**Sistemas Distribuídos**

Ano letivo 2019/2020

**Relatório**

**Projeto de Sistemas Distribuídos**

**Meta 1**



Trabalho Realizado por:

Alexandre Maria Martins Magalhães Teixeira Serra – 2017248031

João Gabriel de Matos Fernandes – 2017247486

Departamento de Engenharia Informática

Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Arquitetura de software

**Multicast Server**

**Threads:**

* Multicast Server → É a thread principal do multicast server. Em primeiro lugar, obtém o número do Multicast Server. De seguida, vai buscar a informação aos ficheiros e inicia as restantes threads. Após isto, está à escuta de mensagens provenientes do grupo Multicast.
* Multicast Server Action → executa cada pedido recebido pela thread Multicast Server.
* Web Craweler → faz uma indexação pedida por um administrador
* Multicast Server Control → periodicamente, manda mensagem por multicast aos restantes servidores multicast e aguarda, durante 5 segundos, por uma resposta de todos. De seguida, verifica se todos responderam. Se algum não respondeu, manda mensagem ao servidor RMI a avisar que o número do servidor Multicast que foi abaixo.
* Synchronization → periodicamente, sincroniza com os outros multicast servers informação sobre a indexação feita, URL’s, pesquisas efetuadas e utilizadores.
* MulticastAdminPage → periodicamente, atualiza o top 10 das pequisas efetuadas e dos URL’s com mais links a apontar para eles.

**Sockets:**

* Multicast Socket → recebe mensagens do RMI server e de outros multicast servers, respondendo pelo mesmo
* TCP Socket → usado para efetuar a sincronização entre os multicast servers

**Organização do código (main e classes):**

* MulticastServer → Programa a correr para iniciar o multicast server. As funcionalidades são iguais à thread Multicast Server.
* MulticastServerInfo → classe com o número do servidor, endereço TCP, porto TCP e carga do servidor.
* Search → classe com pesquisa e número de vezes que foi pesquisada.
* URL → classe URL com o URL, título e uma citação do texto da página, o número de links e os mesmo que apontam para este URL.
* User → classe com as informações do utilizadores (username, password, se é administrador, histórico de pesquisas, se está logado, o número de cliente e se tem notificação)

**RMI Server**

**Threads:**

**Sockets:**

**Organização do código:**

**RMI Client**

**Threads:**

**Sockets:**

**Organização do código:**