

PROJETO 2 - PLANEAMENTO

Universidade de Aveiro

Domingos Nunes, Dzianis Bartashevich,
Francisco Cunha, Leonardo Oliveira



Versão 1.0

PROJETO 2 - PLANEAMENTO

Departamento de Eletrónica, Telecomunicações e
Informática

Universidade de Aveiro

Domingos Nunes, Dzianis Bartashevich, Francisco Cunha, Leonardo Oliveira
dfsn@ua.pt, bartashevich@ua.pt, franciscomiguelcunha@ua.pt, leonardooliveira@ua.pt

17 de Maio de 2015

Conteúdo

1	Introdução	1
2	Páginas Web	2
3	Servidor	6
4	Som	7
5	Árvore da base de dados	8
6	Fases de desenvolvimento	9
6.1	Fase 1	9
6.2	Fase 2	9
6.3	Fase 3	9
7	Considerações finais	10

Lista de Figuras

2.1	Esboço da página "Adicionar música".	3
2.2	Esboço da página "Adicionar interpretação".	4
2.3	Esboço da página "Play song".	5
5.1	Árvore da base de dados, com as tabelas da música (<i>musics</i>) e interpretação (<i>interpretation</i>).	8

Capítulo 1

Introdução

Este documento contém uma abordagem geral das etapas de realização do projeto 2 de Laboratórios de Informática, de acordo com o enunciado disponibilizado na plataforma *moodle* da universidade [1].

Neste sentido, este documento está dividido em 7 capítulos. Depois desta introdução, no Capítulo 2 é apresentada a estrutura das páginas HyperText Markup Language (HTML) necessárias à interação com a aplicação, no Capítulo 3 são referidos alguns aspetos sobre o servidor, no Capítulo 4, são tratados os módulos necessários ao processamento e geração de ficheiros de som e no Capítulo 5 é apresentada a árvore da base de dados a ser criada. Já no Capítulo 6, estão presentes as datas planeadas para o desenvolvimento do projeto, assim como a distribuição de tarefas da fase inicial. Finalmente, no Capítulo 7 são apresentadas algumas considerações finais.

Capítulo 2

Páginas Web

Para este projeto decidimos optar por três páginas web, para tornar a aplicação menos sobrecarregada visualmente e mais interativa do ponto de vista do utilizador.

A página que nos aparece ao iniciar a aplicação chama-se "Adicionar música"(ver Figura 2.1), na qual vamos introduzir o código Ring Tone Transfer Language (RTTTL) da respetiva música na primeira caixa de texto e depois pressionamos o botão em baixo para podermos enviar o código para o servidor.

Na página "Adicionar interpretação"(ver Figura 2.2), como o nome indica, vamos adicionar uma nova interpretação a uma música. Temos um *dropdown* que nos vai colocar em lista as músicas guardadas, uma caixa de texto para adicionarmos um registo e outro *dropdown* para acrescentar um efeito à nossa escolha, como ilustra a figura. Após estas escolhas, temos de adicionar o nome que queremos dar à interpretação e enviar para o servidor.

Finalmente, temos a página "Reproduzir música"(ver Figura 2.3), na qual vamos fazer essencialmente reproduzir música. Para isso temos de escolher a música num *dropdown*, escolher a interpretação e carregar no *play*. Ao lado de cada interpretação temos uma barra de gostos/não gostos para classificar cada música.

No decorrer do projeto vamos adicionar mais alguns botões e *navbars* de modo a facilitar a navegação pela *app*.

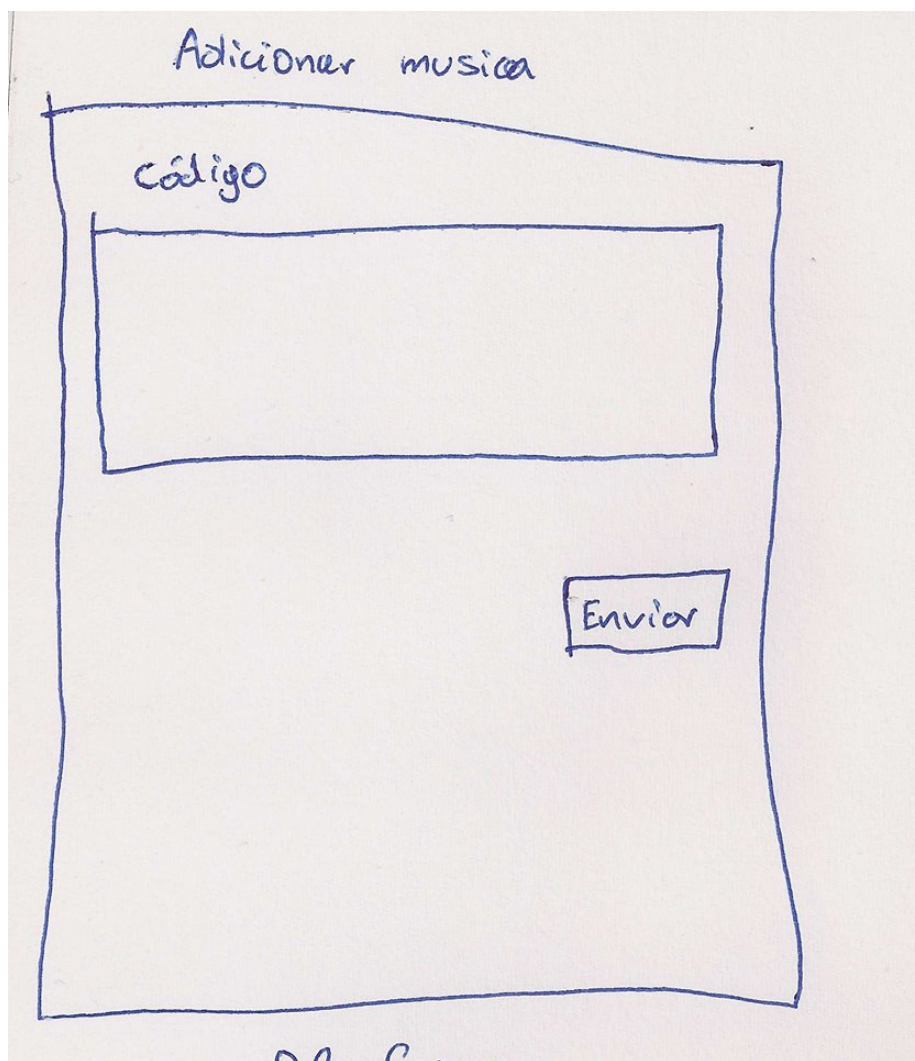


Figura 2.1: Esboço da página "Adicionar música".

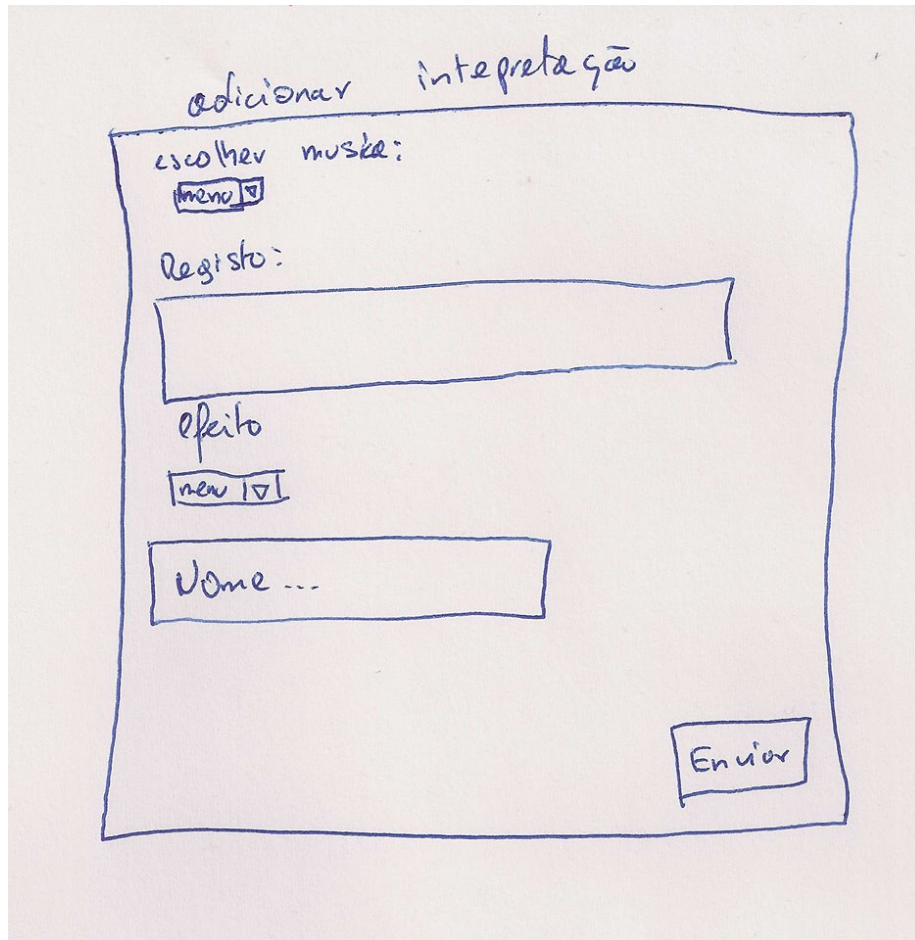


Figura 2.2: Esboço da página "Adicionar interpretação".

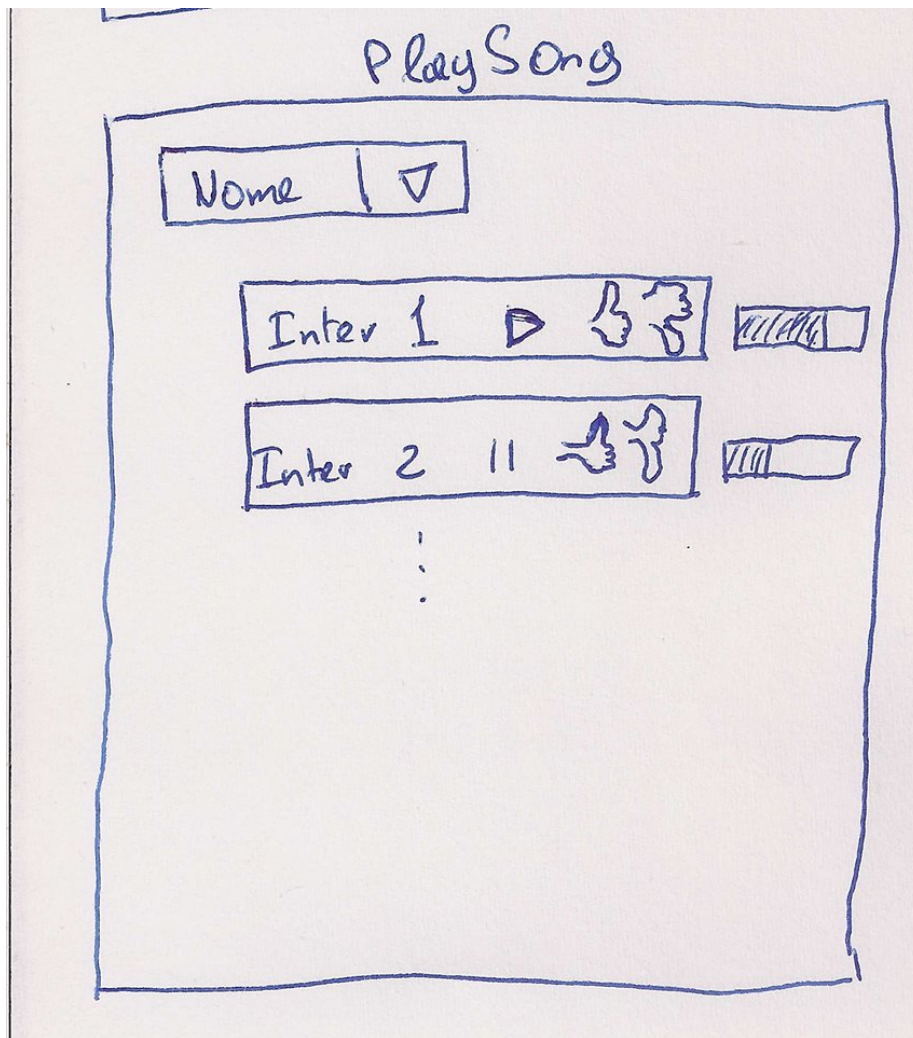


Figura 2.3: Esboço da página "Play song".

Capítulo 3

Servidor

Vai ser utilizado o módulo CherryPy [2] para fazer um servidor HTTP que irá ter funcionalidades como GET e POST. Vai utilizar o JavaScript Object Notation (JSON) como o meio de transporte de informação entre a base de dados e a Aplicação Web.

Os serviços a implementar seguirão o modelo dos colocados no enunciado do projeto.

Estamos a considerar não implementar o */getWaveFile* e */getWaveForm*, visto que, se guardarmos os ficheiros com o nome igual ao seu identificador, a sua procura é facilitada, podendo-se aceder diretamente ao *link* em que se encontram. Por exemplo, para obter o ficheiro de música da interpretação com id 5, acedemos a */interpretations/5.wav*.

Se o utilizador pedir a lista de músicas existentes (*/listSongs*), o servidor acede à base de dados e irá gerar um ficheiro JSON com informação das músicas existentes e a aplicação web irá interpretar esse ficheiro. Um exemplo de ficheiro JSON poderia ser:

```
{
  {
    "id": 1,
    "name": "The Simpsons"
  },
  {
    "id": 2,
    "name": "The Family Guy"
  }
}
```

Capítulo 4

Som

Os módulos ligados ao processamento e geração de ficheiros de som seguirão os moldes do que foi sugerido no enunciado do projeto. Será criado um interpretador de pautas, um sintetizador e um processador de efeitos.

Capítulo 5

Árvore da base de dados

A base de dados associada ao projecto serão duas tabelas **musics**, **interpretation** que estão interligadas entre si e as quais servirão para armazenar informação acerca dos conteúdos. A tabela **musics** conterà informação com o **name** das músicas criadas e as **notes** (pautas) que lhes são correspondentes, sendo que a estes dois campos há um **id** que os identifica.

Por outro lado, a tabela **interpretation** terá campos como **id_music** que estará associado à tabela **musics**. Conterà também campos como **(registration)** (registos) que servirá para a codificação do registo e o campo **effects** (efeitos), servindo para guardar informação sobre e os efeitos utilizados na música. Finalmente, os campos **upvotes** e **downvotes** guardarão respectivamente o número de votos positivos e negativos.

A Figura 5.1 apresenta o esquema da árvore da base de dados.

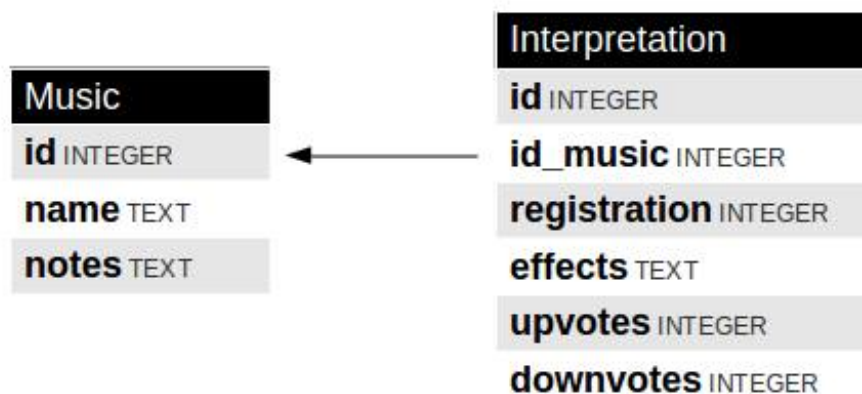


Figura 5.1: Árvore da base de dados, com as tabelas da música (*musics*) e interpretação (*interpretation*).

Capítulo 6

Fases de desenvolvimento

Estão previstas três fases de desenvolvimento do projeto:

- Fase 1 - 13 de Maio a 21 de Maio (a decorrer);
- Fase 2 - 22 de Maio a 29 de Maio;
- Fase 3 - 30 de Maio a 7 de Junho.

6.1 Fase 1

Inicialmente, será desenvolvida cada uma das partes da aplicação em separado e a distribuição das tarefas é a seguinte:

- Domingos - Implementação do *mockup* das páginas HTML;
- Dzianis - Implementação do servido;
- Francisco - Implementação da base de dados;
- Leonardo - Implementação do sintetizador e interpretador de pautas.

6.2 Fase 2

Neste fase a aplicação será testada no seu todo, sendo efetuados os ajustes necessários ao seu correto funcionamento, podendo ser acrescentadas algumas funcionalidades úteis que não tenham sido previstas.

6.3 Fase 3

Finalmente, serão efetuados os ajustes finais, sendo também completado o módulo processador de efeitos, para que este comece efetivamente a criar efeitos nas músicas: Echo, Tremolo, Distorção, Percursão, Chorus e Envelope.

Capítulo 7

Considerações finais

As considerações feitas neste documento estão abertas a alterações, pelo que poderão ser efetuados ajustes nas funcionalidades implementadas ou na distribuição de tarefas caso seja necessário.

Embora não tenha sido aqui referido com pormenor, serão efetuados alguns testes (manuais e automáticos) a cada um dos componentes desenvolvidos e que serão, sempre que possível, da responsabilidade de um elemento que não tenha desenvolvido esse componente, para que seja mais ágil a deteção de erros e para que todos possamos ter uma melhor perseguição do que foi feito pelos restantes colegas.

Acrónimos

HTML HyperText Markup Language

RTTTL Ring Tone Transfer Language

JSON JavaScript Object Notation

Bibliografia

- [1] *Laboratórios de informática*, [Online; acedido em Maio de 2015]. endereço: <http://elearning.ua.pt/course/view.php?id=3470>.
- [2] *Cherrypy*, [Online; acedido em Maio de 2015]. endereço: <http://www.cherrypy.org/>.