

# Trabalho Prático 2 – Aplicação Web em Jakarta EE



Departamento de  
Informática

## Trabalho realizado por:

Cristiano Santos, nº 43464

Bruno Monteiro, nº43994

Alexandre Monteiro, nº 44149

Manuel Magalhães, nº44604

03/06/2022

# Conteúdo

<b>Introdução .....</b>	<b>3</b>
<b>Estrutura da Aplicação .....</b>	<b>4</b>
<b>Estrutura da Base de Dados .....</b>	<b>7</b>
<b>Manual de Configuração e Instalação.....</b>	<b>9</b>
<b>Notas a acrescentar.....</b>	<b>10</b>
<b>Conclusão .....</b>	<b>10</b>

## Introdução

No âmbito da unidade curricular de Sistemas Distribuídos da Licenciatura em Engenharia Informática, foi nos proposto a realização deste trabalho prático que acaba por refletir a implementação do primeiro trabalho prático de uma forma distinta.

Este trabalho compreende a realização de uma aplicação web via utilização de Java Spring, que compreenderá a apresentação, subscrição e publicação de notícias bem como o estabelecimento de diversos papéis de utilizadores para a gestão das mesmas.

Para a realização deste projeto foram aplicados conhecimentos lecionados em aulas práticas e teóricas da unidade curricular.

# Estrutura da Aplicação

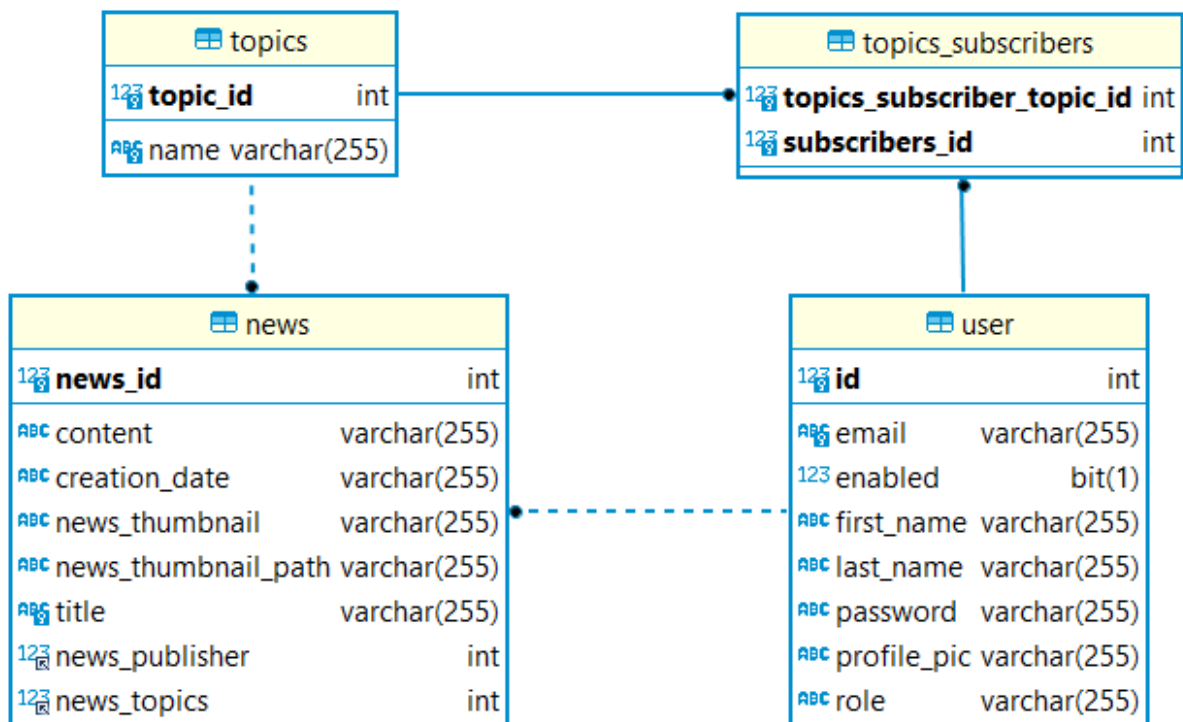
A aplicação desenvolvida em Java Spring encontra-se estruturada nos seguintes packages:

- **auth:** Esta package compreende 3 classes MvcConfig, WebSecurityConfig e FavIconConfiguration. O **MvcConfig** é uma classe que efetua o tratamento da especificação de caminhos para elementos da página como fotografias de perfil de utilizadores e imagens associadas às notícias. O **WebSecurityConfig** é uma classe que efetua operações de tratamento de credenciais, de papéis de utilizadores bem como permissões que estão atribuídas a cada um desses papéis. Além disso também se define o tratamento de **Cookies**, *sendo esta parte importante para guardar o estado de ligação caso o utilizador não efetue **logout***. O **FavIconConfiguration** é uma classe que trata do icon da aplicação desenvolvida.
- **controllers:** Esta package compreende todos os controladores designados para a funcionalidade da aplicação Web. Os controladores estão divididos nas seguintes classes:
  - **ErrorRedirectController:** O controlador responsável para redirecionar o utilizador para a página principal caso exista algum **erro relativamente a mapping de URL**;
  - **IndexController:** Este controlador redireciona o utilizador para a página que lhe está atribuída de acordo com o seu papel na aplicação;
  - **loginController:** Controlador responsável pelo tratamento de operações de login;
  - **nonSubscriberController:** Controlador responsável por todas as operações que um utilizador não registado (sem papel definido na aplicação) possa efetuar no contexto da aplicação;
  - **profileController:** Controlador que redireciona o utilizador para o seu perfil;
  - **publisherController:** Controlador responsável por todas as operações que um utilizador com o papel de **jornalista (publisher)** pode efetuar no contexto da aplicação;

- **subscriberController:** Controlador responsável por todas as operações que um utilizador com o papel de **assinante** (**subscriber**) pode efetuar no contexto da aplicação;
  - **registerController:** Controlador responsável pelo registo de novos utilizadores na aplicação.
- **entities:** Este package **engloba as entities** da nossa aplicação:
    - **News:** Entity que representa a tabela das notícias da aplicação bem como as relações da mesma;
    - **Topics:** Entity que representa a tabela dos tópicos das notícias bem como as relações da mesma;
    - **User:** Entity que representa a tabela dos utilizadores da aplicação bem como as relações da mesma;
- **repository:** Este package abarca diversas interfaces que efetuam operações **CRUD(Create, Read, Update, Delete)**. As interfaces subdividem-se nas seguintes:
    - **NewsRepository:** Interface que possui métodos que permitem operações relativas às notícias;
    - **TopicsRepository:** Interface que possui métodos que permitem operações relativas aos tópicos;
    - **UserRepository:** Interface que possui métodos que permitem operações relativas a utilizadores.
- **service:** Este package abarca diversos serviços que acedem aos diferentes repositórios. Estes serviços permitem que diversos controllers acedam à informação da base de dados. Os diversos elementos deste package subdividem-se em:
    - **EmailService e EmailServiceImpl:** A interface **EmailService** contém os protótipos dos métodos que a classe **EmailServiceImpl** irá implementar. Estes métodos dizem respeito a operações de envio de notificações para email dos utilizadores;
    - **NewsService e NewsServiceImpl:** A interface **NewsService** contém os protótipos dos métodos que a classe **NewsServiceImpl** irá implementar. Estes métodos dizem respeito a diversas operações relacionadas às notícias;

- **SecurityService:** Classe que efetua redirecionamento de login quando um utilizador se regista pela primeira vez;
  - **TopicService** e **TopicServiceImpl:** A interface **TopicService** contém os protótipos dos métodos que a classe **TopicServiceImpl** irá implementar. Estes métodos dizem respeito a diversas operações relacionadas com os tópicos;
  - **UserService** e **UserServiceImpl:** A interface **UserService** contém os protótipos dos métodos que a classe **UserServiceImpl** irá implementar. Estes métodos dizem respeito a diversas operações relacionadas com os utilizadores;
- **forms:** Este package contém classes auxiliares que ajudam no tratamento de operações de diversos elementos da aplicação desenvolvida:
    - **NewsForm:** Classe auxiliar para operações de tratamento de conteúdo relativo às notícias da aplicação;
    - **TopicForm:** Classe auxiliar para operação de adição de tópicos por parte do **jornalista** e remoção de subscrições por parte do **assinante**;
    - **TopicFormSearch:** Classe auxiliar para a operação de procura de uma notícia de um dado tópico num certo timestamp a definir;
    - **TopicFormSubscriber:** Classe auxiliar para a operação de subscrição de um ou mais tópicos por parte do assinante;
    - **UserRegistrationDto:** Classe auxiliar para a operação de registo de um dado utilizador.
    - **EmailDetails:** Classe auxiliar para tratamento de conteúdo de email **para notificação** dos utilizadores.
- **resources:** Este package contém os ficheiros .html que são utilizados na aplicação desenvolvida. Para a elaboração dos ficheiros .html foi escolhida a framework **bootstrap** para fornecer um design mais consistente e equilibrado do **front-end** da aplicação.

## Estrutura da Base de Dados



- Relação 1 – N entre **Tópicos** e **Notícias** respetivamente
- Relação N – N entre **Tópicos** e **Utilizadores** respetivamente => Esta ligação subdivide-se em 2 ligações **1-N** e **N-1** criando a tabela **topics\_subscribers**
- Relação 1 – N entre **Utilizadores** e **Notícias** respetivamente

Os utilizadores subdividem-se nos seguintes papéis:

**Jornalista (publisher):** Pode efetuar operações de **adição de tópico** e **notícia** e operações de **pesquisa de tópicos disponíveis** e **consultar e editar notícias publicadas por ele**. Pode também alterar informações sobre o seu perfil de utilizador;

**Assinante (subscriber):** Pode efetuar operações de **subscrição de tópicos** (adicionar e remover), pode **consultar uma notícia de um dado tópico através do timestamp** e **consultar a última notícia de um dado tópico**. Pode também alterar informações sobre o seu perfil de utilizador;

**Utilizador não registado (non-subscriber):** Pode **consultar uma notícia de um dado tópico através do timestamp** e **consultar a última notícia de um dado tópico**.

É de salientar que **todos os utilizadores** conseguem visualizar as notícias publicadas na página inicial onde são carregadas 10 notícias de cada vez, sendo possibilitado ao utilizador a opção de avançar ou recuar entre páginas. Para isto foi implementado **paginação**.

Por fim é de acrescentar que quando uma notícia é adicionada por um **jornalista**, todos os assinantes que **subscreveram aquele tópico são notificados** via email.



## Manual de Configuração e Instalação

- Numa primeira instância deve ser criado um novo projeto Java EE com as seguintes dependências base: **Spring Web, Spring Boot Dev Tools, Thymeleaf, Spring Data JPA, MySQL Driver**. Deve ser utilizada a versão **JDK 17** para o projeto.
- Numa segunda instância deve se colocar o conteúdo da pasta **src** enviada na pasta **src do projeto criado**. Deve-se também colocar as pastas **news-thumbnail** e **profile-pics** na pasta do projeto.
- Numa terceira instância deve-se criar um utilizador com o username **SD\_TP\_02** e com a palavra passe **root**. Este utilizador deve ser criado no **MySQL** e dar todas as permissões (através da opção **“Administration”** -> **“Users and Privileges”** -> **“Add Account”** -> (criar utilizador) -> **“Administrative Roles”** -> (dar *check* na opção **“DBA”**) -> confirmar criação do utilizador (**“apply”**), estas operações devem ser efetuadas no **“MySQL Workbench”**).
- Numa quarta instância a aplicação pode ser executada após as configurações anteriores.
- Numa quinta instância para aceder ao website deve-se introduzir no URL de um browser: **http://localhost:80**

## Notas a acrescentar

A aplicação pode ser executada fora do editor através da utilização de um **proxy** fornecido pelo **ngrok**.

O **ngrok** fornece um canal seguro de comunicações através de um endereço que pode ser acedido remotamente por outras máquinas. Este próprio serviço disponibiliza um certificado que permite tornar a ligação segura (por https).

## Conclusão

Com a realização deste trabalho foi possível aplicar diversos conhecimentos lecionados em aulas práticas e teóricas da unidade curricular de Sistemas Distribuídos. Além disso foi nos possível aprender uma nova framework sendo neste caso a framework **Java Spring**, bem como a possibilidade de enriquecer o nosso conhecimento de desenvolvimento de **front-end** e **back-end**.