



Projeto 2

Objetivos:

- Planeamento e preparação do projeto 2.

Este guião apresenta as regras do segundo projeto da disciplina de Laboratórios de Informática.

1.1 Regras

O trabalho deve ser realizado por um grupo de 4 alunos e entregue, via a plataforma <http://elearning.ua.pt>, dentro do prazo lá indicado. A entrega deverá ser feita por **apenas um dos membros** do grupo e deve consistir de **um único ficheiro** PDF com o código e relatório final.

Na elaboração do relatório recomenda-se a adopção do estilo e estrutura de relatório descrito nas aulas teórico-práticas e a utilização de recursos de escrita como: referências a fontes externas, referências a figuras e tabelas, tabela de conteúdo, resumo, conclusões, etc.

O objectivo do relatório é descrever a estrutura da solução, a sua implementação, apresentar provas que comprovem o seu funcionamento correto (p.ex, capturas de ecrã) e analisar os resultados obtidos. Recomenda-se vivamente a utilização de uma estrutura baseada nos objetivos do relatório.

É obrigatório incluir uma secção “Contribuições dos autores” onde se descrevem resumidamente as contribuições de cada elemento do grupo e se avalia a percentagem de trabalho de cada um. Esta auto-avaliação poderá afetar a ponderação da nota a atribuir a cada elemento.

É obrigatório identificar claramente os autores, indicando também o seu número mecano-gráfico.

É obrigatório identificar claramente a área utilizada no servidor XCOA e o projeto na plataforma Code.UA.

1.2 Avaliação

A avaliação irá incidir sobre:

1. **cumprimento dos objetivos através das funções implementadas,**
2. **qualidade do código produzido e dos comentários,**
3. **testes realizados,**
4. **estrutura e conteúdo do relatório,**
5. **utilização das funcionalidades de tarefas do code.ua e git,**
6. **apresentação e discussão do trabalho**

Relatórios meramente descritivos sem qualquer descrição da aplicação, apresentação dos resultados obtidos, testes efetuados, ou discussão serão fracamente avaliados.

Só serão avaliados trabalhos enviados via a plataforma <http://elearning.ua.pt>. Ficheiros corrompidos ou inválidos não serão avaliados à posteriori e não será permitido o reenvio.

Deve ser utilizado um projeto na plataforma Code.UA com um identificador no formato `labi2021-p2-gN`, com o número N que lhe foi atribuído na folha de inscrição do grupo.

1.3 Tema Proposto

O objetivo do projeto é a criação de um sistema que permita criar músicas através da composição de pedaços de música. Existem vários exemplos online deste género tal como o <https://drumbit.app/>. O projeto deve ficar a funcionar no servidor XCOA.

O sistema proposto tem de ser composto pelos seguintes componentes:

- **Interface Web**
- **Aplicação Web**
- **Persistência**
- **Gerador de Músicas**
- **Bónus: Gerador de Imagens**

1.3.1 Interface Web

Este componente deve ser composto por um mínimo de 3 páginas (funcionalidades), desenvolvidas através de HyperText Markup Language (HTML)[1], JavaScript (JS)[2] e Cascading Style Sheets (CSS)[3], fornecendo o único interface para interação com o sistema. Desde que as funcionalidades sejam mantidas, os alunos poderão adicionar outras páginas.

A primeira página irá listar todas as músicas presentes no sistema, ordenadas por ordem ascendente do nome, permitindo a sua reprodução. As músicas em si estarão armazenadas no sistema de ficheiros da aplicação Web, sendo que são pedidas pelo browser quando se invoca a sua reprodução (ver a marca `audio` da linguagem HTML). Assume-se assim que esta página terá uma lista de músicas e botões de controlo da reprodução. Esta página permitirá igualmente que se vote na música (para cima ou para baixo), apresentando o número de votos e os utilizadores que votaram.

A segunda página irá listar todos os excertos de sons disponíveis para a construção de músicas. Deverá ser possível a sua reprodução e a adição de novos excertos, realizando-se um envio de um ficheiro WAVEform audio file format (WAVE), identificado por um nome, para a aplicação Web.

A terceira página permitirá a composição de músicas através da combinação dos excertos existentes, potencialmente também com outras músicas. Esta página deverá permitir selecionar um conjunto de excertos e definir quando é que estes excertos são reproduzidos e qual o efeito a aplicar (se algum). É deixado ao critério do grupo qual o esquema do interface, assim como os seus detalhes de implementação.

A título de exemplo, é sugerido que o interface seja baseado numa matriz de elementos (checkboxes), que representam os instantes temporais e o efeito a tocar nesse instante. Campos de seleção permitirão selecionar o excerto a reproduzir e o efeito a aplicar. Poderá ser igualmente possível prever a música, reproduzindo os excertos nos instantes definidos. Um exemplo é apresentado na Figura 1.1.

O intervalo de tempo entre posições é algo definido para toda a música e medido em Batidas por Minuto. Por definição, uma música a 60 Batidas por Minuto permite reproduzir um excerto a cada segundo. Como referência, considere que a música *Wake Me Up* de Avicii possui batidas a uma taxa aproximada de 124BPM, *Hello* de Adele possui 158 BPM, e *We Will Rock You* de Queen possui 81 BPM. Também é comum existirem repetições de alguns elementos de ritmo a cada 4 intervalos, mas isto varia muito entre músicas. Um excerto pode durar mais do que um segundo, podendo assim existir sobreposição de excertos.

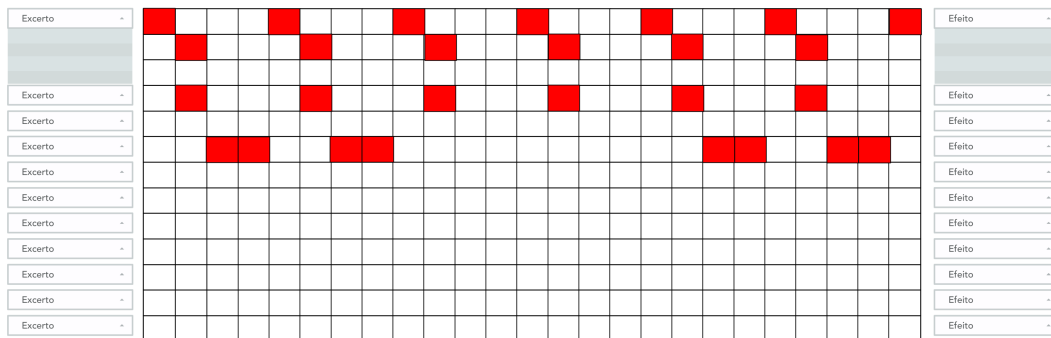


Figura 1.1: Exemplo do interface de definição da música.

No exemplo apresentado na Figura 1.1, para se criar a música é necessário iterar sobre todos os elementos (células) e definir se o excerto toca ou não no instante temporal definido. O resultado é uma espécie de pauta, definindo que efeitos são usados e em que instantes são reproduzidos. Esta pauta deverá ser enviada para a aplicação Web para que a música seja gerada.

De seguida apresenta-se um exemplo de uma pauta em que é indicado o tempo (**bpm**), os excertos utilizados **samples** e os tempos de reprodução de cada excerto (**music**). Neste caso, a música é composta por uma lista de instantes e cada instante é novamente uma lista de excertos. O excerto de nome **sample1** será tocado a cada 4 tempos, sendo que alguns instantes não possuem qualquer excerto, ou podem mesmo existir dois ([2, 4]). Existe igualmente uma lista de efeitos (**effects**) para serem aplicados a cada um dos instantes. Neste caso, no primeiro instante e no último instante (11).

```
{
  "bpm": 61,
  "samples": ["sample1", "sample2", "sample3", "sample..n"],
  "effects": {
    0: "fadein", 11: "fadeout"
  },
  "music": [
    [0], [], [], [0], [1], [], [0], [], [2], [0], [], [2, 4]
  ]
}
```

1.3.2 Aplicação Web

A aplicação web consiste num programa **python** que serve conteúdos estáticos (**html**, **css**, **js**). Apresenta métodos que permitem o fluxo de informação entre os diversos componentes.

Em particular, esta aplicação deverá expôr as seguintes funções (podendo expôr mais):

- **/list?type=songs**: Quando invocado, devolve um objeto JavaScript Object Notation (JSON)[4] com a lista de todas as músicas já criadas. Cada música deverá conter a data de criação, o seu nome, o identificador da música, a sua duração em segundos e os votos. O identificador deverá ser um valor único criado com base no conteúdo (uma síntese);

```
[
  {
    "name": "Loop0",
    "id": "a28d4b24452hfE357",
    "length": 101,
    "date": "2021-05-01 16:00:00",
    "votes": -1,
  },
  {
    "name": "Loop1",
    ....
  }
]
```

- **/list?type=samples**: Quando invocado, devolve um objeto JSON com a lista de todas os excertos existentes. Cada excerto deverá conter a data de criação, o seu nome, o identificador e a sua duração em segundos;
- **/get?id=identificador**: Permite obter um excerto ou uma música com base num identificador fornecido. Este identificador deve ser uma síntese do conteúdo;
- **/put**: Permite enviar a pauta de uma nova música para que esta seja criada. Deve ser utilizado o método POST, passando como argumento a pauta e o nome da música;
- **/vote?id=identificador&points=1**: Permite a um utilizador a emissão de um voto numa música. O campo **id** identifica a música e o campo **points** especifica o número de pontos a atribuir (+1 ou -1).

1.3.3 Persistência

Este componente deverá ser composto por métodos que permitem o registo de informação numa base de dados relacional e a obtenção de informação da mesma. Os métodos serão utilizados pela Aplicação Web para registar músicas, excertos e votos.

Deverá ser utilizada uma base de dados **SQLite3**, localizada no mesmo diretório da aplicação.

Os ficheiros contendo as músicas deverão ser armazenados no sistema de ficheiros, sendo que a base de dados relacional apenas terá informação sobre as músicas.

1.3.4 Gerador de Músicas

Este componente 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.

1.3.5 Bónus: Gerador de Imagens

Este módulo, de carácter opcional, permitindo obter pontos de avaliação adicionais (até +2), tem como função a geração de uma imagem que represente a estrutura da música. Por estrutura entende-se os excertos existentes e os tocados em cada instante temporal. Um exemplo de uma imagem é a zona central apresentada na Figura 1.1. De notar que o módulo deverá ser implementado em **python**, usando a pauta como base de partida para a geração. O resultado deverá ser uma imagem Portable Network Graphics (PNG), para apresentação na página web.

É deixado a cargo dos alunos a definição deste elemento e a sua integração na página web. Considera-se que irá ser necessário alterar os objetos trocados com a aplicação web ou os métodos utilizados.

1.4 Notas Relevantes

1.4.1 Preparação

Teste o acesso ao servidor XCOA com a devida atencendência. É comum existirem questões a resolver e adquirir conhecimento relacionado com a utilização da VPN da Universidade de Aveiro, acesso remoto ao servidor e execução de um script **python** com a porta correta. Teste estes pontos assim que possível. Caso encontre algum problema, contacte os docentes.

1.4.2 Portas

Como estudado, o sistema Linux apenas permite criar um socket em cada porta da mesma família. Não será possível executar múltiplas aplicações, num mesmo servidor, sem que se tomem precauções adequadas. Desta forma, é obrigatório que cada grupo execute a sua aplicação numa porta distinta. Neste caso estarão atribuídas as portas $10000 + n$ em que n é o número do grupo.

Depois será possível aceder ao XCOA de forma a que o servidor redirecione os pedidos HyperText Transfer Protocol (HTTP)[5] para a aplicação `cherrypy` correta.

Como exemplo, no caso do grupo 8, será possível aceder a

`http://xcoa.av.it.pt/labiproj2g8,`

sendo que estes pedidos serão autenticados e reenviados para uma aplicação à escuta na porta 10008.

Um acesso à listagem de músicas deste grupo deverá ser feita acedendo ao URL

`https://xcoa.av.it.pt/labiproj2g8/list?type=songs`

(ou também `/labiproj2g8/list?type=songs` se a página base for servida do mesmo servidor), sendo que o servidor XCOA irá reenviar a informação para uma aplicação no endereço

`http://127.0.0.1:10008/list?type=songs.`

De forma a iniciar uma aplicação numa porta alternativa, o seguinte excerto pode ser utilizado como base:

```
import cherrypy
cherrypy.config.update({'server.socket_port': 10008,})

class HelloWorld(object):
    def index(self):
        return "Hello World!"

    index.exposed = True

cherrypy.quickstart(HelloWorld())
```

1.4.3 Execução remota

De forma a deixar o seu script a executar no servidor após terminar a sessão de SSH, utilize o programa `screen`. Para mais informação, consulte este tutorial sobre o programa `screen`: <https://linuxhint.com/linux-screen-command-tutorial/>.

Glossário

CSS	Cascading Style Sheets
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JS	JavaScript
JSON	JavaScript Object Notation
PDF	Portable Document Format
PDF	Portable Document Format
PNG	Portable Network Graphics
URL	Uniform Resource Locator
WAVE	WAVEform audio file format

Referências

- [1] W3C. (1999). HTML 4.01 Specification, URL: <http://www.w3.org/TR/1999/REC-html401-19991224/>.
- [2] ECMA International, *Standard ECMA-262 – ECMAScript Language Specification*, Padrão, dez. de 1999. URL: <http://www.ecma-international.org/publications/standards/Ecma-262.htm>.
- [3] W3C. (2001). Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification, URL: <http://www.w3.org/TR/2011/REC-CSS2-20110607/>.
- [4] E. T. Bray, *The JavaScript Object Notation (JSON) Data Interchange Format*, RFC 7159, Internet Engineering Task Force, mar. de 2014.
- [5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach e T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, RFC 2616 (Draft Standard), Updated by RFCs 2817, 5785, 6266, Internet Engineering Task Force, jun. de 1999.