

# Report

## SDIS – Distributed Backup Service

**Turma 3**  
**Grupo 06**

Bruno Rafael Leite Ribeiro – 201505817  
Cleverson Veloso Nahum – 201710783

Em cada instância do servidor são executados 4 canais diferentes representando o Canal de Backup (**MDB**), Canal de Recuperação (**MDR**), Canal de Controle (**MC**) e por último o canal de comunicação com o cliente utilizando RMI, chamado de Canal Unicast (**UC**).

Cada canal citado é implementado numa thread diferente no servidor, de forma que todos os canais possam atuar de forma independente e sem interferir na comunicação dos outros canais, quando executado qualquer tipo de ação que exija bloqueio devido à espera por alguma ação (por exemplo, atividades de escrita e leitura em disco).

Basicamente, os canais estão divididos em dois grupos diferentes, o grupo de canais que interagem com a interface multicast e o grupo que interage com o cliente baseado em RMI. Fazem parte dos canais que interagem com a interface multicast o **MDB**, **MDR** e **MC**. Esses três canais herdam as suas características duma classe pai denominada Channel que é a responsável por implementar o código referente ao sistema de Threads e a função padrão para o envio de mensagens para outros canais. Ainda nessa classe, temos a função `select()` que

será substituída nas classes filhas de Channel por funções que se adequem ao tratamento de mensagens recebidas em cada canal.

A classe do canal UC não herda da classe Channel, pois ela é executada na thread principal. Ela herdará da classe FileChunk, que por sua vez é uma implementação da interface RMI contendo todas as funções necessárias para atender as chamadas do cliente no peer principal (isto é, o peer que interage diretamente com o cliente), como as funções de backup, restore e delete de arquivos completos no armazenamento do peer.

Uma instância do objeto FileChunk é instanciada na classe UC, através dessa instância todas as funções definidas na classe serão registradas no RMIRRegistry do JAVA, de forma a que todos os servidores conectados na rede do RMIRRegistry possam usar as funções disponibilizadas pelo objeto e fazer chamadas ao mesmo. Toda a conexão TCP, chamada de funções e transporte de argumentos entre os servidores com RMIRRegistry conectados é abstraída pela implementação da biblioteca RMI do Java.