UNIVERSIDADE DE ÉVORA

Disciplina: Sistemas Distribuídos

Pedro Emílio Nº52649

Luís Carvalho Nº51817

Introdução

Para este trabalho foi-nos pedido para fazer um upgrade à versão do primeiro trabalho, já realizado, de modo a tornar a aplicação mais segura

Este upgrade visa não apenas melhorar o desempenho e a eficácia da aplicação, mas também a garantir uma experiência de utilização mais segura e adaptável a um maior volume de usuários.

Desenvolvimento

A interface escolhida para estabelecer a conexão entre o servidor e o cliente foi a interface RMI. Graças à facilidade de uso do RMI, a implementação das funções para gerenciar a base de dados foi realizada de forma eficiente.

Foi criado o ficheiro Servidor, que serve para inicializar o objeto remoto e de base de dados. De seguida foi criado também o ficheiro ConexaoBD que serve para conectar à base de dados criadas, sendo que lá têm as queries todas que permitem consultar, inserir e atualizar campos das bases de dados . Por fim foi criado os ficheiros que Artistas, Atuacoes, Classificacao, Donativo e Utilizador servem para organizar e facilitar o envio de dados para o servidor.

Base de Dados

```
CREATE TABLE utilizadores (
        username varchar(255) NOT NULL,
        email varchar(255) NOT NULL,
        password varchar(255) NOT NULL,
        tipoDeUtilizador varchar(255) NOT NULL,
        PRIMARY KEY(username)
);
CREATE TABLE artistas (
        nomeartista varchar(255) NOT NULL,
        tipodearte varchar(255) NOT NULL,
        latitude DOUBLE PRECISION,
        longitude DOUBLE PRECISION,
        aatuar BOOLEAN, estado varchar(255) NOT NULL,
        artistid integer,
        PRIMARY KEY(artistid)
);
CREATE TABLE atuacoes (
        iddoartista INTEGER REFERENCES,
        latitude DOUBLE PRECISION,
        longitude DOUBLE PRECISION,
        datadaatuacao DATE,
        idatuacao integer, PRIMARY KEY(idatuacao)
);
CREATE TABLE donativos (
        idartista integer,
        valordodonativo integer,
        userdodonativo varchar(255) NOT NULL,
        datadodonativo DATE,
        iddoacao integer,
        PRIMARY KEY(iddoacao)
);
```

```
CREATE TABLE classificacoes (
        artistid integer,
        rating integer, -- 0 a 10 0-> Muito mau 10 -> Muito bom
        utilizadorClas varchar(255) NOT NULL,
        idclassificacao integer,
        PRIMARY KEY(idclassificacao)
);
```

Execução

Para executar a solução desenvolvida é preciso seguir os seguintes passos:

1. É necessário iniciar o servidor RMI com o seguinte comando: rmi: rmiregistry -J-classpath -Jbuild 9000
2. Para compilar o programa utilizamos o comando seguinte: javac -cp resources/postgresql.jar -d build *.java
3. Para correr o servidor fazemos: java -cp build:resources/postgresql.jar Servidor.java 9000 localhost bdtrab2 user1 umaPass
4. Por último para correr o cliente fazemos: java -cp build:resources/postgresql.jar Cliente.java localhost 9000