

**Licenciatura em Engenharia Informática**  
**Sistemas Distribuídos - Googol**

---

**Hugo Batista Cidra Duarte (2020219765)**  
**Nuno Carvalho do Nascimento (2020219249)**

May 18, 2024

---

## Índice

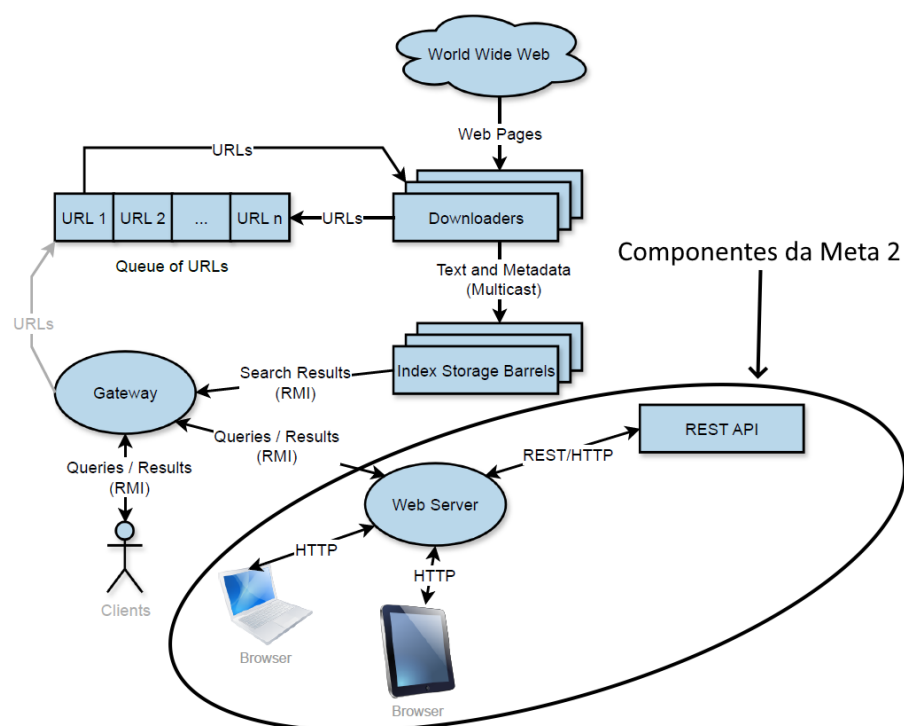
<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Arquitetura</b>	<b>3</b>
<b>3</b>	<b>Spring Boot</b>	<b>4</b>
<b>4</b>	<b>WebSockets</b>	<b>4</b>
<b>5</b>	<b>REST</b>	<b>4</b>
<b>6</b>	<b>Testes</b>	<b>5</b>

## 1 Introdução

Este projeto realizado no contexto da cadeira de Sistemas Distribuídos tinha como objetivo simular as operações realizadas por um motor de busca convencional. No contexto desta meta foi pedido que além das funções previamente efetuadas pelo sistema, o mesmo também tivesse uma interface visível no browser além de integração com o serviço duma REST API.

## 2 Arquitetura

A arquitetura desta meta é bastante semelhante aquela da primeira, com a adição de alguns componentes ligados à componente de front-end e REST.



Na figura apresentada acima é feita uma distinção entre os componentes da arquitetura relativos à primeira meta e os pertencentes à segunda. Com isto é possível ilustrar apropriadamente os elementos a analisar neste relatório.

O Web Server encontra-se responsável por receber e enviar as informações tanto da REST API como da Gateway, servindo como o passo prévio à interface utilizada pelo browser dos utilizadores. O Web Server prepara portanto a informação de modo a que ela possa depois ser vista o browser de modo a que os utilizadores possam então interagir com o projeto Googol. O outro novo componente a mencionar nesta arquitetura é a introdução da REST API. O objetivo da REST API é providenciar um método de nos conectarmos com a OpenAI e com a Hacker News, de modo a obter notícias das Hacker News e obter recomendações mais contextualizadas com a ajuda da inteligência artificial generativa da OpenAI.

---

### 3 Spring Boot

O Spring Boot é usado para conectar o RMI ao web server, mais concretamente a gateway. Spring Boot consiste numa ferramenta que tem o objetivo de facilitar o desenvolvimento e a criação de aplicação que sejam feitas a partir de Spring. O Spring Boot agiliza o processo de trabalhar com Spring ao configurar automaticamente o Spring e 3rd party libraries sempre que possível, além de dar opções para um build inicial de forma a tornar o processo mais simples e fácil para o criador do projeto.

No contexto do projeto, o Spring Boot é implementado no RMI de modo a que nos seja possível correr o frontend de forma eficaz. Para atingir este objetivo é necessário remover o main da class RMIClient, de modo a que o mesmo possa ser utilizado como uma classe auxiliar pelo Spring, de modo a que este passa apenas a ser usado para correr as funções. A informação do HTML chega até ao Spring Boot através do thymeleaf, o qual vai posteriormente processar estas do lado do JAVA.

### 4 WebSockets

Websockets são um método para enviar informação dum servidor para um cliente sem que esta seja enviada para um cliente específico diretamente. Os websockets são implementados como uma extensão do método TCP, adicionando a habilidade de estabelecer handshakes, verificar endereços e nomear, tudo através da web. O primeiro passo é realizado através de HTTP para estabelecer a ligação ao servidor. Após isto o servidor enviará um socket, que é consiste num wrapper com a informação pedida. Por fim o web socket é enviado, tendo informação sobre o endereço que é suposto recebê-lo, em que o recetor que tenha o seu endereço igual ao do socket, apanhará o socket e a informação contida nele.

### 5 REST

REST consiste num método que engloba múltiplos princípios de arquitetura de software de modo a agilizar a implementação de web applications, isto deve-se ao facto de o mesmo ser mais simples que outros métodos, como por exemplo SOAP. REST usa HTTP como um método de transporte de modo a incorporar métodos de base de dados (comandos como GET, POST, PUT, entre outros) com o objetivo de realizar operações numa base de dados através da web. REST é stateless, pelo que o estado de cada pedido é mantido pelo próprio, e o mesmo pode ser mantido na cache do browser.

---

## 6 Testes

A fase de testes para este projeto foi realizada seguindo um princípio de whiteboxing, uma vez que os testes foram feitos depois de já haver código feito, devido a isto os testes são feitos de modo a não dar valores impossíveis que nós tendo acesso ao código poderíamos inferir diretamente como impossíveis. Com isto apresenta-se a seguir os testes realizados ao código manualmente, devido a uma incapacidade de propriamente integrar o JUnit no código.

### Teste A - Conexão ao Servidor

**Input:** No Input

**Métrica de sucesso:** Receber a mensagem de que a conexão foi feita com sucesso, e ser capaz de abrir o googol no browser.

### Teste B - Indexar novo URL

**Input:** "https://www.youtube.com" na indexação de links

**Métrica de sucesso:** O link fica propriamente indexado e é possível ver o mesmo na lista de links indexado.

### Teste C - Pesquisar por URL

**Input:** "youtube" na pesquisa

**Métrica de sucesso:** As 10 pesquisas mais apropriadas são exibidas ao utilizador.