

AVALIAÇÃO: Distribuída								
Tipo de		Exame	Tipologia TP					
		Frequência	2° Semestre <u>x</u> ° Trimestre <u> </u>					
Avaliação	Х	Teste	Época Normal	Х	Época Recurso		Época Especial	
		Outro:						

Ano Letivo:	2023/2024
Curso:	Licenciatura em Saúde Digital
Unidade	Paradigmas de Programação em Saúde Digital
Curricular:	
Ano:	1° x 2° □ 3° □ 4° □
Docente:	Mário Pinto
Data:	2o de junho de 2024
Duração:	90 Minutos

#### A preencher pelo aluno

Nome:	Número:	

#### A preencher pelo docente

Classificação:	Assinatura:	
Valores	Assinatura:	

## Instruções

O código escrito por cada estudante será comparado com os restantes através de ferramentas de deteção de plágio. NÃO é permitido o recurso a ferramentas de geração de código baseadas em AI, sob pena de anulação do exame.

Em caso de dúvida da originalidade do código, haverá lugar a uma breve prova oral para o estudante justificar o código que escreveu. Com consequências diretas na avaliação do exame.

- Descarregue do Moodle os ficheiros do objeto **EXAME NORMAL**.
- Clique sobre a pasta (no disco) e no botão direito do rato e escolha *Extrair todos*
- Altere o nome da pasta **EXAME NORMAL** para o seu **número de aluno**.
- No final do teste compacte a pasta com o seu número e submeta-a no Moodle, para avaliação

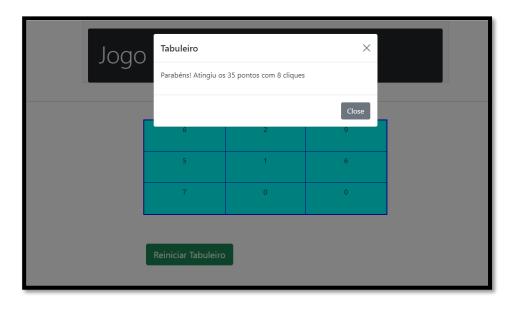


### **EXERCICIO 1**

O código disponibilizado para o exercício 1 cria uma tabela semelhante à abaixo apresentada:



- 1. Sempre que clicar numa célula da tabela, o seu programa deve gerar um número inteiro e aleatório entre 1 e 9, apresentando-o na respetiva célula (podem ser gerados números repetidos).
- 2. Quando todas as células da tabela somarem pelo menos 35 pontos deve:
  - 2.1. Alterar o fundo das células da tabela para a cor cyan
  - 2.2. Abrir uma modal como a abaixo apresentada, onde indica o número de cliques que foram necessários para atingir a pontuação desejada.



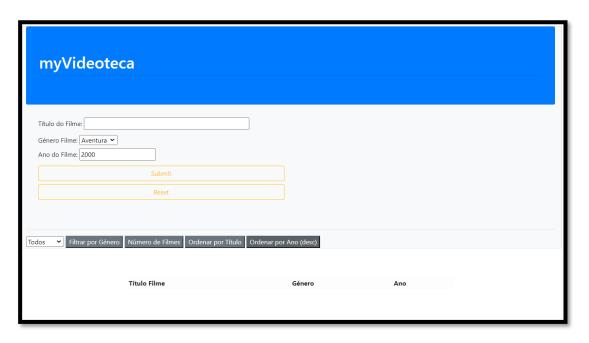
3. O button **Reiniciar Tabuleiro** deve repor o estado da tabela inicial, isto é, todas as células com o valor 0.

ESTSP.004.MO.160.01 Página 2 de 4



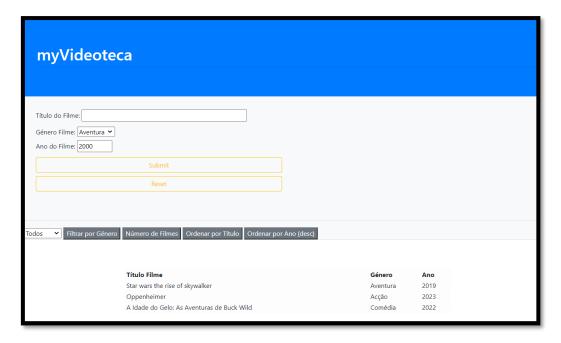
# **EXERCICIO 2**

O código disponibilizado para o exercício 2 cria a *interface* de uma pequena aplicação de gestão de uma videoteca, semelhante à abaixo apresentada:



### Notas:

- O ficheiro init.js já contém um array com 3 objetos (filmes)
- O ficheiro index.js já inclui a importação dos dados do init.js, assim como a declaração do array films com os dados carregados a partir do ficheiro init.js.
- Crie a função *renderTable(films)* de forma a renderizar na tabela todos os filmes do array films:



ESTSP.004.MO.160.01 Página 3 de 4



- 2. Ao submeter um novo filme no formulário, verifique se o filme já existe no array films:
  - 2.1 Caso exista, deve abrir uma alert com a informação de que "O filme já existe".
  - 2.2 Se o filme não existir, invoque a função **addFilm()** a qual deve adicionar o novo filme ao array de objetos films. Atualize a tabela com o novo filme.
- 3. Implemente as funcionalidades associadas aos buttons existentes abaixo do formulário:
  - 3.1 Button Filtrar por Género: deve filtrar e apresentar, na tabela abaixo, os filmes do género selecionado. Note que se selecionar "Todos" deve apresentar na tabela todos os filmes.
  - 3.2 Button Número de Filmes: Deve apresentar numa alert o número de filmes existentes.



- 3.3 Button Ordenar por Título: deve apresentar na tabela os filmes por ordem alfabética (título)
- 3.4 Button Ordenar por Ano (desc): deve apresentar na tabela os filmes por ordem descendente de ano.

Bom trabalho!

ESTSP.004.MO.160.01 Página 4 de 4