

ESCOLA
SUPERIOR
DE MEDIA
ARTES
E DESIGN
POLITÉCNICO
DO PORTO



PROVA DE AVALIAÇÃO

ATIVIDADE LETIVA

Tecnologias e Sistemas de Informação para a Web						
ANO LETIVO	DATA	HORA	DURAÇÃO			
Programação Orientada a Ol	1º ano					
UNIDADE CURRICULAR			ANO			
Mário Pinto / Ricardo Queiro	Época Normal					
DOCENTE			ÉPOCA			

Observações:

- O exame é individual e deve ser resolvido através da linguagem JavaScript.
- Podem consultar apenas as seguintes referências externas: MDN, W3Schools, Moodle e GitHub (repo da UC e repo pessoal)
- A avaliação do exame poderá implicar convocar os estudantes para uma sessão <u>de avaliação</u>
 <u>oral</u>, onde podem ter de explicar o código entregue
- Serão usadas ferramentas anti-plágio na avaliação dos testes
- Descarregue a pasta NR_ALUNO_EXAME_EX2 do Moodle. Altere o nome da pasta para o seu número de aluno.
- No final do teste submeta as pastas (compactadas) dos dois exercícios no moodle, em objeto próprio de submissão



Grupo I (30%)

Crie uma pasta **NR_ALUNO_EXAME_EX1,** em que NR_ALUNO é o seu número de aluno. Resolva o exercício nessa pasta.

Crie uma página Web com uma tabela de 3 linhas e 3 colunas.

Ao clicar em cada célula da tabela deve gerar um número inteiro e aleatório entre 1 e 9. Deve ser possível clicar mais do que uma vez em cada célula, substituindo o seu valor pelo novo valor gerado.

Quando uma linha somar na totalidade 10 a linha deve ficar com o fundo verde.

Quando toda a tabela ficar verde deve surgir uma caixa de alerta com o texto: "PARABÉNS! VOCÊ GANHOU COM X TENTATIVAS" onde X é o número de cliques efetuados na tabela.

1	5	4
4	4	2
2	5	4

Grupo II (70%)

Descarregue a pasta **NR_ALUNO_EXAME_EX2** do Moodle. Altere NR_ALUNO para o seu número de aluno. Use esta pasta como ponto de partida para resolver o exercício proposto.

Crie um programa de gestão de álbuns musicais, que deve permitir aos utilizadores criar e gerir álbuns assim como as respetivas músicas. O seu programa deve ser constituído por 4 ficheiros:

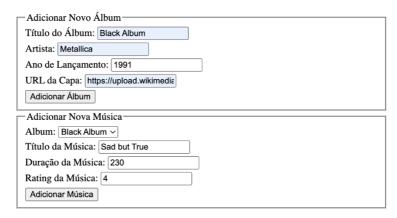
- **index.html** (inclui toda a interface gráfica)
- index.js (gere os eventos do rato e invoca os métodos da classe Album)
- **musica.js** (gere a classe **Musica**)
- album.js (gere a classe Album e respetivas funções)

Requisitos do programa:

1. No ficheiro **index.html** deve conter toda a interface em HTML para interagir com o programa. A interface deve permitir ao utilizador adicionar novos álbuns, adicionar músicas aos álbuns existentes e exibir a lista de Álbuns e de músicas de um álbum selecionado.



Álbuns Músicais



Álbuns



Músicas

	Album:				
ı	Título	Duração	Rating		
ı	Enter Sandman	200	4		
	Sad but True	230	4		

 No ficheiro index.js inclua todo o código necessário para gerir os eventos do rato (cliques nos botões) e respetivas chamadas aos métodos codificados na classe Album.
 Crie um array Albuns, inicialmente vazio, que lhe permita gerir as diversas instâncias da classe Album.

3. Na classe Musica

- a. Inclua as seguintes propriedades a serem inicializadas no construtor da classe:
 - titulo (string): representa o título da música.
 - duração (number): representa a duração da música em segundos.
 - rating (number): representa um rating que deve estar entre 0 a 5, atribuído pelo utilizador.
 - Visualizações (number). Por omissão 0
- b. Crie os métodos "get" e "set" para as propriedades da classe **Musica**. Note que a duração deve situar-se abaixo dos 300 segundos, e o rating não pode ser superior a 5.
- c. Crie os métodos
 - i. incVisualizacoes () método que incrementa o número de visualizações em 1
 - ii. **alteraRating (valor)** método que incrementa o rating de uma música, caso seja inferior a 5. Caso contrário, informa o utilizador que o rating não pode ser incrementado.

4. Na classe Album

- a. Inclua as seguintes propriedades a serem inicializadas no construtor da classe:
 - titulo (string): representa o título do álbum.
 - artista (string): representa o nome do artista ou banda.



- anoLancamento (number): representa o ano de lançamento do álbum.
- capa (string): url para uma imagem que representa a capa do álbum.
- musicas (array): um array vazio, que será preenchido com instâncias da classe **Música**.

b. Crie os métodos:

- i. **criarAlbum**(). Certifique-se de que não existe nenhum álbum com o mesmo título, caso contrário deve enviar uma mensagem de erro;
- ii. **criarMusica():** recebe uma instância da classe Musica e adiciona a música ao array de músicas do álbum (a função deve receber como parâmetros o titulo do álbum e a instancia da classe Musica). Certifique-se que o Album (título) existe.
- iii. **renderizarAlbuns**(): renderiza numa tabela os álbuns existentes, como na imagem acima;
- iv. **verMusicas**(): renderiza numa tabela as músicas de um determinado Album, como na imagem acima.

Notas importantes:

- Deve conectar todos os ficheiros entre si usando as funcionalidades de import/export
- Para todos os métodos criados exiba numa **caixa de alerta** os resultados e as notificações de sucesso e insucesso