UNIVERSIDADE
DE ÉVORA

Engenharia Informática
(2023/2024)

Sistemas Distribuídos

Sistema de Gestão de Artistas de Rua

SeekArtist 2.0

Pedro Bacalhau - 52043
António Carvalho - 51483

Objetivos

O objetivo do trabalho consistia num upgrade a versão do primeiro trabalho porposto de modo a tornar a aplicação mais segura, robusta e escalavel.

Base dados

A base dados implementamos consiste num upgrade a base dados antiga de modo a podermos adquirir novas funcionalidades, ficamos assim com as seguintes tabelas:

```
CREATE TABLE users (
    username VARCHAR(255),
    email VARCHAR(255) unique ,
    password VARCHAR(255) NOT NULL,
    type VARCHAR(20),
    PRIMARY KEY(username)
);

CREATE TABLE artists (
    id SERIAL,
    name VARCHAR(255) unique,
    type VARCHAR(255),
    latitude DOUBLE PRECISION,
    longitude DOUBLE PRECISION,
    performing BOOLEAN,
    approved BOOLEAN,
    PRIMARY KEY(id)
);

CREATE TABLE donatives (
    id SERIAL PRIMARY KEY,
    artistid INTEGER,
    donation_date DATE,
    username VARCHAR(255) REFERENCES users(username),
    value INTEGER
);

CREATE TABLE performances (
    actuation_id SERIAL,
    artist_id INTEGER REFERENCES artists(id),
    latitude DOUBLE PRECISION,
```

```
    longitude DOUBLE PRECISION,
    actuation_date DATE,
    PRIMARY KEY(actuation_id,longitude,latitude,actuation_date)
);

CREATE TABLE rating(
    rating_id serial primary key,
    artist_id INTEGER REFERENCES artists(id),
    username VARCHAR(255) REFERENCES users(username),
    rating INTEGER
);
```

Desenvolvimento

Fizemos uso da interface ***Rmi*** para fazer a ligação servidor-cliente. Começámos por criar as tabelas numa base de dados local e de seguida a ligação do servidor com o cliente. A partir do momento que tínhamos esta ligação dada a simplicidade do RMI foi bastante direto a implementação das funções de gestão das bases de dados. Sendo possível o envio e pedido de informação do lado cliente para a base de dados de seguida tivemos que fazer os pedidos de informações ao utilizador da aplicação do cliente.
Nesta versão da aplicação ambos os clientes(user e admin) ficam no mesmo ficheiro e são separados pela autenticação requerida no início da aplicação.
Para que isto funcionasse como pedido foi implementado um sistema de autenticação e registro.

Após o registro do lado do **cliente normal** é possível:

- registar um artista,
- Listar e filtrar artistas(localização e arte),
- Listar locais com atuações no momento,
- Listar as atuações passadas de um determinado artista como a próxima atuação,

- Fazer uma doação e observar as doações recebidas por cada artista.
- Dar classificação a um Artista, como apresentar o rating deles na listagem de artistas.

Do lado do **cliente administrador** é possível:

- Apresentar os artistas por estado e aprovar aqueles que ainda não se encontram aprovados,
- Dar permissão de administrador a um user,
- Consultar e alterar informação de um Artista,
- Como todas as operações permitidas por um User normal.

O servidor (**server**) tem apenas a inicialização do objeto remoto (**remoteObjectImpl**) da interface (**remoteObject**) e da base de dados. O objeto remoto apenas contém as funções que chamam as funções da classe que gere a base de dados (**BDserver**). Esta por fim tem as funções que permitem consultar, inserir e atualizar campos das bases de dados.

Compilação e Execução

Para compilar e executar o trabalho decidimos proceder da maneira tavez mais habitual e mais básica, pelo terminal diretamente na pasta do trabalho.
Para **compilar** usámos o comando:
javac -cp src/main/java/resources/postgresql.jar -d build sr c/main/java/.java*
Para inicializar a interface rmi -
rmiregistry -J-classpath -Jbuild 9000
para executar o servidor -
java -cp build:src/main/java/resources/postgresql.jar -Djdbc.drivers=org.postgresql.Driver server 9000 local

host (nome base dados) (utilizador da bd) (palavra-passe)

para executar o cliente - *java -cp build:src/main/java/resources/postgresql.jar client localhost 9000*

Conclusão

Este trabalho permitiu-nos aplicar de uma maneira mais complexa algumas das bases aprendidas, mesmo assim não fizemos uso de nenhum utilitário.