



Projeto 02

Sistemas Distribuídos

Alunos:

Leonardo Cerce Guioto

Rodrigo Augusto Valeretto

NUSP: 10716640

NUSP: 10684792



Aplicação *mobile* com comunicação *front/back-end*

O nosso objetivo para esse segundo projeto da disciplina de Sistemas Distribuídos (SSC 0904) é o desenvolvimento de um aplicativo que utilizasse um modelo de cliente-servidor fazendo uso das interfaces RPC/RMI.

A partir disso o nosso grupo decidiu por implementar um aplicativo de *upload* de arquivos para um banco de dados.



RPC

Chamada remota de procedimento (RPC, acrônimo de *Remote Procedure Call*) é uma tecnologia de comunicação entre processos que permite a um programa de computador chamar um procedimento em outro espaço de endereçamento (geralmente em outro computador, conectado por uma rede). O programador não se preocupa com detalhes de implementação dessa interação remota: do ponto de vista do código, a chamada se assemelha a chamadas de procedimentos locais.



RMI

O RMI (*Remote Method Invocation*) é uma interface de programação que permite a execução de chamadas remotas no estilo RPC em aplicações desenvolvidas em Java. É uma das abordagens da plataforma Java para prover as funcionalidades de uma plataforma de objetos distribuídos. Esse sistema de objetos distribuídos faz parte do núcleo básico de Java desde a versão JDK 1.1, com sua API sendo especificada através do pacote *java.rmi* e seus subpacotes.


Através da utilização da arquitetura RMI, é possível que um objeto ativo em uma máquina virtual Java possa interagir com objetos de outras máquinas virtuais Java, independentemente da localização dessas máquinas virtuais.



Nosso Projeto: *Storage Files*

O *front-end* da nossa aplicação foi desenvolvido na linguagem *javascript* se utilizando de um *framework* chamado *React Native*, o qual permite a passagem de código *javascript* para código nativo *mobile* tanto de Android quanto de IOS.

Sendo assim, o paradigma utilizado pelo nosso projeto é o RPC pois o RMI exige que se tenha uma comunicação através de *virtual machines* do Java, o que ocorre apenas em Java mesmo e não em *javascript*. Logo, o nosso programa é composto de inúmeras chamadas de procedimentos que requisitam funções a serem executadas no *back-end* como fazer *upload* e *download* de um arquivo para um banco de dados ou até mesmo pegar um *array* de todos os arquivos disponíveis e retornar para o *front*.



Back-end

A parte do *back-end* do nosso programa foi feita utilizando *javascript* também. Fizemos uma integração com um banco de dados *Mongo*, no qual armazenamos os arquivos enviados pelo *back* (o qual recebe uma requisição do *front*).

A conexão entre *back* e *front* foi feita utilizando um *framework* de conexão chamado *Axios*.



Middleware e proteções

Middleware é o software de computador que fornece serviços para *softwares* aplicativos além daqueles disponíveis pelo sistema operacional.

É utilizado para mover ou transportar informações e dados entre programas de diferentes protocolos de comunicação, plataformas e dependências do sistema operacional. É geralmente constituído por módulos dotados com APIs de alto nível que proporcionam a sua integração com aplicações desenvolvidas em diversas linguagens de programação e interfaces de baixo nível que permitem a sua independência relativamente ao dispositivo.



Multer no back-end

O *multer* é um *framework* utilizado no *back-end* que serve justamente para fazer a comunicação com o *middleware* do *Mongo*. A partir dele podemos fazer as validações necessárias para o acesso ao banco e também colocar certas condições específicas para as funções caso desejado. Por exemplo, poderíamos limitar nosso programa a um programa de *uploads* de arquivos de texto (.txt).



Demonstração prática do *App*

O app vai ser rodado em um emulador de Android.

O *back* está hospedado na *web* num servidor de aplicação chamado *Heroku*.



Projeto 02

Sistemas Distribuídos

Alunos:

Leonardo Cerce Guioto

Rodrigo Augusto Valeretto

NUSP: 10716640

NUSP: 10684792



Link da apresentação

Link: <https://github.com/RodrigoValeretto/ProjetoSD>



Autoavaliação

Cremos que houve certa dificuldade na hora de explicar alguns conceitos, a dizer RPC/RMI. Além disso cremos que existem certas funcionalidades que omitimos do projeto por motivos de dificuldade como por exemplo excluir um arquivo do sistema.

Dessa forma, acreditamos que uma nota justa para o projeto dois seria 7.