

Licenciatura em Engenharia Informática
Projeto Googol

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

7 de Abril de 2024

Índice

1	Arquitetura	3
1.1	SAVE	3
1.2	Downloaders	3
1.3	URL Queue	3
1.4	Webpage	3
1.5	Barrel	3
1.6	RMI Gateway	3
1.7	RMI Client	4
2	Multicast	4
3	RMI	4
4	Distribuição	4
5	Testes	4

1 Arquitetura

1.1 SAVE

A arquitetura do nosso projeto, começa por um ficheiro .txt onde são armazenadas as informações relativas aos nossos barrels.

1.2 Downloaders

Os Downloaders feitos com recurso à ferramenta JSoup permitem-nos ir à internet buscar informação relativa às páginas que pretendemos usar nas nossas funções e para obtermos os atributos das nossa múltiplas webpages, o qual fazemos com a ajuda da nossa classe que tem o mesmo nome.

1.3 URL Queue

É no nosso URL Queue que nós guardamos os URLs para eles serem usados entre outras funções. O nosso servidor permite-nos indexar novos URLs. O URL Queue também é responsável pela organização dos nossos downloaders, e a informação que este originalmente recebem é dada pelos downloaders, mas também pode ser transmitida através da nossa Gateway.

1.4 Webpage

Esta classe tem simplesmente a função de servir para podermos armazenar as informações relativas a cada uma webpage, sendo depois estas guardadas em listas de modo a puderem ser usadas para outras funções mais tarde.

1.5 Barrel

Os nossos barrels é o que nós usamos para armazenar e transmitir informação. Os barrels irão originalmente receber informação dos Downloaders por multicast, a qual será depois passada por RMI para a Gateway. Os barrels são o que nos permite realizar a transmissão de dados entre diferentes componentes do programa servindo como o principal método de transmissão de informação, recorrendo ao uso de multicast para esta comunicação. O mesmo também executa múltiplas funções que nos permitem extrair e atribuir múltiplos tipos de informação relativo aos múltiplos de atributos que se encontram armazenadas dentro dos nossos barrels.

1.6 RMI Gateway

A gateway é a componente deste projeto que nos permite fazer as nossas buscas, recorrendo a ajuda dos barrels, sendo o componente que desempenha a função mais superficial do nosso projeto, sendo aquele que, fora o RMI Client que será mencionado a seguir, mais diretamente interage com o utilizador sendo que muitas das funções usadas no RMI Client, são chamadas a funções implementadas na RMI Gateway, a qual já fez ou irá fazer o tratamento dos dados que esta recebe dos barrels.

1.7 RMI Client

O RMI Client é o último componente da nossa arquitetura e este é responsável por interagir diretamente com o utilizador e apresentar as suas possibilidades de escolha. O RMI Client comunica diretamente com a RMI Gateway e é a componente mais superficial da arquitetura pelo motivo previamente mencionado, funcionando como um pseudo front-end no contexto desta meta.

2 Multicast

Tal como é indicado no enunciado o multicast dos Barrels e dos Downloaders no nosso projeto é feito através de UDP. O UDP é por sua vez mais rápido mas menos confiável que o seu método semelhante o TCP, devido à inexistência do handshake, por este motivo é necessária a criação de failsafes caso a conexão não seja bem sucedida. Para tal é feita a garantia de que um pacote só pode chegar caso todos eles o tenham feito, na eventualidade de nem todos os pacotes chegarem ao destinatário os mesmos são enviados para trás. O nosso barrel aceitará portanto os pacotes de forma a que não haja pacotes que sejam descartados no processo e garantindo sempre que eles chegando chegam na sua totalidade.

3 RMI

Fora o multicast o RMI é o outro método de comunicação usado no nosso projeto. O servidor RMI é o método utilizado para realizar as comunicações efetuadas com a nossa gateway, o qual também engloba os nossos Downloaders. Devido ao modo como o nosso servidor está feito, é possível fazer chamadas das funções do Url Queue pela Gateway. A Gateway também serve como um servidor para o Client e os Barrels sendo, como já foi previamente mencionado, através disto o principal método de comunicação entre estes. O RMI é uma componente crucial neste projeto, devido a toda a informação que é transmitida pelo mesmo, há semelhança do multicast.

4 Distribuição

Neste projeto grande parte do trabalho foi feito de um modo conjunto, mas houve um maior foco na componente dos Downloaders e do URL Queue da parte do aluno Hugo Duarte enquanto que o aluno Nuno Carvalho envolveu-se mais na Gateway e no Client.

5 Testes

Os testes neste projeto foram efetuados com recurso às funcionalidades para teste do Maven, a qual corria o programa para determinados cenários e caso o programa não apresentasse falhas nesse cenário, considerava-se como tendo passado no teste.