Relatório trabalho prático

Sistemas Distribuidos

Trabalho realizado por:
Ruben Farinha nº48329
Diogo Ferreira nº43075

Docentes:
Professor José Saias
Professor Pedro Patinho

1. Introdução

Neste trabalho o objetivo é implementar as aplicações servidor, cliente geral e cliente de gestão que permitam, desde as mais variadas localizações, o acesso ao serviço para gerir as perfomances de artistas de rua, permitindo registar artistas e localizá-los quando estão a fazer as suas atuações.
Neste trabalho era pedida uma aplicação cliente-servidor, através de uma solução de middleware apresentada nas aulas.

2. Base de Dados

Iniciamos o projeto pela implementação da base de dados, estabelecendo duas tabelas fundamentais para o desempenho otimizado das aplicações: uma destinada aos artistas e outra às doações. Desenvolvemos essas tabelas com os atributos essenciais para atender às especificações dos pedidos que planejamos realizar.

```sql
CREATE TABLE artista (
    artist_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    artist_type VARCHAR(255) NOT NULL,
    location VARCHAR(255) NOT NULL,
    is_acting BOOLEAN NOT NULL,
    approval_status VARCHAR(20) DEFAULT 'Não Aprovado' CHECK (approval_status IN ('Aprovado', 'Não Aprovado')),
);

CREATE TABLE donation (
    donation_id SERIAL PRIMARY KEY,
    id INTEGER REFERENCES artista(artist_id),
    donation_amount DECIMAL(10,2) DEFAULT 0.00,
);
```

3. Servidor-Cliente

Estabelecemos a comunicação entre o servidor e o cliente por meio de JavaRMI, utilizando o objeto remoto Tanaforja. A implementação inicial do servidor foi desenvolvida em grande parte durante a aula prática, mas precisou passar por algumas adaptações para se alinhar ao contexto específico do projeto. Da mesma forma, o cliente também passou por ajustes necessários, incluindo a implementação de um menu que é exibido no terminal.

4. Métodos:

• **registarArtista,** na qual podemos registar um novo artista inserindo o seu nome, tipo de arte, localização e se está a atuar.
• **obterListadeArtistas,** na qual obtemos a lista de todos os artistas que estão registados.
• **obterListaLocal,** na qual obtemos a lista de localizações onde temos artistas a atuar.
• **obterListadeArtistasPorArte,** na qual escrevendo um tipo de arte conseguimos obter todos os artistas que lhe correspondem.
• **enviarDonativo,** que nos permite enviar um donativo a algum artista. Para isto escrevemos o id do artista e também o valor monetário do donativo a ser enviado.
• **obterListadeDonativos,** em que recebemos uma tabela com os valores dos donativos já enviados, bem como o seu id (id da doação) e o id do artista a quem foi doado. Para isto inserimos também o id do artista que queremos consultar.

• **obterListadeArtistasEstado,** na qual recebemos uma tabela com todas as informações dos artistas que foram aprovados ou não aprovados, dependendo do que pretendemos. Ao selecionar esta opção é pedido que seja inserido o estado atual do artista (aprovado ou não aprovado).
• **aprovarArtistas,** que nos permite (apenas se estivermos no painel admin) alterar o estado de um artista de não aprovado para aprovado. Para isto temos de inserir o id do artista que queremos aprovar.
• **obterListadeArtistasLocalizacao,** em que recebemos uma tabela com a lista de todos os artistas a atuar na localização que escrevermos.

5. Comandos para execução do programa

Comandos para executar o programa: Utilizámos o NetBeans para compilar o projeto e os seguintes comandos (cada um num terminal diferente):

rmiregistry -J-cp -J. 9000

Servidor: java -classpath postgresql-42.7.1.jar:. SDWork1.Tanaforja.Server 9000

ClienteGeral: java -classpath build/classes SDWork1.Tanaforja.ClienteGeral localhost 9000
ClienteAdmin: java -classpath build/classes SDWork1.Tanaforja.ClienteAdmin localhost 9000

6. Observações

Durante a execução deste projeto, deparamo-nos com um dos maiores desafios ao lidar com interfaces remotas, uma vez que não tínhamos experiência prévia nesse tipo de tecnologia, exceto durante as aulas práticas. No que diz respeito à manipulação da base de dados, conseguimos realizar as consultas com certa facilidade, mas enfrentamos algumas dificuldades ao estabelecer a comunicação entre a base de dados e a aplicação. Isso se deveu, em parte, às nossas dificuldades na correta utilização dos comandos do terminal. Posteriormente, em algumas instâncias, enfrentamos problemas de sintaxe que resultavam em exceções no código.