

Misturador de Músicas

Universidade De Aveiro

André Alves, Daniel Correia, Pedro Almeida,
Pedro Valente



Misturador de Músicas

Departamento de Electrónica, Telecomunicações e
Informática

Universidade De Aveiro

André Alves, Daniel Correia, Pedro Almeida, Pedro Valente
(88811) andr.alves@ua.pt, (88753) dcorreia@ua.pt,
(89205) pedro22@ua.pt, (88858)pedro.valente@ua.pt

15/06/2018

Conteúdo

| | | |
|----------|----------------------------------|----------|
| 1 | Introdução | 3 |
| 2 | Desenvolvimento | 4 |
| 2.1 | Interface Web | 4 |
| 2.2 | Aplicação Web | 5 |
| 2.2.1 | list | 5 |
| 2.2.2 | get | 5 |
| 2.2.3 | sheet | 5 |
| 2.2.4 | vote | 5 |
| 2.3 | Persistência | 6 |
| 2.4 | Gerador de Músicas | 7 |
| 3 | Conclusões Finais | 8 |
| 4 | Contribuições dos autores | 9 |

Lista de Figuras

| | | |
|-----|---------------------------------|---|
| 2.1 | Misturador de músicas | 7 |
|-----|---------------------------------|---|

Capítulo 1

Introdução

O objetivo do projeto é a criação de um sistema que permita criar músicas através da composição de pedaços de música. O projeto funciona a partir de um servidor do XCOA. O sistema é composto por vários componentes, componentes estes que são:

- Interface Web
- Aplicação Web
- Persistência
- Gerador de Músicas

A Interface Web (Seção 2.1) é composta por quatro páginas HTML cada uma com um propósito próprio a explicar mais à frente.

A página Início é uma pequena página inicial na qual o utilizador poderá ver aquilo que pode fazer neste site, assim como algumas informações sobre os autores e o local onde o projeto foi desenvolvido.

A Aplicação Web consiste num programa python que serve conteúdos estáticos, esta parte do trabalho serve como ponte para todos os outros componentes, para além disto tem também objetivos próprios a ser explicados mais à frente na Seção 2.2.

A parte da Persistência consiste numa base de dados relacional e obtenção de dados da mesma, tudo isto será depois utilizado pela parte da Aplicação Web para registar as músicas criadas, os excertos que existem e os votos de cada um dos utilizadores, explicado em maior detalhe em Seção 2.3.

A parte do Gerador de Música deverá aceitar uma pauta na forma de um dicionário, gerando a música especificada. Caso a geração tenha sucesso, a música deverá ser escrita no sistema de ficheiros. Deverá igualmente ser gerado um identificador com base no conteúdo, sendo este devolvido pelo módulo para persistência na base de dados. Este componente deverá suportar pelo menos 5 efeitos a aplicar no momento da geração da música.

Capítulo 2

Desenvolvimento

2.1 Interface Web

A Interface Web é composta por quatro páginas desenvolvidas através de e HyperText Markup Language (HTML), JavaScript (JS) e Cascading Style Sheets (CSS).

A página *Início* é uma página inicial, onde se encontra informação sobre o que poderá ser feito no site. Além de servir como ligação para as outras páginas, a página *Início* tem também informação sobre os elementos que desenvolveram o projeto e o local onde o mesmo foi desenvolvido.

A segunda página do site é a página *Músicas*. Nesta página estão todas as músicas produzidas/editadas por todos os utilizadores. É também nesta página que os utilizadores poderão votar músicas, apenas um voto por utilizador. O voto pode ser *upvote* ou *downvote* e pode ser alterado. No projeto esta página apenas vai ter uma música de exemplo mostrando que os votos funcionam. Isto deve-se a um problema no Gerador de Músicas.

Na terceira página estão todos os excertos de sons que podem ser utilizados para criar músicas no Gerador de Músicas, ordenados por ordem alfabética. Os excertos podem também ser transferidos individualmente.

A quarta página do site diz respeito ao *Misturador*. É nesta página que o utilizador pode misturar excertos criando uma música. Essa música depois de criada aparece na página das *Músicas*, já em cima descrita. Ao misturar as músicas, o utilizador controla o número de batimentos por minuto da música assim como a adição de efeitos ao som e o volume.

No projeto, ao selecionar os excertos, o números de batimentos, o volume e os efeitos e ao clicar no botão *Mistura-te!* é criado um ficheiro JSON (o ficheiro JSON é criado no ficheiro *get_sheet.js* através da função *generate_json()*) com todas as especificações selecionadas mas não é possível criar de facto a música.

2.2 Aplicação Web

A aplicação Web é o programa em python que serve como ponte entre todas as partes do projeto. Tem as funções *list*, *get*, *sheet* e *vote*:

2.2.1 list

A função *list* está dividida nas *samples* e nas *songs*.

Samples Este módulo quando usado vai devolver o JSON de todas as samples com o *name*, *date*, *id*, *length* das samples guardado na base de dados na tabela de Samples.

Songs À semelhança do módulo das Samples, este irá devolver o JSON de todas as músicas criadas com o *name*, *date*, *id*, *length*, *uses*, *votes*, *author* que também se encontra guardado na base de dados.

Aceder Para aceder deve-se escrever */list?type=songs* para o JSON de músicas e */list?type=samples* para os das samples

2.2.2 get

Esta função devolve todos os atributos de uma sample ou song através do seu id.

Não se encontra funcional.

Aceder Para aceder deve-se escrever */get?id=id* em que o id é uma string de 16 caracteres de 0 a 9 e de A a F.

2.2.3 sheet

Esta função deveria fazer a mistura das samples para criar músicas.

Não se encontra funcional.

2.2.4 vote

Função onde recebendo o id da música e os pontos(-1 ou 1) adiciona um voto negativo ou positivo na música. A função averigua se o utilizador já votou ou não. Caso já tenha votado dá para tirar o voto clicando no botão contrário ao usado para votar e poderá votar denovo. A função adiciona os votos diretamente na base de dados.

2.3 Persistência

Este projeto contém uma base de dados que recorre à utilização de três tabelas, todas as tabelas são alteradas consoante as funções de Python existentes no resto do projeto. Tabelas estas que vão de seguida enviar os dados para as páginas HTML através de JavaScript. A tabela das samples serve para guardar todos os excertos que colocámos disponíveis. Esta tabela tem como componentes:

- Name -> do tipo TEXT
- Date -> do tipo TEXT
- ID -> do tipo TEXT
- Length -> do tipo INTEGER
- Uses -> do tipo INTEGER

A tabela das músicas serve para guardar todas as músicas criadas pelos utilizadores Misturador, esta tabela deverá conter apenas uma amostra exemplar, isto porque o nosso misturador não funcionava pelo que esta tabela não irá receber dados. Esta tabela terá também uma contagem de votos assim como o autor de cada uma das músicas. Esta tabela tem como componentes:

- Name -> do tipo TEXT
- Date -> do tipo TEXT
- ID -> do tipo TEXT
- Length -> do tipo INTEGER
- Uses -> do tipo INTEGER
- Votes -> do tipo INTEGER
- Author -> do tipo TEXT

A tabela dos votos irá ter a função de guardar os votos que são inseridos em cada uma das músicas, cada utilizador apenas pode ter um voto numa música(pode votar 1, -1 ou deixar 0 se se quiser manter imparcial). É permitido ao utilizador alterar o seu voto. Esta tabela tem como componentes:

- Email -> TEXT
- sID -> TEXT
- Points -> INTEGER

2.4 Gerador de Músicas

Como já foi referido Seção 2.1 o Misturador de Músicas não está totalmente funcional. A seleção dos excertos, dos efeitos, do volume e dos batimentos por minutos está operacional e é criado um ficheiro JSON com toda essa informação. O ficheiro JSON é criado no ficheiro *get_sheet.js* através da função *generate_json()*. A informação que entra no ficheiro JSON é conseguida através das funções *get_samples()* (excertos), *get_effects()* (efeitos) e *get_vol()* (volume).

O aspeto do misturador é o seguinte:

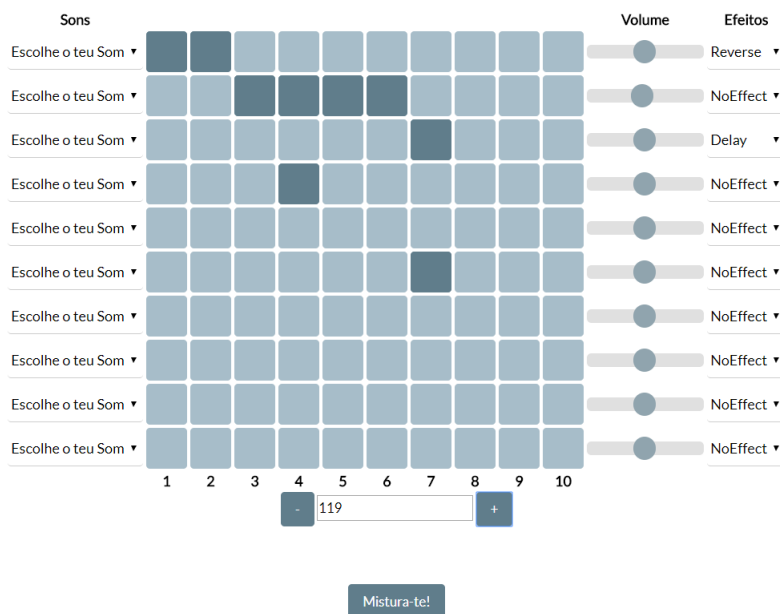


Figura 2.1: Misturador de músicas

O que falta para o Misturador de Músicas estar a funcionar é passar a informação do ficheiro JSON para python e usar a função *mix()* (esta já criada no ficheiro *mixer.py*).

Capítulo 3

Resultados

3.1 Demonstração do funcionamento dos Votos

| Músicas | Reprodução | Votos |
|---------|--|--|
| teste |  0:00 <input type="text"/> |  -1  |

Figura 3.1: Resultado dos votos para: -1

| Músicas | Reprodução | Votos |
|---------|---|---|
| teste |  0:00 <input type="text"/> |  0  |

Figura 3.2: Resultado dos votos para: 0




| Músicas | Reprodução | Votos |
|---------|---|---|
| teste |  0:00 <input type="text"/> |  1  |

Figura 3.3: Resultado dos votos para: 1

3.2 Demonstração do funcionamento da página Excertos

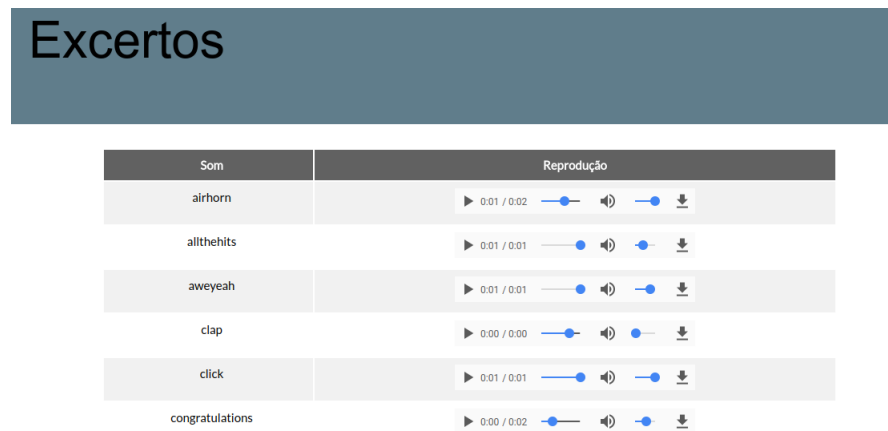


Figura 3.4: Funcionamento de todos os componentes da página excertos

3.3 Demonstração do funcionamento da Base de Dados

```
sqlite> .open database.db
sqlite> .tables
samples  songs      votos
sqlite> SELECT * FROM samples;
airhorn|00:07 09/06/2018|e2d6c924e2c3cf84|0|0
allthehits|00:07 09/06/2018|5e8e127dddcf669d|0|0
aweyeah|00:07 09/06/2018|b3d97b42e2457cd5|0|0
clap|00:07 09/06/2018|d9f661bb40cfacd8|0|0
click|00:07 09/06/2018|35a938b772c25858|0|0
congratulations|00:07 09/06/2018|ea610325ef744e01|0|0
cops|00:07 09/06/2018|e73d135ca7700b6a|0|0
damnson|00:07 09/06/2018|0212513e0c7fabcd|0|0
di|00:07 09/06/2018|0ddd313eb970e08b|0|0
dino|00:07 09/06/2018|bc72c814efbdd8fb|0|0
```

Figura 3.5: Funcionamento da tabela samples

```
sqlite> .open database.db
sqlite> .tables
samples  songs      votos
sqlite> SELECT * FROM songs;
teste|0123456789ABCDEF|2|00:07 14/06/2018|0|1|autor1
```

Figura 3.6: Funcionamento da tabela songs

```
sqlite> .open database.db
sqlite> .tables
samples  songs      votos
sqlite> SELECT * FROM votos;
andr.alves|0123456789ABCDEF|1
sqlite> █
```

Figura 3.7: Funcionamento da tabela votos

Capítulo 4

Conclusões Finais

Com este projeto de gerador de músicas foi possível consolidar os conhecimentos adquiridos em aulas pois foi necessário tudo o que foi realizado ao longo semestre.

Infelizmente, o projeto não está completamente funcional. Ainda assim, com o que foi feito, mudou a forma como os elementos do grupo veem as aplicações como esta de gerador de músicas, todo o trabalho que está por detrás do que é apresentado ao utilizador.

No futuro o que o grupo pode melhorar é a gestão de tempo pois foi isso que impossibilitou a conclusão do projeto na íntegra.

Capítulo 5

Contribuições dos autores

André Alves - 25%
Daniel Correia - 25%
Pedro Almeida - 25%
Pedro Valente - 25%