

SHIFT

 FIAP



# FRONT-END JOURNEY

DESENVOLVIMENTO WEB COM ANGULAR & REACT

# FRONT: **PROGRAMAÇÃO**

HTML - CSS - BOOTSTRAP - SASS

---



## ISRAEL MARQUES JÚNIOR

PROFESSOR

---

- Israel é pós-graduado em Engenharia Web e trabalha com educação há 27 anos. Trabalhou no desenvolvimento de sistemas para desktop e, em seguida, migrou para a criação de aplicações para a Internet.
- Na FIAP, é professor nos cursos de: Sistemas de Informação, Análise e Desenvolvimento de Sistemas, Sistemas para Internet e Jogos Digitais para as disciplinas focadas em front end.

✉ [profisrael.copi@fiap.com.br](mailto:profisrael.copi@fiap.com.br)

# AGENDA

---

1

## AULA 1

CONTEÚDO – Baixando o Editor + Introdução HTML + CSS

2

## AULA 2

CONTEÚDO – Listas + Links + Imagens + Divs + Posicionamento

3

## AULA 3

CONTEÚDO – Semântica – Background – Flexbox

4

## AULA 4

CONTEÚDO – Tabelas – Formulários

5

## AULA 5

CONTEÚDO – Design Responsivo + Media Queries + Mobile First

# AGENDA

---

6

## AULA 6

CONTEÚDO – BootStrap: Introdução + Containers + Utilitários

7

## AULA 7

CONTEÚDO – BootStrap: Grid + NavBar + Cards + Carousel

8

## AULA 8

CONTEÚDO – BootStrap: Accordion + Modal + Formulários

9

## AULA 9

CONTEÚDO – Sass: Introdução + Instalação + Conceitos Iniciais

10

## AULA 10

CONTEÚDO – Sass: Estilização + Variáveis + Funcionalidades

## AULA 5

# DESIGN **RESPONSIVO**

---





# DESIGN RESPONSIVO

---

Design responsivo consiste em como as páginas web se comportarão em telas diferentes, principalmente nos dispositivos móveis. Por meio dele será definido como o layout da sua página será exibido em telas pequenas, médias, grandes e até extragrandes, ou seja, como o layout deve **responder** a diferentes tipos de dispositivos que fazem acesso às nossas aplicações.

# DESIGN RESPONSIVO

---

Em Maio de 2010, o designer e desenvolvedor web **Ethan Marcotte** publica o seguinte artigo: “Responsive Web Design”, <https://alistapart.com/article/responsive-web-design/>. É, portanto, nesse exato momento que foi criado o **termo design responsivo**.

Os sites responsivos exibem os conteúdos da mesma forma, independentemente do tipo de dispositivo que está sendo usado. Eles também respondem a qualquer mudança de posição do dispositivo: **retrato** (vertical) ou **paisagem** (horizontal).

# PILARES DO DESIGN RESPONSIVO

---

Para que uma aplicação baseada em layout responsivo seja desenvolvida, devemos entender os três pilares que fundamentam esse conceito:

- Layout fluído;
- Imagens flexíveis;
- Media queries.

# LAYOUT FLUÍDO

---

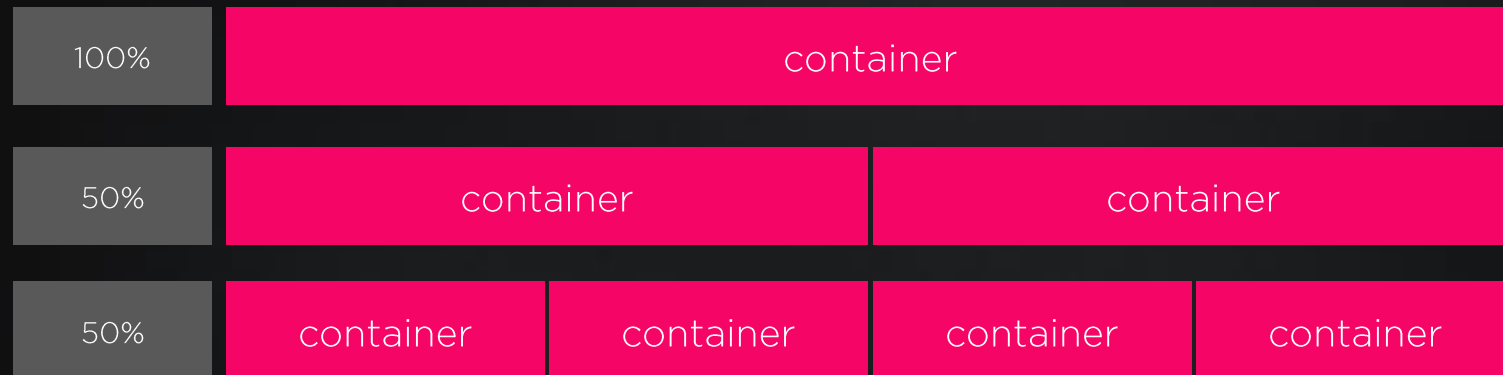
O layout fluído permitirá que os seus elementos se redimensionem conforme o tamanho da tela do usuário. Isso acontece porque ele foi projetado para utilizar a porcentagem (%) como medida relativa.

Com o uso da porcentagem, definimos tamanhos proporcionais para os containers existentes no código. Essas proporções ajudarão no redimensionamento, permitindo que o conteúdo seja exibido de forma correta.

# LAYOUT FLUÍDO

## PROPORÇÕES

---



## IMAGENS FLEXÍVEIS

---

Não dá para pensar em layout responsivo sem pensar em imagens flexíveis, elas também precisam se ajustar a telas diferentes.

Precisamos que as imagens fiquem no tamanho compatível com a tela e não percam a sua resolução. Para isso, usaremos a propriedade **max-width**, tendo como valor **100%**. A imagem permanecerá com seu tamanho máximo e se tornará flexível a telas menores.

# MEDIA **QUERIES**

# MEDIA QUERIES

---

Quando você começa a testar sua página web, pode acontecer que ela não fique tão legal quando mudamos o tamanho da tela ou o dispositivo.

É exatamente para isso que existem as **media queries**, elas são os pontos quebra do layout (breakpoints), e através deles podemos definir como o conteúdo será visualizado em dispositivos diferentes.



# MEDIA QUERIES

---

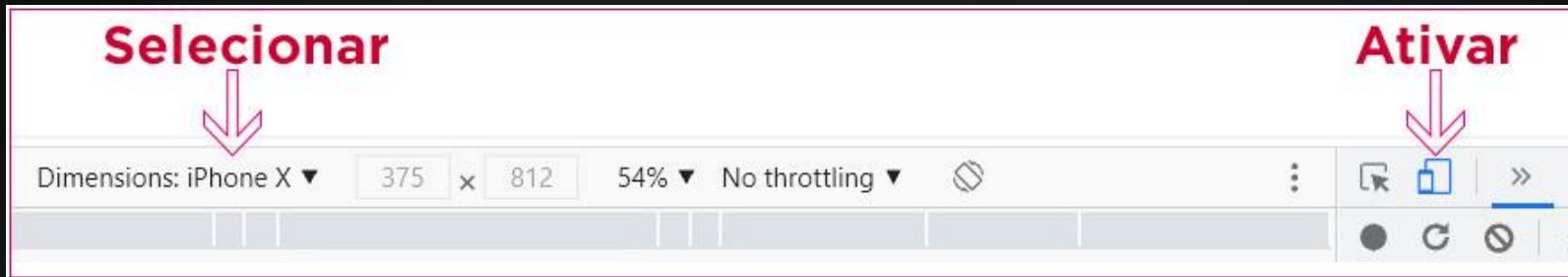
As media queries representam uma faixa específica de pixels que definem como o layout deve ser apresentado em diferentes tamanhos de tela, tudo isso dentro do seu arquivo CSS. Basicamente funciona assim:

- O browser detecta o viewport do dispositivo que você está fazendo o acesso.
- O browser consulta o seu arquivo CSS para ver se existe alguma regra definida para aquele **breakpoint**.
- Encontrando a regra, ele faz a renderização conforme a declaração, caso não encontre, ele executa a renderização padrão.

## MEDIA QUERIES

---

Você pode testar como sua página será exibida em diferentes dispositivos. Para isso, basta dar um clique direito na página e ativar a opção **Inspecionar**. Agora, clique no botão que tem um celular e um tablet juntos e escolha um dos dispositivos apresentados.



## MEDIA QUERIES

---

Os breakpoints serão usados por regras CSS específicas para atender a diferentes resoluções. Imagine que temos dois usuários acessando a sua aplicação em dispositivos com resoluções diferentes: 480px e 960px. Os breakpoints poderão ser os tamanhos dessas resoluções, ou seja, teremos uma regra CSS para os dispositivos com 480px e outra regra para dispositivos com 960px.

# MEDIA QUERIES

---

É muito simples declarar media queries: basta definir o valor para o breakpoint, ou seja, em quais tamanhos a regra será usada.

É aqui que entram as propriedades CSS **min-width** e **max-width**, definindo os valores mínimos ou máximos para que a regra seja aplicada.

```
@media(max-width: 900px){
```

Regras que serão aplicadas

```
}
```

```
@media(min-width: 1200px){
```

Regras que serão aplicadas

```
}
```

MOBILE **FIRST**

# MOBILE FIRST

---

A ideia do Mobile First é iniciar o desenvolvimento de sites pelos dispositivos móveis e, a partir dessa etapa, o layout é adaptado para dispositivos maiores como notebooks e desktops.

Esse conceito foi criado por **Luke Wroblewski** em 2009, e até hoje segue como uma das boas práticas no desenvolvimento Web.

## MOBILE FIRST

---

O conceito de Mobile First inverte o padrão inicial de desenvolvimento web. Ao invés de iniciarmos pelas telas grandes (desktops e notebooks), passamos a priorizar as telas menores (celulares e tablets). Dessa forma, estamos também priorizando os usuários que acessam por dispositivos móveis, ou seja, a grande maioria deles.

## MOBILE FIRST

---

Começar a pensar em como o conteúdo será exibido em telas menores ajuda a hierarquizar os elementos que serão prioridade para o usuário. Lembre-se que temos pouco espaço, sendo assim, o layout deve privilegiar o posicionamento das informações, deixando-as fáceis de se localizar, objetivas e visualmente bonitas.

Para que o usuário tenha uma ótima experiência é preciso que o layout e toda a sua aplicação possuam um cuidado todo especial com a usabilidade e com a acessibilidade, **pense sempre no usuário**.



## BIBLIOGRAFIA BÁSICA

---

- **DUCKETT**, Jon. HTML&CSS projete e construa websites. 2014.
- **SILVA**, Maurício Samy. HTML5 A linguagem de marcação que revolucionou a Web. 2011.
- **SILVA**, Maurício Samy. CSS3: Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. 2012.
- **TERUEL**, Evandro Carlos. HTML5 guia prático. 2011.
- **SANDERS**, Bill. Smashing Html5. 2012.
- **CASTRO**, Elizabeth; **HYSLOP**, Bruce. HTML5 e CSS3: Guia Prático e Visual. 2013.

# OBRIGADO



## FIAP

Copyright © 2021 | Professor Israel Marques Cajai Júnior

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.



# SHIFT

 FIAP



00000