

Exercício 1

Estrutura de arquivos:

Arquivo Ex01.html

Possui toda a parte estática do exercício, incluindo o link com a CSS do Bootstrap, você deve usar esse arquivo para exibir os dados existentes no arquivo JSON. Leia o arquivo HTML, localize os elementos necessários para receber as informações que serão exibidas na página. Esse arquivo não deverá ter sua estrutura alterada.

Arquivo unidades.js

Possui os dados que montarão o seu front-end. Esse arquivo possui a seguinte estrutura:

nome: String, endereco_logradouro: String, endereco_numero: String, endereco_bairro: String, endereco_municipio_nome: String, endereco_uf_sigla: String, clinica: String, media_consultas_mes: Number

Parte 1:

Você deve montar uma função que leia o arquivo **unidades.js** e monte o front-end, retornando **APENAS** as unidades localizadas na **Cidade de São Paulo**. O seu front deve ficar igual à imagem abaixo, após a execução de seu JS:

Lista de Unidades de Atendimento

Nome	Endereço	Bairro	Cidade / UF	Clínica	Consultas Mês
CC BELEM	R CESARIO ALVIM, 174	BELENZINHO	São Paulo / SP	Cardiologia	645
CC ZONA SUL	AV JOAO DIAS, 76	STO AMARO	São Paulo / SP	Clínica Geral	418
CENTRO MEDICO IPIRANGA	R MAURICIO CASTILHO, 416	VL MONUMENTO	São Paulo / SP	Cardiologia	756
CC ZONA NORTE	AV NOVA CANTAREIRA, 229	TUCURUVI	São Paulo / SP	Clínica Geral	609
CENTRO MEDICO SAO GABRIEL	AV AMADOR B, VEIGA, 195	PENHA	São Paulo / SP	Ortopedia	444
CENTRO CLINICO BARRA FUNDA	AV FRANCISCO MATARAZZO, 612	ÁGUA BRANCA	São Paulo / SP	Clínica Geral	673
GREEN LINE SISTEMA DE SAUDE	AV REBOUCAS, 2011	PINHEIROS	São Paulo / SP	Cardiologia	214
CC TATUAPE	R PLATINA, 215	VL AZEVEDO	São Paulo / SP	Neurologia	333
CC HERMINIO LEMOS	R HERMINIO LEMOS,, 385	CAMBUCCI	São Paulo / SP	Neurologia	312

É importante que você leia ao arquivo **ex01.html** para entender onde as informações deverão ser inseridas. As classes CSS do Bootstrap já estão inseridas nos arquivos, sua única tarefa é com o JS.

Solução composta por dados mocados, será zerada.

Valor do exercício = 3,5

Parte 2:

Logo abaixo da inserção dos dados na tabela, seu arquivo JS deverá calcular e exibir os totais para: número de consultas realizadas na cidade de São Paulo, total de unidades na cidade de São Paulo e o total de unidades em outras cidades. Para isso já temos as estruturas montadas apenas esperando os dados que serão retornados de seu JS. Leia o arquivo HTML e faça a correta exibição dos dados encontrados. Veja como deverá ficar o front-end populado pelo JS:

Número de consultas realizadas:

4404

Total de Unidades - Cidade São Paulo

9

Total de Unidades - Outras Cidades

8

Exercício 2

Estrutura de arquivos:

Arquivo Ex02.html

Possui um formulário e uma estrutura inicial de uma tabela para receber os dados que forem digitados e posteriormente processados pelo seu JS. Não mudar a estrutura do arquivo, seu código deverá se adaptar a ele.

Você deve montar uma função em Javascript que:

- Recupere as informações passadas no formulário;
- Faça o cálculo do valor mensal a ser pago na contratação dos convênios;
- Exiba no front-end, os dados digitados e processados pela sua função.

Abaixo temos a representação de como seu front-end deverá ficar, após os dados serem processados:

Cotação Plano de Saúde

Dados do usuário

Nome

ex: Fulano de Tal

Tipo de Plano

GNDI Smart 400

Número Dependentes

ex: 3

Realizar Cotação >>

Cotações realizadas

Nome	Plano Selecionado	Valor Plano	Num Dependentes	Valor Mensal
Eu mesmo	GNDI Smart 400	R\$ 245,92	7	R\$ 1.106,64
Eu mesmo	GNDI Advance 700	R\$ 379,31	7	R\$ 1.706,90
Eu mesmo	GNDI Premium 900	R\$ 524,33	7	R\$ 2.359,49
Eu mesmo	GNDI Infinity 1000	R\$ 728,12	7	R\$ 3.276,54

Estrutura do formulário:

- Campo do tipo texto para o nome do usuário;
- Campo do tipo select para o tipo de plano selecionado, já possui nome do plano e valores correspondentes;
- Campo do tipo texto para o número de dependentes;
- Botão para gravar os dados na tabela.

Assim que o botão do formulário for pressionado, e antes do formulário ser enviado, sua função deverá:

- Cancelar o evento padrão do formulário;
- Recuperar os valores passados no formulário. Não esquecer que o retorno deles é no formato de **String**;
- Calcular o valor mensal a ser pago para o convênio, cuidado com as Strings;
- Passar os dados para a tabela, gerando uma nova linha para cada simulação executada.

Tipos de planos e valores mensais(dados meramente ilustrativos):

- GNDI Smart 400 - R\$ 245.92
- GNDI Advance 700 - R\$ 379.31
- GNDI Premium 900 - R\$ 524.33
- GNDI Infinity 1000 – R\$ 728.12

Para conseguirmos pegar o nome do plano e o seu valor correspondente precisamos de um array. Para isso, estou disponibilizando o arquivo **planos.txt**. Ele possui um exemplo de um elemento HTML **select**, para três carros e um pequeno JS que recupera as informações selecionadas pelo usuário. O exemplo JS possui:

- Uma constante que recupera o elemento do formulário através de seu id.
- Uma constante que recebe o texto que deve ser exibido no HTML, no meu exemplo o nome do carro.
- Uma constante que recebe o valor atribuído, no meu exemplo o valor do carro.

Perceba que as constantes recebem a propriedade **options[select.selectedIndex]**, junto ao nome do array que está recebendo o elemento select. Cada uma das constantes receberá uma informação que você usará: o texto que indica o nome do plano, e o valor que indica o preço mensal do plano. Altere esse código para que ele funcione no seu exercício.

Para calcular o valor mensal a ser pago, você deve usar a seguinte fórmula:

$$\text{Valor do plano} + ((\text{valor do plano} * \text{número de dependentes})/2)$$

Com os dados recuperados e os valores processados, basta exibir na tabela. Você pode usar Template String ou o método `createElement()`, entendo que com Template String, ficaria mais rápido. Não esqueça de limpar os dados do formulário. Perceba que os valores de moeda possuem o formato correto, use um arquivo para deixar nesse formato. Lembre-se da última aula de JS.

Solução composta por dados mocados, será zerada.

Valor do exercício = 5,0

ATENÇÃO:

Cada arquivo HTML deverá ter no elemento `<title>` o nome e o rm de cada integrante da equipe. Não tendo esses dados, **será descontado 1.0 da nota final de cada arquivo.**

Montar o arquivo `integrantes.txt` contendo o nome e o rm de cada integrante da equipe. Não tendo esse arquivo, **será descontado 2.0 da nota final.**

Você pode consultar o material disponível na área de apostilas do portal.

FORMATO DA ENTREGA:

Arquivo compactado contendo os arquivos originais e os arquivos JS criados por você. O nome do arquivo deve conter o RM de cada integrante da equipe.

Entrega deverá ser feita até **o final da aula** pelo portal da Fiap, área de entrega de trabalhos. Não existe outra forma de envio dos arquivos.

Apenas um integrante da equipe deverá fazer o envio.

Você é o único responsável pela entrega. Caso o arquivo enviado não corresponda à solução de sua prova, a nota será zerada.

Qualquer tipo de tentativa de cola, a nota será zerada.