**Distributed Systems**

**Continuous Assessment (CA) Type: Project**

**Smart Home**

**Due Date: 19th April 2024**

**Lecture: Mark Cudden**

Repository:

<https://github.com/sergio-oliveira-br/CA1-DistributedSystems>

Sergio Vinicio da Silva Oliveira

[**x23170981@student.ncirl.ie**](mailto:x23170981@student.ncirl.ie)

Table of Contents

1. Domain Description ............................................................................................................ ?

2. Service Definition ....................................................................................................................... ?

3. Service Implementations................................................................................................................... ?

4. Advanced Features ................................................................................................. ?

5. Command Line & GUI ................................................................................................................... ?

6. GitHub.......................................................................................................................... ?

**1. Domain Description**

Para aplicar os conceitos apresentados em sala de aula, optei em abordar os quatros serviços gRPC, na qual são associados dentro de um contexto de uma Smart Home. Além disso, na tentativa de aprimorar e reforçar os meus conceitos nos serviços gRPC, e partindo do princípio que a prática leva a perfeição, três versões foram criadas.

As duas primeiras versões se diferem apenas na apresentação gráfica, onde uma aborda os conceitos de linha de comando, e a segunda abordando a interface de usuário gráfica GUI por meio dos recursos Java Swing.

A terceira versão já não esta associada diretamente com a ideia principal do contexto Smart Home, entretanto foi através desta que reforcei a abordagem dos serviços gRPC e fui introduzido ao JavaFX.

**2. Service Definition**

Como mencionado anteriormente, o projeto consiste em três arquivos na qual os dois primeiros se diferem apenas na apresentação gráfica.

Para estas versões nomeadas Smart Home v1.0 e Smart Home GUI v1.1 proponho os seguintes serviços.

Unary RPC, propõe ao usuário envia uma única solicitação, e obtém uma única resposta do servidor. A aplicação deste conceito está associada ao comando de ligar e desligar luzes, e a abertura e fechamento de portas.

Client Streaming RPC, sugere o cliente enviar uma sequência de mensagens ao servidor, que por sua vez responde com uma única mensagem. Neste projeto este conceito está sendo associado no procedimento de verificação de conexão, onde o usuário envia seus dados ao Servidor para confirmar sua conexão.

Bidirectional Streaming RPC: Este serviço propõe uma interação contínua entre cliente e o servidor. E neste projeto está sendo abordado no tópico Smart Thermostats, onde o cliente solicita ajuste de temperatura ambiente e a previsão do tempo, obtendo as respostas como confirmação do ajuste de temperatura ambiente e a previsão do tempo do dia seguinte.

Referente a versão nomeada SmarOven v2.0 fora proposto, os serviços

Unary RPC para uma simples saudação e personalização de acesso ao disposto Smart Oven.

Server Streaming RPC, enviando dados fictícios que simulam a temperatura de um forno.

**3. Services Implementations**

Para implementar os serviços mencionados, fora utilizado Proto Buffers, especificamente a sintaxe proto 3.

Uma vez que os serviços e métodos foram definidos, o compilador Java gera os arquivos necessários com as classes para cada mensagem incluindo obter e definir valores de campo, serializar suas mensagens em um fluxo de saída e analisar suas mensagens de um fluxo de entrada.

A estrutura de implementação tem com base a extensão das classes geradas através do Proto Buffers, onde inicialmente construímos um objeto de resposta para retornar ao cliente ou servidor conforme declarado na definição de serviço. Além disso, temos incluso os métodos de resposta do observador, que por meio dos metodos onNext( ), onError( ) e onCompleted ( ) podemos acompanhar o status atual do serviço.

Além da implementação dos métodos mencionados anteriormente, a implementação e inicialização do servidor gRPC é fundamental para que ocorram essa comunicação entre cliente e servidor.

De modo resumido, esta implementação do Servidor é construída por meio da especificação de um endereço e porta para efetivar a comunicação, sendo possível adicionar este serviço ao construtor por meio do método addservice( ), e por fim, iniciando o servidor RPC por meio dos métodos build( ) e start( ).

Referente ao Client-Side, é fundamental a criação de um canal gRPC para o Stub, que é por meio do método ManagedChannelBuilder( ) que especificamos a porta ou endereço do servidor a ser utilizado para a comunicação com o servidor.

Uma vez que o canal foi criado, os métodos newStub( ) e newBlockingStub fornecidos através do Proto Buffers permitem a associação com o Stub, que neste caso utilizei o de bloqueio/síncrono, na qual a chamada RPC aguarda a resposta do servidor, retornando uma resposta ou uma exceção.

Feito isso, implementação do cliente e servidor estão concluídas e prontas para chamar os métodos de serviços declarados.

**4. Testing & Responsiveness**

Talk about website responsiveness here along with **Optimisation, SEO and Testing**. You can talk about how you made your website responsive. You could also mention how you used JavaScript: loops, if statements, etc. How you deployed your website?

**5. Command-Line and GUI**

Devido aos desafios que incialmente não estava claro, a versão Smart Home v1.0 foi desenvolvida para interagir com o usuário por meio de command-line, não contemplando nada de especial, e focando exclusivamente no desenvolvimento e implementação dos serviços gRPC.

Durante o desenvolvimento da versão inicial, percebi que o tempo disponível para conclusão deste projeto possibilitava a incrementação de novos desafios, e foi ai que optei em migrar a versão inicial para o GUI aplicando os conceitos Java Swing.

Nesta versão Smart Home GUI v1.1, busquei reforçar os conhecimentos adquiridos na implementação gRPC, e aplicar esforços no aprendizado Java Swing.

Durante uma aula

**6. References**

Finance, B. &., 2015. *Flexible working worth €2.5bn to Irish economy.*[Online]    
Available at: http://businessandfinance.com/news/flexible-working-worth-e2-5bn-to-irish-economy/    
[Accessed 8 March 2023].