

PROVA 2 – PROGRAMAÇÃO WEB

NOME:

Tendo sempre como base o arquivo HTML abaixo, responda as perguntas a seguir, utilizando apenas sua folha de consulta (**que deve ser entregue juntamente com a prova**).

01	<!doctype html>
02	<html>
03	<head>
04	<title>Sistema ABC</title>
05	<meta charset="utf-8"/>
06	<link href="telas.css" type="text/css" rel="stylesheet"/>
07	<script src="telas.js" type="text/javascript"></script>
08	</head>
09	<body>
10	<form>
11	<label for="buscaCliente">Cliente</label>
12	<input type="text" id="buscaCliente" placeholder="Busque pelo nome"/>
13	</form>
14	<form method="get">
15	<input type="hidden" name="id"/>
16	<label for="clienteInpt">Cliente</label>
17	<input type="text" id="clienteInpt" name="cliente"/>
18	<label for="cpfInpt">CPF</label>
19	<input type="text" id="cpfInpt" name="cpf"/>
20	<label for="saldoInpt">Saldo</label>
21	<input type="number" id="saldoInpt" name="saldo"/>
22	<button type="button">Editar</submit>
23	<button type="button">Remover</submit>
24	</form>
25	<section class="clientes devedores"></section>
26	<section class="clientes inativos"></section>
27	<section class="clientes credores"></section>
28	</body>
29	</html>

Questão 1 (1 ponto)

Preencha a tabela com o número de elementos selecionados por cada comando JavaScript abaixo:

document.querySelector("form[method]")	
document.getElementsByClassName("clientes")	
document.querySelectorAll("input:nth-of-type(odd)")	
document.querySelectorAll("form>input")	

NOME:

ATENÇÃO: As questões a seguir devem ser respondidas como sendo parte do mesmo arquivo de script “telas.js” (invocado na linha 07 do HTML).

Questão 2 (1 ponto)

Escreva uma função chamada **iniciar** que seja invocada ao carregar a página, e adicione as funções de call-back **editar** e **remover** aos seus respectivos botões. Adicione também a função **buscar** como call-back para cada vez que o **buscaCliente** for alterado.

Questão 3 (1 ponto)

Escreva a função **remover** (executada toda vez que o botão **Remover** for clicado) que remove os valores dos campos do formulário de cadastro. Antes de apagar, porém, todos os valores dos campos do formulário de cadastro devem ser copiados para a **section** de **clientes inativos**, onde será adicionado um **article** para cada cliente removido com o texto “[ID] :: [CPF] :: [CLIENTE] :: [SALDO]”.

Questão 4 (1.5 pontos)

Escreva a função **editar** (executada toda vez que o botão **Editar** for clicado) que verifica se o valor do campo **saldo** é positivo, e copia todos os valores dos campos do formulário de cadastro para uma das **section** de **clientes devedores** ou **clientes credores** (de acordo com o saldo verificado). Para cada cliente deverá ser adicionado um **article** com o texto “[ID] :: [CPF] :: [CLIENTE] :: [SALDO]”. Os campos do formulário devem ser limpos em seguida.

Questão 5 (2 pontos)

Escreva a função **buscar**, invocada cada vez que o campo **buscaCliente** é alterado, que crie uma requisição assíncrona, em método GET, para a URL local “**servlets/BuscaClientes**” (utilizando a API XMLHttpRequest ou Fetch), passando como parâmetro **query** o valor preenchido no campo. Como call-back, invoque uma função de **processar** quando a requisição for completada.

Questão 6 (3.5 pontos)

Escreva a função **processar**, que deve parsear o resultado textual (ver **exemplo** abaixo), e preencher o formulário de cadastro com as respectivas informações recebidas (lembrando de calcular o saldo com base nos dados de créditos e débitos).

EXEMPLO:

```
{ "nome" : "João Silva", "nascimento" : "1980-12-10", "CPF" : "99.999.999-99", "id" : 1234, "debitos" : 2000, "creditos" : 45000 }
```

NOME:

NOME: