



Faculdade de Ciências e Tecnologia da Universidade de Coimbra
Licenciatura em Engenharia Informática

Sistemas Distribuídos
2022/2023

Trabalho Prático

Googol: Motor de pesquisa de páginas Web

Projeto realizado por:

Miguel Fazenda – 2019222229

Tiago Henriques – 2020237060

Professor orientador:

Hugo Amaro

Índice

1. Introdução	3
2. Arquitetura do Software.....	3
2.1. Descrição geral	4
2.1.1. Models.....	4
2.1.2. Views	4
2.1.3. Threads.....	5
2.1.4. Sockets.....	5
3. Spring Boot e o Servidor RMI	5
4. Serviço REST	5

1. Introdução

O objetivo da segunda meta do projeto passa pela utilização dos programas implementados na meta anterior de modo a desenvolver um website com as suas funcionalidades e outras adicionais como, por exemplo, a integração da aplicação com serviços REST externos e a utilização de WebSockets para comunicar assincronamente com os clientes.

2. Arquitetura do Software

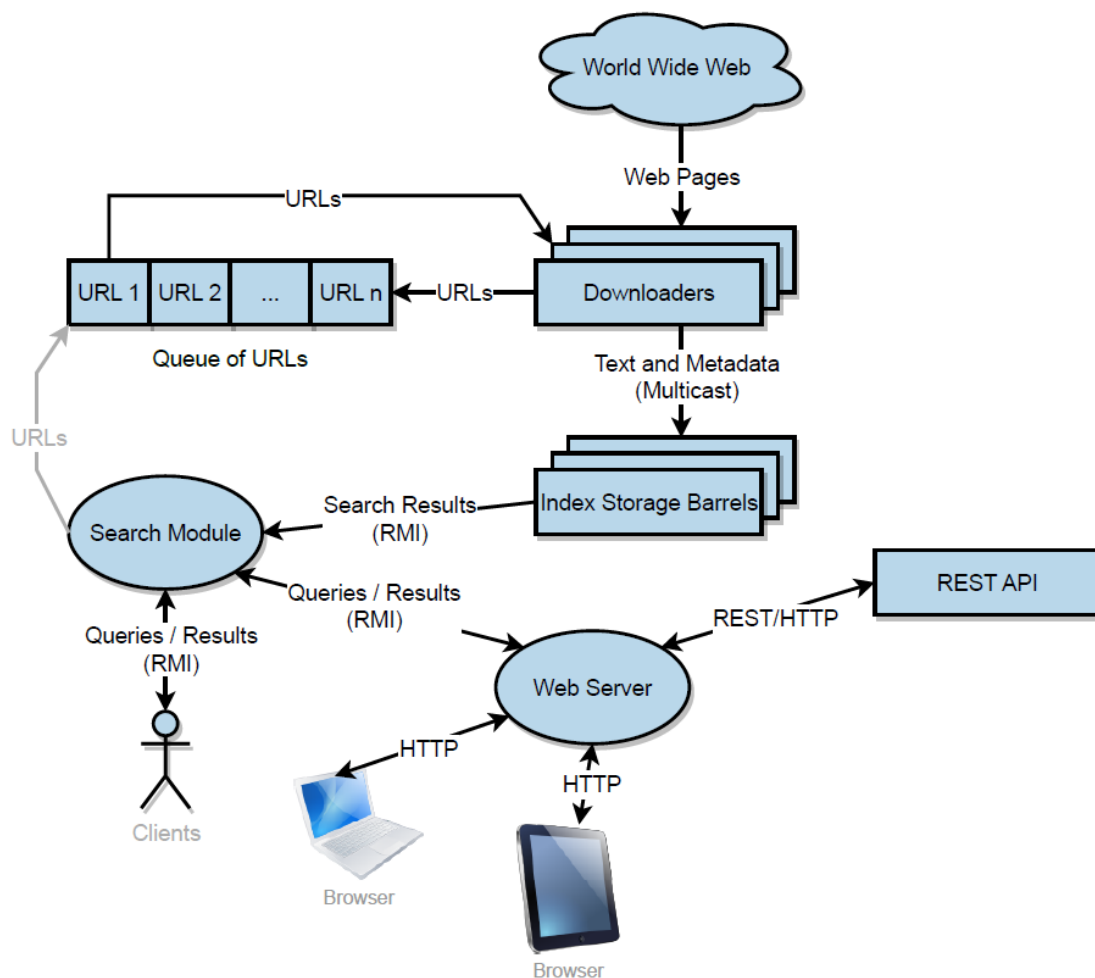


Figura 1 - Arquitetura da aplicação, incluindo os componentes da Meta 1.

2.1. Descrição geral

2.1.1. Models

Foram criados cinco modelos que são utilizados ao longo do projeto para representar, receber e enviar informação das páginas. Estes são o Url, User, Keywords, HackerKeywords e HackerUser. Estes modelos são classes Java que são compostas por parâmetros, um ou mais construtores e getters/setters.

2.1.2. Views

Ao conectar-se ao localhost no seu web browser, o utilizador é imediatamente redirecionado para a página “GoogolHome”, definida pelo template Home.html, onde existem dois botões dizendo “Login” e “Register”, respetivamente.

Como o nome indica, se o utilizador clicar no botão “Login” este é redirecionado para a página “GoogolLogin” onde é possível inserir os dados da sua conta (username e password) para avançar para a aplicação. Caso os dados forem incorretos, a página é reiniciada de modo a dar ao utilizador outra oportunidade de entrar (este processo é repetido até os dados forem validos).

Por outro lado, ao escolher o botão “Register”, o utilizador é encaminhado para a página “GoogolRegister” que lhe permite escolher um username e password que iram ficar guardados no sistema e podem ser usados para entrar na aplicação em futuras utilizações. Se o username escolhido pelo utilizador já se encontrar usado, a página é reiniciada para este poder tentar novamente (este processo é repetido até os dados forem validos).

Após realizar o login ou o register com sucesso, o utilizador é redirecionado para a página da aplicação, “Googol”. Nesta o utilizador tem cinco opções, pode escolher adicionar um novo URL, pesquisar por links que contenham certas keywords, adicionar URLs pertencentes às top stories de Hacker News especificando keywords, adicionar URLs de utilizadores de Hacker News e, finalmente, ver as estatísticas do admin.

Caso pressione “Add a new URL”, “Hacker News” ou “Hacker News Users”, o utilizador irá para uma página onde poderá indicar a informação necessária para indexar os novos URLs e seguidamente é redirecionado de volta para a página da aplicação.

Após escolher o botão “Search for keywords”, o utilizador pode indicar as keywords a serem pesquisadas e, de seguida, os primeiros dez resultados vão aparecer na página “GoogolSearchResults”. Nesta página há dois botões “Next 10 links” e “Show result connections” que iriam reencaminhar o utilizador para outra página mas não foi possível recuperar o arrayUrl.

NOTA: a implementação de ver as estatísticas do admin não foi possível a tempo da entrega.

2.1.3. Threads

Em comparação com a entrega anterior o nosso grupo decidiu melhorar o desempenho do código através da utilização de mais threads para correr mais processos em paralelo. Para alcançar isso, as mensagens enviadas por Multicast foram divididas entre várias threads, uma para cada tipo de mensagem. Os downloaders contêm uma thread que envia as words de um url, uma thread que envia a UrlInfo e outra que envia os links. Do outro lado, os Storage Barrel têm o mesmo número de threads para receber essas mensagens e mais uma que controla quando estes devem ser guardados.

2.1.4. Sockets

Em termos de sockets, as funcionalidades implementadas na primeira meta foram mantidas com a diferença que agora são utilizadas 3 sockets Multicast, tendo cada uma delas um Multicast group de modo a responderem melhor às threads presentes no Downloader e nos Storage Barrel.

3. Spring Boot e o Servidor RMI

Na integração do Spring Boot com a componente RMI, o nosso grupo utilizou o GetMapping e PostMapping e, nas funções associadas a essas annotations fizemos as comunicações RMI entre as páginas e os servidores. Foram utilizados o GetMapping e o PostMapping para receber os parâmetros necessários dos utilizadores, e de seguida enviar para as páginas web os resultados, caso existam.

4. Serviço REST

Integração com a REST do Hacker News, no caso das topstories, depois do utilizador enviar as keywords ou um username, cria-se uma HttpURLConnection e faz-se uma Request do método "GET", O programa lê a input stream desta connection, e de seguida separa as storyIds, e envia as stories para uma função, onde esta sim fará a indexação.

Esta função cria outra HttpURLConnection e pede outra vez o método "GET" de maneira a ler a story do link. De seguida, o Jsoup é utilizado para ser realizada a parse à resposta Json. Verifica-se que as keywords pedidas pelo utilizador estão presentes no body e se estiverem são selecionados os urls presentes nessa story e, através de uma

ligação RMI e de seguida TCP, enviamos os urls para a Queue. Para indexar as stories de um utilizador o processo é o mesmo, exceto o passo de verificar as keywords que é desnecessário, e processamos todas as stories.