

Trabalho Prático nº2 – Sistema de Reserva de Bilhetes de Teatro

Engenharia Informática

Sistemas Distribuídos

**Ivan Miguel Serrano Pires**

**Hugo Paredes**

**Autores**

Luís Pimenta al70827

Pedro Guerra al70596

Tiago Morais al71395

Tomás Silva al70680

# Introdução

No âmbito da disciplina de Sistemas Distribuídos, foi-nos proposto a construção de um sistema de reserva de bilhetes de teatro, que visa a retratar uma aplicação de bilheteira virtual de teatros. Com o tema já definido, tivemos que usar a framework gRPC de forma a criar uma comunicação cliente/servidor. O gRPC (Google Remote Procedure Call) é um modelo de comunicação, que permite que o cliente e o servidor comuniquem de modo transparente. Por último, a aplicação foi construída em C#.

# Servidor e Serviços RPC

É no servidor que está presente a conexão à base de dados. A função principal do servidor é guardar toda a informação, por exemplo guardar uma compra, um user e ou uma sessão. Também tem a implementação dos serviços. Os serviços são classes, com um conjunto de métodos.

Nesta aplicação, os serviços são:

* O Login, é responsável por verificar se o user tem permissão para aceder à página que está a tentar aceder;
* O Register, que contém as funções responsáveis por registar os diferentes tipos de clientes;
* O UserService, que retorna a informação do user conectado bem como a lista dos utilizadores, e tem a capacidade de adicionar dinheiro à conta virtual;
* O LocalizationService, o serviço contem funções capazes de retorna as localizações consoante a localização do user, bem como adicionar localizações novas;
* O TheaterService, com funções responsáveis por retornar teatros, mudar o estado de um teatro, retornar a informação de um teatro em especifico a partir do ID, atualizar os dados de um teatro e adicionar um teatro;
* O ShowService, com funções responsáveis por retornar espetáculos, mudar o estado de um espetáculo, retornar a informação de um espetáculo em especifico a partir do ID, atualizar os dados de um espetáculo e adicionar um espetáculo;
* O SessionService, com funções responsáveis por retornar sessões, mudar o estado de uma sessão, retornar a informação de uma sessão em especifico a partir do ID, atualizar os dados de uma sessão e adicionar uma sessão;
* O CartService, com a capacidade de ver que há lugares disponíveis, bem como reservar um lugar e cancelar a reserva do lugar;
* O CompraService, com funções responsáveis por comprar as sessões previamente adicionadas ao carrinho de compras, retornar o histórico de compras de um user, retornar a informação de uma compra, cancelar a compra caso seja possível e retornar todas as compras já feitas.

Exemplos de código:Anexos 1-4 & 7

# Cliente para Utilizador

O utilizador antes de realizar qualquer movimento, tem que fazer login na aplicação, caso já esteja registado. O utilizador não está constantemente ligado ao servidor, só quando efetua algum movimento é que é criado um canal que é responsável pela conexão cliente/servidor. Para haver uma comunicação, o utilizador a partir de uma interface gráfica é capaz:

* Visualizar a lista de espetáculos já assistidos;
* Visualizar bilhetes comprados;
* Adicionar fundos à conta virtual;
* Comprar bilhetes para sessões;
* Pesquisar teatros, espetáculo e sessões;
* Anular compra do bilhete;
* Guardar espetáculos já assistidos;

# Cliente para Administração

O administrador antes de realizar qualquer movimento, tem que fazer login na aplicação, caso já esteja registado. O administrador não está constantemente ligado ao servidor, só quando efetua algum movimento é que é criado um canal que é responsável pela conexão cliente/servidor. Para haver uma comunicação, o administrador a partir de uma interface gráfica é capaz:

* Visualizar a lista de teatros;
* Visualizar a lista de espetáculos;
* Visualizar a lista de sessões;
* Visualizar a lista de bilhetes comprados;
* Visualizar o histórico de pedidos de logging;
* Adicionar managers e users.

Exemplos de código: Anexos 5-6

# Cliente para Gestão

O administrador antes de realizar qualquer movimento, tem que fazer login na aplicação, caso já esteja registado. O gestor não está constantemente ligado ao servidor, só quando efetua algum movimento é que é criado um canal que é responsável pela conexão cliente/servidor. Para haver uma comunicação, o gestor a partir de uma interface gráfica é capaz:

* Adicionar teatros;
* Editar teatros;
* Adicionar espetáculos;
* Editar espetáculos;
* Remover espetáculos.

**Anexos**

Uma imagem com texto

Descrição gerada automaticamente

Fig 1 – Conexão

Uma imagem com texto

Descrição gerada automaticamente

Fig 2 - UserConnected

Uma imagem com texto

Descrição gerada automaticamente

Fig 3 - Serviços

Uma imagem com texto

Descrição gerada automaticamente

Fig 4 – Mensagens

Uma imagem com texto

Descrição gerada automaticamente

Fig 5 – Logs Error

Uma imagem com texto

Descrição gerada automaticamente

Fig 6 – Logs Warning

Uma imagem com texto

Descrição gerada automaticamente

Fig 7 – Configurar o servidor