UNIVERSIDADE D
COIMBRA

Meta 2 - Integração Web

Sistemas Distribuídos

FCTUC – Departamento de Eng. Informática 2023/24

Trabalho realizado por:

- Johnny Fernandes PL5 2021190668 uc2021190668@student.uc.pt
- Miguel Leopoldo PL5 2021225940 uc2021225940@student.uc.pt

Índice

Introdução

Nesta meta do projeto de Sistemas Distribuídos, o objetivo foi criar uma interface web para interagir com os componentes desenvolvidos na meta anterior. Optámos por utilizar o Spring Boot como framework de desenvolvimento, aproveitando a robustez e flexibilidade que este oferece para a criação de aplicações web.

Ao longo deste relatório, serão apresentados os desafios encontrados durante o processo de integração da nova aplicação web com as funcionalidades existentes, bem como as soluções adotadas para superá-los. Será fornecida uma visão clara e concisa do trabalho realizado, destacando a eficiência e eficácia proporcionadas pelo uso do Spring Boot.

O relatório está estruturado da seguinte forma: inicialmente, será descrita a arquitetura da aplicação, seguida pela apresentação detalhada de cada um dos controladores implementados (Search, Results, Stats e Error). Por fim, serão apresentadas as conclusões e considerações finais sobre o projeto.

Arquitetura

A arquitetura da aplicação foi implementada conforme solicitado no enunciado, mantendo todos os componentes desenvolvidos na primeira meta praticamente inalterados e adicionando uma nova componente que comunica exclusivamente com o gateway. Esta abordagem garante a integração e funcionalidade contínuas do sistema existente enquanto expande as suas capacidades. A arquitetura é demonstrada abaixo pela figura 1.

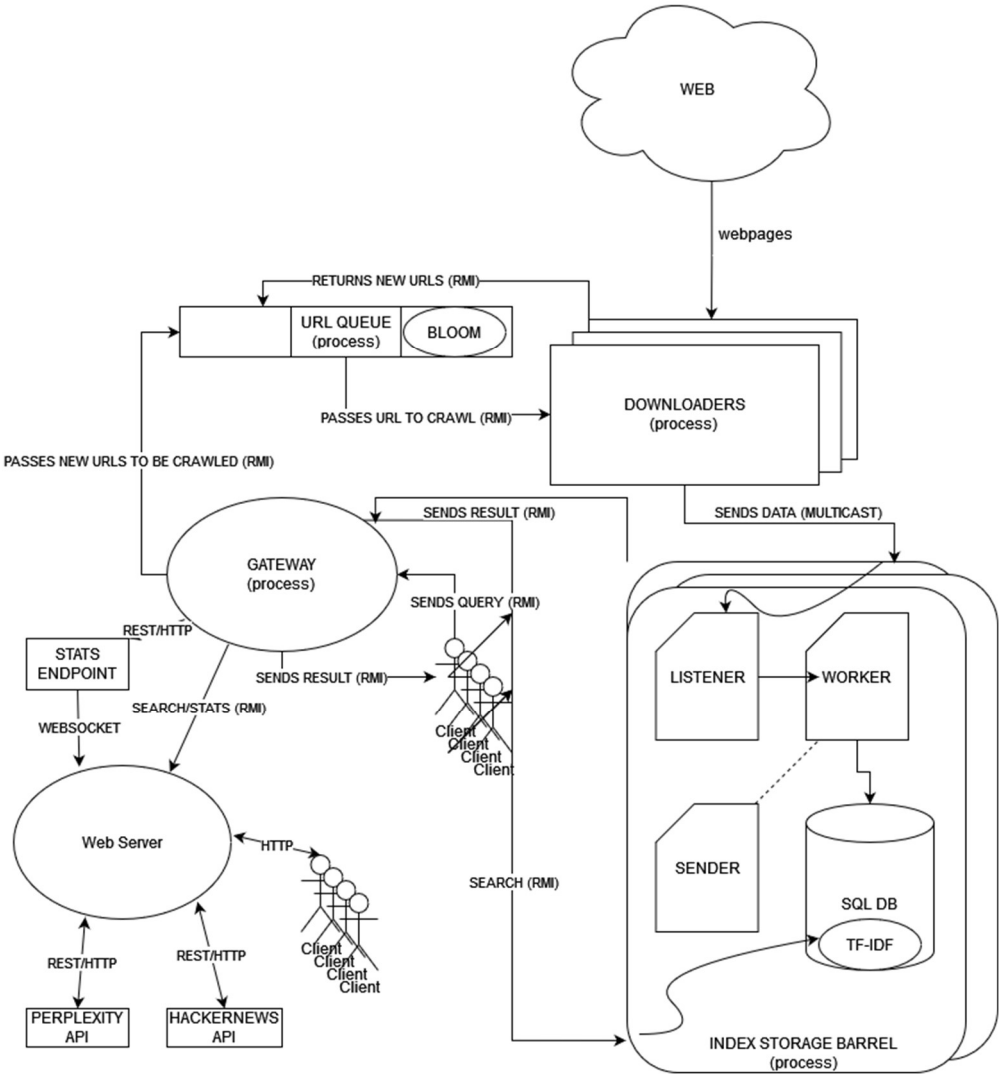

Figura 1 - Arquitetura da meta 2

A seguir – e partindo desta arquitetura -, serão apresentados os diversos controladores utilizados para construir a aplicação web, detalhando as suas características e como cada um contribui para a comunicação entre o frontend e o backend através do gateway. Sendo esta segunda meta focada na componente web com recurso a uma framework, a quantidade de elementos novos no sistema é menor e por isso apenas nos concentraremos num conjunto de controladores necessários ao funcionamento da interface.

Search Controller

Este controlador é responsável pela página inicial da aplicação, sendo a página padrão. A página inicial contém uma caixa de pesquisa e três botões.

- **Caixa de Pesquisa**: Permite ao utilizador introduzir uma query para pesquisa.
- **Botões**:
    - **Search**: Redireciona o utilizador para a página de resultados, onde são apresentadas páginas que contêm a query, ordenadas pelo número de ligações para cada página.
    - **Search Linked URLs**: Redireciona o utilizador para a página de resultados, onde são apresentadas URLs que estão ligadas ao URL submetido na caixa de pesquisa.
    - **Stats**: Redireciona o utilizador para a página de estatísticas, onde são apresentadas informações e métricas sobre a utilização da aplicação.

É um controlador totalmente independente do backend, não necessitando por isso de integração com o gateway.

Results Controller

O acesso normal a este controlador é feito através do `Search Controller`, onde um elemento HTML do tipo `Form` no controlador anterior é responsável pelas variáveis passadas no URL. As variáveis existentes são:

- `query`: Indica qual foi a query submetida na página de pesquisa.
- `page`: Indica o número da página.
- `urlsLinked`: Indica se os resultados são URLs que mencionam a query ou não.

Para obter os resultados, foi necessário integrar o gateway neste controlador. Para isso, foi utilizado RMI, invocando os métodos do gateway `searchQuery` quando `urlsLinked` é falso, ou `getWebsitesLinkingTo` quando é verdadeiro. Os resultados são apresentados usando ferramentas do Thymeleaf, que permitem inserir elementos no HTML de forma dinâmica.

Para mudar de página, a página tem dois botões no canto inferior esquerdo que permitem navegar entre páginas. Estes botões simplesmente incrementam ou decrementam o valor da variável `page`.

A página também contém um botão "Index HackerNews Top Stories" que acede à API do HackerNews. Ao clicar neste botão, é enviado um pedido para o endpoint "`/app/index-top-stories`" através de uma WebSocket, onde é tratado pelo método `onIndexHackerNewsTopStoriesPress`. Este pedido inclui a `query` e, dentro desta função, são buscadas as "Top Stories" do HackerNews e verificadas para ver se contêm a query. Se contiverem, os seus URLs são indexados na fila (queue), que é acedida através de uma ligação RMI ao gateway. Para poupar tempo de processamento, são selecionadas aleatoriamente 10 "Top Stories" para verificação.

Também é criada uma análise contextualizada pela API do Perplexity. Quando os resultados são enviados, é também feito um pedido a esta API, que devolve a análise. Este resultado é depois inserido no HTML usando o Thymeleaf.

Por último, existe um botão que permite ao utilizador regressar à página de pesquisa.

Stats Controller

Para consultar as informações do sistema, foi criado este controlador. Ele apresenta duas tabelas que são populadas dinamicamente usando Thymeleaf. A tabela dos barris apresenta informação relevante sobre os barris, e a tabela das "top" pesquisas contém as consultas mais realizadas.

Ao aceder a esta página, é feito um pedido ao Gateway através de RMI para garantir que as tabelas estão populadas inicialmente. Em seguida, através de um script JavaScript, a aplicação fica à escuta de informação enviada para "`/topic/update-stats`". Esta informação é enviada pelo Gateway quando há alguma alteração no estado do sistema.

Foi implementado um endpoint REST, "`/trigger-stats`", no frontend, ao qual o Gateway envia informação quando esta é atualizada. Esta informação é então transmitida através de WebSockets para todos os clientes que estão a escutar no endpoint "`/topic/update-stats`" mencionado anteriormente. Quando esta informação chega aos clientes, é tratada com recurso a um script JavaScript, de forma a atualizar a informação sem necessitar de um "refresh" da página.

Para tentar garantir alguma segurança no endpoint "`/trigger-stats`", é utilizado um token de autorização que apenas o Gateway e o controlador conhecem. Apesar de não ser a solução mais robusta, oferece uma proteção mínima ao sistema.

O único botão presente nesta página serve para voltar à página de pesquisa.

Error Controller

Este último controlador tem uma função simples: redirecionar o utilizador para esta página em caso de erro. A sua finalidade é puramente informativa.

Conclusão

A implementação da interface web com recurso ao Spring Boot para integrar com os componentes desenvolvidos da meta anterior do projeto foi um sucesso.

Com uso do Spring Boot como framework de desenvolvimento web foi possível ter um ambiente robusto e flexível que facilitou a integração com o backend existente. A arquitetura manteve os componentes da primeira meta praticamente inalterados, adicionando uma nova camada de comunicação exclusiva com o gateway que permitiu uma integração, garantindo a continuidade e expansão das funcionalidades do sistema.

Ao longo do desenvolvimento, enfrentamos alguns desafios, como a integração com o gateway via RMI, a apresentação dinâmica dos resultados usando Thymeleaf e a implementação de atualizações em tempo real das estatísticas usando WebSockets.

Além disso, a integração com APIs externas, como a do HackerNews e a do Perplexity, adicionou funcionalidades interessantes e relevantes à aplicação

Estamos orgulhosos do trabalho realizado e satisfeitos com os resultados alcançados. A aplicação web desenvolvida não apenas atende aos requisitos estabelecidos no enunciado, mas também oferece uma base sólida para futuras melhorias e expansões.