UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA
Licenciatura de Engenharia Informática

Sistemas Distribuidos

Relatório do 1º Trabalho de Sistemas Distribuidos 2023/2024

Gustavo Dias Gomes nº48392 Bárbara Loureiro nº48469

Introdução

O sistema em questão é um aplicativo de gestão de artistas de rua baseado em Java, usando a tecnologia RMI (Remote Method Invocation) para comunicação entre clientes e servidor. O sistema permite a gestão de artistas, das suas atuações e donativos associados.

Desenvolvimento

Estrutura Geral do Código:

O código está organizado em várias classes: Server.java, Metodos.java, MetodosImp.java, Cliente_geral.java, e Cliente_gestao.java.

Server.java:

- Carrega as configurações do servidor a partir de um arquivo configs.properties.
- Estabelece uma conexão com o banco de dados PostgreSQL usando a classe PostgresConnector.
- Cria e registra o objeto remoto MetodosImp no Registro RMI.

Metodos.java: define a interface RMI com métodos remotos.

MetodosImp.java: implementa a interface e contém a lógica para operações como registro de artistas, listagem, aprovação, etc.

Cliente_geral.java:

- Implementa a lógica do cliente geral, interagindo com o objeto remoto para registrar artistas, listar artistas, doar, etc.
- Apresenta um menu interativo para o usuário fazer escolhas.

Cliente_gestao.java:

- Implementa a lógica do cliente de gestão, permitindo a gestão de artistas aprovados, não aprovados, e aprovação de artistas.
- Também apresenta um menu para escolhas do usuário.

Base de Dados:

O sistema utiliza um banco de dados PostgreSQL com três tabelas: artistas, performances, e donativos. Essas tabelas armazenam informações sobre artistas, suas atuações e donativos recebidos.

```
CREATE TABLE artistas (
    artistid serial PRIMARY KEY,
    nome varchar NULL,
    tipoarte varchar NULL,
    localizacao varchar NULL,
    atuar bool NULL,
    estado bool NULL
);

CREATE TABLE performances (
    performanceid serial NOT NULL,
    artistid int NOT NULL,
    data date NOT NULL,
    localizacao varchar NULL,
    FOREIGN KEY (artistid) REFERENCES artistas(artistid)
);

CREATE TABLE donativos (
    donativoid serial PRIMARY KEY,
    artistid int NOT NULL,
    valor decimal NOT NULL,
    data date NOT NULL,
    FOREIGN KEY (artistid) REFERENCES artistas(artistid)
);
```

Execução do programa

Para executar o programa no terminal da pasta do projeto, isto, após garantida a criação das tabelas sql :

```
javac -d build/classes -classpath build/classes src/main/java/project/*.java

rmiregistry -J-classpath -Jbuild/classes 9000

java -classpath build/classes:recursos/postgresql-42.5.0.jar project.Server

java -classpath build/classes:recursos/postgresql-42.5.0.jar project.Cliente_geral

java -classpath build/classes:recursos/postgresql-42.5.0.jar project.Cliente_gestao
```

Conclusão

Foi um trabalho desafiante, que melhorou muito as nossas capacidades tanto sobre Java RMI como conexão Servidor - Cliente. Conseguimos implementar o pedido, tentando sempre que o sistema apresentado ao utilizador ficasse simples, intuitivo e eficaz.