Departamento de Engenharia Informática - FCTUC

Licenciatura em Engenharia Informática

**Sistemas Distribuídos**

Relatório Intermédio do Trabalho Prático

**Googol: Motor de pesquisa de páginas Web**

Uma imagem com texto, Tipo de letra, logótipo, tipografia

Descrição gerada automaticamente

Afonso da Silva Duarte - nº 2020221554

Miguel Aires Rainho – nº 2021222571

3 de abril, 2024

**Índice**

1. **Introdução**……………………………………………………………………….….**3**
2. **Arquitetura**……………………………………………………………………........**3**
   1. Threads e Sockets…………………………………………………………..…...X
   2. Organização do Código………………………………………………………....X
3. **Multicast**……………………………………………………………………………X

3.1 Downloaders e Storage Barrels…………………..….…...…....….……….....…X

1. **Componente RMI**…………....…………………………………..….….….………X 4.1 Métodos Remotos Disponibilizados………………………………………….....X
   1. Call-Backs………………………………………………………………………X
   2. Solução para o Failover…………………………………………………………X
2. **Estruturas de Dados Utilizadas**…………………………………………………...X
3. **Distribuição de Tarefas**…………………………………………………………….X
4. **Testes Realizados**…………………………………………………………………...X
5. **Introdução**

No âmbito da cadeira Sistemas Distribuídos foi pedido o desenvolvimento do projeto Googol tem como objetivo desenvolver um motor de pesquisa de páginas web com funcionalidades semelhantes aos motores de busca mais usados e conhecidos. Para isto acontecer há que garantir a existência de recursos e ferramentas que os assegurem.

O sistema indexará páginas web, armazenando a informação sobre os mesmos (como o URL da página, o título, uma pequena citação e algumas palavras encontradas na página). A informação é armazenada num índice invertido onde temos, para cada palavra as páginas web em que a mesma ocorre.

Na sua pesquisa o utilizador introduzirá palavras que deseje procurar, recorrendo o motor a este índice invertido para mostrar as páginas que contêm a palavra em causa.

O presente relatório irá aprofundar a arquitetura do software desenvolvido, como estão a funcionar as componentes Multicast e RMI, a solução encontrada para o Failover e os testes realizados.

1. **Arquitetura**

\*inserir esquema – fazer no draw.io\*

\*Threads e Sockets\*

**2.2** **Organização do Código**

O código encontra-se subdividido em várias partes, temos uma pasta onde estão contidas as diversas interfaces disponíveis no projeto.

Barrel\_C\_I – interface que define como é que os clientes vão interagir com a informação armazenada no barrel.

Barrel\_I – interface que é implementada pelo server onde estão os métodos que irão ser usados pelo server.

Client\_I – interface que

Downloader\_I – interface

Gateway\_I – interface

Noutra pasta encontramos os recursos

Temos ainda uma pasta destinadas aos servers, que permitem

Por fim temos os 4 programas principais

Barrel –

Client –

Downloader –

Gateway Server –

1. **Multicast**
   1. **Downloaders e Storage Barrels**

A comunicação entre os Downloaders e os Storage Barrels é estabelecida através de um protocolo Multicast

1. **Componente RMI**

**4.1 Métodos Remotos Disponibilizados**

1. **Estruturas de Dados Utilizadas**
2. **Distribuição de Tarefas**

Foi seguida a distribuição de tarefas sugerida no enunciado do projeto com o aluno Afonso Duarte tendo assumido o papel de Elemento 1 e o Miguel Rainho de Elemento 2. O Miguel Rainho também ajudou a implementar as funcionalidades do Elemento 1 em alguns momentos.

1. **Testes Realizados**