



universidade  
de aveiro

---

## Projecto 2 - Aplicações Web

---

Universidade de Aveiro

Rui Machado  
João Vieira  
Inês Moreira  
Pedro Cruzeiro

# Projecto 2 - Aplicações Web

Departamento de Electrónica Telecomunicações e Informática

Universidade de Aveiro

Rui Machado, N<sup>o</sup> Mec.: 65081

ruimmachado@ua.pt

João Vieira, N<sup>o</sup> Mec.: 50458

joaopvieira@ua.pt

Inês Moreira, N<sup>o</sup> Mec.: 100084

inesfm@ua.pt

Pedro Cruzeiro, N<sup>o</sup> Mec.: 103173

pedrocruzeiro@ua.pt

## Resumo

O presente relatório aborda o desenvolvimento de um sistema que permita a criação e reprodução de músicas e sons fazendo recurso ficheiros de audio, usando conceitos e temas abordados no decorrer da Unidade Curricular de Laboratórios de Informática da Universidade de Aveiro, nomeadamente servidores aplicativos web, bases de dados, transformação e transmissão de informação sonora e aplicações web para serviços móveis (android/ios...). A aprendizagem destes conceitos anteriormente referidos torna-se imperativa para os estudantes de cursos superiores ligados às tecnologias de informação, já que é evidente o papel fundamental que estas desempenham no desenvolvimento moderno de tecnologias web.

O projecto foi realizado por uma equipa de quatro pessoas, tendo como objectivo o desenvolvimento das capacidades de trabalho em grupo, desenvolvimento e estruturação de código, realização de testes funcionais e planeamento através da plataforma **code.ua.pt**.

Foi possível desenvolver a implementação uma interface web para comunicação entre o utilizador e os ficheiros servidos para interação, uma aplicação web (API) para servir e comunicar diversos ficheiros e informações entre todos os elementos programáticos, uma base de dados desenvolvida em SQL para armazenar informação sobre diversos ficheiros de audio, respectivos autores, sistemas de votação por utilizadores e outras informações relevantes e a implementação de um gerador/compositor virtual de música. Foi possível corrigir todos os erros encontrados através de testes executados por todos os elementos e o projecto foi terminado pela equipa de forma satisfatória.

Serve ainda o presente relatório como base de documentação para o projecto.

# Índice

0.1	Funcionamento e Objectivo . . . . .	1
0.2	Metodologia e Estrutura . . . . .	1
<b>1</b>	<b>Interface Web</b>	<b>2</b>
<b>2</b>	<b>Aplicação Web</b>	<b>3</b>
2.1	O que usamos . . . . .	3
<b>3</b>	<b>Persistência</b>	<b>4</b>
<b>4</b>	<b>Gerador de Músicas</b>	<b>5</b>
<b>5</b>	<b>Testes e Resultados</b>	<b>6</b>
<b>6</b>	<b>Conclusão:</b>	<b>7</b>

# Lista de Figuras

# Acrónimos

**UA** Universidade de Aveiro

**UC** Unidade Curricular

**LABI** Laboratórios de Informática

**HTTP** HyperText Transfer Protocol

**SQL** Structured Query Language

**HTML** HyperText Markup Language

**JS** JavaScript

**CSS** Cascading Style Sheets

**JSON** JavaScript Object Notation

**API** Interface de programação de aplicações

**Web** World Wide Web

# Introdução

Devido ao elevado avanço tecnológico da última década e sendo os serviços de aplicações web uma tecnologia essencial para o funcionamento actual da sociedade, torna-se imperativo o estudo aprofundado sobre o desenvolvimento de websites, Interface de programação de aplicações (API), bases de dados e salvaguarda e transmissão segura de diferentes tipos de informação. Assim, no âmbito da Unidade Curricular (UC) de Laboratórios de In-

formática (LABI) da Universidade de Aveiro (UA), vem o presente relatório descrever, demonstrar e analisar a elaboração do Projecto Final 2, cujo tema consiste no desenvolvimento de um sistema que permita a criação e reprodução de músicas e sons fazendo recurso ficheiros de audio, usando as tecnologias anteriormente referidas.

## 0.1 Funcionamento e Objectivo

O projecto consiste no desenvolvimento de uma página World Wide Web (Web) a qual será servida de diversos ficheiros de programação por uma API desenvolvida num framework **CherryPy** fazendo recurso de uma linguagem de transmissão entre os mesmos, sendo neste caso a linguagem JavaScript Object Notation (JSON). A aplicação do sistema em causa deve permitir a composição, reprodução e transmissão (upload/download) de musicas, sons e excertos dos mesmos, por forma a serem reutilizáveis por um gerador musical. As informações dos diversos ficheiros audio, devem ser registadas numa base de dados desenvolvida em Structured Query Language (SQL), a qual deve permitir o acesso e alteração dos dados referidos.

## 0.2 Metodologia e Estrutura

Devido à natureza do trabalho, do envolvimento de uma equipa de quatro elementos e sendo o tempo de desenvolvimento relativamente curto, decidiu-se utilizar uma metodologia **Agile**. Esta metodologia permite tornar uma ideia realidade de forma rápida, sendo que a prioridade foi na foi a divisão do projecto em pequenas partes e identificação/desenvolvimento da parte mais essencial do projecto, obtendo um trabalho funcional de forma rápida, para posteriormente ser melhorado e iterado com novas funcionalidades requeridas. Assim, foi possível à equipa ter reuniões regulares sobre dúvidas ou dificuldades, responder a mudanças de código com facilidade, permitir a aprendizagem através de erros, executar pequenos updates tornando o desenvolvimento flexível ajustando o código e tempo disponível consoante necessário. Em suma, o uso do **Agile** tornou-se a escolha ideal para o desenvolvimento deste projecto já que pequenos bugs (que devem ser corrigidos) não põe em causa o funcionamento do projecto ou em risco qualquer utilizador do mesmo, sendo este metodo muito usado para o desenvolvimento de redes sociais, websites de entretenimento ou até reprodutor de música.

O presente relatório foi dividido em diversos capitulos, sendo que cada capitulo aborda uma parte ou funcionalidade implementada no projecto.

# Capítulo 1

## Interface Web

A Interface Web é o **Frontend** do Projecto desenvolvido, sendo que esta permite a interação do utilizador com a base de dados e ficheiros de audio fornecidos para reprodução, alteração, composição e download.

Esta é composta por vários ficheiros desenvolvidos por diversas tecnicas e linguagens de programação, para criar conteudos estáticos e dinâmicos, tendo sido usadas as seguintes linguagens:

- HyperText Markup Language (HTML)
- Cascading Style Sheets (CSS)
- JavaScript (JS)
- JSON

Estes ficheiros foram distribuidos e organizados em diversas páginas com nomes iguais ao da sua linguagem de programação por forma a serem executados, encontrados ou servidos de forma mais intuitiva e versátil. Assim, HTML foi usada para criar a base da interface, CSS e JS foram usadas para criar conteúdos estaticos e dinâmicos a aplicar ao HTML e a linguagem JSON desempenha uma função de comunicação entre interações na interface web e a aplicação web API e base de dados desenvolvidas.



## Capítulo 2

# Aplicação Web

Utilizamos a aplicação web para, mais concentradamente, ligar o servidor ao cliente através do protocolo HyperText Transfer Protocol (HTTP). A web tem 3 tecnologias:

- Sistema global de identificadores únicos/uniformes: URL e URI;
- Linguagem que representa a informação: HTML;
- Uma comunicação cliente/servidor: HTTP.

No nosso trabalho usamos a linguagem de programação python e para o formato de documentos JSON, como, por exemplo: o generate de músicas, o ratchet e os sons.

### 2.1 O que usamos

#### HTML

O HTML é usado para criarmos o nosso site. Começamos por procurar um template e modifica-lo ao nosso gosto, de seguida fomos desenvolvendo o código HTML nas restantes páginas que fazem parte do site.

Esta linguagem vai representar a informação do nosso site e é usada como conexão o HTTP.

#### CherryPy

É um servidor simples, mas poderoso. Sendo que pode ser usado sozinho ou através de um servidor web. No caso específico (o nosso trabalho) foi utilizado sozinho.

Usámos o CherryPy para conectarmos a acesso às páginas index, sounds, musics, create e about. Estas páginas vão ser abertas, o seu conteúdo vai ser lido e os dados lidos retornados. Também foi usado por nós um método mais simples que define regras para servir ficheiros individuais onde indicamos ao CherryPy que todos os seus conteúdos no diretório são estáticos.

## Capítulo 3

# Persistência

Uma base de dados é como um género de repositório onde é possível armazenar e consultar informações e é uma mais valia, porque, por exemplo para os pontos a baixo:

- Grande quantidade de dados;
- Consulta dos dados de forma não sequencial;
- Ocasionalmente acrescentar dados;
- Cruzar dados de várias naturezas.

a base de dados acaba por ser mais eficiente do que um ficheiro normal.

No nosso trabalho utilizamos uma base de dados relacional, que é a que organiza os dados em forma de tabela, sendo esta caracterizada por um nome, um identificador e um tipo de dados de cada atributo. As tabelas por nós formadas estão relacionadas com a parte das músicas, tendo sido criadas 5 tabelas na nossa base de dados: a tabela 'votes', 'musics', 'sounds', 'effects' e 'creation'.

Para criar uma base de dados é necessário executar a ferramenta "sqlite3", que é uma SGBD.

## SQL

SGBD - Sistemas de gestão de bases de dados: são programas que permitem fazer pesquisas e alterações de registos numa base de dados, de forma eficiente.

O SQL Server é um armazenador e recuperador de dados, efetua estas operações de forma super detalhada.

### **Alguns comandos utilizados:**

Para inserir os dados utiliza-se o comando **INSERT**

Para consultar os dados utiliza-se o comando **SELECT**.

Para atualizar os dados utiliza-se o comando **UPDATE**

Para se apagar linhas inteiras utiliza-se o comando **DELETE**

Os comandos SQL são os que codificam as operações sobre a base de dados.

## Capítulo 4

# Gerador de Músicas

O nosso gerador de músicas foi criado através da linguagem pyaudio. Permite que seja possível obter os dados sonoros e reproduzir o som diretamente de uma forma programática. Uma maneira simples de implementar uma tabela de frequências seria através de um dicionário, em que cada símbolo possui uma lista com as frequências utilizadas.

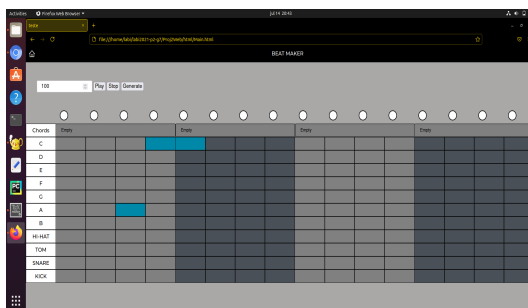
A leitura/construção de ficheiros WAVE é o que gera os tons. Por exemplo, para controlo do volume (sabendo que o som depende da amplitude), no caso do ficheiro WAVE o som é representado pelo valor absoluto de cada impulso.

## Capítulo 5

# Testes e Resultados

Os resultados do nosso site foram satisfatórios, embora tenha havido algumas complicações e dificuldades pelo caminho. O trabalho foi concluído com sucesso e apesar de haver vontade para melhorar e até conseguirmos usar o gerador de imagens, ficámos satisfeitos com o resultado.

O nosso site é facilmente navegável, reproduz as músicas colocadas, e é também possível criarmos os nossos próprios sons e guarda-los, tal como pretendido.



Representação do beat maker do nosso trabalho.

## Capítulo 6

## Conclusão:

A realização deste projeto deu-nos a oportunidade de aprofundar os nossos conhecimentos em python, pyaudio, cherrypy, data bases e um pouco mais em html. O projeto encontra-se no codeUa com o nome labi2021-p2-g7.



O link do nosso projeto, no code ua, pode ser acedido clicando na figura acima.

# Contribuições dos Autores

Tabela 1: Contribuições dos Autores.

Contribuição	Rui Machado	João Vieira	Inês Moreira	Pedro Cruzeiro
Produção de Relatório	25 %	25 %	25 %	25 %
Aplicação Web	25 %	25 %	25 %	25 %
Persistência	25 %	25 %	25 %	25 %
Gerador de Músicas	25 %	25 %	25 %	25 %
Gerador de Imagens	25 %	25 %	25 %	25 %
Testes e Resultados	25 %	25 %	25 %	25 %

# Bibliografia

- [1] Allen Downey *Think Python - How to Think Like a Computer Scientist.* | Green Tea Press, 2nd Edition, Version 2.4.0, 2015
- [2] [www.stackoverflow.com](http://www.stackoverflow.com). | acedido a 29/06/2021