



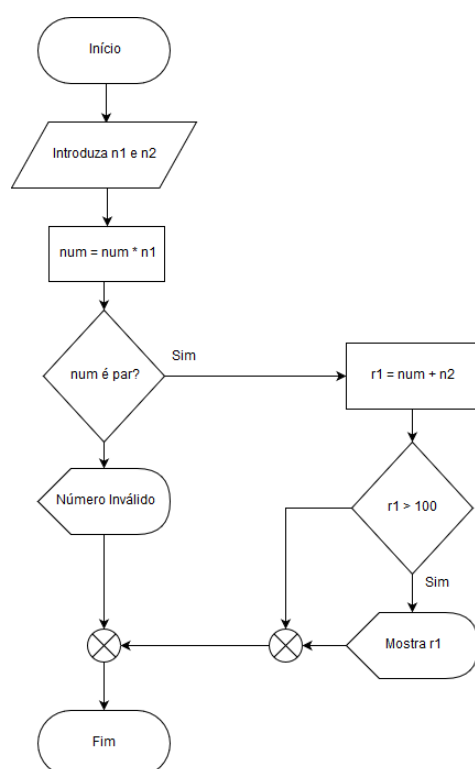
Nome: \_\_\_\_\_

N.º Mec. \_\_\_\_\_

### Exercício 1 (2 valores)

Represente o fluxograma seguinte em Javascript. Para tal, deve pressupor a existência de dois campos de input com os id `numero1` e `numero2`, onde o utilizador introduz os valores, e deve utilizar a consola para mostrar as mensagens.

A execução deste script não necessita de eventos.



## Exercício 2 (3 valores)

Tendo por base o consumo de um veículo, pretende-se verificar se um veículo é híbrido, *plug-in* ou de combustão interna (gasolina/gasóleo).

A partir de um valor introduzido pelo utilizador (litros por 100km), se o consumo for menor que 2 deverá apresentar a mensagem *Veículo Plug-in*; se for entre 2 e 5 deverá apresentar a mensagem: *Veículo híbrido*; se for entre 5 e 7,5 deverá apresentar a mensagem: *Veículo a gasolina*; e se for superior a 7,5 apresentar a mensagem: *Veículo a gasóleo*.

Caso o veículo seja de combustão interna, uma vez que queremos promover melhoria da qualidade do ar, deverá sugerir um *test drive*. Para tal, deverá solicitar ao cliente a marca do veículo anterior e apresentar uma mensagem composta pela marca e pelo modelo do veículo elétrico de referência (consultar tabela abaixo).

Marca do Veículo	Modelo elétrico de referência
Hyundai	Ionic
Opel	Ampera
Nissan	Leaf
Renault	Zoe
Nenhum dos casos...	Bicicleta

Desenvolva o código Javascript necessário para estas operações. Deve pressupor a existência de um campo de input com id `consumo`, onde o utilizador introduz o valor, um input com id `marca` e pode utilizar a consola para mostrar as mensagens.

A sua execução deste script não necessita de eventos.

### Exercício 3 (5 valores)

Considere os seguintes blocos HTML e JS:

#### HTML

1. ``
2. `<h1 id="id1">Parágrafo</h1>`
3. `<input id="textoInput" type="text" />`
4. `<button id="calculo"> Calcular </button>`

#### Javascript

1. `var elemento = document.getElementById("id1");`
2. `elemento.innerHTML = "Novo parágrafo";`
3. `elemento.style.fontSize = "40px";`
4. `elemento.style.color = "red";`
5. `document.getElementById("imagem").src = "boneco.jpg";`
6. `document.getElementById("imagem").style.display = "none";`

1. Descreva em detalhe qual o resultado da execução do código JS.
2. Desenvolva o código Javascript necessário de modo a que:
  - 2.1. Seja alterado o conteúdo do elemento com id `calculo` para *a calcular*;
  - 2.2. Verifique se o valor do elemento com id `textoInput` convertido em número inteiro é um número. Se não for deve ser aplicada a classe `validation_error` ao elemento com id `textoInput`;
  - 2.3. Seja alterada a *source* da imagem com id `pic` para `superpic.png`;
  - 2.4. O elemento com id `id1` fique oculto (mas conservando a sua posição no DOM) e limpe o conteúdo do elemento com id `calculo`.

#### Exercício 4 (4 valores)

Considere o seguinte bloco HTML:

```
1. <div id="inicio">
2.     <button id="entrar">Entrar</button>
3. </div>
4. <div id="principal">
5.     <button id="apagar">Botão Apagar</button>
6.     <button id="calcular">Botão Calcular</button>
7.     <button id="verificar">Botão Verificar</button>
8.     <div id="mensagem">valor gerado</div>
9.     <div id="mensagem_nova"></div>
10. </div>
```

Sabe-se que o código será guardado num ficheiro externo mas não se sabe qual a posição na página HTML onde este será invocado. Desenvolva o código Javascript necessário de modo a serem implementadas as alíneas seguintes.

1. Ao clicar no botão `entrar` esconda a `div inicio` e mostre a `div principal`.
2. Ao clicar no botão `calcular` calcule a soma de dois números aleatórios entre -3 e 3 e mostre o resultado num novo parágrafo na `div mensagem`. O valor já existente deverá ser mantido.
3. Ao clicar no botão `verificar` deverá validar se o valor gerado na alínea anterior é menor do que 5. Caso seja verdade deverá mostrar na `div mensagem_nova` a mensagem: *O valor aleatório é inferior a 5.*

## Exercício 5 (6 valores)

Pretende-se criar uma base para um jogo de sorte *online*, onde são apresentadas ao jogador duas cartas aleatórias. Sendo que, o tempo máximo de jogo é de 10 segundos e o jogador começa com 20 pontos.

Com base no código HTML abaixo, deverá desenvolver o código Javascript necessário para implementar as alíneas seguintes:

```
1. <button id="start">JOGAR!</button>
2. <div>
3.   <img id="left">
4.   <img id="right">
5. </div>
6. <button id="cards">gerar cartas</button>
7. <div>Tempo restante <input type="text" id="tempo" /></div>
```

1. Quando a página HTML estiver carregada deve:
  - 1.1. Definir a imagem dos elementos das cartas, `left` e `right`, respetivamente como `'imgs/back_blue.png'` e `'imgs/back_red.png'`;
  - 1.2. Definir o evento clique para o elemento `start`, o qual deve invocar a função **`inicia_jogo`**;
  - 1.3. Definir o evento clique para o elemento `cards`, o qual deve invocar a função **`gerar_cartas`**;
2. Desenvolver a função **`inicia_jogo`** que deve:
  - 2.1. Esconder o elemento `start`;
  - 2.2. Criar um temporizador que execute a função **`conta_tempo`** a cada 1 segundo;
3. Desenvolver a função **`conta_tempo`** que deve:
  - 3.1. Descontar um segundo ao tempo total de jogo e mostrar o tempo total no elemento `tempo`;
  - 3.2. Quando o tempo chegar a 0, deve terminar o temporizador.
4. Desenvolver a função **`gerar_cartas`** na qual deve:
  - 4.1. Gerar aleatoriamente dois números entre 1 e 10;
  - 4.2. Com recurso aos números gerados mostre as imagens das cartas correspondentes, nos elementos `left` e `right`, sabendo que os nomes dos ficheiros estão numerados de `'imgs/carta1.png'` até `'imgs/carta10.png'`.

