



# PROGRAMAÇÃO PARA WEB

---

Luis Fernando Planella Gonzalez – Aula 03

# Professores

## **ANDREA KONZEN**

Professora Convidada

Formada em Ciência da Computação, com mestrado em Ciência da Computação na área de Inteligência Artificial pela (PUCRS) e doutorado em Informática na Educação na área de Inteligência Artificial com aplicação em Sistemas Educacionais pela (UFRGS). Além disso, possui pós-doutorado na área de Machine Learning em projetos voltados para exploração autônoma com múltiplos robôs. Atua como professora adjunta na Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) e como coordenadora de projetos na área de Inteligência Artificial na Foreducation EdTech. É secretária regional da Sociedade Brasileira de Computação do Estado (SBC), pesquisadora de projetos na área de Inteligência Artificial com ênfase em Sistemas Inteligentes e Aprendizagem de Máquina aplicados em áreas como Saúde, Educação e Agricultura, com o propósito de reverter benefícios concretos e significativos para a sociedade.

## **LUIS FERNANDO PLANELLA GONZALEZ**

Professor PUCRS

Doutor Ciências da Computação (PUCRS, 2018). Desenvolvedor e arquiteto Java com experiência profissional desde 1999, certificado pela Sun como programador e desenvolvedor de componentes web na plataforma Java. Entusiasta de software livre.

# *Ementa da disciplina*

Estudo do desenvolvimento de aplicações com HTML, CSS e JavaScript.  
Estudo sobre Document Object Model (DOM). Utilização de forms em aplicações WEB.  
Desenvolvimento de aplicações responsivas e acessíveis.

# Aula 3 - O que vamos trabalhar

- Mais detalhes do protocolo HTTP
- Exemplo de cada tipo de aplicação web
  - MPA (Multi Page Application)
  - SPA (Single Page Application)
  - PWA (Progressive Web Application)
- Desenvolvendo aplicações web responsivas e acessíveis
  - Web design responsivo
  - SVG (Scalable Vector Graphics)
  - Tema escuro
  - Animação reduzida
  - Recomendações de acessibilidade
  - Uso de elementos semânticos
- Componentes personalizados

# Mais detalhes do protocolo HTTP

# O protocolo HTTP (HyperText Transfer Protocol)

- O protocolo HTTP é utilizado para a transferência de documentos
  - Mas pode ser utilizado para baixar qualquer tipo de arquivo
- Possui distintas versões:
  - HTTP/1.1 (1997): Baseado em texto simples
  - HTTP/2 (2015): Adiciona compressão dos cabeçalhos, *server push* (o servidor pode enviar arquivos que sabe que serão utilizados futuramente), priorização de requisições e multiplexação de requisições
  - HTTP/3 (submetido para padronização em 2022): Não utiliza TCP, mas QUIC (sobre UDP), apresenta menor latência e requer HTTPS

# A estrutura de uma requisição HTTP/1.1

MÉTODO caminho HTTP/1.1

Cabeçalho-1: Valor 1

Cabeçalho-2: Valor 2

...

<linha em branco>

Linha 1 do corpo

Linha 2 do corpo

...



POST /upload HTTP/1.1

Host: dominio.com

User-Agent: exemplo

Content-Type: image/png

Content-Length: 50123

<bytes da imagem>

# A estrutura de uma resposta HTTP/1.1

HTTP/1.1 cod-status linha-status

Cabeçalho-1: Valor 1

Cabeçalho-2: Valor 2

...

<linha em branco>

Linha 1 do conteúdo

Linha 2 do conteúdo

...



HTTP/1.1 200 OK

Server: Apache

Content-Type: text/plain; charset=utf-8

Content-Length: 23

A imagem foi recebida com sucesso



# Métodos HTTP

Os métodos HTTP são verbos:

- GET: Baixa um recurso do servidor
- POST: Envia um recurso ao servidor / cria um recurso no servidor
- PUT: Atualiza um recurso no servidor
- DELETE: Remove um recurso do servidor
- PATCH: Atualiza um recurso parcialmente no servidor
- Há também outros de uso mais específico, como HEAD, OPTIONS, TRACE e CONNECT

# Códigos de status HTTP

Os códigos de status (numéricos) indicam a situação da requisição:

- 100 - 199: Informação
- 200 - 299: Sucesso
- 300 - 399: Redirecionamento
- 400 - 499: Erro do cliente
- 500 - 599: Erro do servidor

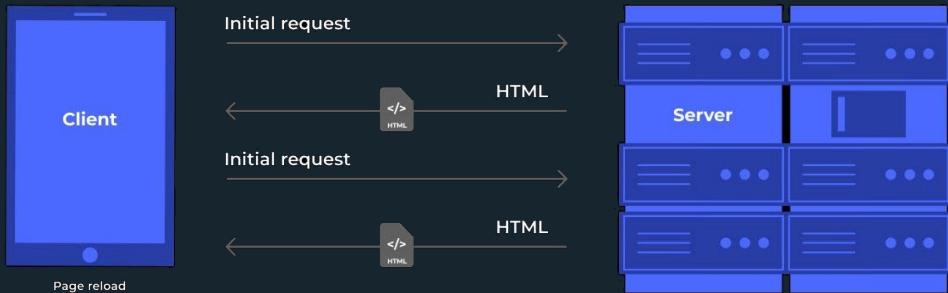
# Tipos MIME - Multipurpose Internet Mail Extensions

- Originalmente definidos para e-mails
- Amplamente usados no HTTP
- Define o tipo de arquivo sendo transmitido
- O navegador usa o MIME type, não a extensão!
- Formado por 2 componentes: tipo/subtipo
- Exemplos:
  - text/plain, text/html, text/javascript, text/css
  - image/jpeg, image/gif, image/png
  - video/mpeg, video/mp4
  - audio/mpeg, audio/wav
  - application/octet-stream, application/pdf

# MPA vs SPA

Multi x Single Page Application

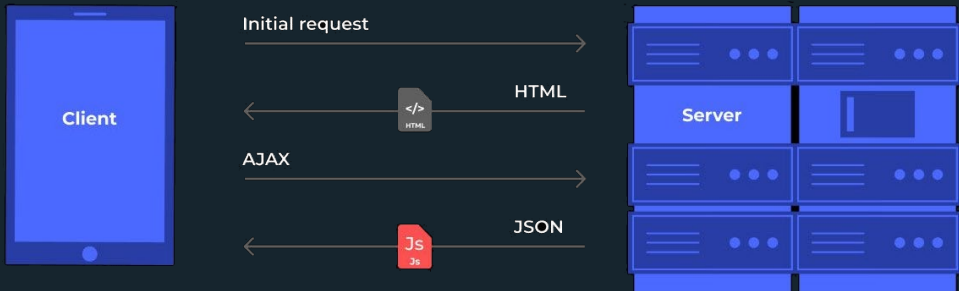
# Interações em uma MPA



# Interações em um MPA

- MPA é a forma clássica de se trabalhar com aplicações web
  - PHP, JSP, ASP...
- A cada interação com o servidor, é baixado um documento HTML
- O navegador substitui o documento atual pelo novo
- O uso do cache para arquivos comuns é muito importante!
- Exemplo: <https://luisfpg.github.io/exemplo-web/mpa/>

# Interações em uma SPA



# Atualização do DOM

- Como visto anteriormente, o DOM é uma estrutura em árvore cujos nodos são cada **elemento** no documento HTML
  - Os **atributos** também são nodos na árvore DOM
- É a base de uma SPA (*Single Page Application*)
  - O DOM é atualizado quando o usuário interage com a aplicação
  - A atualização ocorre de forma dinâmica
- Exemplo: <https://luisfpg.github.io/exemplo-web/spa/>



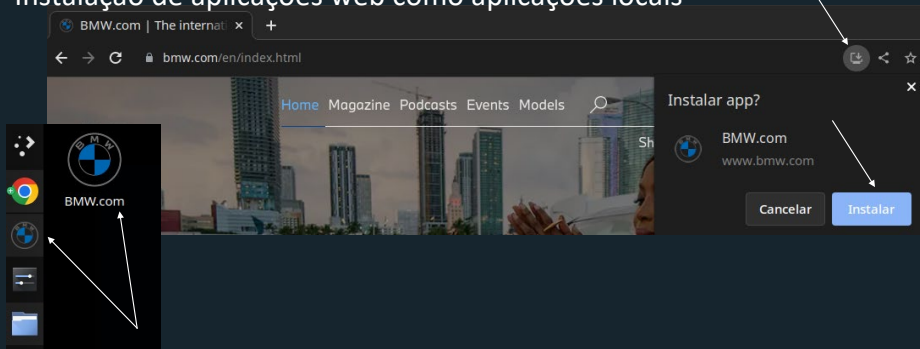
PWA - Aplicações web progressivas

# Aplicações web progressivas (PWA)

- A web é a plataforma de desenvolvimento mais portátil e acessível
- Muitas das aplicações que os usuários necessitam são web
- Mas, para acessar a aplicação, o usuário precisa
  - Abrir o navegador
  - Digitar a URL (ou clicar em um favorito)
  - Só então a aplicação é acessada
  - Mas fica 'contida' no navegador, com seu menu e barra de endereços
- Mas há uma forma de fazer aplicações web que se integram com o sistema operacional como qualquer outra aplicação
- Temos as Progressive Web Apps (PWA)

# Aplicações web progressivas (PWA)

## Instalação de aplicações web como aplicações locais



# Requisitos para a instalação de uma PWA

- A página é servida por HTTPS (ou do localhost)
- Inclui um manifesto
- Registra um *service worker* capaz de fornecer uma página offline

# Manifesto web

- É um arquivo JSON (com extensão .webmanifest ou .json)
- Provê metadados sobre a aplicação web
- Referenciada através de um link:

```
<link rel="manifest" href="pwa.webmanifest">
```

- Contém atributos como:
  - short\_name, name, description, icons, id, start\_url, scope, display, background\_color e theme\_color
  - Os ícones requeridos variam de navegador para navegador
  - O site web.dev (mantido pelo Google) recomenda ícones de 192x192, 384x384, 512x512 e 1024x1024 pixels, mas ao menos um de 512x512

# Service worker

- É um arquivo JavaScript que é registrado pela página
- Ele adiciona funções de tratamento de eventos
  - Eventos disponíveis: install, activate, fetch
- Funciona como um proxy para requisições
- Pode definir políticas, como:
  - Sempre servir páginas do cache
  - Sempre servir páginas da rede
  - Servir páginas alternativas em caso de erro de rede
  - ...
- Podem ser complexos quando provêem funcionalidades avançadas

# Exemplo de PWA

- Exemplo disponível em:

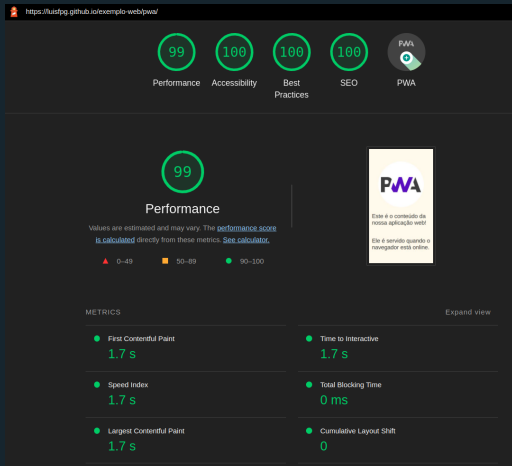
<https://luisfpg.github.io/exemplo-web/pwa/>

- Possui um manifesto
- Registra um service worker
- Serve uma página offline quando há erro na requisição

# Lighthouse

Ferramenta integrada ao Google Chrome que analisa métricas

- Performance
- Acessibilidade
- Melhores práticas
- Otimização para mecanismos de busca
- Adequação da página como PWA

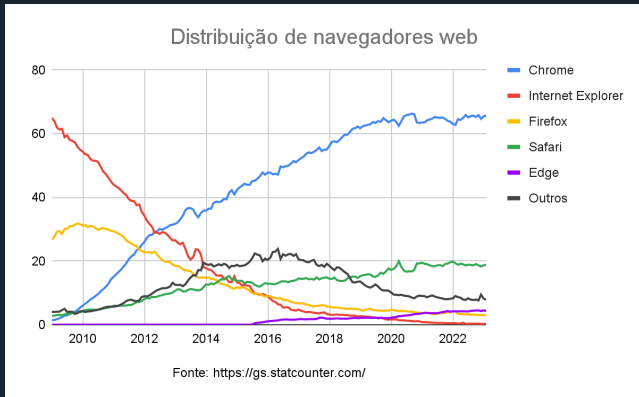




# Aplicações web responsivas e acessíveis

# Navegadores

- Google Chrome
- Safari
- Firefox
- Internet Explorer
- Edge
- ...



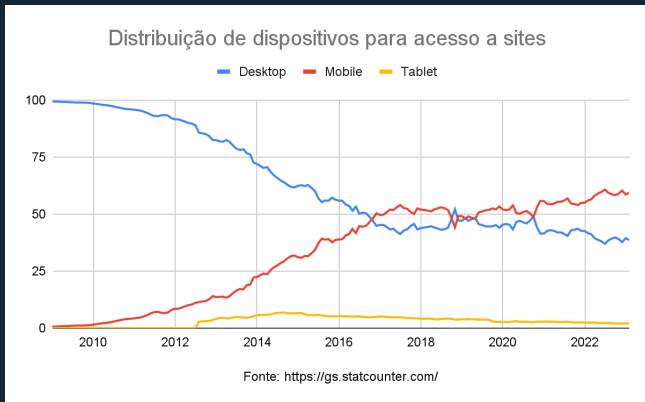
# Sistemas operacionais

- Linux
- Windows
- Mac
- Android
- iOS
- KaiOS
- ...



# Tipos de telas

- Monitor
- Smart TV
- Tablets
- 'Phablets'
- Smartphones
- Feature-phones
- Wearables
- CLI (terminal)
- ...



# Dispositivos para interação

- Mouse / teclado
- Somente teclado
- Tela de toque
- Teclado numérico
- Controle remoto
- Voz
- ...

Pesquisando no Google por PUCRS usando um navegador de console (links)

Google PUCRS - Pesquisa Google (p1 of 2)

PUCRS [ Pesquisar ]

TODAS NOTÍCIAS IMAGENS VÍDEOS

Pós-Graduação PUCRS Online - Conheça os Cursos Anônimo - online.pucrs.br/pós-graduação

Estude na melhor universidade privada do país pelo ranking da Folha. Conheça Agora.

PUCRS [www.pucrs.br](http://www.pucrs.br)

A Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) é uma instituição de ensino superior privada e católica brasileira, com campus localizado em ...

Pontifícia Universidade Católica do Rio Grande do Sul Instituição de ensino superior em Porto Alegre, Rio Grande do Sul

A Pontifícia Universidade Católica do Rio Grande do Sul é uma instituição de ensino superior privada e católica brasileira, com campus localizado em Porto Alegre. [Wikipédia](#)

Imagens Ver tudo

Pontifícia Universidade Católica do Rio Grande do Sul - Wikipédia ... Home - Estude na PUCRS Conheça a PUCRS e seus cursos de graduação Novos estudantes internacionais iniciam o semestre na PUCRS ... Saúde e Bem-estar: PUCRS é eleita a melhor universidade da área ... Case de Sucesso Gestão Ambiental PUCRS- Biota Geom

Ver tudo

[LINK] PUCRS Twitter - PUCRS

#PUCRSPesquisa I Projeto do professor da PUCRS sobre mulheres negras é selecionado em universidade alemã Confira: [www.pucrs.br/blog/projeto...](http://www.pucrs.br/blog/projeto...) há 1 dia Mídia postada por PUCRS

Pesquisadores debatem os impactos da pandemia na Educação em evento na PUCRS O encontro é promovido pela Rede Recupera Brasil, consórcio que une dez programas de pós-graduação e oito universidades do País para conduzir pesquisas sobre essa temática. [www.pucrs.br/blog/pesquis...](http://www.pucrs.br/blog/pesquis...) há 2 dias Mídia postada por PUCRS

Um novo semestre de possibilidades, encontros e reencontros se iniciou aqui no Campus. E, se você quer ingressar em uma graduação presencial na Universidade, ainda dá tempo de se inscrever no vestibular. • Acesse: [pucrs.br/vestibular](http://pucrs.br/vestibular) • Prova online: dias 15 e 16/3 há 2 dias Mídia postada por PUCRS

Voluntariado da PUCRS está com inscrições abertas • Atividades estão disponíveis nas modalidades presencial e remota (híbrida) - Confira as instituições participantes do projeto neste semestre: [www.pucrs.br/blog/inscri...](http://www.pucrs.br/blog/inscri...) há 2 dias Mídia postada por PUCRS

PUCRS (@pucrs) • Instagram photos and videos [www.instagram.com/pucrs](http://www.instagram.com/pucrs)

PUCRS. Uma universidade do tamanho do futuro. [linktr.ee/pucrs](https://linktr.ee/pucrs). Vestibular's profile picture. Vestibular. Eventos.

PUCRS Online (@pucrsonline) • Instagram photos and videos [www.instagram.com/pucrsonline](http://www.instagram.com/pucrsonline)

Somos pioneirismo, originalidade e inovação no ensino. Descubra o que é ser PUCRS Online: [online.pucrs.br/gratuito/jornada-da-inteligencia-profissional](http://online.pucrs.br/gratuito/jornada-da-inteligencia-profissional).

# Capacidade de interação

- Usuário regular
- Usuário “poweruser” (teclas de atalho, gestos...)
- Pessoas com pouca experiência computacional
- Pessoas surdas
- Pessoas cegas, ou com visão limitada
- Pessoas daltônicas
- ...



# Resumindo

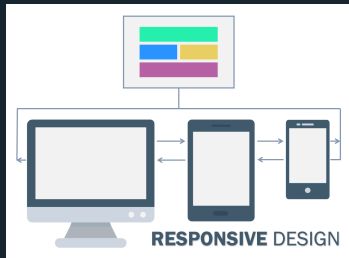
- Há dois tipos de acessibilidade a considerar
  - Dispositivos
  - Pessoas
- O W3C mantém o Web Accessibility Initiative (WAI)
  - Grupo que define padrões de acessibilidade na Web
  - Indica que as restrições para o conteúdo em aplicações mobile são semelhantes àsquelas enfrentadas por pessoas com deficiência <sup>1</sup>

[1] <https://www.w3.org/WAI/standards-guidelines/wcag-mobile-overlap/>

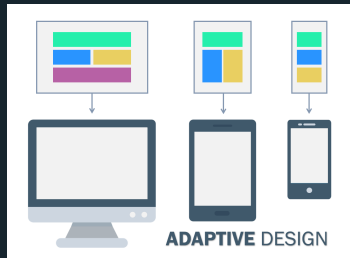
# Web design responsivo



# Layout responsivo vs layout adaptativo



Mesmo conteúdo. O layout é definido através de CSS.



Conteúdo distinto é gerado (no servidor) para o layout.

Fonte: <https://webflow.com/blog/adaptive-vs-responsive-design>

# Web design responsivo

“O conteúdo é como a água.

Você coloca água em uma xícara, ela se torna a xícara.

Você coloca água em uma garrafa, ela se torna a garrafa.

Você coloca água em um bule, ela se torna o bule.”

Josh Clark (originalmente, Bruce Lee). Ilustração de Stéphanie Walker.



## Abordagem “*mobile-first*”

- Abordagem proposta por Luke Wroblewski <sup>1</sup> em 2009
- Amplamente adotado hoje em dia
- O design é inicialmente projetado para dispositivos móveis
- Há pouco espaço na tela
  - Foco no conteúdo, que deve ser reduzido ao essencial
- Iterativamente considera-se tamanhos de tela maiores
- Cuidado para não tornar-se “*mobile-only*”

[1] <https://www.lukew.com/ff/entry.asp?933>

# CSS responsivo - media queries

- Media queries são declarações CSS condicionais
- Permitem aplicar estilos de acordo com o dispositivo / ambiente
- Há duas formas de uso:

Em um elemento <link>:

```
<link rel="stylesheet"
      media="print"
      href="estilos-impressao.css" />
```

Dentro de uma declaração CSS:

```
<style>
  @media print {
    .menu {
      display: none;
    }
  }
</style>
```

## CSS responsivo - media queries

- Usá-lo como atributo do elemento link tem a vantagem de baixar menos dados para o navegador quando o seletor não é aplicado
- Mas a desvantagem de requerer diversos arquivos
- Já o uso dentro da declaração (*inline*) CSS é mais prático
- Como as definições de CSS para media queries específicos tendem a ser pequenas, geralmente vale mais a pena o estilo *inline*
  - Claro, podem haver exceções que justifiquem separar os arquivos

# Media queries - largura de tela

- `width`: Largura exata do *viewport* (pouco usado)
- `min-width`: Largura mínima do *viewport*
- `max-width`: Largura máxima do *viewport*

Sempre utilizar entre parêntesis! Exemplo:

```
button { width: 100%; }
```

```
@media (min-width: 600px) {  
  button { width: auto; }  
}
```

# Design responsivo - media breakpoints

- A tela de cada dispositivo apresenta determinada dimensão
- Os dispositivos mapeiam pixels lógicos para pixels físicos
  - Exemplo: Samsung Galaxy s20
  - Resolução física: 1080 x 2400 (563 PPI)
  - *Viewport* web: 360 x 800 (141 PPI) (pixel ratio = 3x)
- Utilizamos larguras para diferenciar os tipos de dispositivo
- Cada “quebra” é chamada de *media breakpoint*
- Não há consenso nos tamanhos!



# Media breakpoints - Bootstrap

O Bootstrap (ferramenta CSS bem conhecida) define:

- Extra pequena - celulares (retrato):  $< 576\text{px}$
- Pequena - celulares (paisagem):  $\geq 576\text{px}, < 768\text{px}$
- Média - tablets (retrato), desktops:  $\geq 768\text{px}, < 992\text{px}$
- Grande - tablets (paisagem), desktops:  $\geq 992\text{px}, < 1200\text{px}$
- Extra grande - desktops, tvs:  $\geq 1200\text{px}, < 1400\text{px}$
- Extra extra grande - desktops, tvs:  $\geq 1400\text{px}$



# Media breakpoints - Material design

O Material Design (especificação usada no Android / Google) define:

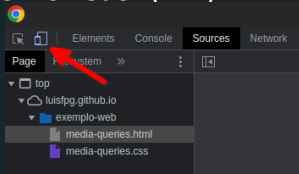
- Extra pequena - celulares:  $< 600\text{px}$
- Pequena - tablets:  $\geq 600\text{px}, < 1240\text{px}$
- Média - laptops:  $\geq 1240\text{px}, < 1440\text{px}$
- Grande - desktops / tvs:  $\geq 1440\text{px}$

# Exemplo de media queries: larguras de tela

- O código de marcação (HTML) é o mesmo!
- O CSS é único, mas se adapta
- Exemplo disponível em:

<https://luisfpg.github.io/exemplo-web/media-queries/>

- Ao abrir a página, abra as ferramentas de desenvolvedor (F12)
- Utilize o modo Dispositivo



# Elementos de interface

- *Mobile-first* não é somente largura de tela
- Tamanho de texto razoável
- O layout deve ser usável na tela pequena
- Menus com poucas opções ou inicialmente ocultos
- Pense o macro-layout da página (cabeçalho, conteúdo, rodapé, etc)
- ...

# Exemplo de layout responsivo

- O código de marcação (HTML) é único
- O CSS também é único, mas contém *media queries*
- Exemplo disponível em:  
<https://luisfpg.github.io/exemplo-web/layout-responsivo/>
- Ao abrir a página, abra as ferramentas de desenvolvedor (F12)
- Utilize o modo Dispositivo

# SVG - Scalable Vector Graphics

## SVG - Imagem vetorial (escalável)

- O mesmo grupo que mantém as especificações do HTML, CSS, JavaScript, etc, também define muitas outras
- Entre elas, o Scalable Vector Graphics - SVG
- Permite definir uma imagem com elementos vetoriais
- Assim, é garantido que a mesma imagem possa ser exibida em distintos tamanhos sem perder qualidade
- Já utilizamos ícones SVG no exemplo anterior

# SVG - Imagem vetorial (escalável)

```
<svg xmlns="http://www.w3.org/2000/svg"
width="16" height="16" fill="currentColor"
class="bi bi-list" viewBox="0 0 16 16">
  <path fill-rule="evenodd"
    d="M2.5 12a.5.5 0 0 1 .5-.5h10a.5.5 0 0 1 0 1H3a.5.5 0 0 1-.5-.5zm0-
4a.5.5 0 0 1 .5-.5h10a.5.5 0 0 1 0 1H3a.5.5 0 0 1-.5-.5zm0-4a.5.5 0 0 1 .5-
.5h10a.5.5 0 0 1 0 1H3a.5.5 0 0 1-.5-.5z" />
</svg>
```

Esta marcação representa a figura  
que utilizamos para abrir o menu  
<https://icons.getbootstrap.com/>



# SVG - Imagem vetorial (escalável)

- Parece bem complicado!
- Mas possui elementos mais simples também:
  - circle, rect, ellipse, line, polygon, text, ...
- Pode ter estilo aplicado da mesma forma que qualquer outro elemento dentro da página!
  - Inclusive a cor!
- Possui um sistema de coordenadas (viewbox), e todas as posições são relativos a ele
  - Independe do tamanho de exibição



# SVG - Imagem vetorial (escalável)

```
<svg viewBox="0 0 50 50">  
  <circle style="stroke:silver;fill:white" cx="25" cy="25" r="7" />  
  <rect style="fill:red" x="10" y="5" width="10" height="10" />  
  <polygon style="stroke:green;fill:none" points="30,45 40,35 40,45" />  
</svg>
```

Este exemplo é bem mais fácil de entender...

Principais aplicações do SVG:

- Ícones
- Gráficos dinâmicos



Tema escuro

# Tema escuro

- Os sistemas operacionais modernos permitem alternar entre temas
  - Claro (modo dia)
  - Escuro (modo noite)
- O fundo escuro com conteúdo claro pode:
  - Ajudar na questão de fadiga visual dos olhos
  - Reduzir a exposição à luz azul (afeta o sono)
  - Parecer mais minimalista e moderno (percepção)
- O fato é que cada vez mais pessoas usam o tema escuro
- E os sistemas web?!?

# Tema escuro

- Temos o media query prefers-color-scheme!
  - @media (prefers-color-scheme: light) {...}
  - @media (prefers-color-scheme: dark) {...}
- Bastante poderoso se combinado com variáveis CSS
- Mas temos atualmente um problema: como o usuário pode alternar se nos baseamos puramente no media query?
- Solução: JavaScript :-{

# Exemplo de tema escuro

- Exemplo disponível em:

<https://luisfpg.github.io/exemplo-web/modo-escuro/>

- Por JavaScript, quando o usuário prefere o tema escuro, o mesmo é usado por padrão
- O botão no canto superior alterna o tema
- Utiliza-se o LocalStorage
  - Na próxima visita, o tema é restaurado

# Animação reduzida

- Algumas pessoas preferem desativar as animações no sistema
  - Seja por questão de atenção ou preferência
  - Alguns dispositivos, especialmente leitores de e-books com tela de e-ink, possuem uma taxa de atualização muito reduzida
- Nestes casos, as animações atrapalham!
- Temos o media query `@media(prefers-reduced-motion: reduce)`
- Exemplo disponível em:  
<https://luisfpg.github.io/exemplo-web/animacao-reduzida/>
  - Esse exemplo utiliza a configuração do navegador

Acessibilidade

# Acessibilidade

- Devemos desenvolver um sistema / site acessível a todos
- O que vimos até aqui é considerados acessibilidade:
  - prefers-color-scheme
  - prefers-reduced-motion
- Mas são mais voltados às preferências do usuário
- E quando ao invés de preferência temos uma necessidade?
- O W3C mantém o Web Content Accessibility Guidelines (WCAG)
  - <https://www.w3.org/WAI/standards-guidelines/wcag/>
- São definidos níveis de conformidade (AAA é a melhor, AA, ... F)  
Se aplicado com rigor, muitos sites são reprovados! Cuidado!!!



Algumas das recomendações do  
WCAG para páginas acessíveis

# Texto

- Não há um tamanho mínimo de texto definido
  - É consenso que seja ao menos 16px (1rem)
- Deve poder ser aumentado a 200% sem que o usuário necessite rolagem horizontal para ler uma frase
- Usar uma unidade relativa ajusta o texto pela preferência do usuário
- Largura máxima recomendada de 80 caracteres por linha
- Espaçamento entre linhas é ao menos 1,5
- Espaçamento entre parágrafos é ao menos 1,5x maior que linhas
- Não utilizar o alinhamento justificado
  - Dificulta a leitura por apresentar espaçamento variável

# Contraste de cor

- O contraste é a relação entre a cor de fundo e a cor do texto
- Ranking de contraste:
  - Texto menor que 18pt (~24px): grau AA se contraste  $\geq 4.5:1$
  - Texto 18pt ou maior: grau AA se contraste  $\geq 3.0:1$
  - Grau AAA: contraste  $\geq 7.0:1$

fg: #222 bg: #000 taxa: 1.32	fg: #444 bg: #000 taxa: 2.16	fg: #666 bg: #000 taxa: 3.66	fg: #888 bg: #000 taxa: 5.92	fg: #aaa bg: #000 taxa: 9.04	fg: #ccc bg: #000 taxa: 13.08	fg: #eee bg: #000 taxa: 18.10
fg: #222 bg: #fff taxa: 15.91	fg: #444 bg: #fff taxa: 9.74	fg: #666 bg: #fff taxa: 5.74	fg: #888 bg: #fff taxa: 3.54	fg: #aaa bg: #fff taxa: 2.32	fg: #ccc bg: #fff taxa: 1.61	fg: #eee bg: #fff taxa: 1.16

# Utilização de cores

- As cores não podem ser a única forma de diferenciar elementos!
- Daltônicos podem ter dificuldade em perceber a diferença
- Exemplo, links apenas em cores distintas
  - Utilizar sublinhado, ou negrito
- Em outros elementos, apresentar um elemento textual indicativo
- A ferramenta de desenvolvimento dos navegadores pode emular diferentes tipos de daltonismo

# Navegação

- A página deve ter um título descritivo
- A página deve ser operável apenas com teclado
- A ordem de foco dos elementos deve ser predizível
- Os indicadores de foco de todos os elementos devem ser visíveis
- O propósito de um link deve ser identificado pelo texto do mesmo

Diretivas da WAI para conteúdo  
acessível a pessoas e dispositivos

# Resumo das recomendações WCAG

- <https://www.w3.org/WAI/standards-guidelines/shared-experiences/>
- Essa página define a experiência compartilhada do desenvolvimento de páginas acessíveis a dispositivos móveis e pessoas
- Dividido em 4 grupos - POUR
  - *Perceptible* - Perceptível
  - *Operable* - Operável
  - *Understandable* - Inteligível
  - *Robust* - Robusto

## Conteúdo perceptível. Evitar:

- Informação transmitida unicamente através de cor
- Páginas ou imagens muito grandes
- Elementos multimídia sem legendas
- Notificações apenas sonoras para eventos importantes (erros, ...)
- Elementos não textuais (multimídia) sem alternativas textuais
- Digitação muito específica de textos
- Conteúdo tabular ou posicionado com CSS não possui uma ordem de leitura correta quando linearizado
- Informação transmitida unicamente com CSS (elementos visuais)



## Conteúdo operável. Evitar:

- Requerer mouse para interação ou navegação
- Scripting (JavaScript) requerido para operar o conteúdo
- Plugin ou extensão do navegador requerido para operar o conteúdo
- Título da página faltante ou incorreto
- Inconsistência da ordem de foco (tabulação) e a sequência lógica do documento
- Texto não descritivo de links

## Conteúdo inteligível. Evitar:

- Palavras ou frases muito longas, jargões
- Sequência de conteúdo em distintas janelas sem avisar o usuário
- Conteúdo piscando, em movimento, rolando ou mudando

## Conteúdo robusto. Evitar:

- Código de marcação inválido ou não suportado
- Scripting (JavaScript) requerido para gerar o conteúdo

Uso de elementos semânticos

# Uso de elementos semânticos e nativos

- A especificação HTML define muitos elementos
- Quase todos podem ser emulados com <div>, CSS e JavaScript!
- Mas isso é ruim para a acessibilidade
- Os elementos nativos HTML são implementados pelos navegadores levando a acessibilidade a sério
- Sempre que um componente nativo estiver disponível, ele é preferível a um componente personalizado
- Implementar um componente acessível é bem difícil!

# Exemplos

Utilizar: Ao invés de:

<header> <div class="header">

<footer> <div class="footer">

<nav> <div class="menu">

<main> <div class="principal">

<aside> <div class="lateral">

<h1>, <h2>, ... <h6> <div class="cabecalho">

<dialog> <div class="dialogo">

<template> <div style="display:none">

# Cuidados ao desenvolver componentes

- O HTML define diversos componentes de formulário e interação
- Mas às vezes, tornam-se necessários outros...

Toppings  
Mushroom, Onion ▼

- ☐ Extra cheese
- ☒ Mushroom
- ☒ Onion
- ☐ Pepperoni
- ☐ Sausage
- ☐ Tomato

☒ Slide me!

Next page

Items per page: 10 ▼ Page 1 of 10 < >

First Second Third

▼ Fruits

▼ Apple

Fuji

Macintosh

Orange

Banana

> Vegetables

# Alguns pontos a considerar

- Se possível, utilizar uma biblioteca de componentes “aprovados”
  - Bootstrap, Material design, Ionic, ...
- Certifique-se que o componente seja acessível pelo teclado
- Utilize os atributos ARIA (Accessible Rich Internet Applications) para ajudar os leitores de tela para deficientes visuais:
  - `role`: define a função desempenhada pelo elemento
  - `aria-label`: define um rótulo curto para o elemento
  - `aria-description`: descrição detalhada do elemento
  - ...



# Web Components - Componentes personalizados

# Componentes personalizados

- É possível estender a funcionalidade das páginas criando-se componentes reusáveis - *Web Components*
- Eles funcionam de forma semelhante aos elementos padrão do HTML, como `<input>` e `<button>`, mas adicionando funcionalidade
- Embora existam frameworks como o React e o Angular, os *Web Components* são um padrão W3C
  - E a maioria desses frameworks podem integrar-se com o padrão

# Componentes personalizados

- Podem utilizar o *Shadow DOM* ('DOM fantasma')
- Permitem isolar o componente do resto da página
- Classes CSS e definições não conflitam com outros

# Criando um componentes personalizado

- Necessário criar uma classe que estenda a classe `HTMLElement`
- Métodos chamados em distintos pontos do ciclo de vida:
  - `connectedCallback()`: O elemento foi adicionado ao documento
  - `disconnectedCallback()`: O elemento foi removido do documento
  - `adoptedCallback()`: O elemento foi movido para outro documento
  - `attributeChangedCallback()`: O valor de um atributo foi modificado
    - Requer um getter estático da propriedade `observedAttributes`: `string[]`
- Pode gerar eventos como qualquer outro elemento

# Exemplo de Web Component

- Exemplo disponível em:

<https://luisfpg.github.io/exemplo-web/web-components/>

- Utiliza 2 componentes personalizados
  - Via tag no HTML
  - Via `document.createElement()`
- Também utiliza um componente externo de gráfico

**PUCRS** online  **uol** edtech.