Relatório

Meta 2

Unidade Curricular:

Sistemas Distribuídos

Licenciatura em Engenharia e Ciência de Dados

Realizado por:

Pedro Fonseca nº2021228824

João Tomás nº 2021219036

Ano Letivo 2023/2024

Introdução

O projeto desenvolvido na unidade curricular de Sistemas Distribuídos tem como objetivo criar um motor de pesquisa de páginas Web denominado Googol. Este motor de busca possui funcionalidades de indexação automática (Web crawler) e busca (search engine), similar a serviços como Google.com, Bing.com e GMX.com. Este relatório detalha o desenvolvimento da segunda meta do projeto, que consiste em criar uma interface Web para a aplicação, integrando-a com a aplicação desenvolvida na primeira meta, utilizando Spring Boot e seguindo a arquitetura MVC.

Objetivos do Projeto

Os objetivos desta segunda meta são:

- Desenvolver uma interface Web para a aplicação Googol.
- Integrar a interface Web com a aplicação desenvolvida na primeira meta.
- Aplicar tecnologias de programação para a Web, nomeadamente Spring Boot.
- Seguir a arquitetura MVC para desenvolvimento Web.
- Integrar a aplicação com serviços REST externos.

Arquitetura

A arquitetura do projeto inclui vários componentes distribuídos que se comunicam via RMI. A arquitetura geral do projeto inclui os seguintes componentes:

- **Downloader**: Obtém páginas web, analisa-as usando jsoup e envia os resultados para os Index Storage Barrels via RMI.
- **Index Storage Barrel**: Servidor principal que armazena todos os dados do sistema, incluindo o índice invertido. Recebe informações dos Downloaders e serve como cliente para o Search Module.
- **RMI Search Module**: Interage com o Index Storage Barrel para realizar buscas.
- **Servidor Web**: Implementado com Spring Boot, liga-se ao servidor de dados via RMI, garantindo interoperabilidade.

O código é organizado em várias classes e interfaces Java que implementam a lógica do sistema. Foram utilizados 4 ficheiros HTML e um ficheiro CSS para o frontend.

Implementação Técnica

Estrutura do Código

O projeto é composto por várias classes e interfaces que seguem a arquitetura MVC. A seguir estão algumas das principais classes e suas responsabilidades:

1. **HackerNewsController**: Controlador responsável por buscar e exibir as histórias do Hacker News. Utiliza o serviço `HackerNewsService` para obter as histórias e adicioná-las ao modelo para exibição na view `hacker-news`.

2. **SearchController**: Controlador responsável pelas funcionalidades de busca e adição de URLs. Ele interage com o serviço RMI para realizar buscas e indexar novos URLs.
3. **RMIConnect**: Serviço responsável por se conectar ao módulo de busca via RMI. Facilita a comunicação entre a aplicação web e o servidor RMI.
4. **OpenAIService**: Serviço que integra a API da OpenAI para gerar análises textuais baseadas nos termos de pesquisa.
5. **HackerNewsService**: Serviço que integra a API do Hacker News para obter as top stories.

Integração com Serviços REST

Este projeto foi integrado com as APIs do Hacker News e da OpenAI. As documentações das APIs permitiram construir funcionalidades adicionais através de REST, incluindo:

- **Indexar URLs das top stories que contenham os termos da pesquisa**: Após uma pesquisa no Googol, o utilizador pode solicitar a indexação dos URLs das "top stories" do Hacker News que contenham os termos pesquisados.
- **Gerar uma análise contextualizada com a API da OpenAI**: A página de resultados do Googol inclui uma análise textual baseada nos termos da pesquisa, gerada pela API de inteligência artificial generativa da OpenAI.

Frontend

A aplicação frontend foi desenvolvida utilizando Thymeleaf e inclui as seguintes páginas HTML:

- **home.html**: Página inicial onde os utilizadores podem realizar buscas e adicionar URLs.
- **hackernews.html**: Exibe as top stories do Hacker News.
- **result.html**: Exibe mensagens de sucesso ou erro após adicionar um URL.
- **search-results.html**: Exibe os resultados da pesquisa e a análise gerada pela OpenAI.

Funcionalidades Implementadas

1. Interface Web com Spring Boot

A interface Web foi desenvolvida utilizando Spring Boot com Thymeleaf, seguindo a arquitetura MVC. Foram implementados controladores principais:

- **AuthController**: Gerencia o login e o registo de utilizadores.
- **SearchController**: Facilita a comunicação entre o website e o RMISearchModule, gerenciando as pesquisas dos utilizadores.
- **UrlController**: Permite aos utilizadores adicionar URLs para indexação.

2. Integração com Serviços REST

Foram integradas duas APIs externas:

- **Hacker News API**: Permite buscar URLs das "top stories" que contêm termos pesquisados no Googol.
- **OpenAI API**: Utilizada para gerar análises textuais contextuais baseadas nos termos de pesquisa.

3. Funcionalidades de Pesquisa

As funcionalidades de pesquisa permitem que os utilizadores realizem buscas por termos específicos, retornando uma lista de páginas web contendo esses termos. Cada resultado de pesquisa inclui o título da página, o URL e uma citação curta do texto.

4. Indexação de URLs

Os utilizadores podem adicionar manualmente URLs para serem indexados. O indexador automático visita os URLs encontrados em páginas previamente visitadas, mantendo uma fila de URLs a serem processados.

5. Ordenação de Resultados por Relevância

Os resultados de pesquisa são apresentados por ordem de relevância, determinada pelo número de ligações de outras páginas para cada página.

6. Consulta de Ligações para uma Página

Utilizadores registados podem consultar todas as ligações conhecidas que apontam para uma página específica.

Testes Realizados

A aplicação foi submetida a vários testes para assegurar o seu funcionamento correto. Os testes incluíram:

Funcionalidades

Funcionalidade	Passou	Falhou
Indexar um novo URL	X	
Indexar recursivamente todos os URLs	X	
Pesquisar páginas que contenham termos	X	
Resultados de pesquisa ordenados por importância	X	
Consultar lista de páginas com ligações	X	
Indexar URLs das top stories que contenham os termos da pesquisa	X	
Gerar uma análise contextualizada com a API da OpenAI	X	

Requisitos Não-funcionais

Requisito	Passou	Falhou
Tratamento de exceções	X	
Failover		X

Conclusão

O projeto Googol, desenvolvido no âmbito da unidade curricular de Sistemas Distribuídos, cumpriu os objetivos propostos. Através da integração de várias tecnologias e a implementação de uma interface Web robusta, foi possível criar um motor de pesquisa funcional e eficiente. Este projeto permitiu aprofundar conhecimentos em threads, sockets, comunicação via RMI e Spring Boot, culminando numa aplicação prática e de grande utilidade.