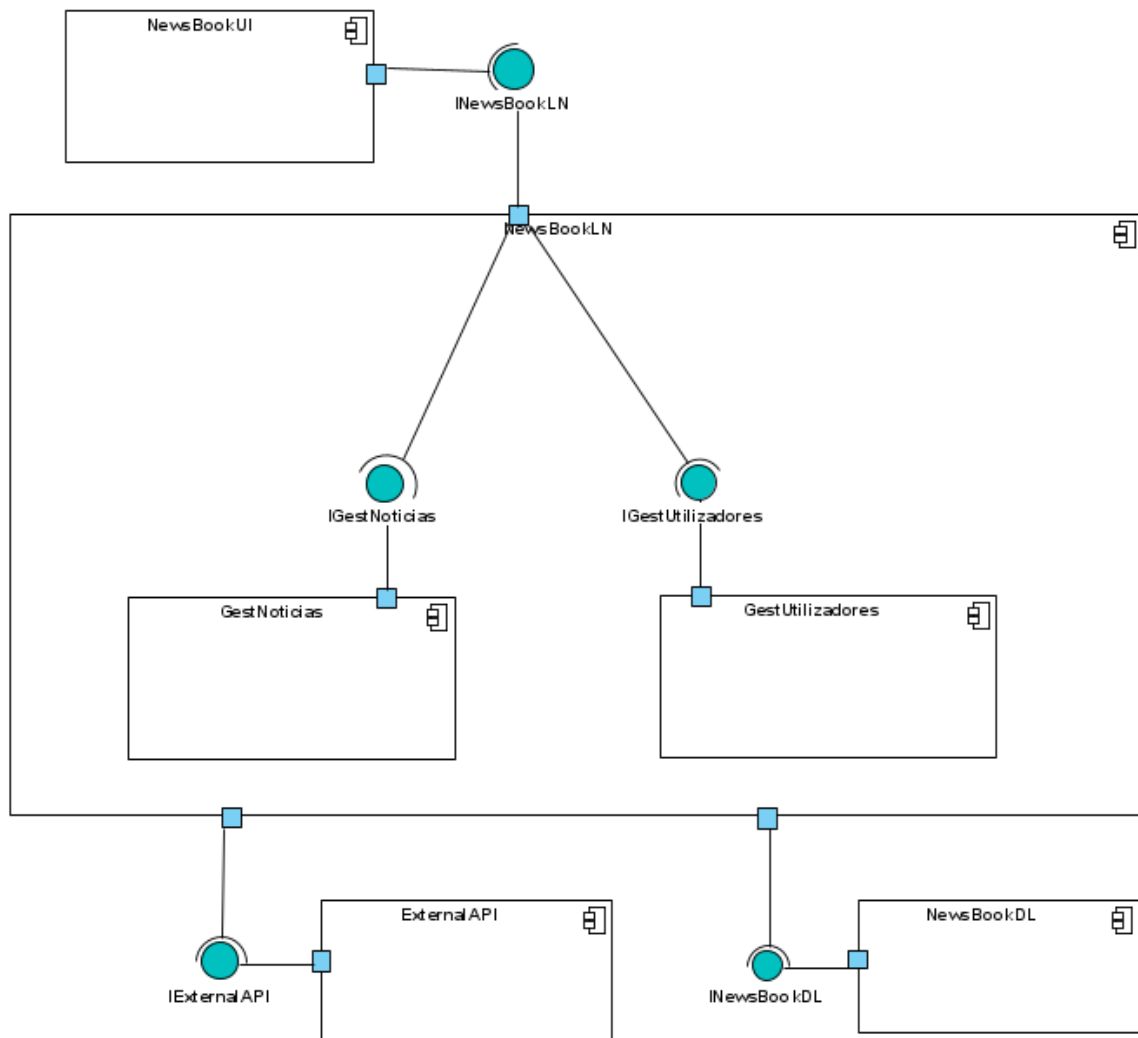


Figura 9 - Diagrama de Componentes

11. Arquitetura da Camada de Negócio

A partir da análise dos requisitos e responsabilidades levantadas, e ainda dos modelos UML para já apresentados, foi possível não só identificar as principais classes de cada subsistema, como também caracterizá-las.

11.1. Dicionário das Principais Classes

Utilizador – Trata-se da representação dos utilizadores da aplicação e inclui a informação pessoal correspondente.

Jornalista – Esta classe representa a especialização da classe Utilizador, contendo, portanto, toda a informação adicional que este tipo de utilizador possui.

Notificação – Representa os alertas que o utilizador recebe, e o sistema envia.

Categoria – Classe que representa um grupo de notícias que contém um tema em comum.

País (Country) – Representa o país onde o utilizador vive, ou de onde uma notícia é oriunda.

Publicação – Esta classe representa a generalização das classes Notícia, Media e Artigo. Contendo, portanto, todos os atributos que estas têm em comum.

Notícia – Representa a própria notícia em si, possuindo, desta forma, todas as características da mesma, tal como o seu autor, ou a categoria onde pertence.

Media – Representa todo o conteúdo disponibilizado em forma vídeo ou áudio.

Artigo – É a representação dos artigos de opinião publicados pelos jornalistas da aplicação.

Comentário – Trata-se da representação dos comentários deixados pelos utilizadores numa publicação, seja a sua opinião pessoal, ou então, uma pequena avaliação do conteúdo da notícia.

11.2. Diagramas de Classes

GestPublicações

O seguinte diagrama demonstra as várias classes presentes neste subsistema e a relação entre as mesmas.

Tal como é possível observar, a classe *Publicacao* trata-se de uma generalização das classes *Noticia*, *Media* e *Artigo*, onde a mesma poderá conter zero ou mais comentários feitos por vários utilizadores. Além disso, também foram incluídas as API externas: **NewsApi** e **YouTubeAPI**, de onde a aplicação irá adquirir as notícias e os conteúdos media.

A interface *IGestPublicacoes* contém todos os métodos que irão ser usados pelo sistema, de modo a obter os dados necessários para satisfazer os pedidos feitos pelo utilizador do sistema. As classes do tipo DAO, são aquelas que irão fazer a conexão entre camada de negócios e a camada de dados, permitindo assim a persistência de objetos na Base de Dados. Além disso, permitem também a criação de objetos a partir da informação disponível na BD.

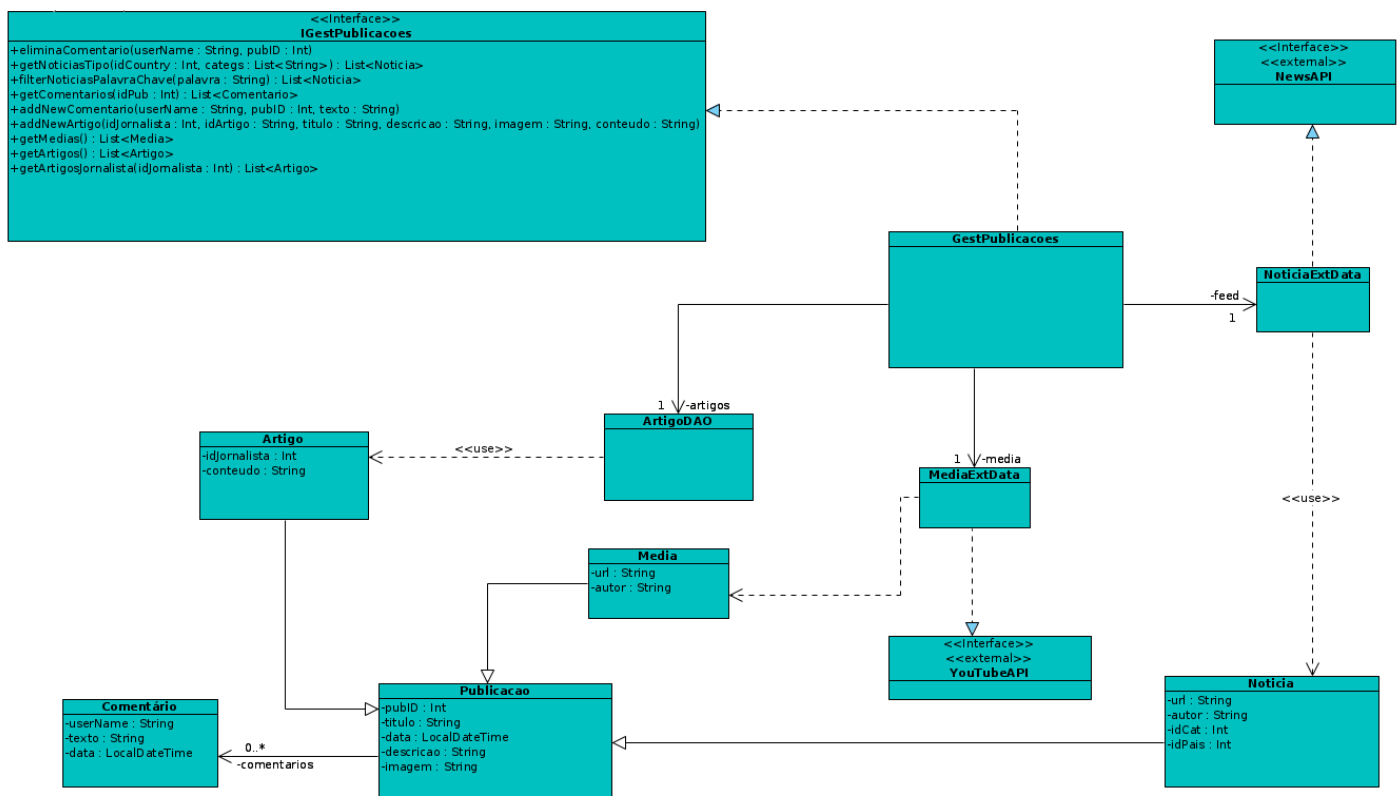


Figura 10 - Diagrama de Classes do Subsistema GestPublicacoes

GestUtilizadores

O diagrama de classes representativo deste subsistema, irá conter as principais classes relacionadas com o utilizador.

Em primeiro lugar, demonstra-se que a classe *Jornalista* é uma especialização da classe *Utilizador*, sendo assim, um jornalista irá possuir todos os atributos que um utilizador tem. Este por sua vez, além das duas informações pessoais, irá ter o país onde vive atualmente ou, então, aquele pelo qual ele se interessa mais e pretende ver notícias dessa origem. Além disso, irá ter também uma lista com as suas categorias favoritas, e outra com os identificadores dos jornalistas que ele segue. Também foi incluído um *Map<Int, Noticia>*, de forma a armazenar todas as notícias guardadas por um utilizador, sem esquecer ainda as notificações recebidas por esse mesmo utilizador.

Tal como o subsistema anterior, este também contém uma interface, neste caso chamada de *IGestUtilizadores*, com todos os métodos que o sistema irá necessitar.

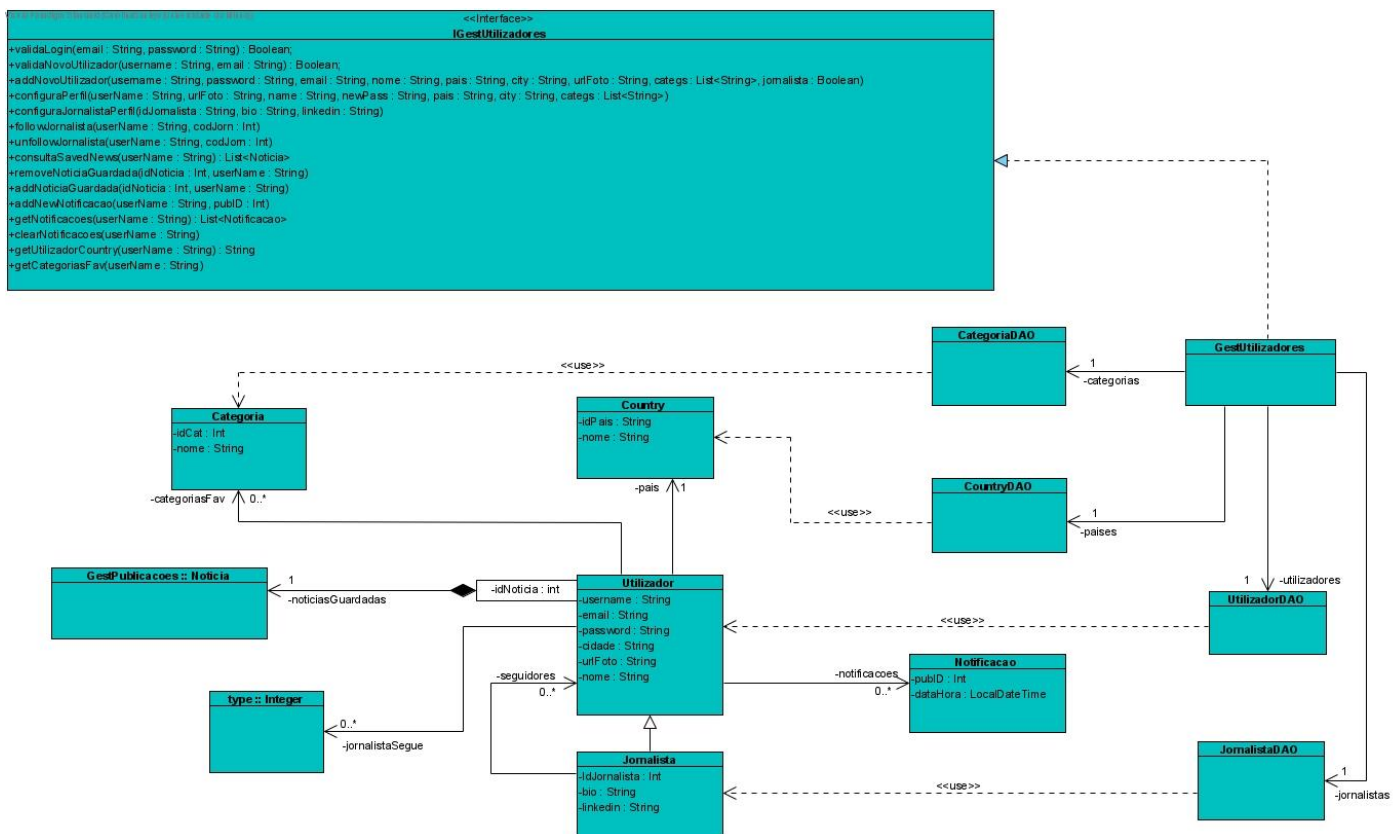


Figura 11 - Diagrama de Classes do Subsistema GestUtilizadores

12. Diagrama de Packages

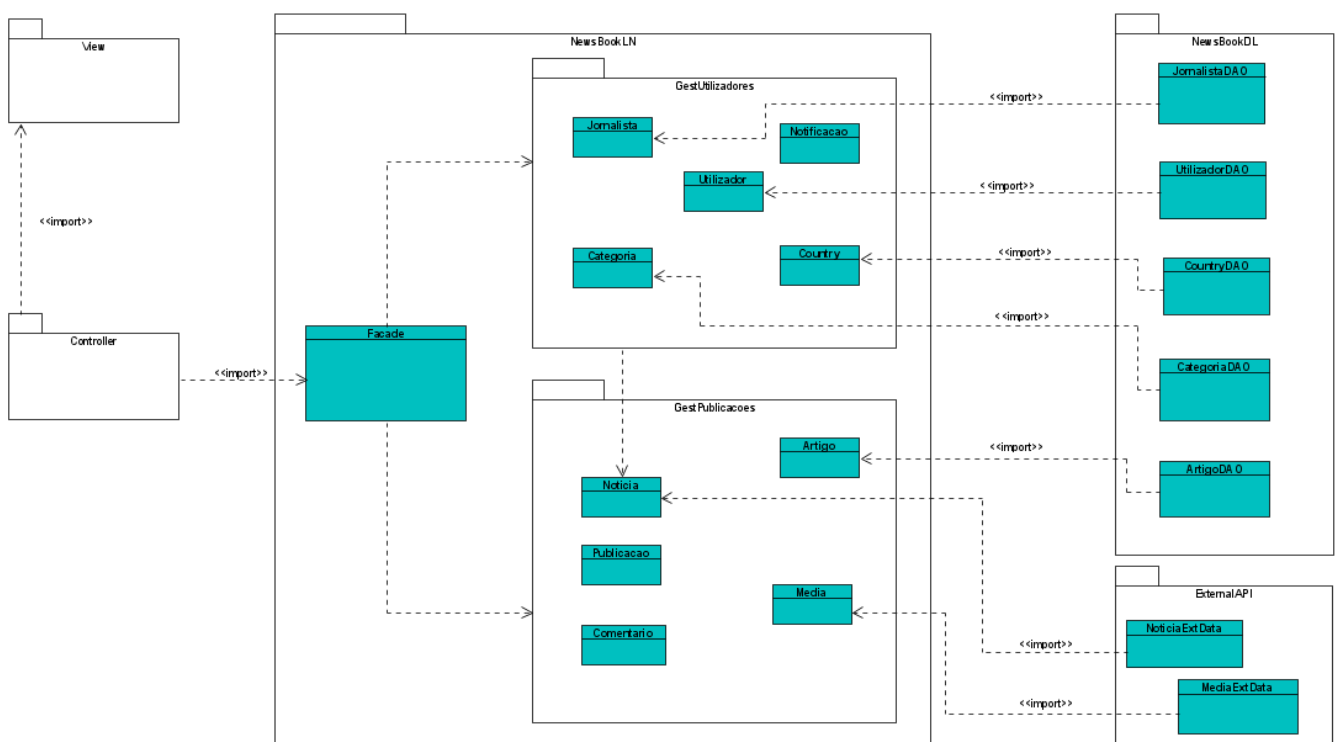


Figura 12 - Diagrama de packages

13. Diagramas de Sequência

Tendo por base os métodos da API da lógica de negócio foram definidos diagramas de sequência para os principais métodos implementados nos dois subsistemas existentes.

13.1. Validar os dados fornecidos no login pelo utilizador

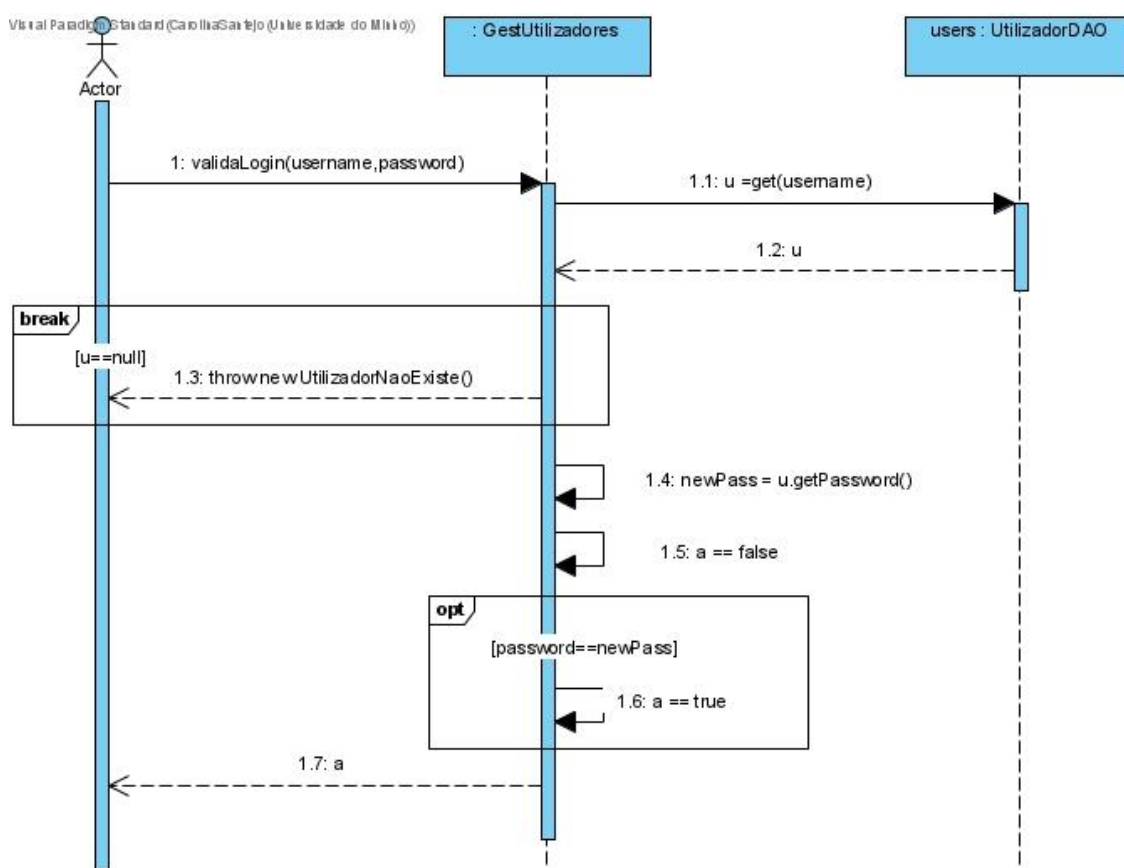


Figura 13 - Método *validaLogin*

13.2. Validar se um dado *userName* e email correspondem a algum utilizador já existente

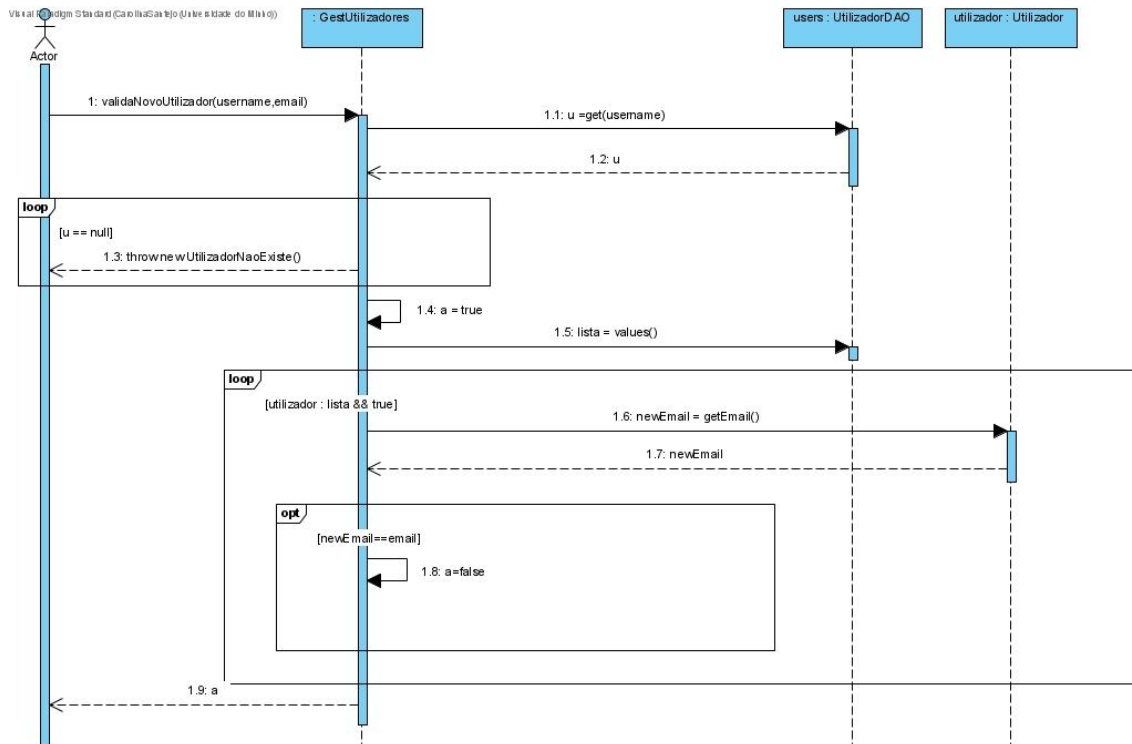


Figura 14 - Método *validaNovoUtilizador*

13.3. Adicionar novo artigo de opinião de um jornalista ao sistema

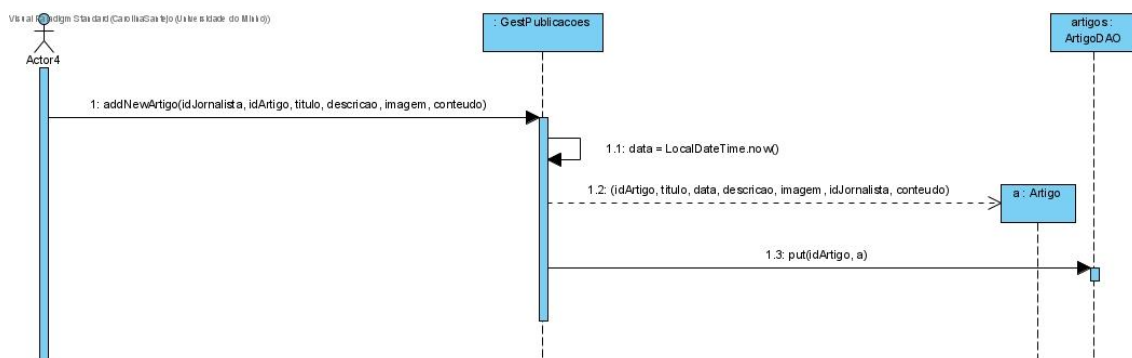


Figura 15 - Método *addNewArtigo*

13.4. Adicionar novo comentário de um utilizador a uma publicação

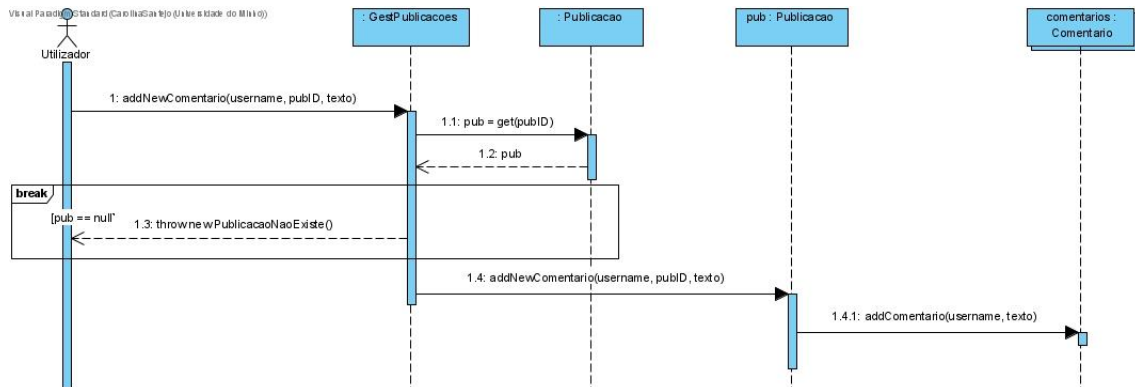


Figura 16 – Método *addNewComentario*

13.5. Adicionar nova notificação à lista de notificações de um utilizador

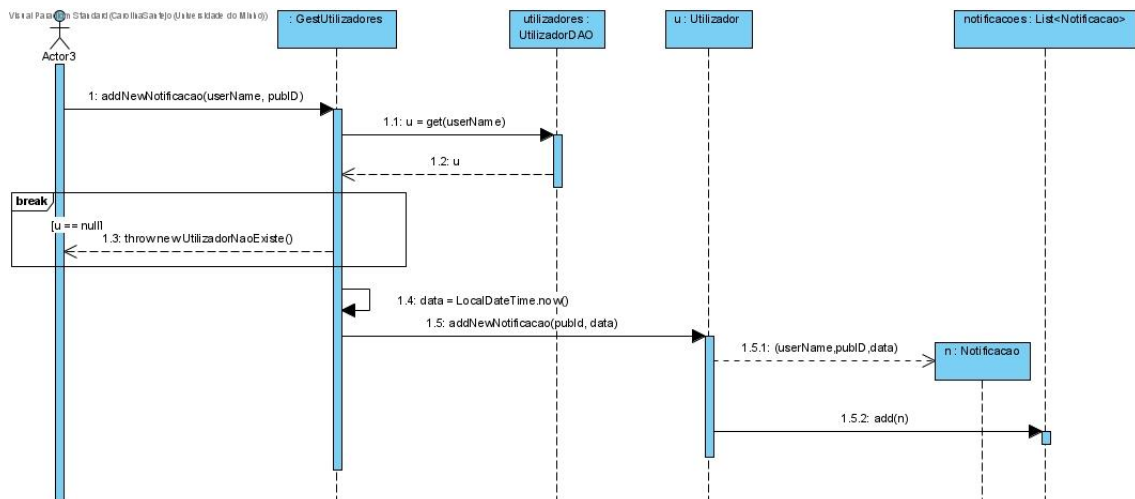


Figura 17 - Método *addNewNotificacao*

13.6. Adicionar novo utilizador (normal ou jornalista) ao sistema

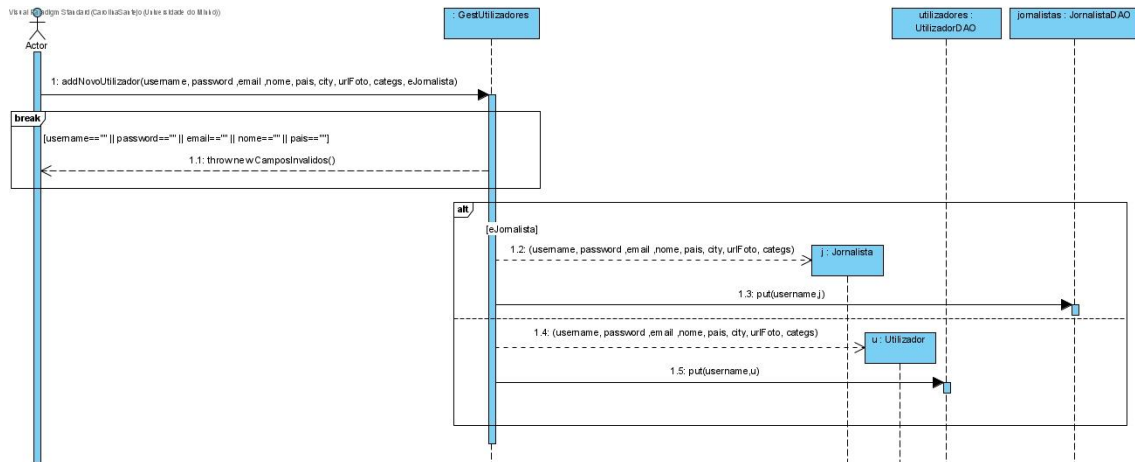


Figura 18 - Método *addNovoUtilizador*

13.7. Configurar atributos de um utilizador normal

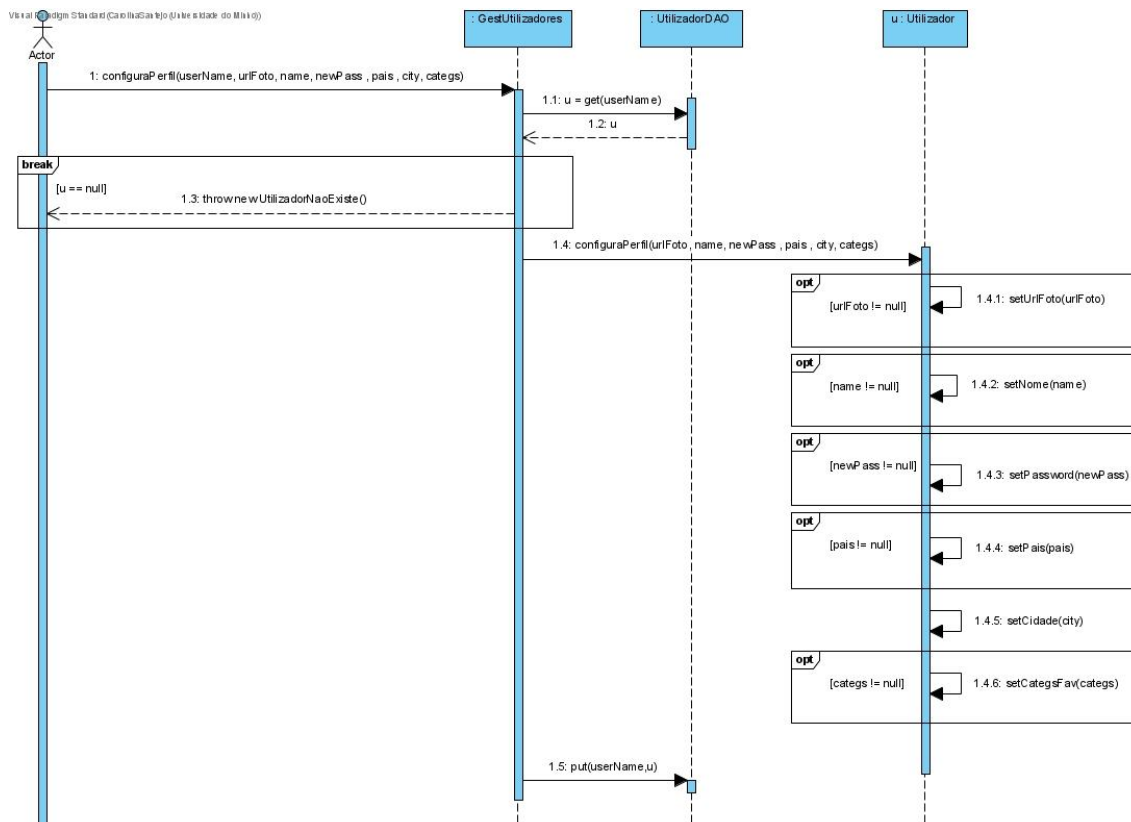


Figura 19 - Método *configuraPerfil*

13.8. Configurar atributos de um jornalista

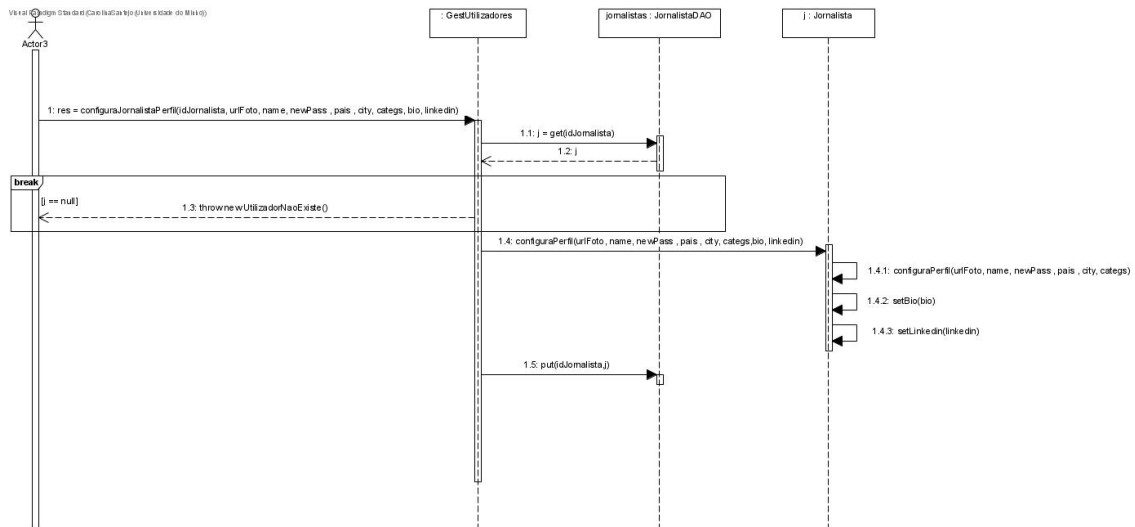


Figura 20 - Método *configurarJornalistaPerfil*

13.9. Obter uma lista de notícias que contém uma dada palavra chave

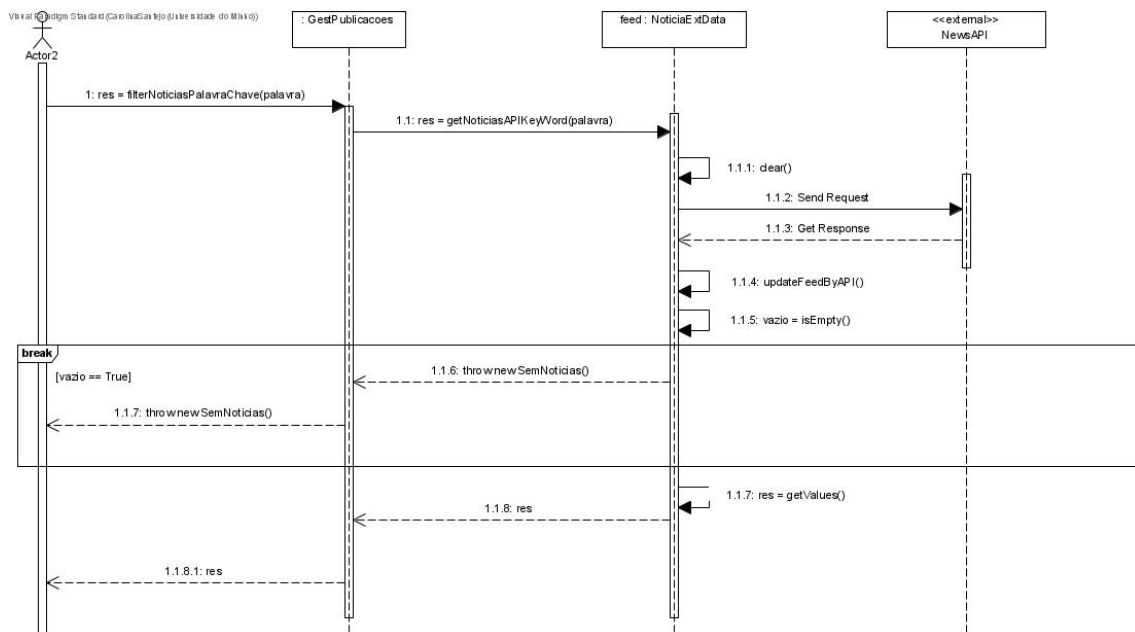


Figura 21 - Método *filterNoticiasPalavraChave*

13.10. Adicionar um seguidor à lista de seguidores de um jornalista

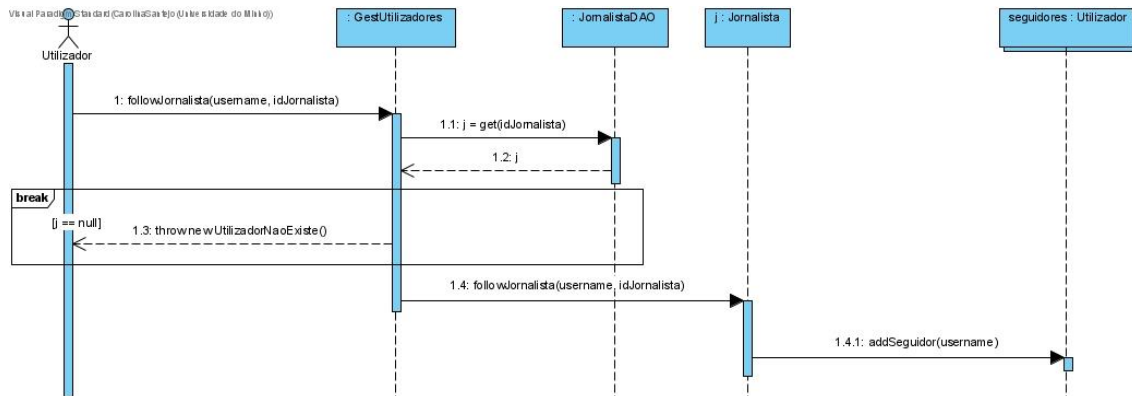


Figura 22 – Método *followJornalista*

13.11. Obter uma lista de todos os artigos de um dado jornalista

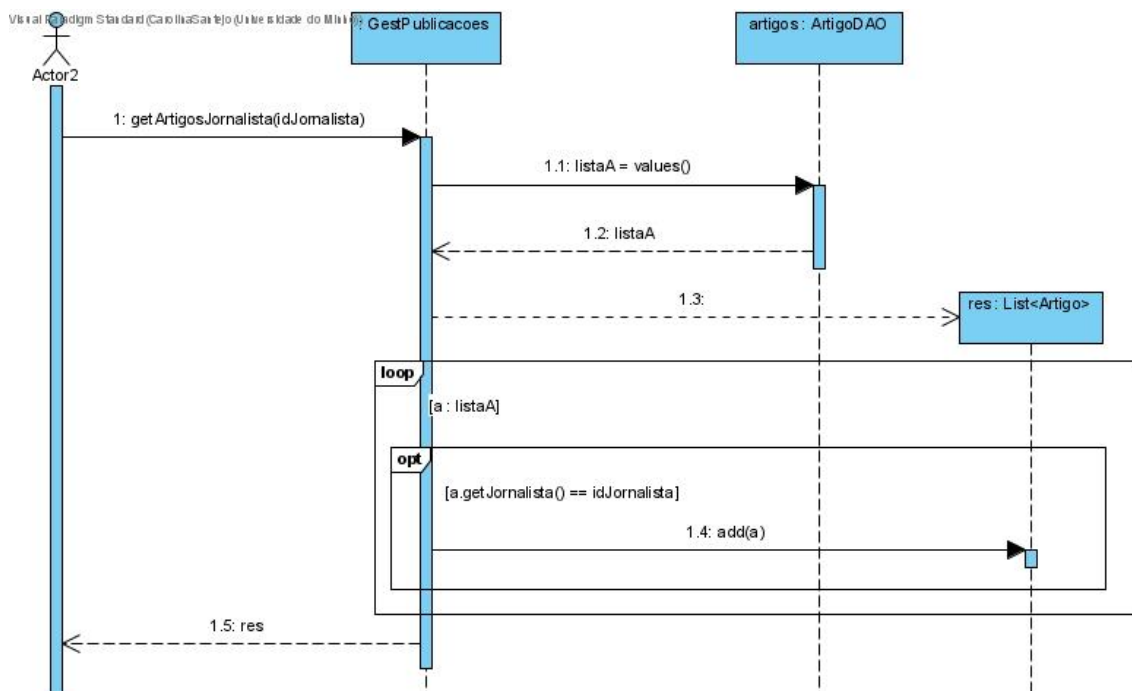


Figura 23 - Método *getArtigosJornalista*

13.12. Obter uma lista de notícias que correspondem a uma das categorias da lista dada para um certo país

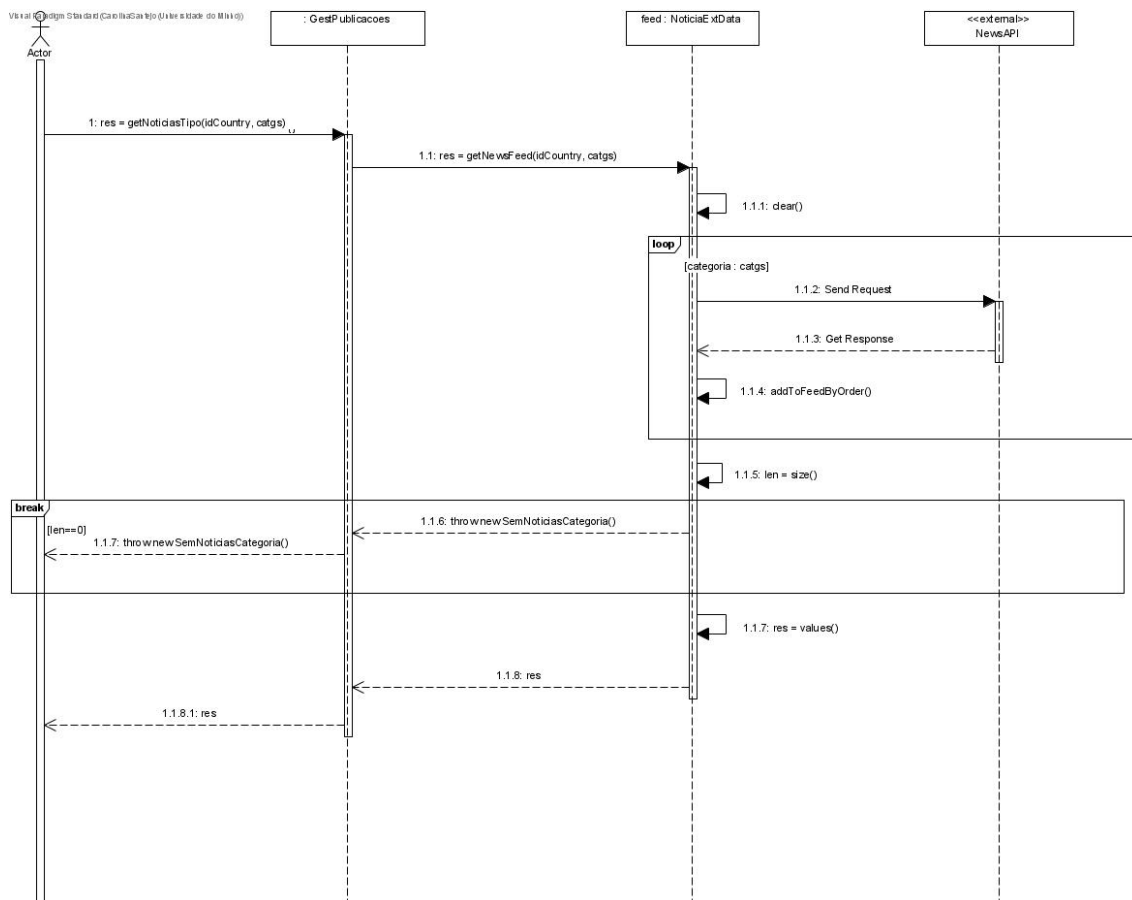


Figura 24 - Método *getNoticiasTipo*

14. Máquina de Estado

Após ter uma ideia bem definida de todos os componentes da aplicação, o grupo considerou essencial utilizar um modelo máquina de estado para representar de uma forma mais ilustrativa o funcionamento do sistema.

Pela análise do diagrama é possível aferir que cada estado corresponde a uma interface da nossa aplicação e, para transitar entre estados, o utilizador precisa de efetuar as ações indicadas pelas interligações.

Inicialmente, o sistema é arrancado através da pagina do login e, caso não esteja registado, o utilizador pode optar por fazer registo. Para qualquer um destes, se forem inseridos dados válidos é feita a transição para a 'HomePage'. Este pode ser considerado o estado principal uma vez que é possível aceder todos os restantes a partir dele, excetuando o caso do 'Configurações Perfil' em que é necessário estar previamente no estado 'Perfil'.

Finalmente, para fazer *logout*, ou seja, sair da aplicação, o sistema precisa de estar em um dos estados 'HomePage' ou 'Perfil'.

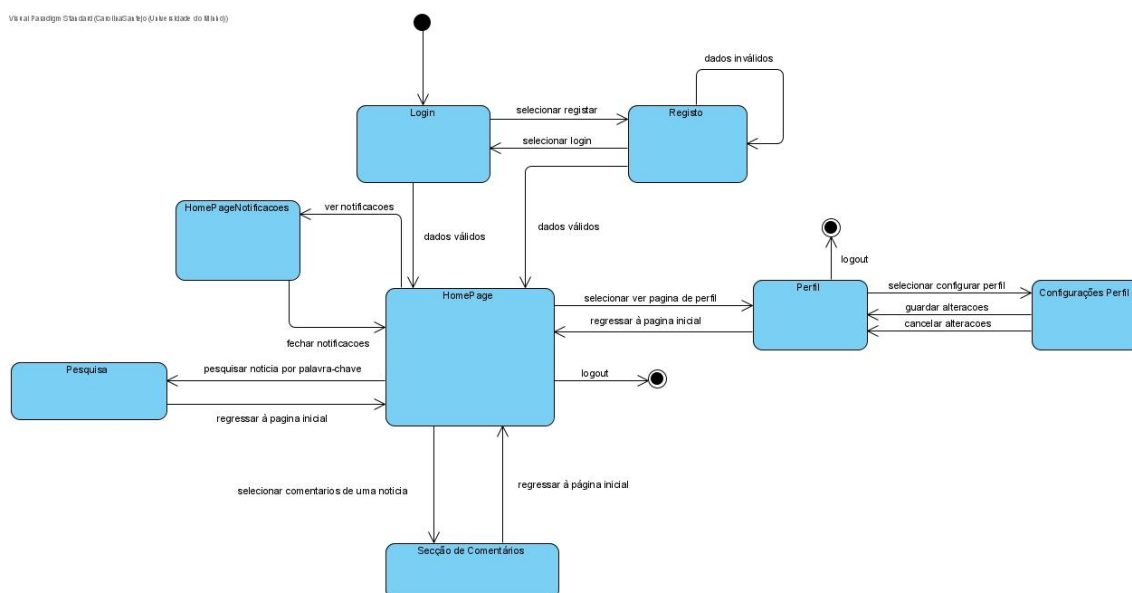


Figura 25 - Modelo máquina de estado

15. Arquitetura da Camada de Dados

15.1. Caracterização das Entidades

Para o desenvolvimento do modelo lógico, foi necessário considerar as entidades mais relevantes ao sistema, e sobre as quais queremos armazenar informação. Desta forma, as entidades presentes são as seguintes: Utilizador, *Country*(País), Jornalista, Categoria, Notícia, Publicação e ArtigoOpinião.

Entidade: Utilizador	
Atributo	Domínio
Username	Varchar(30)
Email	Varchar(45)
Password	Varchar(45)
Nome	Varchar(45)
Cidade	Varchar(45)
Fotografia	Varchar(45)
IdCountry	Varchar(10)

Entidade: Country	
Atributo	Domínio
IdCountry	Varchar(10)
Nome	Varchar(45)

Entidade: Categoria	
Atributo	Domínio
IdCategoria	Int
Nome	Varchar(30)

Entidade: Jornalista	
Atributo	Domínio
IdJornalista	Int
Username	Varchar(30)
Biografia	MediumText
LinkedIn	Varchar(45)

Entidade: ArtigoOpinião	
Atributo	Domínio
IdJornalista	Int
Conteúdo	Longtext
IdPublicação	Int

Entidade: Notícia	
Atributo	Domínio
Url	Mediumtext
Autor	Varchar(45)
IdPublicação	Int

Entidade: Publicação	
Atributo	Domínio
IdPublicacao	Int
Título	Varchar(45)
Data	Datetime
Descrição	Tinytext
Imagem	Mediumtext
Fotografia	Varchar(45)

15.2. Caracterização dos relacionamentos entre entidades

Após a identificação das entidades, é necessário entender como é que elas se relacionam entre si.

Para as relações N:M, foram criadas as tabelas Utilizador_Categoria, utilizador_Jornalista, Utilizador_Noticia, Comentário e Notificação.

Por outro lado, nos relacionamentos 1:N, a entidade X fica com uma chave estrangeira referente à entidade Y. um exemplo presente neste trabalho, é a relação entre o Utilizador e o *Country*, sendo que o Utilizador fica com a chave estrangeira referente ao país que escolheu.

Além disto, é importante realçar que para representar no modelo lógico, o facto de que um artigo de opinião ou notícia são uma publicação, decidiu-se que tanto a entidade Notícia como ArtigoOpinião contém uma chave estrangeira (que também é chave primária) referente à entidade Publicação. O mesmo acontece para o jornalista e o utilizador (um jornalista é um utilizador), à exceção de que a chave estrangeira no jornalista referente ao Utilizador não faz parte da chave primária.

15.3. Diagrama do Modelo Lógico

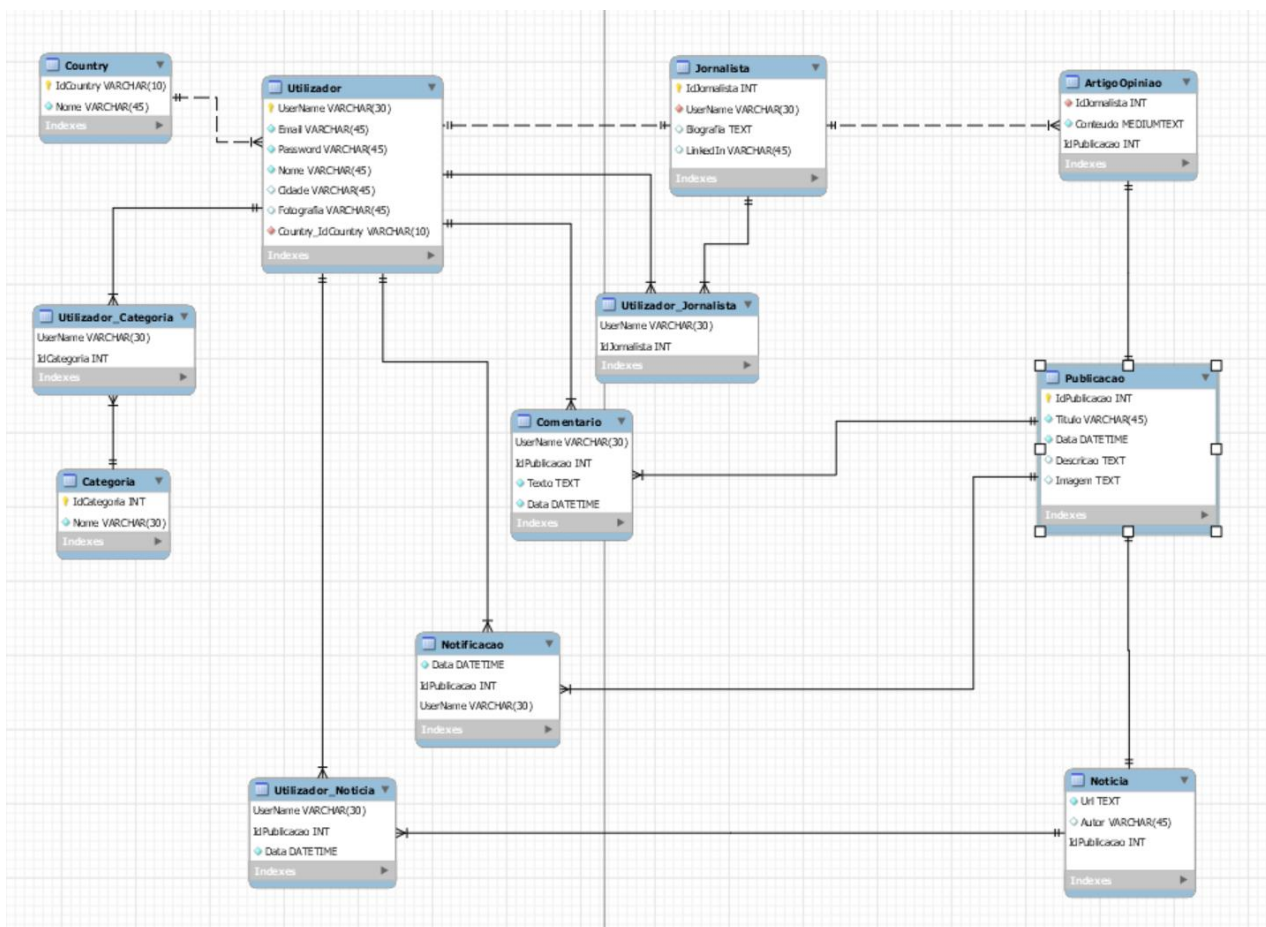


Figura 26 - Diagrama do Modelo Lógico

15.4. Estimativa de Crescimento Anual

Nesta fase do projeto, desenvolveu-se apenas o modelo lógico da base de dados, sendo que o modelo físico e o povoamento serão feitos na próxima fase. No entanto, de forma a obter uma estimativa razoavelmente credível do crescimento anual desta BD, consideremos o seguinte:

- Existem 10 utilizadores no sistema;
- Existem 3 jornalista no sistema;
- Existem 3 países (*country*);
- Existem 6 categorias;
- Existem 10 publicações sendo 5 do tipo notícia e 5 do tipo artigo de opinião;
- Cada um dos utilizadores tem apenas uma categoria preferida;
- Cada utilizador segue apenas um jornalista;
- Cada utilizador fez um comentário numa publicação;
- Cada utilizador recebeu uma notificação;
- Cada utilizador tem uma notícia guardada;

É de realçar que tamanho atual de uma base de dados, obtém-se, para cada tabela existente, multiplicando o tamanho dessa tabela com o seu número de entrada.

Entidade: Utilizador		
Atributo	Tipo	Tamanho
Username	Varchar(30)	30 bytes
Email	Varchar(45)	45 bytes
Password	Varchar(45)	45 bytes
Nome	Varchar(45)	45 bytes
Cidade	Varchar(45)	45 bytes
Fotografia	Varchar(45)	45 bytes
IdCountry	Varchar(10)	10 bytes
265 bytes * 5 users = 1325 bytes		

Entidade: Country		
Atributo	Tipo	Tamanho
IdCountry	Varchar(10)	10 bytes
Nome	Varchar(45)	45 bytes
55 bytes * 3 países = 165 bytes		

Entidade: Categoria		
Atributo	Domínio	Tamanho
IdCategoria	Int	4 bytes
Nome	Varchar(30)	30 bytes
		34 bytes * 6 categorias = 204 bytes

Entidade: Jornalista		
Atributo	Domínio	Tamanho
IdJornalista	Int	4 bytes
Username	Varchar(30)	30 bytes
Biografia	Text	65535 bytes
LinkedIn	Varchar(45)	45 bytes
		65614 bytes * 3 jornalistas = 196842 bytes

Entidade: ArtigoOpinião		
Atributo	Domínio	Tamanho
IdJornalista	Int	4 bytes
Conteúdo	Mediumtext	16 MiB
IdPublicação	Int	4 bytes
		16777224 bytes * 5 artigos = 83886120 bytes

Entidade: Notícia		
Atributo	Domínio	Tamanho
Url	Text	65535 bytes
Autor	Varchar(45)	45 bytes
IdPublicação	Int	4 bytes
		65584 bytes * 5 notícias = 327920 bytes

Entidade: Publicação		
Atributo	Domínio	Tamanho
IdPublicacao	Int	4 bytes
Título	Varchar(45)	45 bytes
Data	Datetime	8 bytes
Descrição	Text	65535 bytes
Imagem	Text	65535 bytes
Fotografia	Varchar(45)	45 bytes
		131172 bytes * 10 publicações = 1311720 bytes

Entidade: Comentário		
Atributo	Domínio	Tamanho
Username	Varchar(30)	30 bytes
Data	Datetime	8 bytes
IdPublicação	Int	4 bytes
Texto	Text	65535 bytes
65577 bytes * 10 comentários = 655770 bytes		

Entidade: Notificação		
Atributo	Domínio	Tamanho
Data	Datetime	8 bytes
Username	Varchar(30)	30 bytes
IdPublicação	Int	4 bytes
42 bytes* 10 notificações = 420 bytes		

Entidade: Utilizador_Noticia		
Atributo	Domínio	Tamanho
Data	Datetime	8 bytes
Username	Varchar(30)	30 bytes
IdPublicação	Int	4 bytes
42 bytes * 10 notícias guardadas = 420 bytes		

Entidade: Utilizador_Jornalista		
Atributo	Domínio	Tamanho
IdJornalista	Int	4 bytes
Username	Varchar(30)	30 bytes
34 bytes * 10 = 340 bytes		

Entidade: Utilizador_Categoria		
Atributo	Domínio	Tamanho
IdCategoria	Int	4 bytes
Username	Varchar(30)	30 bytes
34 bytes * 10 = 340 bytes		

Tendo isto em conta, o tamanho da nossa base de dados é o seguinte:

Tamanho

$$\begin{aligned} &= 340 + 340 + 420 + 420 + 655770 + 1311720 + 327920 + 83886120 \\ &+ 196842 + 204 + 165 + 1325 = 86381656 \text{ bytes} \approx 86,4 \text{ MB} \end{aligned}$$

Consideremos que, anualmente, a nossa base de dados verifica um aumento de 10% no número de utilizadores e de jornalista e que o número de países e de categorias não sofre nenhuma alteração.

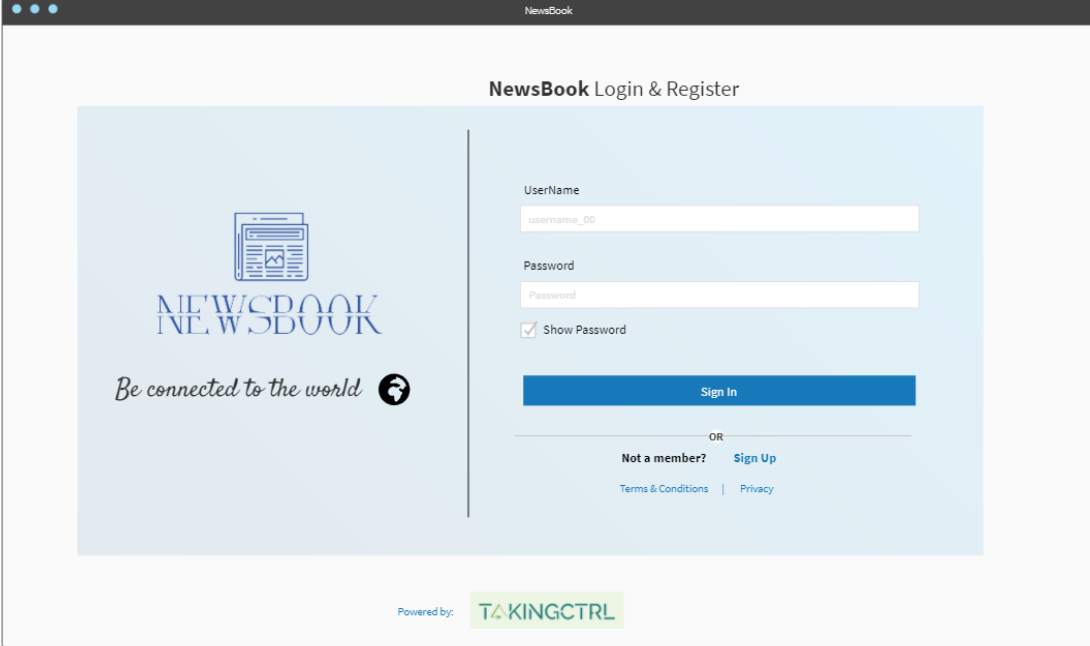
Desta forma, podemos admitir que todas as tabelas à exceção das tabelas Publicação, Notícias, Categoria e País aumentam também cerca de 10%.

Assim, a taxa de crescimento anual da BD será o seguinte:

Taxa de crescimento =

$$\begin{aligned} &0,10 * (1325 + 340 + 196842 + 8388620 + 340 + 655770 + \\ &420 + 420) = 924407,7 \text{ bytes/ano} \end{aligned}$$

16. Caracterização do Sistema de Interface



The screenshot displays a web browser window titled "NewsBook". The main heading is "NewsBook Login & Register". The interface is split into two sections. The left section features the "NEWSBOOK" logo, which includes a newspaper icon, and the tagline "Be connected to the world" with a globe icon. The right section contains the login and registration forms. It includes input fields for "UserName" (with the placeholder "username_00") and "Password" (with the placeholder "Password"). There is a checkbox labeled "Show Password" which is currently checked. A blue "Sign In" button is positioned below the password field. Below the "Sign In" button, the text "OR" is centered. Underneath "OR", there is a link "Not a member? Sign Up". At the bottom of the right section, there are two links: "Terms & Conditions" and "Privacy". At the very bottom of the page, there is a footer that says "Powered by: TAKINGCTRL" with a logo.

NewsBook Login & Register

UserName
username_00

Password
Password

☒ Show Password

Sign In

OR

Not a member? [Sign Up](#)

[Terms & Conditions](#) | [Privacy](#)

Powered by: TAKINGCTRL

Figura 27 - Interface Página de *Login*

NewsBook Login & Register

***Email**
name@domain.com

***Password**
Password
☒ Show Password

***Your name**
First Name & Last Name

***UserName**
username_00

***Country**
Select

City
Lisbon

Favorite Categories
Select

Are you a journalist? ☒

Profile Picture
click here to upload

Powered by: TAKINGCTRL

Already a member? [Sign In](#)
[Terms & Conditions](#) | [Privacy](#)

Figura 28 - Interface página de registo

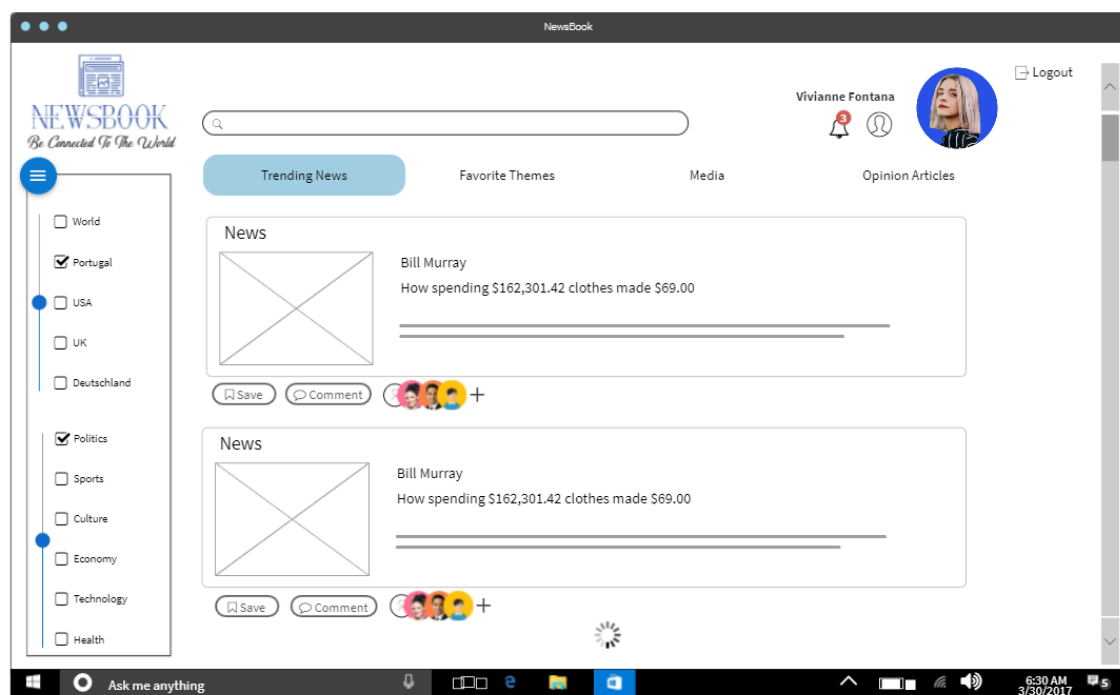


Figura 29 - Interface HomePage

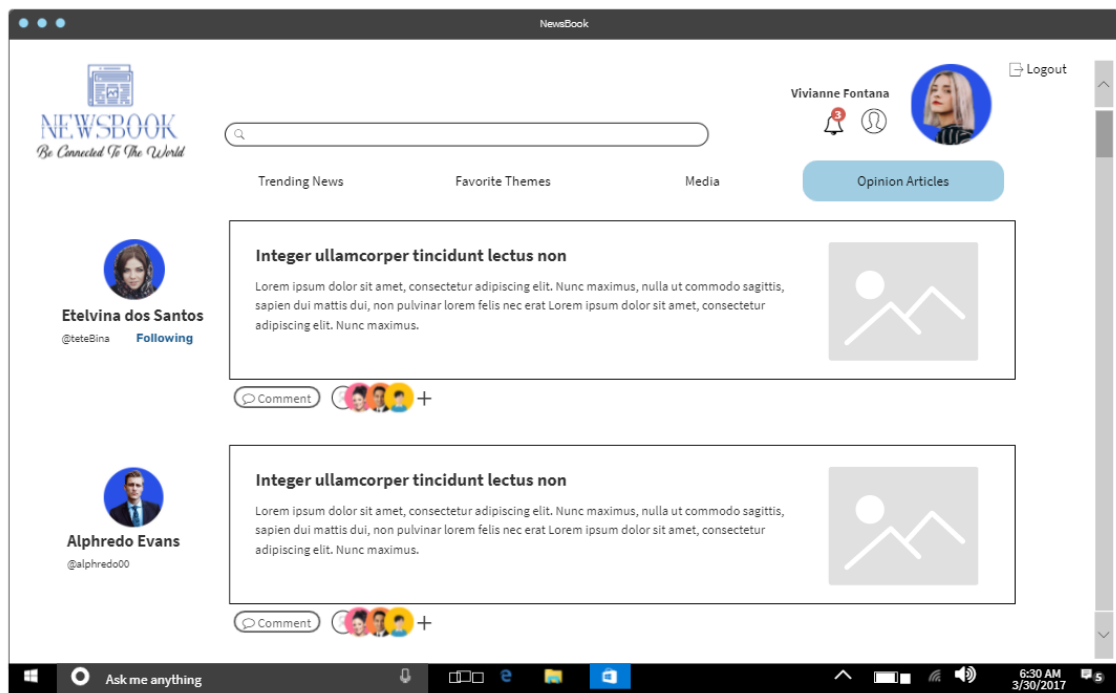


Figura 30 - Interface *HomePage*

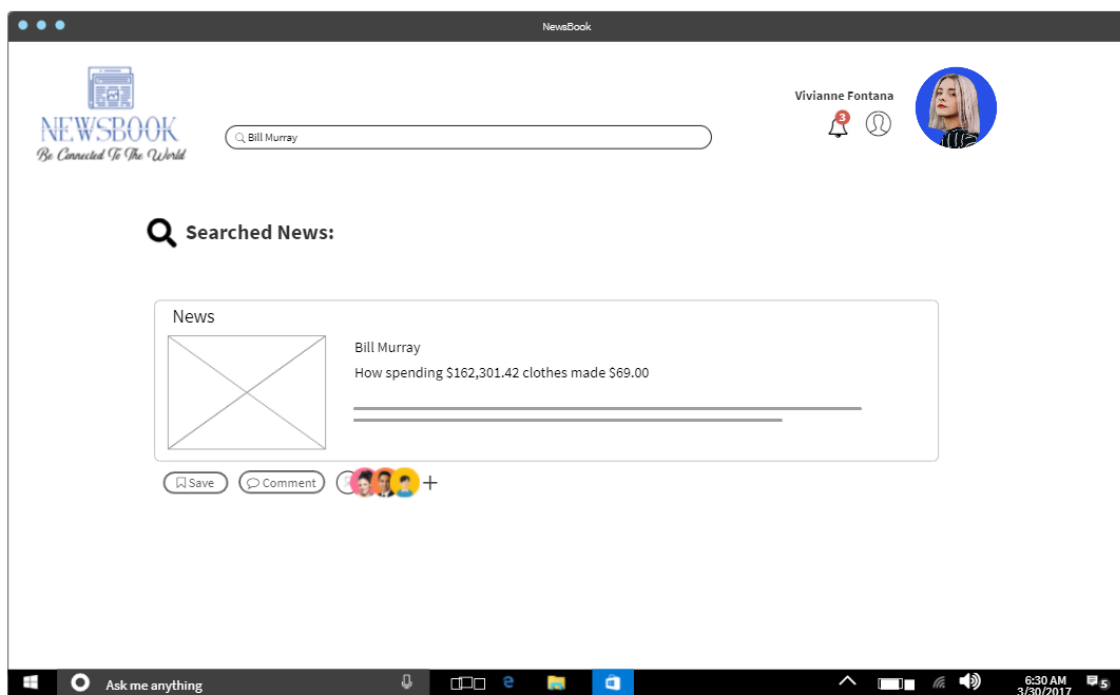


Figura 31 - Interface de procura de notícias por palavra-chave

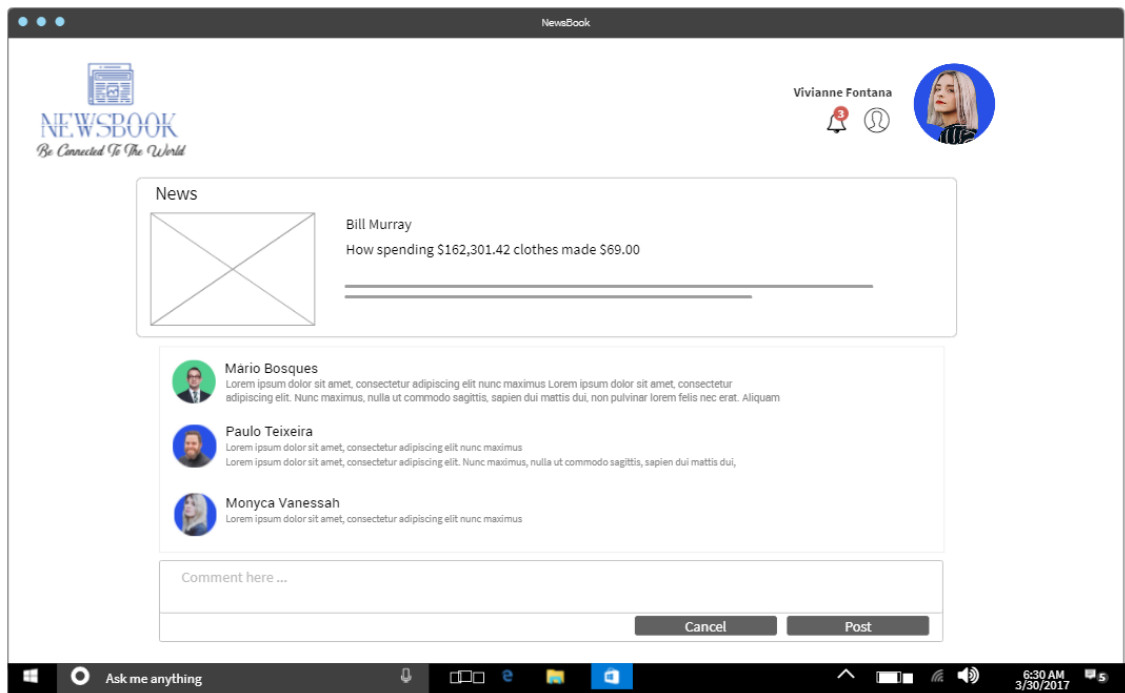


Figura 32 - Interface de secção de comentários

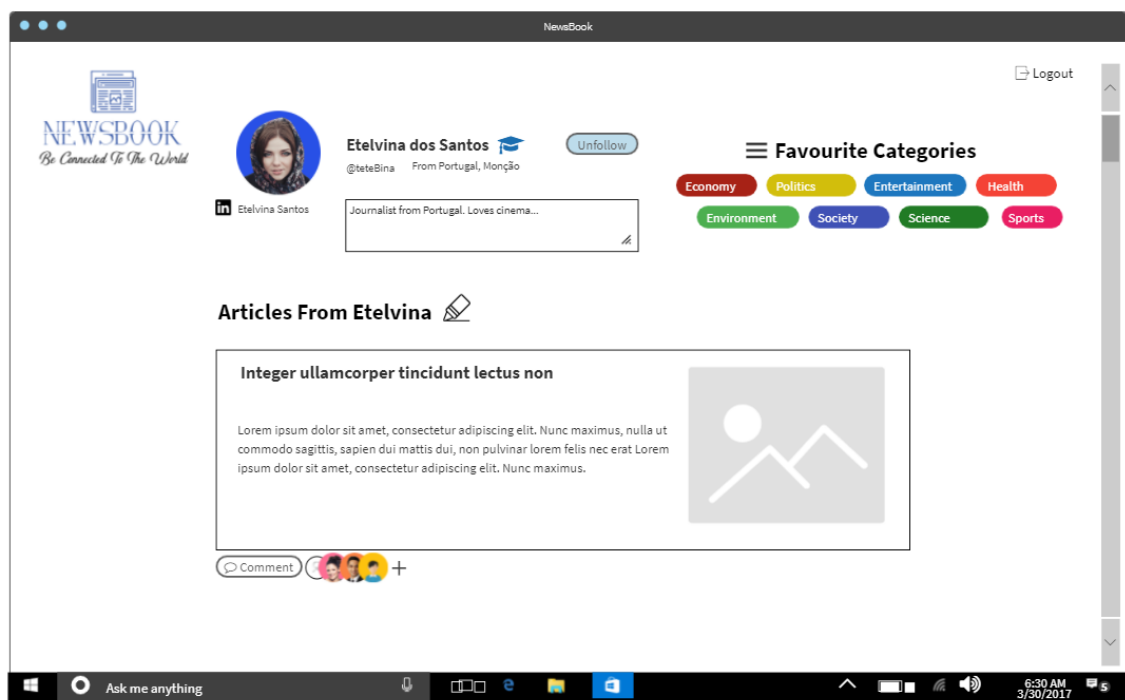


Figura 33 - Interface de Utilizador a visualizar perfil de jornalista

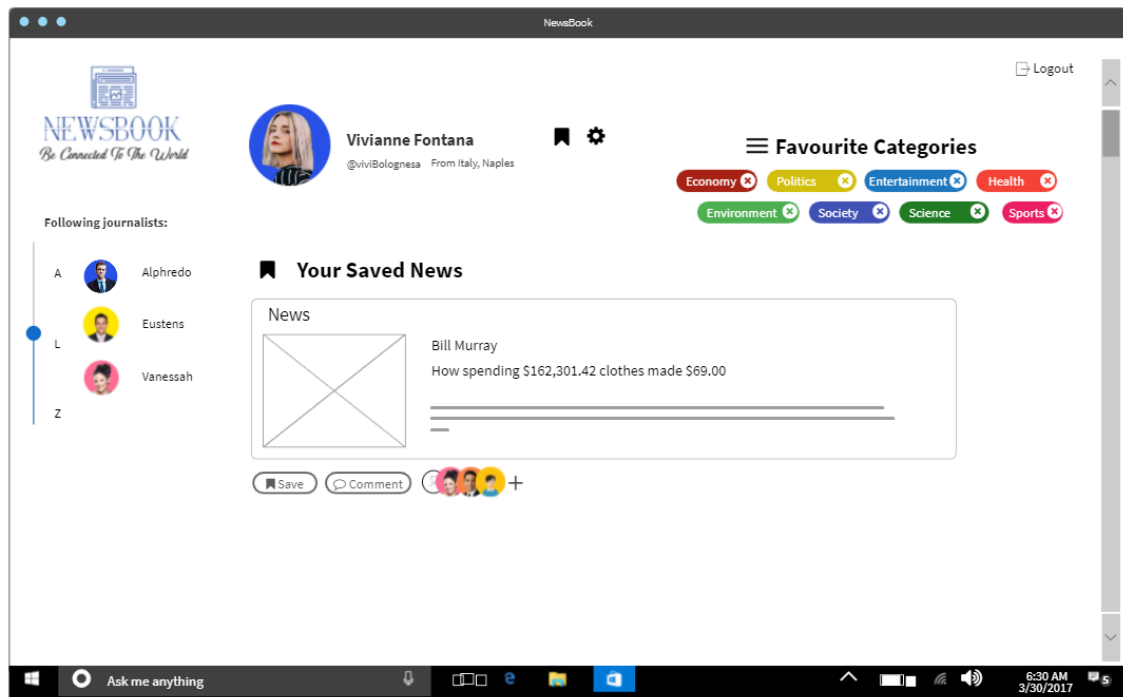


Figura 34 - Interface perfil de utilizador

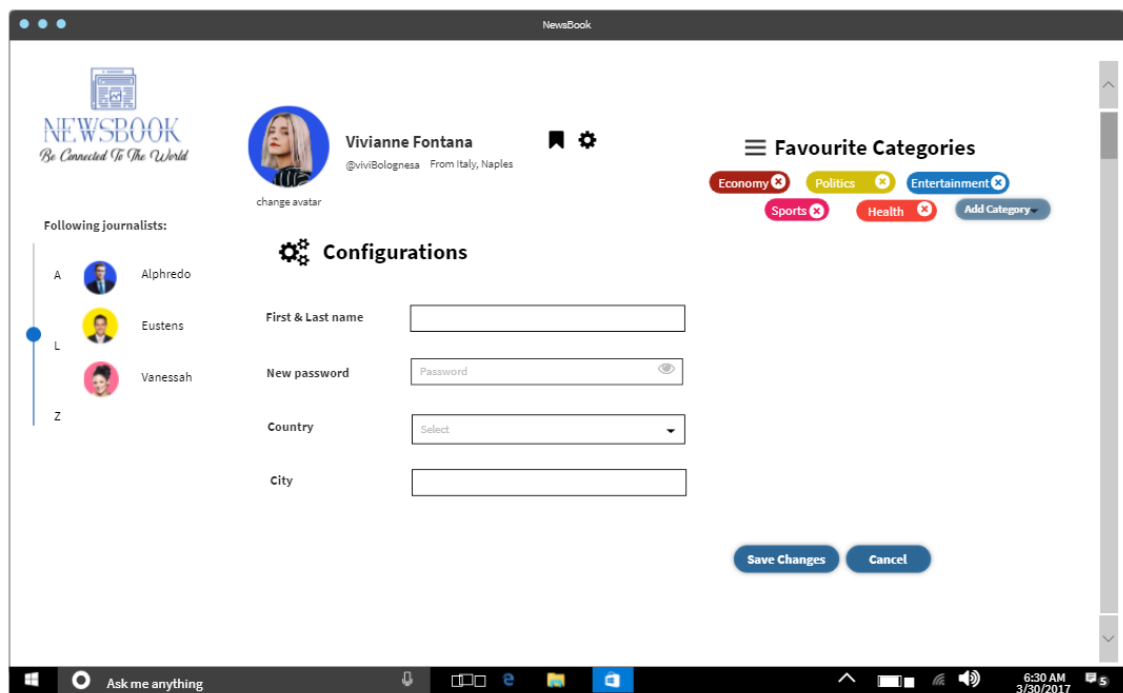


Figura 35 - Interface configuração de perfil de utilizador

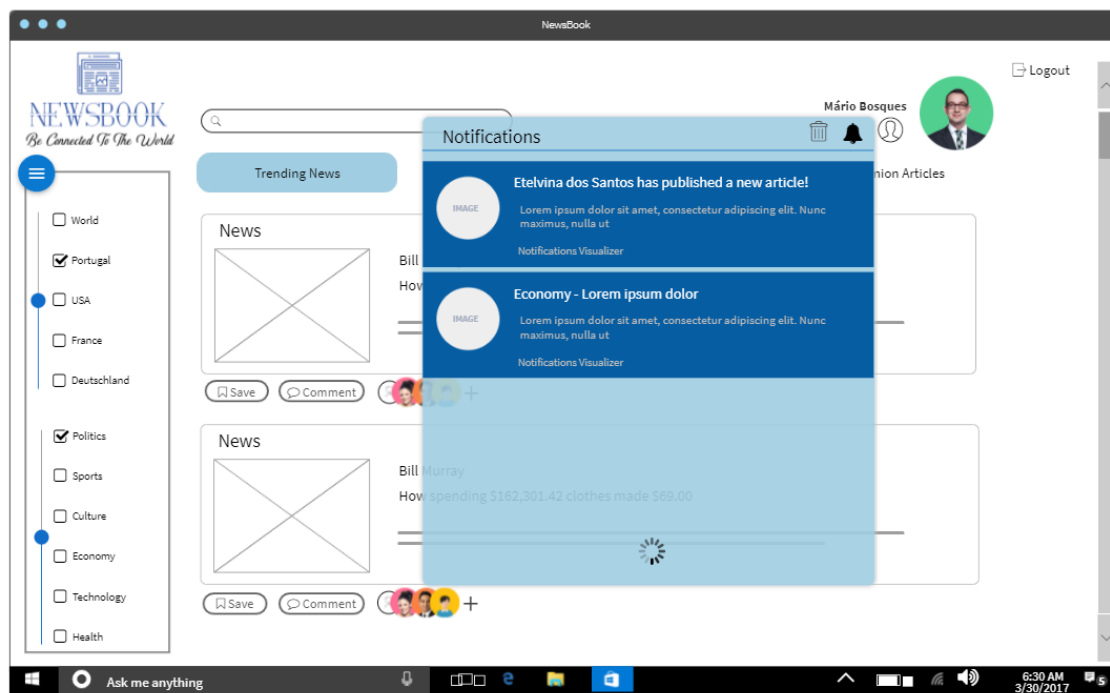


Figura 36 - Interface *HomePage* com notificações

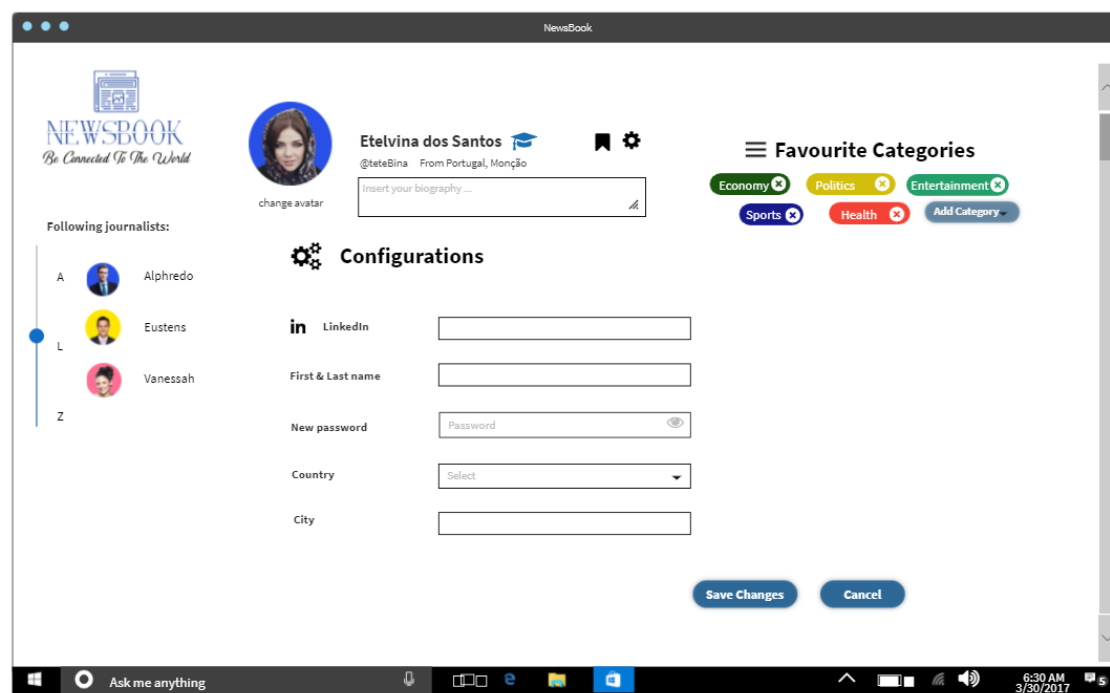


Figura 37 - Interface configuração de perfil de jornalista

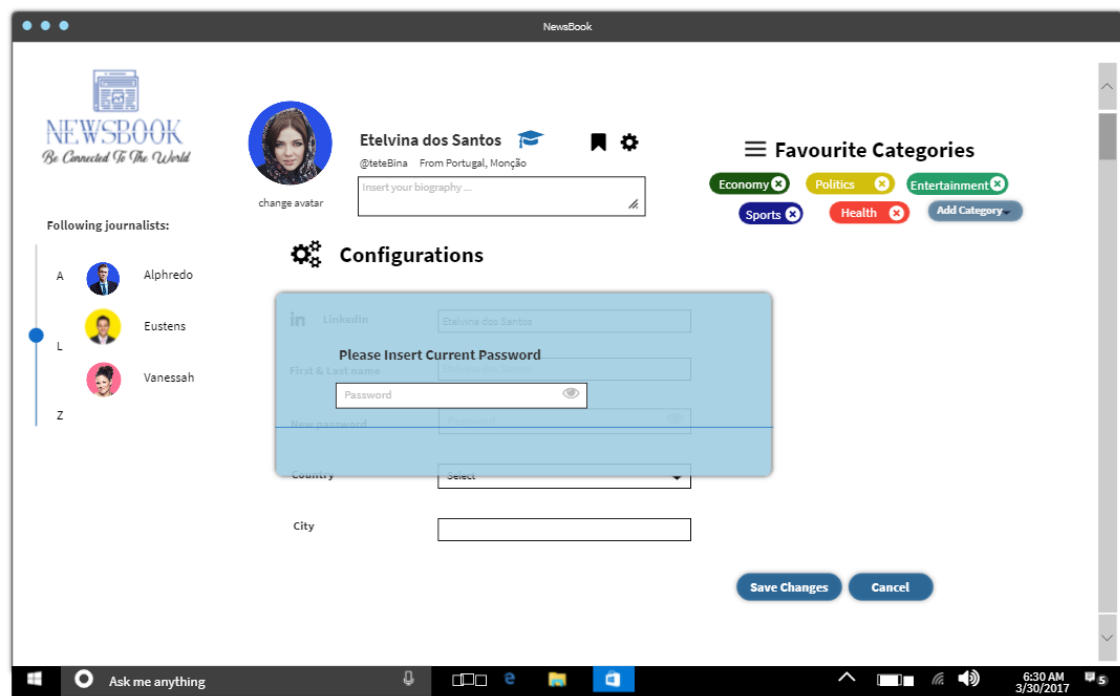


Figura 38 - Interface configuração de perfil de jornalista

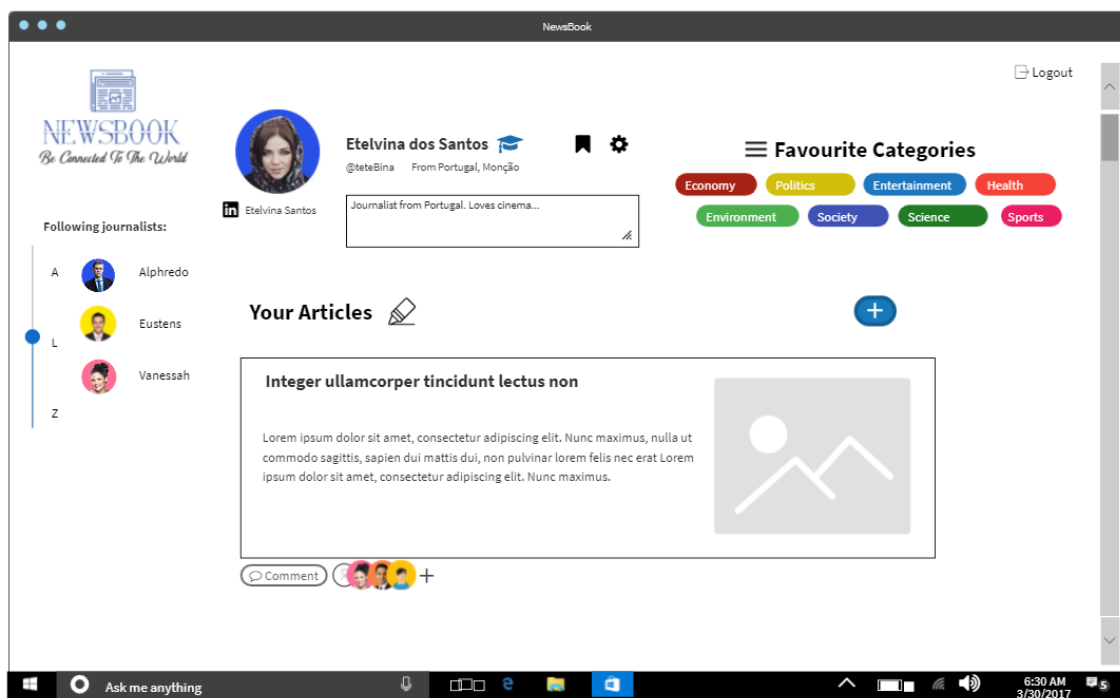


Figura 39 - Interface perfil de jornalista

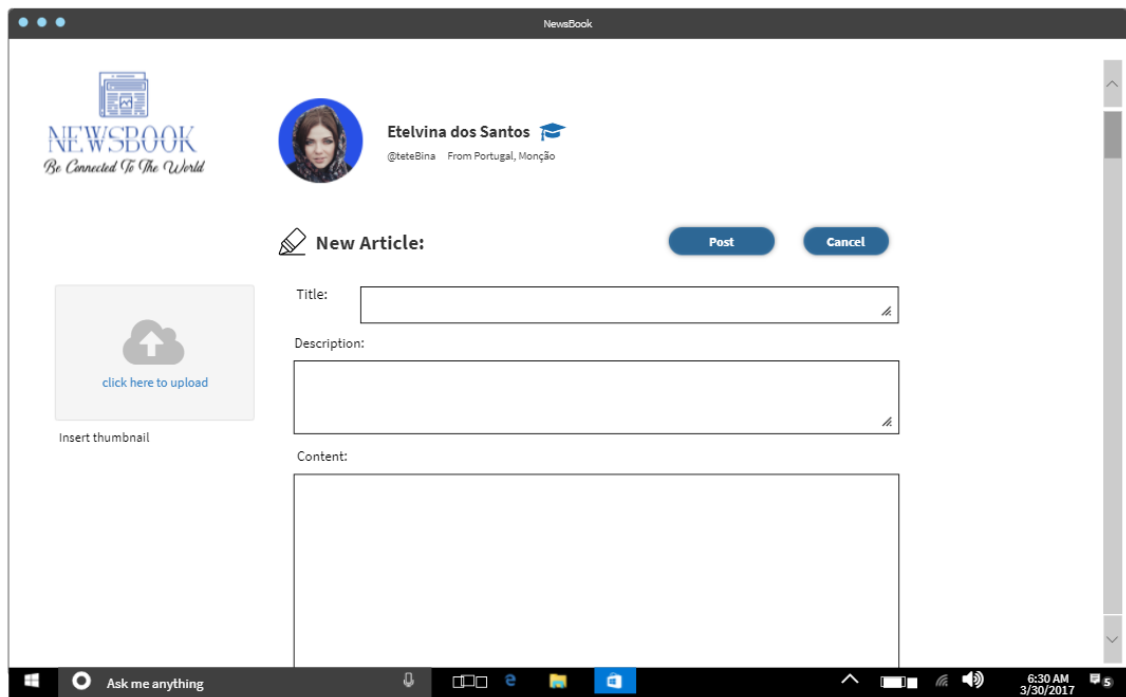


Figura 40 - Interface de adição de um novo artigo de opinião

17. Implementação

Após a conclusão da segunda fase do trabalho prático, passa-se agora para Implementação do projeto até agora planeado, sendo o objetivo implementar as funcionalidades, as interfaces, e a base de dados e tudo aquilo que foi apresentado nas fases anteriores.

Desta forma, foi desenvolvida uma aplicação *Web* intuitiva e fácil de usar, que poderá ser usada tanto por utilizadores como jornalistas, que procurem uma plataforma séria e que preserva, acima de tudo, a veracidade das notícias que publica.

18. Metodologia de implementação

O padrão arquitetural escolhido para a implementação da aplicação foi o MVC (*Model View Controller*) com o objetivo de obter um conjunto de componentes modulares de fácil integração e que facilitam o desenvolvimento em paralelo.

A escolha do MVC, baseou-se, também, no facto de que, este padrão arquitetural, proporciona uma camada de abstração, que facilita a implementação da aplicação em diferentes plataformas, sem que haja alteração daquilo que já está feito. Isto acontece porque a estrutura interna dos dados e procedimentos torna-se independente da interface com o utilizador.

19. Ferramentas utilizadas na implementação

A principal ferramenta usada na implementação da solução de *software* foi as *frameworks* de desenvolvimento ASP.NET Core. Esta ferramenta foi escolhida devido à sua simplicidade de utilização e ao facto de estar muito bem documentada. Recorreu-se, também, a uma *feature* muito útil do ASP.NET, o *Blazor*, para a construção das *web UIs* interativas usando o C# e não o típico *JavaScript*.

Além disto, utilizou-se o motor de base de dados *Microsoft SQL Server* e a plataforma *Azure Data Studio* para a criação das tabelas das várias entidades e relações, para conseguir ver estas tabelas posteriormente, e para ter acesso ao crescimento e tamanho da base de dados.

Para o IDE, escolheu-se o *Visual Studio 2019* visto que, para além de ser um dos IDEs mais usados atualmente e possuir funcionalidades muito úteis ao nível, por exemplo, do *debugging* da aplicação e da partilha de código (*live share*), é também a ferramenta mais aconselhada para o desenvolvimento em ASP.NET.

Por fim, foram usados *browsers* de internet para avaliar a interface visual do utilizador.

20. Desenvolvimento do projeto

20.1. Conexão à base de dados:

Após a instalação e configuração do motor de base de dados *SQL Server da Microsoft*, passou-se à criação da base de dados para a aplicação *Newsbook*.

Quase todo o processo de conexão à Base de Dados foi gerido pela própria *framework* do Asp.Net Core através de alguns ficheiros de configuração que definem como aceder à Base de Dados e as credenciais a utilizar.

Para além disso e para povoar a base de dados, foram introduzidos alguns utilizadores, jornalistas, os países e as categorias das notícias de maneira a garantir desde início existe um programa de base 100% funcional.

20.2. Pesquisa de notícias:

Sendo o principal objetivo da *NewsBook*, a divulgação de diversas notícias de fontes variadas, recorreu-se ao auxílio de uma API externa designada por *NewsApi*, capaz de fornecer essas mesmas notícias.

Com a utilização desta API, foi possível implementar funcionalidades tais como a seleção de notícias de acordo com diversos critérios nomeadamente por país, categoria ou palavra-chave.

21. Produto Final

Ao longo deste semestre a equipa da *TakingCtrl* desenvolveu a aplicação *Newsbook* até que o produto de software final fosse semelhante ao que foi idealizado em fases anteriores. Em seguida, será demonstrado como a aplicação funciona e onde se podem encontrar todas as funcionalidades implementadas.

21.1. Página inicial (Sem sessão iniciada):

A página de *login* é a página inicial da aplicação. Qualquer utilizador que não esteja registado ou que não esteja autenticado, é para aqui redirecionado. Nesta página é possível um utilizador autenticar-se ou efetuar o registo.

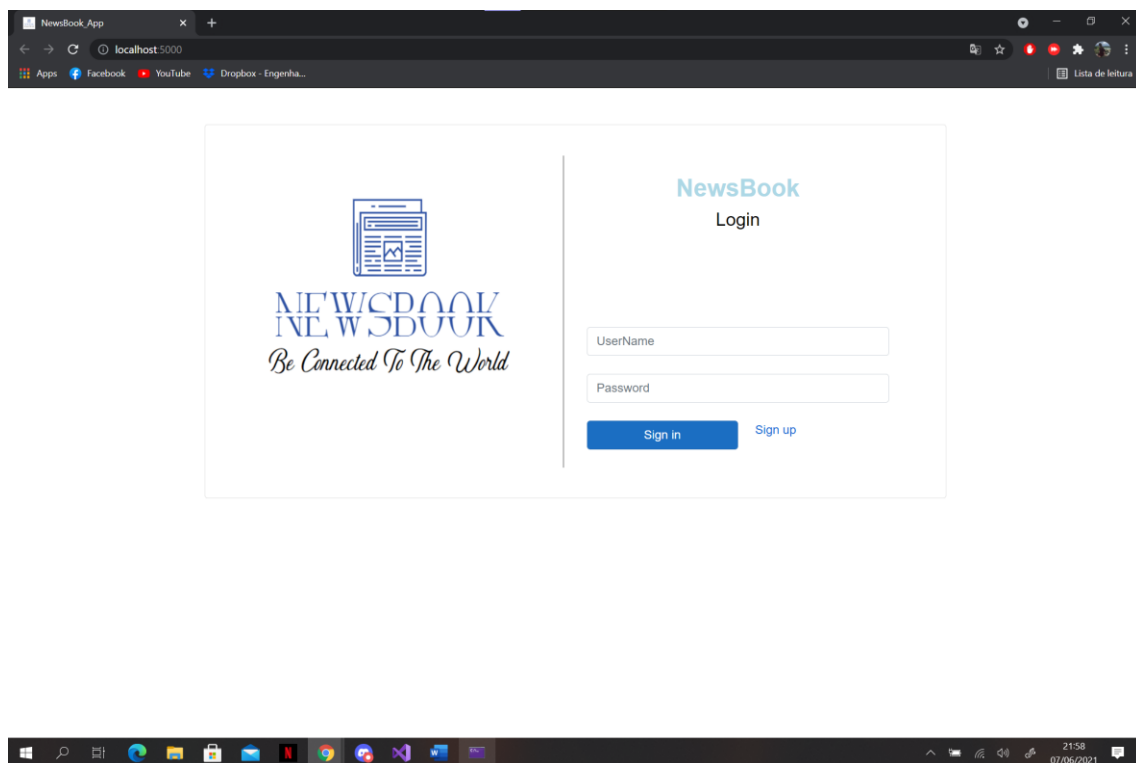
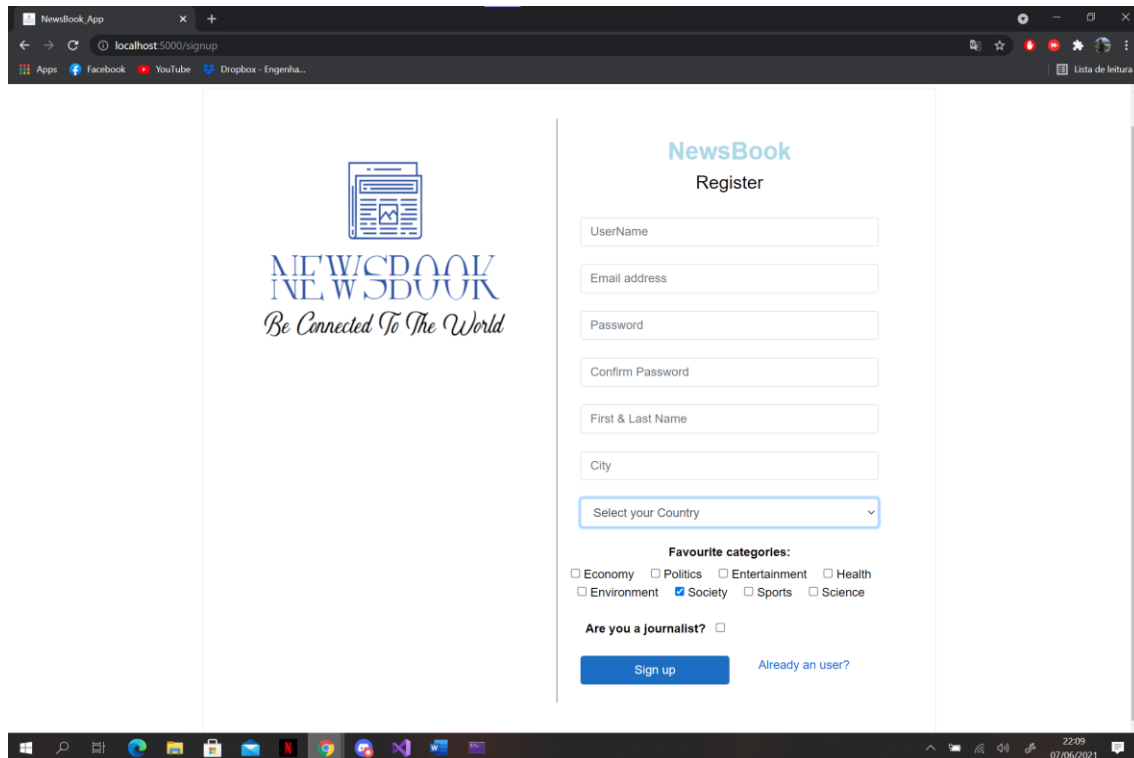


Figura 41 - Login

Se o utilizador seleccionar a opção *Sign up* é então redirecionado para a página onde se efetua o registo. Caso se deseje voltar para a página do *login*, deve-se seleccionar a opção *Already a user?*



The screenshot shows a web browser window with the address bar displaying 'localhost:5000/signup'. The page features the NewsBook logo on the left and a registration form on the right. The form includes input fields for Username, Email address, Password, Confirm Password, First & Last Name, and City. Below these is a dropdown menu for 'Select your Country'. Further down, there is a section for 'Favourite categories' with checkboxes for Economy, Politics, Entertainment, Health, Environment, Society (which is checked), Sports, and Science. At the bottom of the form, there is a checkbox for 'Are you a journalist?' and two buttons: 'Sign up' (highlighted in blue) and 'Already an user?'.

Figura 42 - Registo

21.2. Página inicial (Com sessão iniciada):

Quando um utilizador se regista ou autenticação, é redirecionado para a página principal da aplicação. Aqui é possível ver todas as opções de países e categorias disponíveis, bem como seleccioná-los de forma a filtrar as notícias. Além disto, consegue-se ver as *trending news*, e pode-se aceder a outras páginas da aplicação.

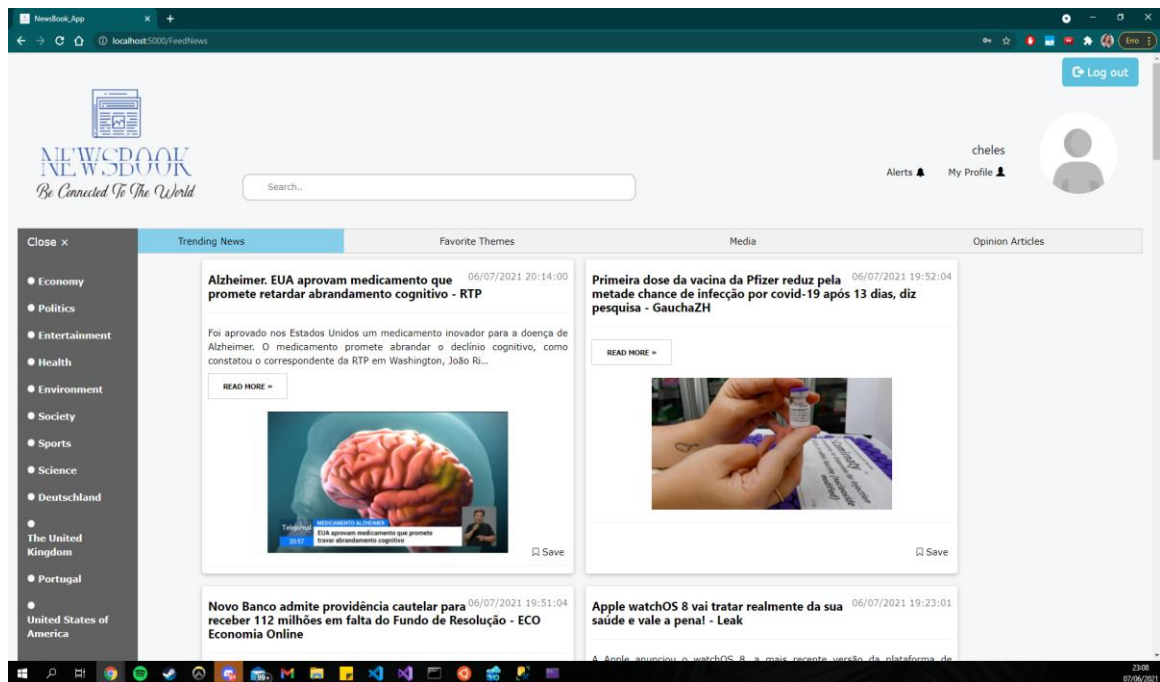


Figura 43 - HomePage (topo)

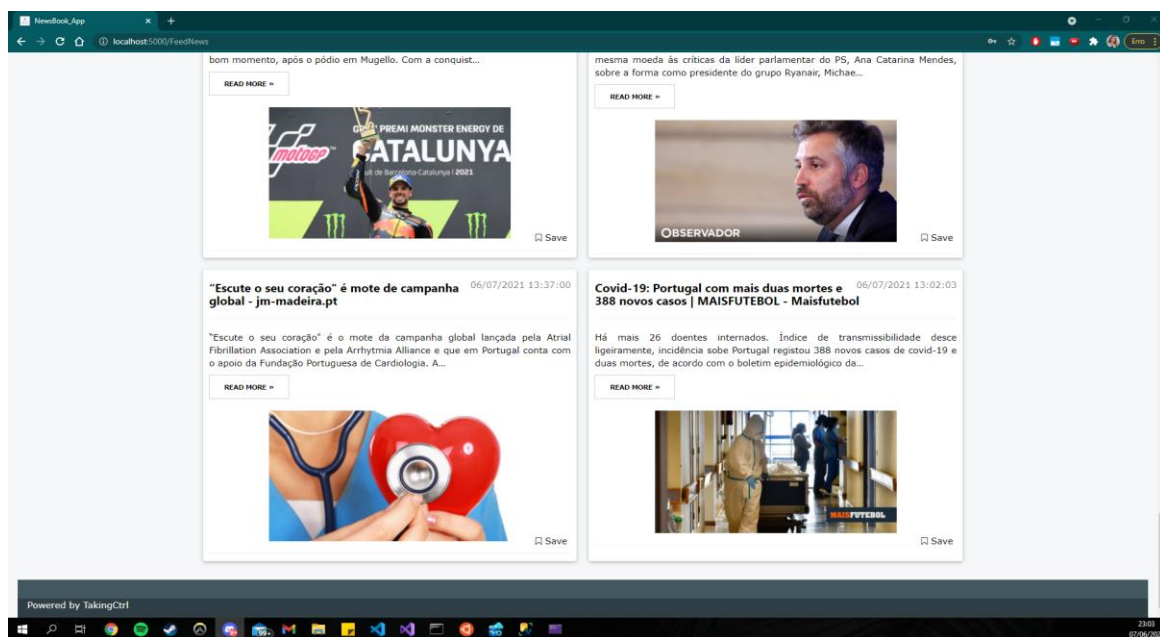


Figura 44 - HomePage (final da página)

21.3. Perfil do utilizador:

Na página do perfil do utilizador, é possível ter acesso a várias informações, nomeadamente, os jornalistas que o *user* segue, as notícias guardadas e categorias favoritas. Nesta página é possível adicionar ou remover categorias favoritas, remover notícias previamente salvas e aceder às configurações do seu perfil.

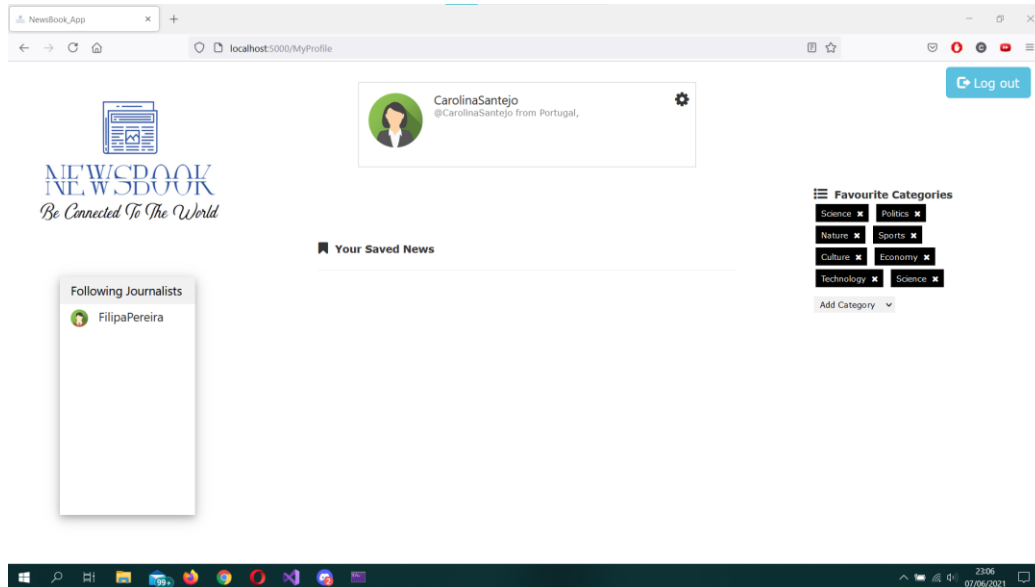


Figura 45 - Perfil de utilizador

21.4. Perfil do jornalista:

O jornalista é também um utilizador, pelo que a sua página de perfil é idêntica à de um utilizador normal. No entanto, difere no sentido em que existe uma secção chamada *your articles* onde poderá ver todos os artigos de opinião que publicou, além de existir um botão onde carrega se quiser escrever um novo artigo. Além disto, o perfil do jornalista pode possuir uma pequena biografia.

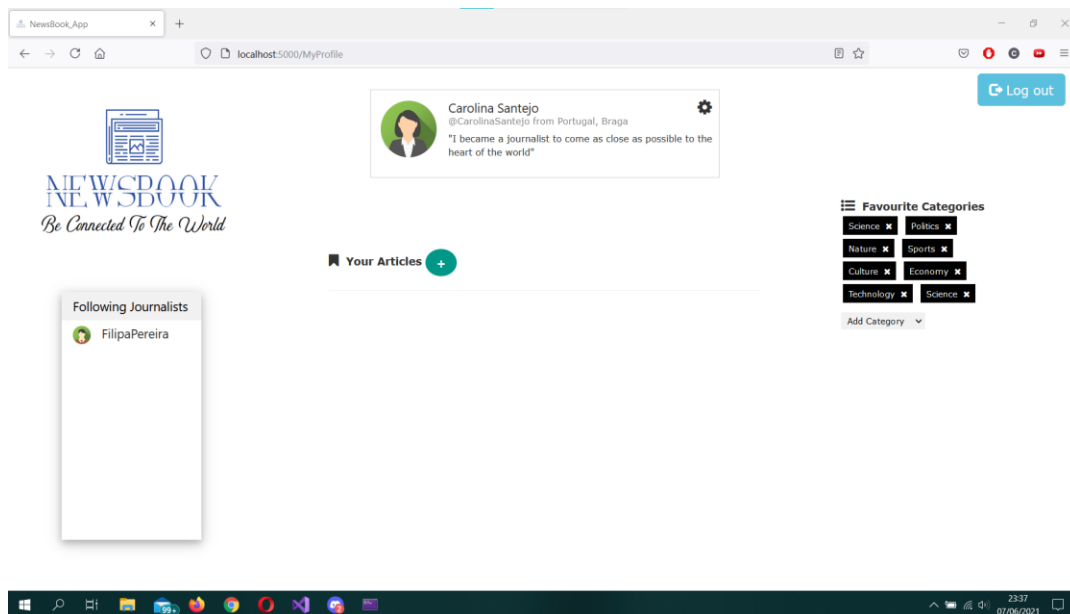


Figura 46 - Perfil de Jornalista

21.5. Configurações do perfil do utilizador:

Quando um utilizador pretende modificar os seus dados pessoais, a partir da página do seu perfil pode dirigir-se à página de configurações. Para salvar alterações é necessário seleccionar a opção *save changes*.

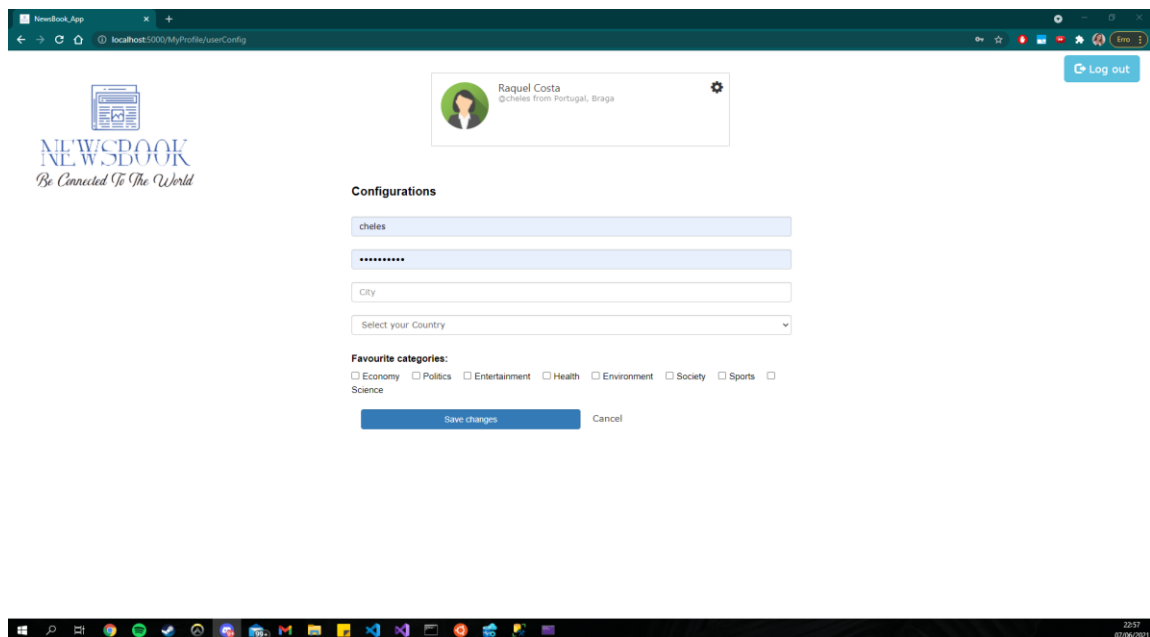


Figura 47 - Configurações de perfil

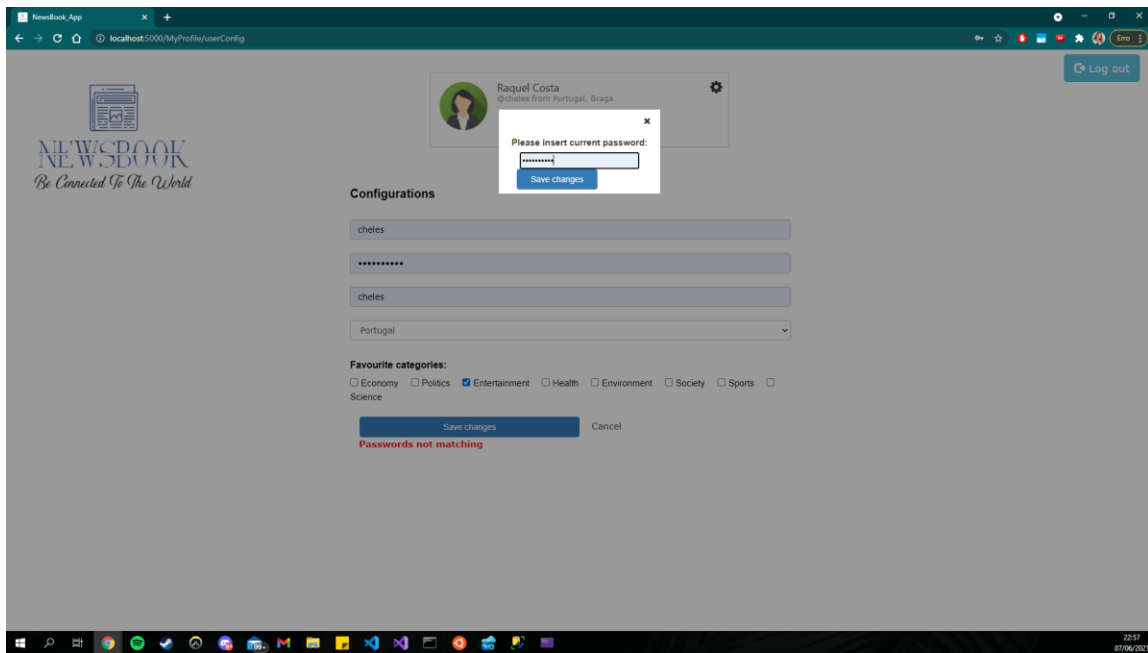


Figura 48 - Confirmação de alterações

21.6. Notificações:

Como já foi referido, os utilizadores possuem categorias favoritas. Desta forma, cada vez que uma notícia de uma categoria favorita de um *user* é publicada, ele recebe uma notificação de aviso. É possível apagar o pop up ou apagar a notificação em si. É também possível apagar todas as notificações já recebidas.

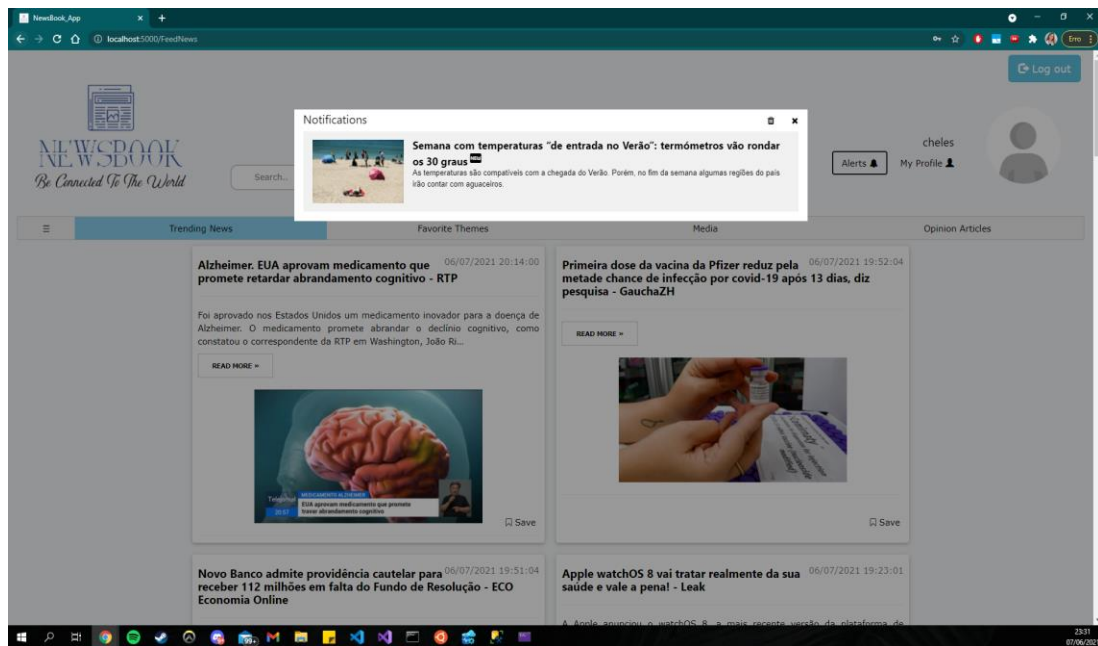


Figura 49 - Notificações

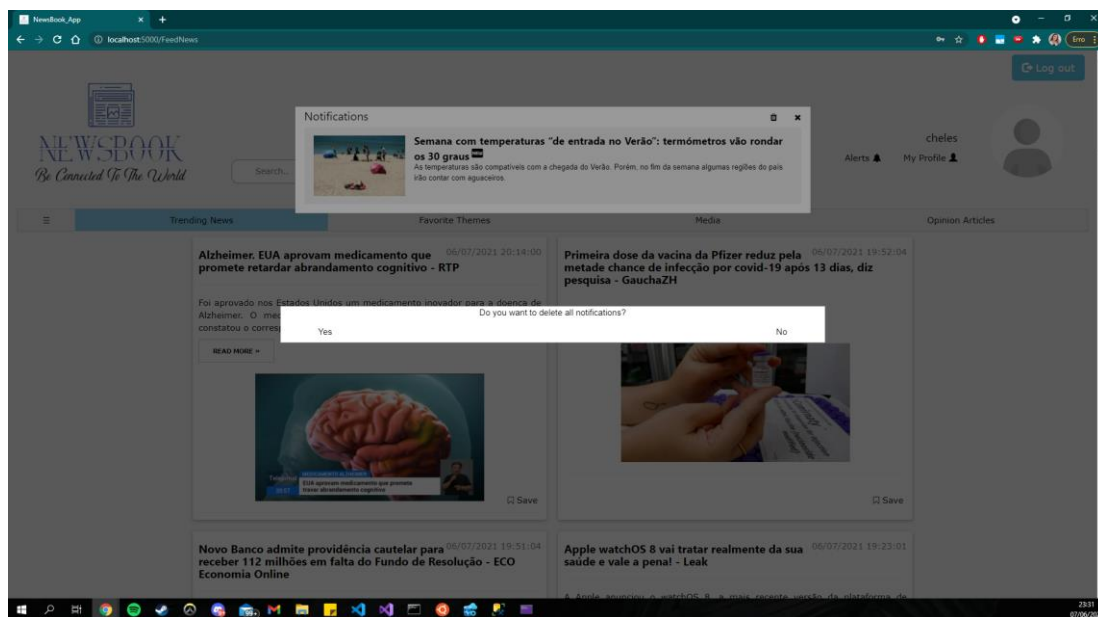


Figura 50 - Limpar notificações

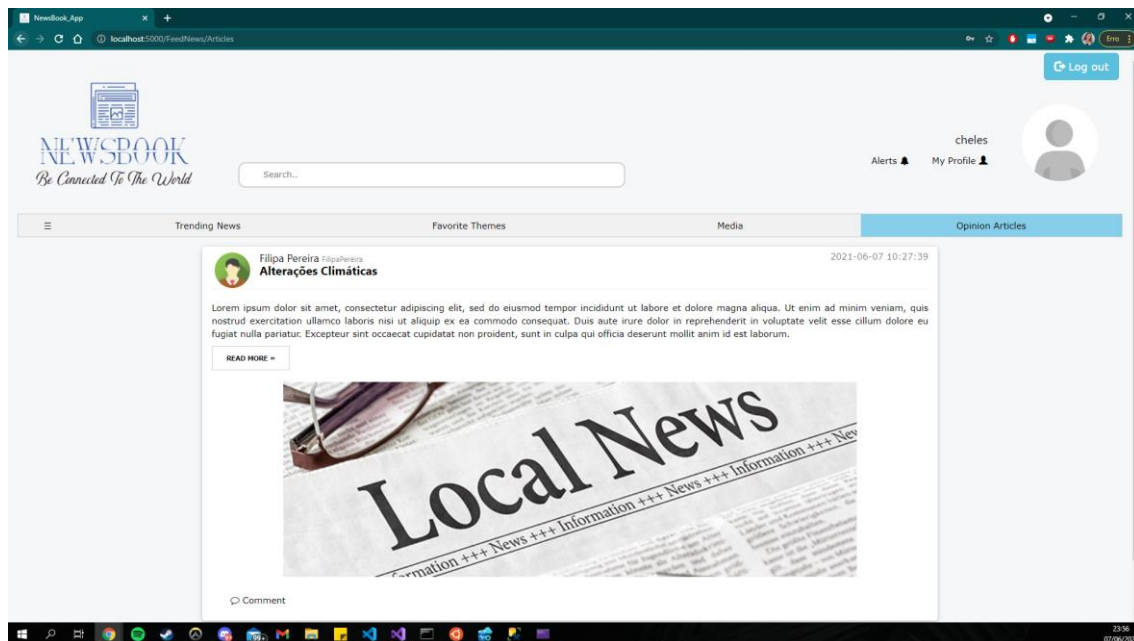


Figura 51 - Interface artigos (parte 1)

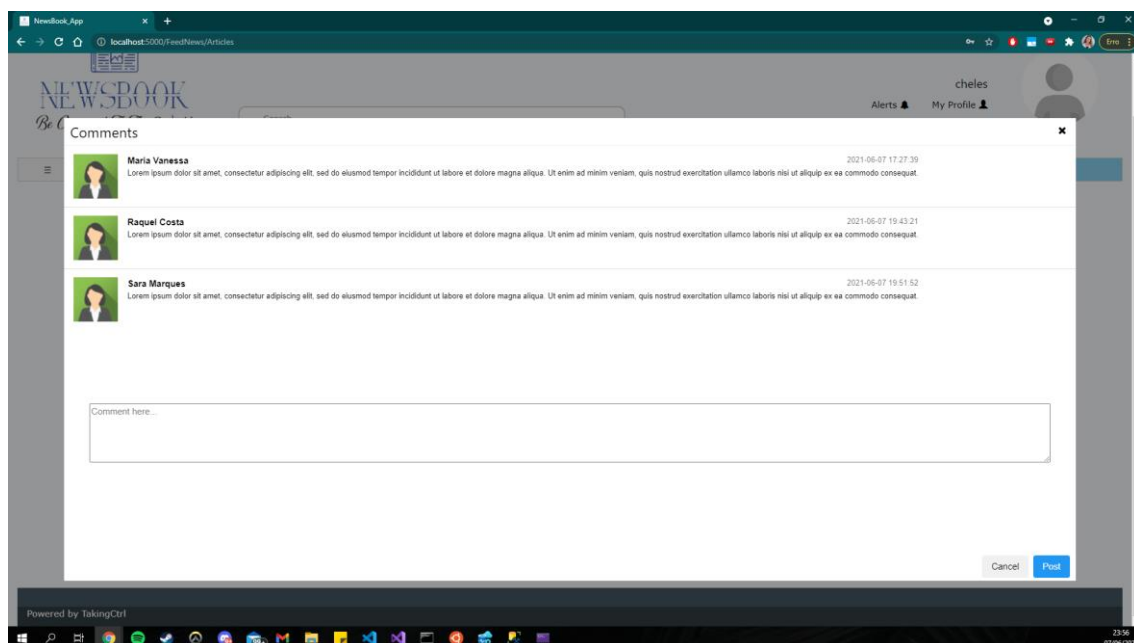


Figura 52 - Interface artigos (parte 2)

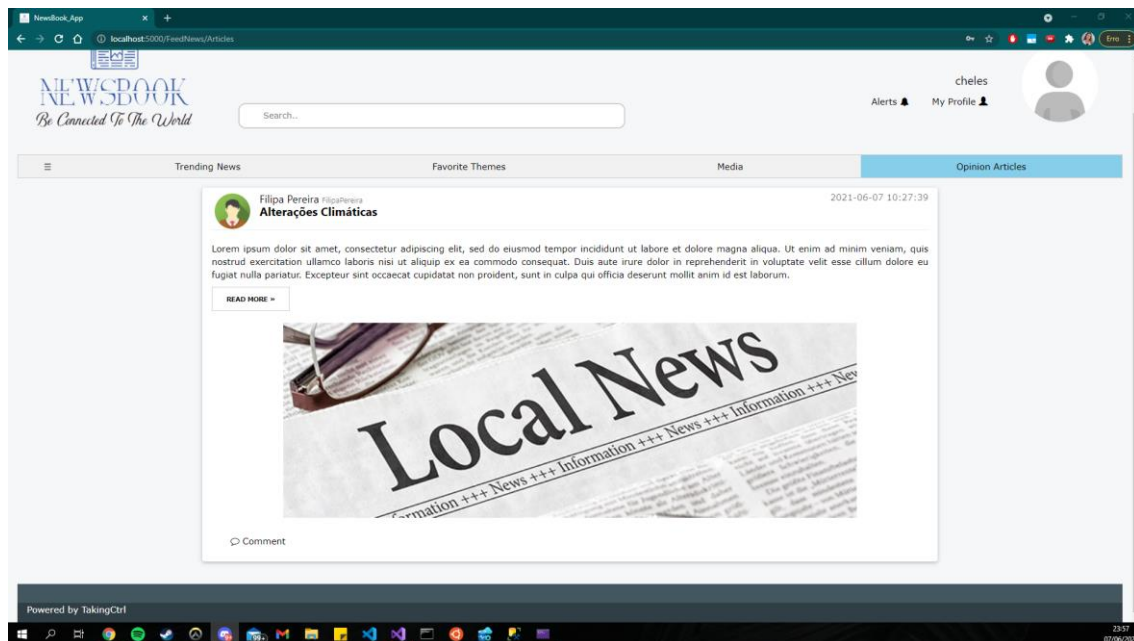


Figura 53 - Interface artigos (parte 3)

22. Conclusões e trabalhos futuros

De forma a incentivar o planeamento de projetos e realçar a sua importância, o desenvolvimento da Newsbook foi feito em três fases distintas.

Na primeira fase – **Fundamentação** – foram definidas as bases principais do nosso trabalho. Foi feita a contextualização do problema, a apresentação do caso de estudo, a análise de viabilidade do projeto, a identificação dos recursos necessários e o modelo do sistema a implementar. Esta é a fase mais importante do desenvolvimento de um projeto, uma vez que uma boa fundamentação é a chave para a criação bem-sucedida do software.

Na segunda fase – **Especificação** – procedeu-se à modelação do projeto de acordo com os princípios ajustados na fase anterior. Foi criado, através da análise detalhada dos requisitos recolhidos, um modelo de domínio, um diagrama de classes, um diagrama de use cases (e as respetivas especificações tabelares), um modelo lógico e uma máquina de estado. Foram feitas, também, as interfaces de apresentação do sistema.

Na terceira e última fase, – **Implementação** – passamos para a produção do software, sendo o resultado final uma aplicação web intuitiva e que contém as funcionalidades mais importantes.

No entanto, é necessário realçar que o projeto não se encontra totalmente finalizado, na medida em que algumas das funcionalidades planeadas acabaram por não ser implementadas na versão final do software. Desta forma, o projeto final é apenas uma “amostra” daquele que foi inicialmente idealizado estando presentes apenas as funcionalidades consideradas mais importantes.

Posto isto, as grandes dificuldades desta fase, consistiram essencialmente no facto de se estar a usar uma nova linguagem de programação e ferramentas de desenvolvimento de software completamente novas, e com as quais nunca tínhamos trabalhado.

Desta forma, e apesar dos diversos obstáculos encontrados durante todo o processo, o grupo considera que concluiu este projeto com sucesso, tendo sido este importante para realçar a relevância de fases de planeamento.

Como sugestão para trabalhos futuros, seria útil implementar todas as funcionalidades inicialmente planeadas e que não estão no produto final, além de ser também interessante desenvolver funcionalidades extra que tornassem a aplicação mais conveniente e útil para o utilizador.

Referências

- Elisa Shearer, M. B. (14 de Julho de 2015). *Pew Research Center Journalism & Media*. Obtido de www.journalism.org: <https://www.journalism.org/2015/07/14/the-evolving-role-of-news-on-twitter-and-facebook/>
- Perez, S. (30 de Julho de 2020). *TechCrunch*. Obtido de techcrunch.com: [WLtlblbw0okBBkGLTfyYkxBIQMhTqNnMYuRb6k23Ww8NxjZe2Tr6HBFAiX2lm6bwBtMhno707ocgwlHmt6t0GeJCJD9aUqjVWYWGb4xr_fatkXxvUo2nz8o3lm_AfJL28aCeRNjjP8Fls_Cchh4d4csiiUqm3x8sy](https://techcrunch.com/2020/07/30/facebook-reports-1-5-billion-active-users-on-its-platforms/)

Lista de Siglas e Acrónimos

UML	<i>Unified Modeling Language</i>
RUP	<i>Rational Unified Process</i>
BD	Base de Dados
DAO	<i>Data Access Object</i>