

## Projeto 2 - Sistemas Distribuídos - Chordzilla

<https://youtu.be/ent6er7JOqM>

Matheus Ferreira Machado - 11201920804

### Executando o projeto

Para compilar os arquivos protobuf, compilar o projeto e rodar o servidor, a partir da raiz do projeto, execute os comandos:

<b><code>sudo apt install -y openjdk-17-jdk</code></b>	# Instala a versão 17 do Java para executar o projeto
<b><code>cd chordzilla</code></b>	# Entra na lib
<b><code>./mvnw clean install</code></b>	# Executa o build da lib que implementa o chord
<b><code>cd ../codehub</code></b>	# Entra na aplicação
<b><code>./codehub/mvnw spring-boot:run</code></b>	# Executa a aplicação que utiliza o chord

Isso vai iniciar uma instância da aplicação, para executar outras instâncias (nós), basta executar novamente o segundo comando.

### Aplicação

A aplicação desenvolvida a partir das funcionalidades do chord é um sistema de arquivos distribuídos que simula o armazenamento de repositórios do *Github*. Ela possui as funcionalidades de armazenar e recuperar diretórios, incluindo todos subdiretórios e seus arquivos. Os arquivos são armazenados nos nós que fazem parte da rede chord.

A partir da raiz do projeto há dois diretórios. O diretório *chordzilla* é onde está toda a implementação da chord, incluindo o protocolo da rede, o envio e recebimento de mensagens GRPC, roteamento e armazenamento dos dados. Essa implementação está é uma biblioteca, que disponibiliza a API para utilização do chord. Já o diretório *codehub* é a aplicação que de fato utiliza o chord, ela utiliza as APIs da biblioteca para armazenar os arquivos dos diretórios.

### Uso

Executando o projeto via CLI, serão apresentadas as opções de comando para interagir com a aplicação. Inicialmente pode-se escolher entre iniciar uma rede nova ou juntar-se a uma rede existente. É possível iniciar uma rede nova a partir de um endereço de IP e porta específicos ou aleatórios. Para juntar-se à rede é possível informar o host específico ou usar a lista de hosts no arquivo `/codehub/src/main/resources/hosts.json`. Se não for possível juntar-se a rede, inicia-se uma rede nova.

Após se juntar à rede, é possível executar os comandos para armazenar ou recuperar um repositório. Para ambos casos é necessário informar o *path* do diretório que será armazenado ou onde será feito o download. Quando desejar sair da rede basta executar o comando `'leave'`. Os detalhes da execução da aplicação estão no vídeo.

### Desafios

A primeira versão da aplicação teve problemas de performance grandes. Devido à natureza da aplicação, que armazena repositórios com possivelmente centenas ou milhares de arquivos, inicialmente foi inviável realizar transferência em larga escala entre os nós da rede, pois havia uma degradação notável de performance. Assim, foi necessário realizar um trabalho de otimização tomando como referência a documentação do GRPC [Performance Best Practices](#).

Um desafio que consumiu um esforço considerável foi a utilização de uma [versão com problemas](#) da biblioteca GRPC. Essa versão da biblioteca para Java apresenta problemas de corrupção das mensagens GRPC enviadas. Encontrei esse problema diversas vezes e foi necessário um esforço de depuração até descobrir a corrupção de mensagens.