

Projeto Final - Gerador de Músicas

Universidade de Aveiro

Alexey Kononov, Ricardo Ermida, Rodrigo
Santos, Rui Santos



Projeto Final - Gerador de Músicas

DETI - Universidade de Aveiro

Universidade de Aveiro

Alexey Kononov, Ricardo Ermida, Rodrigo Santos, Rui Santos
(89187) ricardoermida@ua.pt, (89180) rodrigo.l.silva.santos@ua.pt, (89293) ruipedro99@ua.pt

15 de Junho de 2018

Contribuições dos autores

Em nome de Rui Santos são dados os agradecimentos a:

António Domingues, com quem houve uma colaboração contínua na execução da Persistência.

Professores João Paulo Barraca, José Duarte e Vitor Cunha pela disponibilidade em dar sessões de esclarecimentos de dúvidas.

Resumo

O primeiro passo para o funcionamento da aplicação é a através do qual é efetuada a conexão ao servidor `xcoa.av.it.pt`. Para fazer a interligação dos diversos elementos do projeto é necessário desenvolver uma Aplicação Web em Python que gere todos os componentes. Estes componentes são de diversos tipos como Hypertext Markup Language (HTML), JavaScript, CSS, WAV, etc. Os ficheiros HTML são usados para fazer a Interface Web do nosso projeto, enquanto que os JavaScripts são utilizados para efetuar diversas funções como guardar as músicas geradas no nosso projeto, os ficheiros css, são os estilos que usamos para a aparência dos nossas páginas HTML.

Conteúdo

1	Introdução	1
2	Descrição	2
2.1	Interface Web	2
2.2	Aplicação Web	2
2.3	Persistência	2
3	Conclusões	4

Capítulo 1

Introdução

O tema deste relatório será o código desenvolvido para a criação de uma aplicação web para gerar músicas através de alguns samples.

Será feito uma descrição e discussão acerca dos aspectos mais importantes do funcionamento da aplicação.

Este relatório está dividido em 3 capítulos.

Depois desta introdução, no Capítulo 2 é feita a descrição do funcionamento da aplicação, este capítulo encontra-se dividido em 3 subcapítulos no primeiro é descrita a Interface Web do nosso projeto, no segundo é descrito o funcionamento da Aplicação Web, no terceiro subcapítulo abordamos o funcionamento da base de dados. Por fim, no Capítulo 3 são apresentadas as conclusões retiradas após terminado o projeto.

Capítulo 2

Descrição

O nosso projeto é constituído por uma aplicação web, uma base de dados, elementos CSS e JavaScript. Para as páginas html do nosso site utilizamos um template já construído, o link e autores do template encontram-se na bibliografia.

2.1 Interface Web

Esta componente tem três páginas HTML. Na primeira encontram-se informação acerca dos autores do projeto, na segunda é possível ouvir os samples disponíveis para a criação de músicas e na terceira é apresentada uma tabela que funciona como uma espécie de pauta em que é possível escolher os samples desejados para um determinado instante de tempo, e no fim da construção da música é possível guardar a mesma através de elementos JavaScript.

2.2 Aplicação Web

Este componente constitui uma aplicação em Python que serve conteúdos estáticos como páginas HTML e ficheiros JavaScript e faz a interligação entre todos os componentes do projeto como a base de dados e a reprodução das samples utilizadas.

2.3 Persistência

Este componente é composto por métodos que permitem o registo de informação numa base de dados relacional e a obtenção de informação da mesma. Os métodos são utilizados pela Aplicação Web para registar músicas, excertos e votos. É assim utilizada uma base de dados SQLite3 que contém três tabelas.

Samples. A tabela samples é onde é armazenada a informação dos excertos e contém as colunas:

"ids- Integer que contém os índices dos excertos e existe uma auto-incrementação
"name- Text em que são enunciados os nomes dos samples "length- Integer onde se encontram os segundos de duração do sample "date- Text que se refere à data

do excerto "author- Text, autor do excerto "bpm- Integer, batimentos por minuto "uses- Integer que indica o número de vezes que este excerto foi usado nas músicas.

Relativamente a esta tabela são feitas 4 funções:

"new samples- recebe um dicionário como argumento com os parâmetros name, length, date, author e bpm para que seja registado um novo excerto na tabela samples com as características recebidas "delete sample- recebe um nome como argumento e apaga toda a informação do excerto com este nome da tabela samples "get sample- recebe um nome como argumento e envia uma string com as informações do excerto com o nome recebido "add use sample- recebe um nome de um excerto e o número de usos deste para que sejam adicionados aos usos atuais.

Musics. Na tabela musics é onde fica armazenada a informação das musicas criados pelos utilizadores e contem as seguintes colunas: "id m- Integer que contém os índices das músicas e existe uma auto-incrementação "name- Text em que são enunciados os nomes das músicas "length- Integer onde se encontram os segundos de duração da música "date- Text que se refere à data da música "author- Text, autor da música "bpm- Integer, batimentos por minuto "votes- Integer que indica os votos desta música por parte dos utilizadores

Foram feitas 4 funções relativas à tabela musics:

"new music- recebe um dicionário com os parâmetros name, length, date, author e bpm para que seja registado uma nova música na tabela musics com as características recebidas e vai também adicionar uma nova coluna com o nome name que corresponde a uma nova música em que os utilizadores podem votar "delete music- recebe um nome como argumento e apaga toda a informação da música com este nome da tabela musics e vais também apagar a coluna na tabela users votes correspondente a esta música "get music- recebe um nome como argumento e envia uma string com as informações do excerto com o nome recebido "add vote music- recebe o nome da musica, nome do utilizador e voto deste nesta música para que sejam atualizados os votos desta música e vai também adicionar na tabela users votes o respetivo voto

users votes A tabela users votes guarda o voto que cada utilizador deu a uma determinada música. Contém as colunas seguintes: "id u- Integer que contém os índices dos utilizadores e existe uma auto-incrementação "name- Text, nome do utilizador

As seguintes funções são também relativas a esta tabela:

"add user- recebe um nome e adiciona um novo utilizador na tabela users votes "get vote- recebe como argumentos o nome do utilizador e o nome da musica e devolve o voto que o utilizador em questão deu à música indicada.

Capítulo 3

Conclusões

Durante a elaboração do código necessário para o funcionamento do site, deparamo-nos com algumas dificuldades, tais como o guardar da informação acerca das músicas na base de dados, a interligação entre os elementos integrantes.

Com a realização deste projeto aprofundamos as nossas competências em programação na linguagem Python, bases de dados, JavaScript, CSS, aplicando esses conhecimentos numa situação real. Este projeto final foi uma boa maneira para nos preparar para futuros projetos que possamos vir a estar envolvidos nos quais sejam necessários a utilização de Python, Web Apps, etc.

Contribuições dos autores

No desenvolvimento deste trabalho de aprofundamento, todos os membros do grupo demonstraram o empenho necessário para a sua conclusão.

As páginas html e a aplicação web foram desenvolvidas por Rodrigo Santos, Ricardo Ermida, Alexey Kononov, a base de dados foi elaborada por Rui Santos que também ajudou na organização do trabalho, o relatório foi desenvolvido por todos os membros do grupo.

Assim o trabalho desenvolvido pelo Rodrigo Santos, Ricardo Ermida, Alexey Kononov, Rui Santos, foi de 25% cada um.

Bibliografia

Template utilizado:

Template Name: Lalapeden Author: <a href="http://www.os-templates.com/»OS
Templates Author URI: http://www.os-templates.com/

Acrónimos

HTML Hypertext Markup Language