



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Relatório de SD:

Meta 2: Motor de pesquisa de páginas Web Gogool

Trabalho realizado por:

João Macedo 2021220627

Rafael Vieira 2019214046

Turma: PI7

1. Introdução:

Este relatório detalha o desenvolvimento do motor de pesquisa de páginas web, denominado Gogool, desenvolvido como parte do projeto de Sistemas Distribuídos (SD)

O principal objetivo do Gogool é permitir a pesquisa eficiente de conteúdos na web, utilizando técnicas de crawling e indexação, e integrar diversos serviços web para oferecer uma experiência fluida de pesquisa. Este relatório abordará a arquitetura do projeto, a implementação de modelos, visões e controladores, a utilização de processos, threads e sockets, bem como a integração de serviços REST e RMI (Remote Method Invocation) com o Spring Boot.

2. Arquitetura do projeto Web:

Models:

Representam as classes responsáveis por representar as estruturas dos dados que são manipulados e processados durante a execução da aplicação, tal como ler e escrever nos ficheiros objeto. No nosso caso temos a classe Crawled Data que define como são representados e armazenados os dados descarregados da internet pelo Downloader.

Temos também o Index Storage Barrels, que é responsável por ler e escrever diretamente nos ficheiros objeto os diferentes dados do sistema.

Temos também o gateway que comunica com o Barrels para is buscar os dados necessários à base de Dados, esta componente foi um pouco dúbia a nível de onde colocar na arquitetura estando a equipa indecisa entre o Controller e o Models e tendo acabar por ficar neste ultimo.

Views:

As views no nosso sistema são representadas pelos Templates HTLM que se encontram no "src/main/resources/templates, estes templates servem para gerar as páginas webs que os utilizadores vão visualizar, para além disso parte deles utiliza scripts para que num só ficheiro HTLM seja possível ter uma página antes de uma ação e transformar-se noutra pagina para dar tratar o pedido.

addUrl.html -> Página para adicionar URL

Googol.html -> Menu principal

hackerNewsResults.html ->Página para realizar pesquisa pelas TopStories com a API e exibir o resultado

Index.html -> Página inicial da aplicação.

login.html ->Página para realizar o log-in

register.html ->Página de Registro

Search.html -> Menu para decidir qual pesquisa realizar

searchGateway.html -> Página para realizar a pesquisa por um termo através do Gateway

searchProjects.html -> Página para pesquisar por projetos no site da Hacker News

searchUrl.html ->Página para realizar uma pesquisa por um Url através do Gateway

Controller:

Os controllers são responsáveis por gerir as requisições HTTPS e mapear os URL dos métodos correspondentes no servidor, no nosso caso temos dois controllers implementados, o GogoolController que é responsável por fazer a ligação ao Gateway, através de RMI para poder depois fazer pedidos às componentes internas do sistema.

Existe depois também o WebPageController, é responsável por gerir as requisições das páginas web específicas. Tem duas instâncias uma do HackerNewsService e outra de GogoolController, é através delas que faz os pedidos ao sistema para retornar os dados que posteriormente mostra ao utilizador na página Web.

Processos e Threads:

A utilização de threads permite que o sistema execute múltiplas tarefas simultaneamente, melhorando a performance e a capacidade de resposta.

O serviço Downloader é responsável por descarregar páginas web. Este serviço pode criar múltiplas threads para realizar downloads em paralelo.

O ExecutorService é utilizado para gerir um pool de threads, permitindo que múltiplas tarefas de download sejam executadas simultaneamente.

Sockets:

Os sockets são utilizados para comunicação entre diferentes componentes do sistema, permitindo a transmissão de dados através de uma rede.

São utilizados através de RMI (Remote Method Invocation) para permitir a comunicação remota entre o gateway e o Index Storage Barrel, o contrário também.

Temos também a utilização de sockets TCP para a comunicação entre o gateway e o downloader para que o downloader possa receber os pedidos de addURL dos clientes.

Existe também a utilização do multicast para a transmissão dos dados descarregados nos downloaders para enviar ao Index Storage Barrel.

3. Integração do SpringBoot com o servidor RMI

Primeiro temos a interface RMIGateway a definir os métodos que podem ser invocados remotamente, esta interface da extend no Remote e os métodos dao throw de uma remote Exepction.

Temos então a implementação da interface anterior na Classe Gateway.

Esta classe estende UnicastRemoteObject.

No ficheiro application.properties, foram definidas as propriedades necessárias para a configuração do servidor RMI.

Na classe Gogol Controller é realizada a ligação ao servidor RMI, esta classe é posteriormente instanciada no WebPage Controller, para possam ser feitos os pedidos ao Gateway pelo WebServer.

4. Integração de Rest WebServices no projeto:

Definição do Serviço HackerNewsService: Este serviço utiliza RestTemplate para fazer chamadas à API externa do Hacker News e processar os resultados.

1. Injeção de Dependências: WebPageController injeta HackerNewsService utilizando @Autowired, permitindo o uso dos métodos do serviço no controlador.
2. Definição de Endpoints REST: O WebPageController define endpoints REST que expõem funcionalidades do HackerNewsService aos utilizadores através de requisições HTTP. Estes endpoints permitem pesquisar histórias e projetos no Hacker News.
3. Processamento dos Resultados: Os resultados das chamadas à API são processados e retornados aos clientes como listas de strings ou mapas, dependendo do endpoint.

5. Testes de software

Numero do teste	Descrição do teste	Resultado do Teste
1	Permitir log-in na plataforma com um utilizador válido e dados corretos	Pass
2	Não permitir log-in na plataforma com um utilizador que não existe ou com um utilizador existente e password errada	Pass
3	Não permitir o registro de utilizadores com username que já exista	Pass
4	Realizar uma pesquisa por um termo com a função RMI do search	Pass
5	Realizar pesquisa pelas topStories com um termo através da API da Hacker News	Pass
6	Realizar pesquisa por projetos relacionados a um termo com API da Hacker News	Pass
7	Correr componentes do projeto em diferentes máquinas e continuar funcional	Pass
8	Verificar se a ligação feita do browser com o WebServer utiliza o	Pass

	protocolo HTTPS	
9	Ligar ao WebServer através de um tablet ou ipad	Pass
10	Ligar ao WebServer através de um equipamento a correr numa rede externa à do servidor	Fail

6. Conclusão:

O desenvolvimento do motor de pesquisa Gogool demonstrou a viabilidade de integrar diferentes tecnologias para criar uma aplicação web.

A utilização de Spring Boot facilitou e acelerou a implementação dos controladores e serviços, enquanto o uso de RMI permitiu a comunicação remota entre os componentes do sistema.

A integração com a API Hacker News através do serviço HackerNewsService expandiu as funcionalidades do sistema, permitindo duas novas pesquisas.

Os testes de software realizados confirmaram a funcionalidade e relativa robustez da aplicação, com a maioria dos testes a ter sucesso. A exceção notável foi a falha ao tentar ligar ao WebServer através de uma rede externa, indicando a necessidade de melhorias na configuração de rede e segurança.

No geral, o projeto alcançou os seus objetivos, proporcionando uma base sólida para futuras expansões e melhorias.