

**Lista de exercícios - Programação para Internet - 2020**  
**<https://bit.ly/lista-programacao-internet-2020>**

**Distribuição de pontos**

- Listas 01 até 04: 2 pontos cada (08 pontos no subtotal)
- Listas 05 até 08: 3 pontos cada (12 pontos no subtotal)
- Listas 09 até 13: 4 pontos cada (20 pontos no subtotal)
- Trabalho final 01: 30 pontos (aplicativo)
- Trabalho final 02: 30 pontos (vídeo de apresentação)
- TOTAL: 100 pontos

---

**Trabalho final (aplicativo)**

01) Você vai entregar um aplicativo web com os seguintes requisitos mínimos:

- Usar o Node.js;
- Usar o Socket.io;
- Aplicativo responsivo (deve ficar bem apresentável tanto em telas grandes quanto em telas menores (smartphones));
- Você vai fazer um jogo multiplayer conforme especificação da lista 12 porém com as seguintes melhorias e refinamentos:
  - O usuário pode escolher o seu nome (nickname) que aparece para todos os outros jogadores presentes no jogo;
  - O usuário pode escolher a cor do seu personagem que aparece para todos os outros jogadores presentes no jogo;

**Trabalho final (vídeo de apresentação)**

01) Você vai entregar um vídeo apresentando o seu jogo multiplayer de Programação para Internet.

- Mínimo de cinco minutos.
- Máximo de quinze minutos.
- Mostrar seu rosto (como faço nos meus vídeos) é opcional.
- No início do vídeo:
  - Fale seu nome completo e data do dia da gravação.
- Mostre seu jogo funcionando com todas as funcionalidades pedidas.

Lista 12 - Data: 29/06/2020. Data de entrega: 06/07/2020

01) Assista aos vídeos da playlist do Filipe Deschamps sobre como fazer um jogo multiplayer utilizando Javascript + Node.js + Socket.io ([aqui](#)) e faça um jogo similar ao jogo desenvolvido por ele nos vídeos da playlist. Vamos continuar com esse jogo simples e na próxima semana, vamos fazer algo mais complexo.

Lista 11 - Data: 22/06/2020. Data de entrega: 29/06/2020

01) Vamos começar fazendo um chat real-time com Node.js e Socket.io. Nas próximas semanas vamos fazer jogos multiplayer. Use o github para entregar essa lista.




Lista 10 - Data: 18/05/2020. Data de entrega: 29/05/2020

01) Lembra da lista 04 exercício 02 (um mini-site pessoal com sete páginas)? Então, você vai apresentar esse exercício 02 da lista 04 usando **Node.js** com **Express.js**. Você vai postar seu trabalho em um repositório do github, e postar o link do repositório na atividade Lista 10 do Google Classroom.

Lista 09 - Data: 04/05/2020. Data de entrega: 11/05/2020.

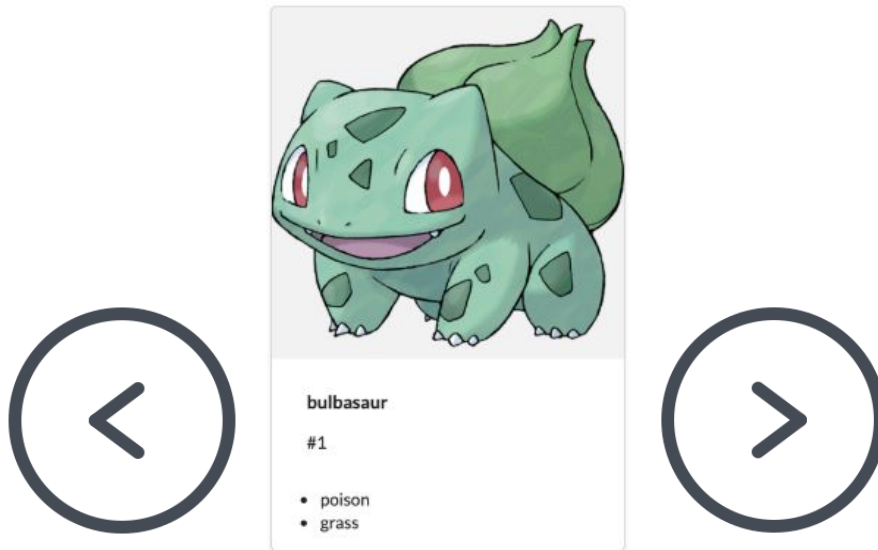
01) Faça um bingo do Pokemon. Pode usar a PokeAPI ou não. Fica a seu critério. Os requisitos do jogo são:

- O jogo deve ter um botão de '*start*' para iniciar o jogo.
- Ao iniciar aparecerá uma cartela (tipo bingo mesmo) com dimensões 4x4 (ou 5x5) com 16 (ou 25) Pokémons aleatórios. Cuidado para não ter Pokémons repetidos na cartela. Imagem 1;
- A cada 5 segundos, será sorteado um Pokémon aleatório e será exibido na tela. Cuidado para não sortear Pokémons que já foram sorteados. Cada Pokémon sorteado, não pode ser sorteado novamente. Imagem 2;
- Caso o Pokémon sorteado, esteja presente na cartela do jogador, o jogador poderá "marcar" em sua cartela o Pokémon sorteado (marcar == clicar em cima e marcar com uma borda colorida, ou um 'x' em cima da imagem, ou qualquer outra forma de destaque); Imagem 3;
- Existe um outro botão chamado "Bingo!". Ao clicar no botão "Bingo!", o jogo confere se todos os Pokémons da cartela estão marcados, se todos os Pokémons marcados realmente foram sorteados. Se essas duas condições forem satisfeitas, o jogo termina com mensagem de sucesso. Se uma ou nenhuma das condições não forem verdadeiras, exibe mensagem de erro, e o jogo não termina.

		
Imagem 1	Imagem 2	Imagem 3

Lista 08 - Data: 27/04/2020. Data de entrega: 04/05/2020

01) Utilize a PokeAPI para fazer um aplicativo que mostre um card de um Pokémon (como abaixo) mostrando informações como nome, foto, número (id), e os tipos deste Pokémon. Coloque duas setas (direita e esquerda) para exibir o Pokémon anterior e o Pokémon posterior.



02) Faça uma Pokédex. Sua Pokédex deve mostrar as seguintes informações de cada Pokémon: foto, nome (primeira letra em maiúscula), stats (ataque, defesa, velocidade de ataque, velocidade de defesa, velocidade, HP, altura, peso). Pode usar sua criatividade para criar uma Pokédex funcional e visual bacana. As imagens a seguir são uma inspiração.

<b>charizard</b> Height: 17 Weight: 905	<b>venusaur</b> Height: 20 Weight: 1000	<b>blastoise</b> Height: 16 Weight: 855	<b>pikachu</b> Height: 4 Weight: 60	<b>slowpoke</b> Height: 12 Weight: 360
				
hp: 78 attack: 84 defense: 78 special-attack: 109 special-defense: 85 speed: 100	hp: 80 attack: 82 defense: 83 special-attack: 100 special-defense: 100 speed: 80	hp: 79 attack: 83 defense: 100 special-attack: 85 special-defense: 105 speed: 78	hp: 35 attack: 55 defense: 40 special-attack: 50 special-defense: 50 speed: 90	hp: 90 attack: 65 defense: 65 special-attack: 40 special-defense: 40 speed: 15

water




Psyduck

52	48	65	50
ATK	DEF	SP-ATK	SP-DEF
55	50	0.8m	19.6kg
SPD	HP	HEIGHT	WEIGHT

Charmander

FIRE

#4



HP	39
Attack	52
Defense	43
Speed	65
Sp Atk	60
Sp Def	50

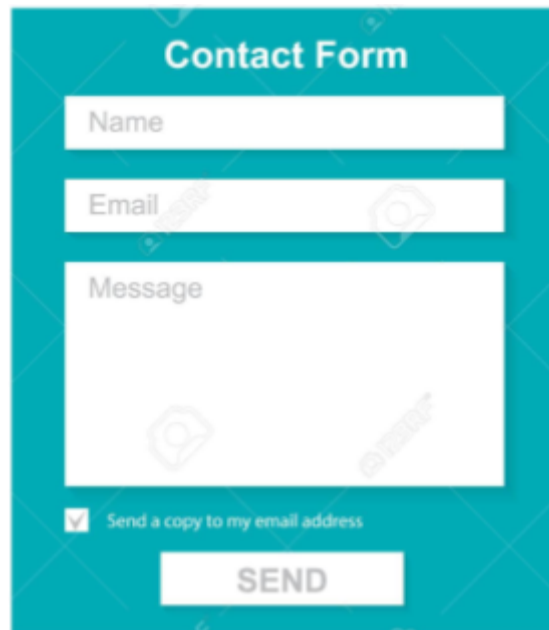
Lizard Pokémon

The fire on the tip of its tail is a measure of its life. If healthy, its tail burns intensely.



Lista 07 - Data: 20/04/2020. Data de entrega: 27/04/2020.

01) Crie um formulário como este da imagem. As informações inseridas no formulário deverão ser armazenadas em um objeto JSON. Exibir o objeto JSON (já preenchido) na tela ao clicar no botão "send".

The image shows a contact form with a teal background. At the top, the title 'Contact Form' is centered in white. Below the title are three input fields: 'Name', 'Email', and 'Message'. The 'Message' field is a larger text area. Below these fields is a checkbox labeled 'Send a copy to my email address'. At the bottom, there is a white button with the text 'SEND' in teal.

02) Construa um site dinâmico utilizando um arquivo JSON. Utilize o objeto JSON abaixo.

```
var videos = [  
  {  
    "youtubeId": "ZtMzB5CoekE",  
    "title": "3 Técnicas Que Eu Uso Para Aprender a Programar Qualquer Coisa  
(mesmo sem ter o dom da programação)",  
    "author": "Filipe Deschamps"  
  },  
  {  
    "youtubeId": "MV3dxDwRgnQ",  
    "title": "Trabalho Remoto: Vagas Disponíveis HOJE para Ganhar em Dólar e  
Real (+9 dicas para ser contratado)",  
    "author": "Filipe Deschamps"  
  },  
  {  
    "youtubeId": "n3tMEOW9KGY",  
    "title": "3 projetos de nível iniciante que geram até $3000 por Mês.",  
    "author": "Programador BR"  
  },  
  {  
    "youtubeId": "nZun1Y_CMJY",  
    "title": "5 maneiras de ganhar dinheiro como programador sem ter um chefe",  
    "author": "Programador BR"  
  }  
]
```

];

Neste exercício, você praticará o uso do JSON para criar HTML dinamicamente. A página inicial já possui os vídeos descritos em JSON e funções que os transformam em uma lista interativa. Siga estas etapas e faça o máximo que puder no tempo que temos:

- Adicione um vídeo adicional ao JSON, para que você tenha mais do que os quatro vídeos listados.
- O seu website dinâmico apresenta os vídeos que estão no seu JSON.
- Pode usar os conceitos que já vimos em outras aulas (CSS grid, layouts, fontawesome).
- Crie um card para cada item do JSON. Seja inteligente, se seu JSON tem cinco itens, é relativamente fácil criar cinco cards. Mas e se seu JSON tiver 1000 itens? Você vai criar 1000 cards na mão? Use a cabeça!
- O link para um vídeo do youtube segue a seguinte sintaxe: `www.youtube.com/watch?v=youtubeID`
- Troque `youtubeID` pelo valor desta chave que está no JSON.

Veja alguns exemplos de cards para os vídeos. Pode se inspirar em um dos cards abaixo, ou criar o seu próprio card.



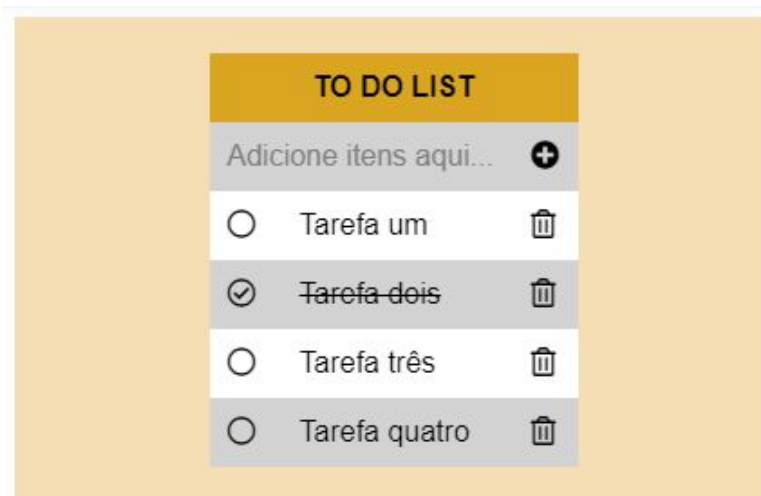
Lista 06 - Data: 13/04/2020. Data de entrega: 20/04/2020.

01) Crie quatro div's. Cada <div> deve conter cinco parágrafos diferentes. Cada <div> deve ter uma cor de fundo diferente. Fora das div's, crie uma lista com <ul>, um elemento <li> para cada <div>. Todas as div's devem começar ocultas. Ao clicar no elemento <li> da lista correspondente a <div>, esta <div> aparece. Ao clicar novamente, a <div> oculta.

02) Crie uma lista <ul> vazia. Crie uma caixa de texto (<input type="text">) e um ícone ( + 'plus sign'). Todo conteúdo inserido na caixa de texto, ao clicar no ícone, será adicionado como um elemento (<li>) da lista (<ul>).

03) Faça uma lista de tarefas utilizando HTML, CSS e JS, similar ao exemplo abaixo. Esta lista de tarefas deve possuir as seguintes funcionalidades:

- Adicionar uma nova tarefa a lista;
- Remover uma tarefa da lista;
- Marcar uma tarefa como concluída (~~tachada~~);
- Linhas zebradas;
- Ícones do fontawesome.



04) Faça um jogo no qual o jogador tem cinco chances para adivinhar um número que foi gerado aleatoriamente. Utilize HTML, CSS, JS. As mensagens que informam ao jogador se o número está alto ou baixo, é uma <ul> com <li>.

## Adivinhe o número!

Adivinhe o número entre 1 e 100.

- 57 é muito alto
- 12 é muito baixo
- 20 é muito baixo
- 45 é muito alto
- **42 é o número certo!**

05) Faça um sistema para geração de crachás similar ao exemplo abaixo. Ao clicar em algum campo (nome da empresa, nome do funcionário, função, foto), este campo se “transforma” em um `<input type="text">` e um `<button>` de confirmação, permitindo assim a atualização do conteúdo do campo.



Lista 05 - Data: 12/03/2020. Data de entrega: 19/03/2020.

01) O sequestro de logotipo. Nenhuma página inicial está a salvo do bandido do logotipo!

- Abra [www.google.com.br](http://www.google.com.br) no navegador e abra o console.
- Encontre o logotipo do Google e armazene-o em uma variável.
- Modifique o source do logotipo IMG para que ele seja um logotipo do Yahoo.
- Encontre o botão de pesquisa do Google e armazene-o em uma variável.
- Modifique o texto do botão para que ele diga "Yahooo!" em vez de "Google Search".

02) Crie um script em Javascript (JS) que será chamado por um botão. Esse script adiciona seu nome em uma tag <p> (previamente vazia) no documento HTML.

03) Faça dois parágrafos com conteúdos distintos. Crie um script em JS que será chamado por um botão. Esse script troca o conteúdo dos dois parágrafos entre si.

04) Adicione uma das imagens abaixo em um documento HTML.

Lâmpada desligada: [https://www.w3schools.com/js/pic\\_bulboff.gif](https://www.w3schools.com/js/pic_bulboff.gif)

Lâmpada ligada: [https://www.w3schools.com/js/pic\\_bulbon.gif](https://www.w3schools.com/js/pic_bulbon.gif)

Adicione dois botões (uma para acender, e outro para apagar). Ao clicar no botão que acende a lâmpada, troque a propriedade "src" da tag <img> para o link da lâmpada ligada. Ao clicar no botão que apaga a lâmpada, troque a propriedade "src" da tag <img> para o link da lâmpada desligada.

05) Crie uma tabela qualquer com 10 linhas com qualquer conteúdo. Crie dois botões (<button>). Um botão ativa as linhas zebreadas da tabela. O outro botão desativa as linhas zebreadas da tabela.

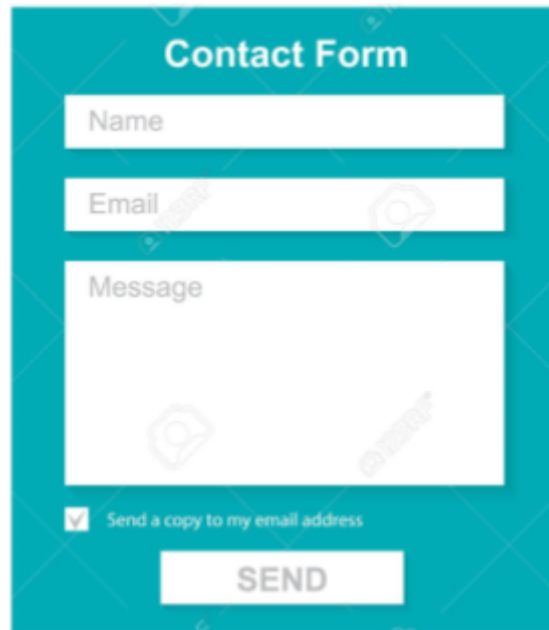
06) Crie uma caixa de texto (<input type="text">). Crie um parágrafo <p> com qualquer conteúdo. Crie um botão. Ao clicar no botão, o texto presente na caixa de texto, substitui o conteúdo do parágrafo. Crie um <input type="color"> para mudar a cor desse parágrafo.

07) Crie dois botões (<button>). Crie um título (<h1>) e três parágrafos com qualquer conteúdo. Cada botão altera uma propriedade do CSS: cor da fonte, e cor do fundo.

08) Faça uma tabela com duas colunas (Nomes e Números). Crie dois botões, um para adicionar uma linha a tabela e outra para remover a última linha da tabela. Quando adicionar uma linha, o 'nome' será aleatório vindo de uma lista (ou array) já pré-definido. O número também será aleatório. A tabela deve ser zebreada.

Lista 04 - Data: 05/03/2020. Data de entrega: 12/03/2020

01) Faça os formulários abaixo em uma página web. O formulário não precisa estar funcionando (enviando e recebendo os dados). Faça o mais similar possível. Faça um formulário por página.



**Contact Form**

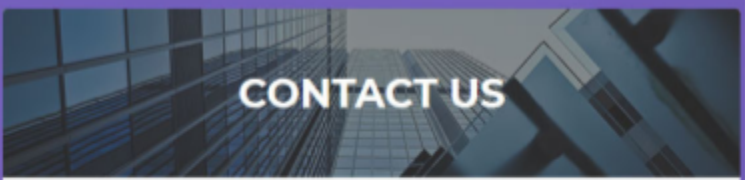
Name

Email


Message


☒ Send a copy to my email address


**SEND**




# CONTACT US

Full name

Eg. example@email.com

Eg. +1 800 000000

Your comments...

SEND NOW

# Colorlib Contact Form

Contact us for custom quote

Type your message here.....



## Address Details

Line 1

Line 2

City

State

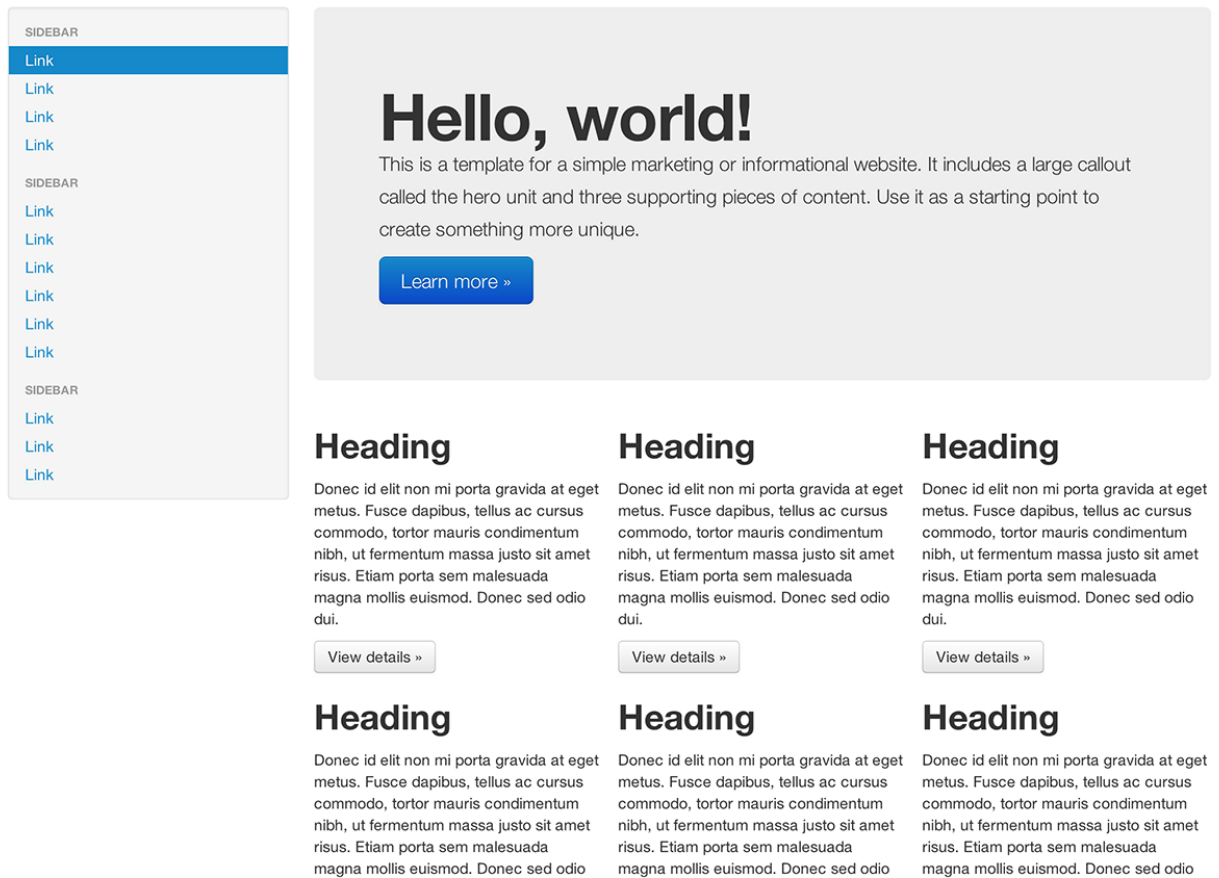
Postcode

Country

2) Faça um site com no mínimo sete páginas. Tema livre (currículo pessoal, time, banda, filme, série, linguagem de programação, carreiras, etc). Deve ter um menu em todas as páginas. Deve ter um formulário de contato (não precisa estar funcionando). Deve ficar bonito.

Lista 03 - Data: 20/02/2020

01) Utilize CSS Grid para criar uma página o mais similar possível ao exemplo abaixo:



2) Utilize CSS Grid para criar uma página o mais similar possível ao exemplo abaixo:

---

# Hello, world!

This is a template for a simple marketing or informational website. It includes a large callout called the hero unit and three supporting pieces of content. Use it as a starting point to create something more unique.

[Learn more »](#)

## Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

## Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

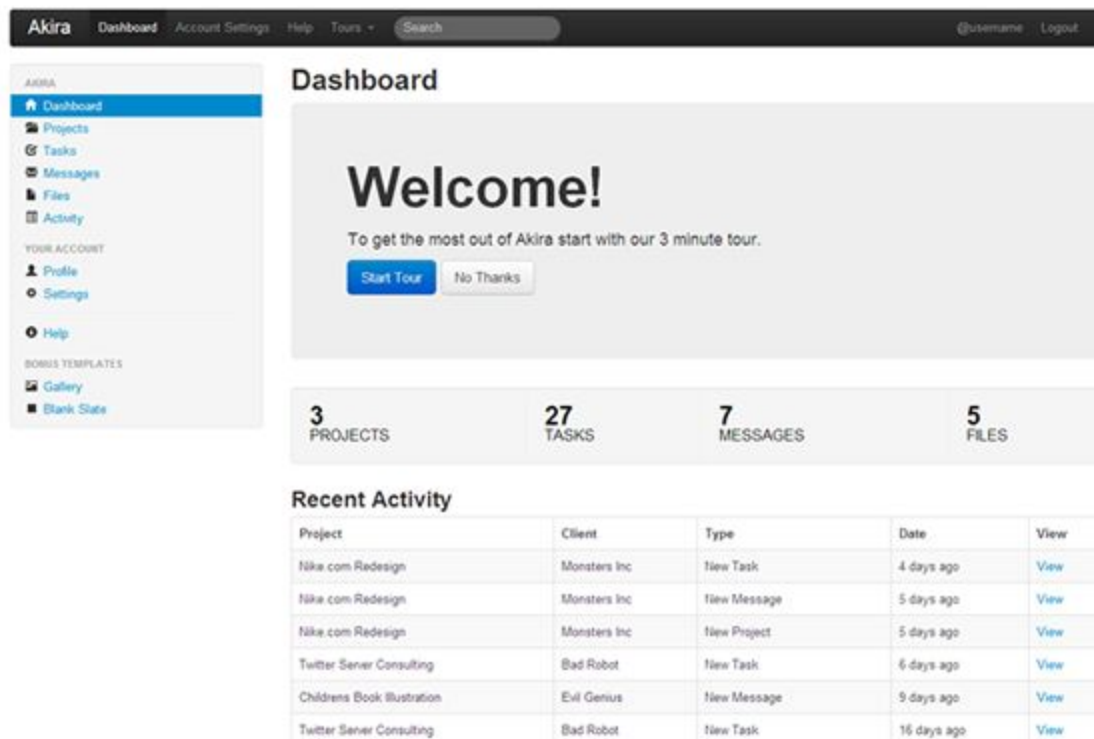
[View details »](#)

## Heading

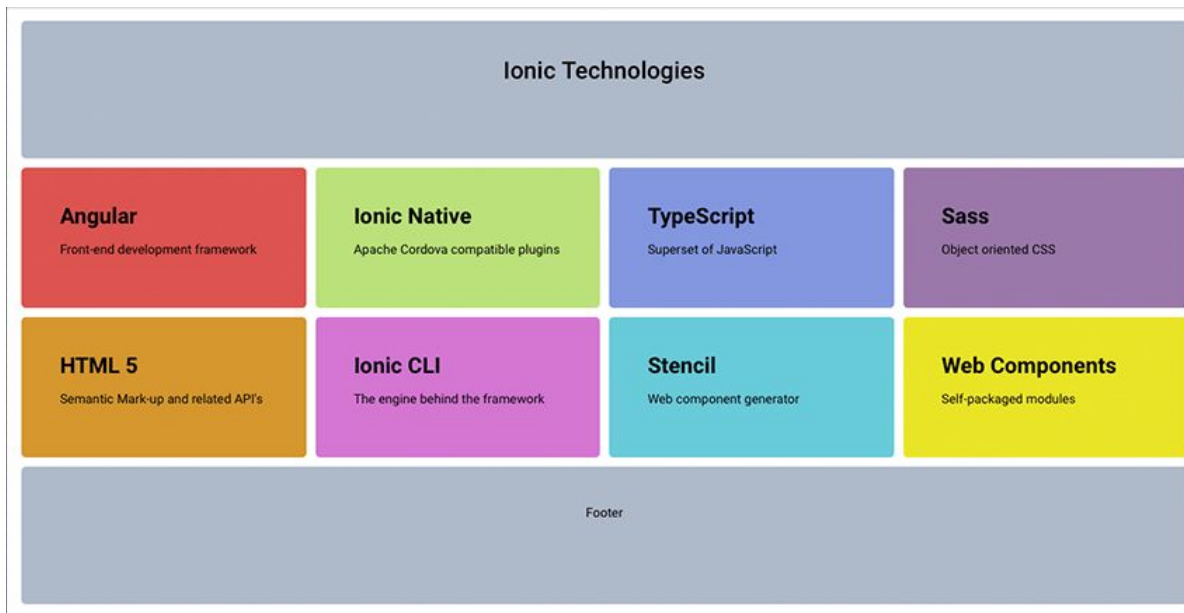
Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

[View details »](#)

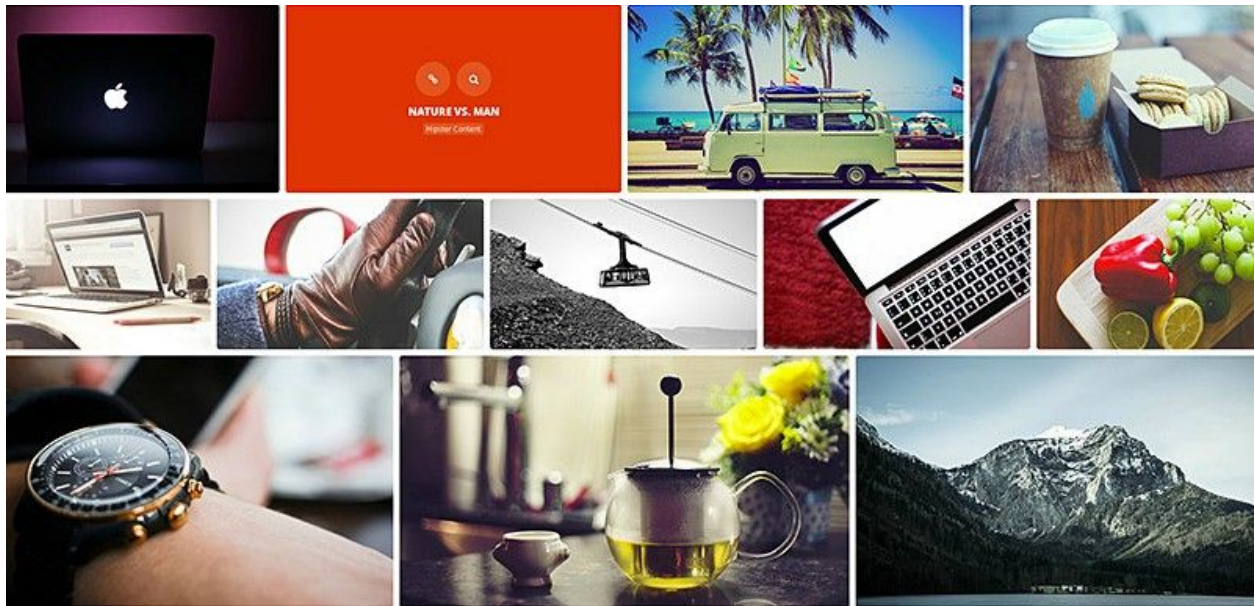
3) Utilize CSS Grid para criar uma página o mais similar possível ao exemplo abaixo. Nesse exercício a parte “Recent Activity” é uma tabela <table>. No menu a direita, utilize o [fontawesome](#) ou [iconmonster](#).



4) Utilize CSS Grid para criar uma página o mais similar possível ao exemplo abaixo:



5) Utilize CSS Grid para criar uma página o mais similar possível ao exemplo abaixo (utilize as imagens que você quiser, não precisa ser exatamente as mesmas imagens do exemplo). Utilize apenas um container:



Lista 02 - Data: 13/02/2020

01) Criar uma página com **seu** currículo pessoal similar a página da imagem abaixo:

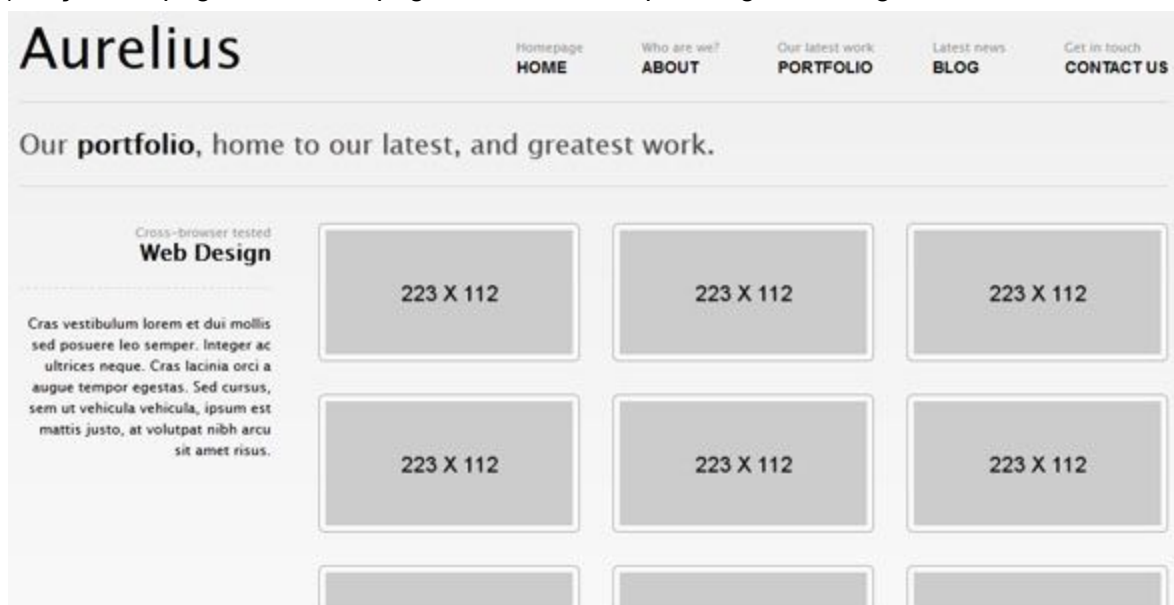
<b>Jamie Peterson Web designer</b>		
<p>34 Walnut Street Springfield, RI 02900 401-555-5982 jpeterson@twdexercise.com www.twdexercise.com</p> <p>A creative, self-motivated web designer with strong typographic, technical, and visual design skills. A team player, able to juggle multiple projects, meet deadlines, communicate clearly with coworkers and clients, and work with minimum supervision. Detail orientated, a quick learner, and a true believer in the importance of accessibility, customer experience, and usability in web design.</p>	<p><b>Professional Experience</b> <b>Website Design Company</b> Springfield, RI</p> <p>An award-winning web design company, specializing in branding and building new initiatives for non-profit organizations.</p> <p><b>Web Designer</b> May 2012-Present</p> <p>Working in a fast-paced environment, responsible for designing and building a variety of websites. Meet with clients to identify design requirements, interact with team of designers, programmers, writers, and clients throughout the project. Test websites for functionality in different browsers and at different resolutions.</p> <p><b>Springfield Pharmaceuticals</b> Springfield, CT</p> <p><b>Web Design Intern</b> June 2011-May 2012</p> <p>A multi-national pharmaceutical company, with products for human and animal consumption.</p> <p>Worked with an in-house design team of designers, copy-writers, photographers, and programmers. Designed landing pages, edited online content, and promoted products both organically and using Google ad words.</p> <p><b>Exclusives Clients</b></p>	<p><b>Skills</b></p> <ul style="list-style-type: none"><li>• Proficient in PC and Mac OS, Adobe Creative Suite, Search Engine Optimization</li><li>• Fluent in HTML, CSS, JavaScript, Flash and ActionScript, Visual Basic, PHP, MySQL</li><li>• Experience creating E-commerce sites</li><li>• Adept in Graphic Design and Typography</li></ul> <p><b>Education</b></p> <p>BFA in Visual Design University of Massachusetts Dartmouth Dartmouth, MA 2007-2011</p>

02) Faça uma página similar a imagem abaixo:





03) Faça uma página similar a página abaixo. Coloque imagens no lugar das caixas cinzas.



04) Crie uma página com uma tabela conforme o *layout* abaixo:

Sudeste x Sul		
Estados		Cidades
Sudeste	São Paulo	São Paulo
		Guarulhos
		Campinas
	Rio de Janeiro	Rio de Janeiro
		São Gonçalo
		Duque de Caxias
	Minas Gerais	Belo Horizonte
		Uberlândia
		Contagem
	Espírito Santo	Vila Velha
Serra		
Cariacica		
Sul	Rio Grande do Sul	Porto Alegre
		Caxias do Sul
		Pelotas
	Paraná	Curitiba
		Londrina
		Maringá
	Santa Catarina	Joinville
		Florianópolis
Blumenau		

Observações:

- Textos: uso de fontes padrões (tipo e tamanho);
- Título da tabela: em formato itálico e negrito;
- Borda, espaçamento entre células e espaçamento interno padrões;
- Tabela com 400px de largura;
- Nome das cidades: em formato itálico.

05) Crie uma página com uma tabela conforme o *layout* abaixo:

<i>Sudeste x Sul</i>		
	Estados	Cidades
Sudeste	São Paulo	<ul style="list-style-type: none"> <li>• <i>São Paulo</i></li> <li>• <i>Guarulhos</i></li> <li>• <i>Campinas</i></li> </ul>
	Rio de Janeiro	1. <i>Rio de Janeiro</i> 2. <i>São Gonçalo</i> 3. <i>Duque de Caxias</i>
	Minas Gerais	<ul style="list-style-type: none"> <li>◦ <i>Belo Horizonte</i></li> <li>◦ <i>Uberlândia</i></li> <li>◦ <i>Contagem</i></li> </ul>
	Espírito Santo	I. <i>Vila Velha</i> II. <i>Serra</i> III. <i>Cariacica</i>
Sul	Rio Grande do Sul	a. <i>Porto Alegre</i> b. <i>Caxias do Sul</i> c. <i>Pelotas</i>
	Paraná	<ul style="list-style-type: none"> <li>• <i>Curitiba</i></li> <li>• <i>Londrina</i></li> <li>• <i>Maringá</i></li> </ul>
	Santa Catarina	1. <i>Joinville</i> 2. <i>Florianópolis</i> 3. <i>Blumenau</i>

Observações:

- Textos: uso de fontes padrões (tipo e tamanho);
- Título da tabela: em formato itálico e negrito;
- Borda, espaçamento entre células e espaçamento interno padrões;
- Tabela com 400px de largura;
- Nome das cidades: em formato itálico.

Lista 01 - Data: 06/02/2020

1) Crie uma página web com o visual abaixo:

<b>Café Starbuzz</b>
<b>Mistura de Casa, RS 1,49</b>
Uma mistura suave e delicada de cafés do México, Bolívia e Guatemala.
<b>Café arábico com leite, RS 2,35</b>
Expresso, leite fervido e xarope de chocolate.
<b>Cappucino, RS 1,89</b>
Uma mistura de expresso, leite fervido e espuma.
<b>Chá Chai, RS 1,85</b>
Uma bebida condimentada com chá preto, especiarias, leite e mel.

2) Crie uma página com o visual abaixo:

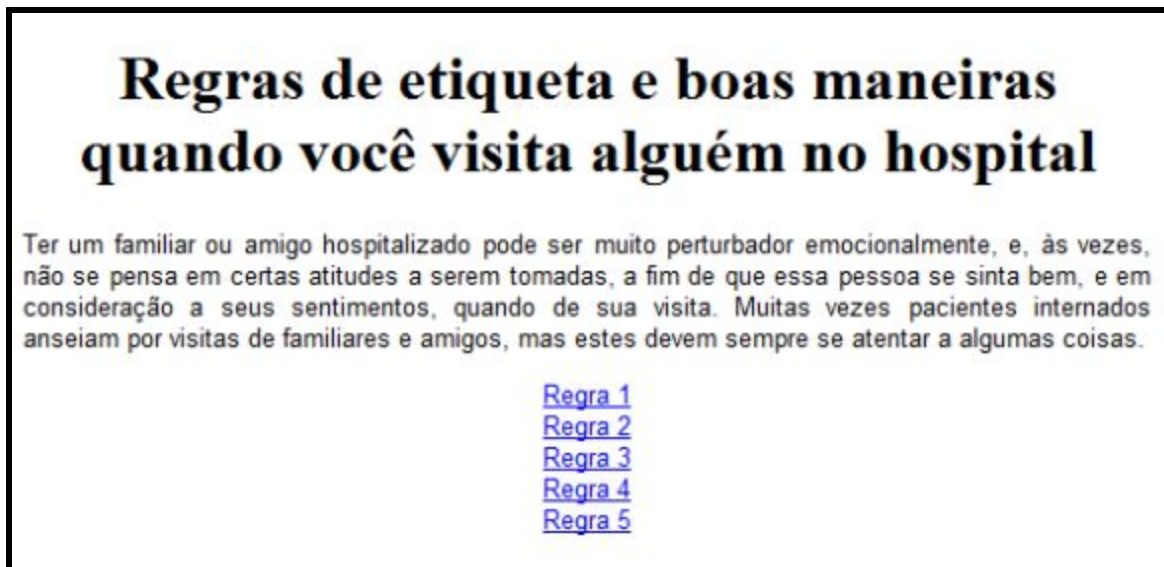
### Mudanças de linha

"No mundo dos negócios todos são pagos com duas moedas: Dinheiro e Experiência... Agarre a experiência primeiro, o dinheiro virá depois." --Harold Geenen

"Há homens que lutam um dia e são bons.  
Há outros que lutam muitos dias e são melhores.  
Mas há os que lutam todos os dias e estes são imprescindíveis."

--Bertolt Brecht

3) Crie uma página com o visual abaixo:



- Título da página: “Etiquetas e Boas Maneiras”;
- Uso de cabeçalho de nível 1 para a primeira linha de texto (“Regras de etiqueta e boas maneiras quando você visita alguém no hospital”);
- Uso de fonte padrão (família e tamanho) para a primeira linha de texto;
- Alinhamento centralizado para a primeira linha de texto;
- Uso de fonte da família “Arial”, com tamanho 20px, para os demais textos;
- Uso de alinhamento justificado para o texto “Ter uma família ... a algumas coisas”;
- Relação de links para as páginas “regra1.html”, “regra2.html”, “regra3.html”, “regra4.html” e “regra5.html”, da forma como exibido na imagem acima.
- 5 páginas, de nomes “regra1.html”, “regra2.html”, “regra3.html”, “regra4.html” e “regra5.html”.
- Diferenças de conteúdo entre as páginas: número da regra e descrição da regra, conforme tabela abaixo:
  - Regra 1 Se não for um familiar próximo, entre em contato com a família da pessoa ou o hospital para saber se, e quando, você pode visitar a pessoa internada.
  - Regra 2 Se for possível, ligue para o paciente e lhe pergunte se ele está disposto a receber visitas.
  - Regra 3 Esteja atento ao fato de que a pessoa poderá ficar cansada à medida que sua visita prossegue. Se perceber que a pessoa está querendo dormir ou ficando cansada, mesmo que esteja há pouco tempo com ela, despeça-se e visite-a noutro dia.
  - Regra 4 Mantenha uma visita positiva, sendo uma pessoa positiva, cordial, simpática - não dê ao paciente notícias perturbadoras. Anime sempre a pessoa, mesmo que ela seja negativa.

- Regra 5 Se o quarto de hospital for compartilhado, cumprimente os pacientes restantes e suas visitas quando chegar e quando sair; se achar oportuno deseje-lhe melhoras também.
- Uso de cabeçalho de nível 1 para a primeira linha de texto (“Regras de etiqueta e boas maneiras quando você visita alguém no hospital”);
- Uso de cabeçalho de nível 3 para a segunda linha de texto (“Regra n”, onde n é o número da regra);
- Título da página: “Etiquetas e Boas Maneiras :: Regra n”, onde n é o número da regra;
- Uso de fonte padrão (família e tamanho) para as duas primeiras linhas de texto;
- Alinhamento centralizado para as duas primeiras linhas de texto;
- Uso de fonte da família “Arial”, com tamanho 20px, para os demais textos;
- Alinhamento centralizado para o texto que contém a descrição da regra;
- Dois links, posicionados em um parágrafo centralizado para as páginas da regra anterior e da próxima regra (em relação à regra descrita na página em questão), respectivamente. Observar duas exceções: na página “regra1.html”, o link à esquerda (texto “Página Inicial”) aponta para a página principal; na página “regra5.html”, o link à direita (texto “Página Inicial”) aponta, igualmente, para a página principal. Sobre o espaçamento entre os links, exige-se uma quantidade mínima de 30 caracteres.
- Importante: certifique-se de que todos os arquivos citados anteriormente estejam armazenados em uma mesma pasta.