

Programação II

2 - Aplicações Web

Páginas estáticas *versus* dinâmicas

- **Página estática:**

- O HTML e conteúdo gráfico é sempre estático, aparecendo de forma idêntica para qualquer visitante;
- Conteúdo somente é alterado quando o administrador manualmente edita o arquivo da página no servidor.

- **Página dinâmica:**

- Pode exibir dados diferentes, dependendo de informações como a hora do dia, usuário que está conectado, a data, o termo de pesquisa que foi utilizado, etc.;
- Ao invés de conter apenas HTML, contém também código dinâmico;
- Utiliza linguagens capazes de interagir com um BD (banco de dados), validar entradas de formulários, etc.

Aplicação web ou site?

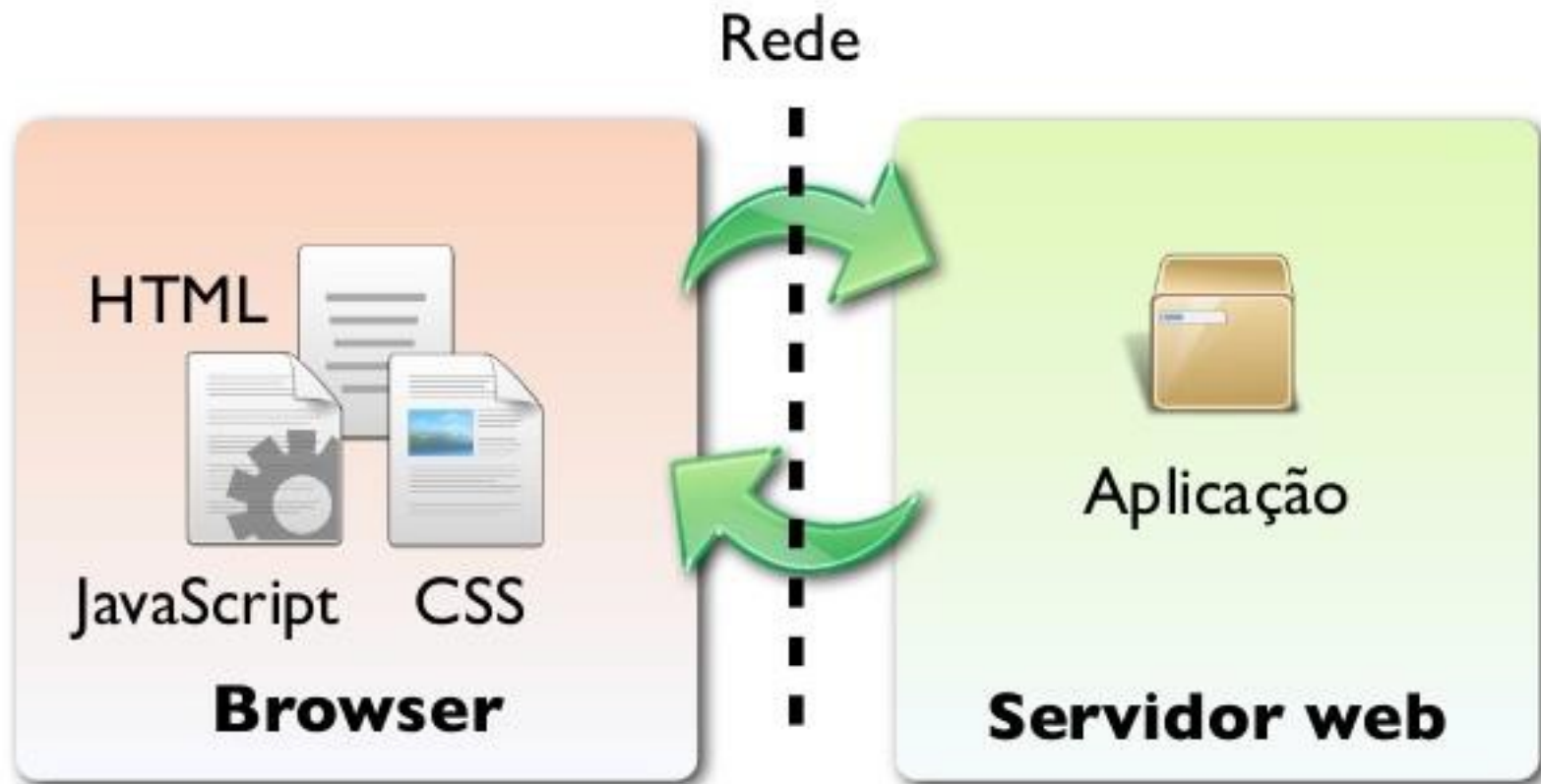
- **Website ou site**
 - Conjunto de páginas WEB com propósito informativo.
 - Exemplos: sites institucionais, pessoais, portfólios, hot-sites promocionais, etc.
 - É possível desenvolver apenas com HTML e CSS, como páginas estáticas.
 - Pode eventualmente envolver alguma linguagem server-side para facilitar a exibição dinâmica do conteúdo, mas não apresenta grandes possibilidades de interação com o usuário (execução de tarefas com formulários, bancos de dados, etc.).

Aplicação web ou site?

- **Aplicação WEB**

- Funciona como um sistema onde pode-se realizar ações como:
 - Cadastrar, consultar, editar e apagar informações em um banco de dados;
 - Interagir com outros usuários;
 - Realizar processos como compras, autorização de pagamento por cartão, geração de boletos, interação com redes sociais, etc.
- As páginas são dinâmicas e personalizáveis em tempo de execução, interagindo com as ações permitidas aos seus usuários.
- Não basta apenas HTML e CSS, pois requer a programação de ações que serão processadas no servidor.

Aplicação Web



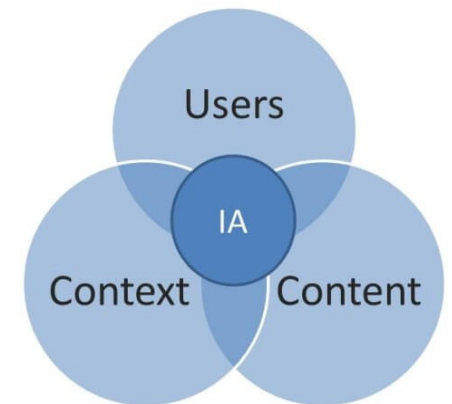
Aplicação Web

- Uma aplicação web possui as seguintes características:
 - Funciona em qualquer dispositivo que possua um navegador web;
 - Pode ser acessada de qualquer lugar com conexão à internet;
 - Não requer instalação/atualização no cliente (diferente de aplicativos *desktop*);
 - É mais segura em relação a vírus;
 - Independe de plataforma e sistema operacional;
 - Em dispositivos móveis, é possível adicionar um atalho à tela (simulando um app nativo). Neste caso, não ocupa memória do aparelho, mas também não é capaz de utilizar as funcionalidades como câmera, GPS, etc.
- Existem ainda os apps híbridos: partes web, partes em linguagem nativa da plataforma.

Aspectos de projeto

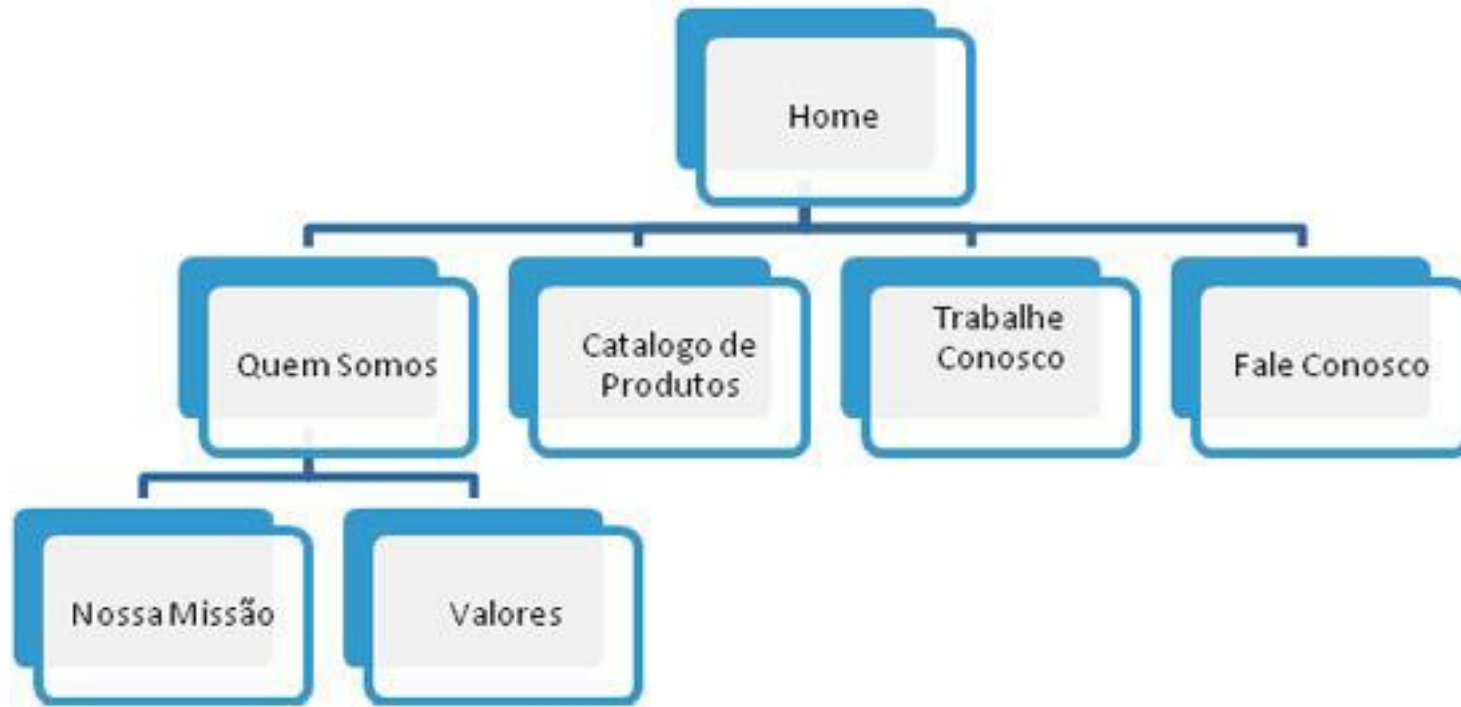
- **Arquitetura da Informação**

- Arranjo e estruturação que organiza os conteúdos em um site/aplicação.
 - Cada informação pode ser apresentada de N maneiras distintas. Diferentes usuários terão diferentes expectativas.
- AI visa facilitar/permitir o consumo de uma informação por um usuário em um determinado contexto.
- Tríade: **Usuário** / **Contexto de uso** / **Informação**
- Preocupações:
 - Estrutura da informação (quais blocos de conteúdo);
 - Organização da informação (hierárquica? linear?);
 - Nomenclatura dos itens;
 - Encontrabilidade da informação (terá caixa de busca? diferentes tipos de usuários terão diferentes menus? usuário logado terá mais funcionalidades?).



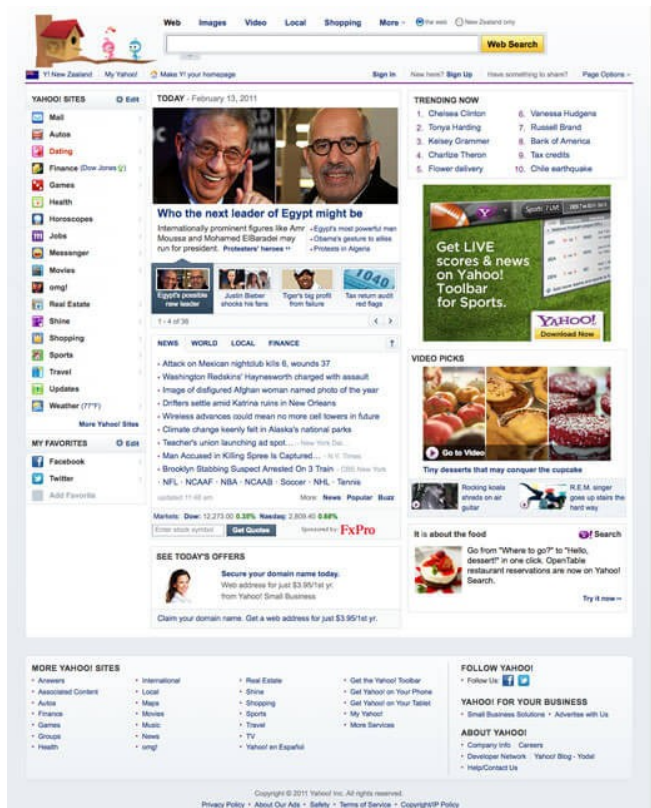
Aspectos de projeto

- **Arquitetura da Informação: Mapa do site**
 - Documenta o escopo, as seções e a hierarquia de um projeto, bem como auxilia outros membros da equipe, como o designer visual e o desenvolvedor.
 - Fornece uma visão do projeto como um todo.



Aspectos de projeto

- **Arquitetura da Informação: Wireframe**
 - Desenho básico (esqueleto) da estrutura do site.
 - Muito utilizado para comunicação entre os membros da equipe.
 - Foca na forma, e não na aparência final do site.

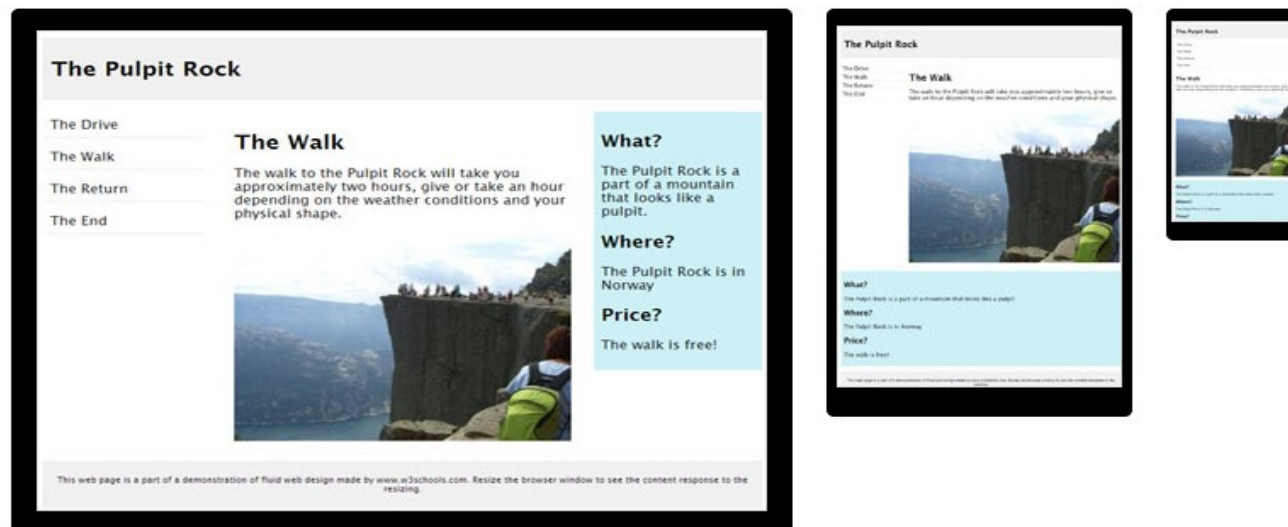


Site do Yahoo, ao lado de seu wireframe.

Aspectos de projeto

- **Responsividade**

- Design capaz de “responder” às características do dispositivo ao qual é servido.
- Um layout responsivo é capaz de expandir ou contrair com a finalidade de se acomodar de maneira usável e acessível à área onde é visualizado ou, mais genericamente, ao contexto onde é renderizado, seja um smartphone, um tablet, um desktop, um leitor de tela, um mecanismo de busca etc.



Aspectos de projeto

- **Usabilidade**

- Grau de facilidade de interação do usuário com uma interface ou sistema.
- O usuário deve ser capaz de completar a tarefa e alcançar seus objetivos no sistema (eficácia), com o menor esforço e sob uma experiência agradável (eficiência).
- Diz respeito à forma como o usuário se comunica com o sistema e como este responde à interação do usuário, considerando as seguintes habilidades (norma ISO 9241):
 - **Facilidade de aprendizado:** a utilização do sistema requer pouco treinamento;
 - **Facilidade de memorização:** o usuário deve lembrar como utilizar a interface depois de algum tempo;
 - **Maximizar a produtividade:** a interface deve permitir que o usuário realize a tarefa de forma rápida e eficiente;
 - **Minimizar a taxa de erros:** caso aconteçam erros, a interface deve avisar o usuário e permitir a correção de modo fácil;
 - **Maximizar a satisfação do usuário:** a interface deve transmitir confiança e segurança.

Aspectos de projeto

- **Acessibilidade**

- Proporcionar o acesso a qualquer usuário, independentemente de eventuais necessidades especiais:
 - Pessoas com deficiências (congenitas, adquiridas ou temporárias);
 - Idosos;
 - Usuários inexperientes;
 - Usuários de dispositivos móveis.
- Sistemas que funcionem com tecnologias assistivas (hardware e/ou software)
- Web Accessibility Initiative (WAI) – Projeto dentro do W3C
 - Conjunto de recomendações para tornar a web acessível, inclusive quando acessada pelos dispositivos móveis.
 - <https://www.w3.org/WAI/>

Projeto de site/aplicações web

- **1.** Definir o público-alvo, requisitos e objetivos da aplicação, por meio da análise de documentos, processos, entrevistas com o cliente, etc.
- **2.** Definir a arquitetura da informação
 - Quais blocos de informação irão compor a estrutura: topo do site, rodapé, sistema de busca interno, menu de navegação, página de contato, home page, formulário de cadastro, página de impressão, enquetes e pesquisas, mapa de localização, mapa do site, nuvem de tags, paginação, dentre outros.
 - Mapa do site.
 - Wireframe (em papel ou em algum software específico).
- **3.** Definir o projeto gráfico
 - Tipografia, cores, tamanhos, imagens, animações.
- **4.** Projeto do *backend* (casos de uso, modelo conceitual, projeto do BD, etc)
- **5.** Desenvolvimento
 - Que tecnologia será usada: ASP.NET, PHP, MySQL, PostgreSQL, MongoDB, etc.
- **6.** Testes, hospedagem

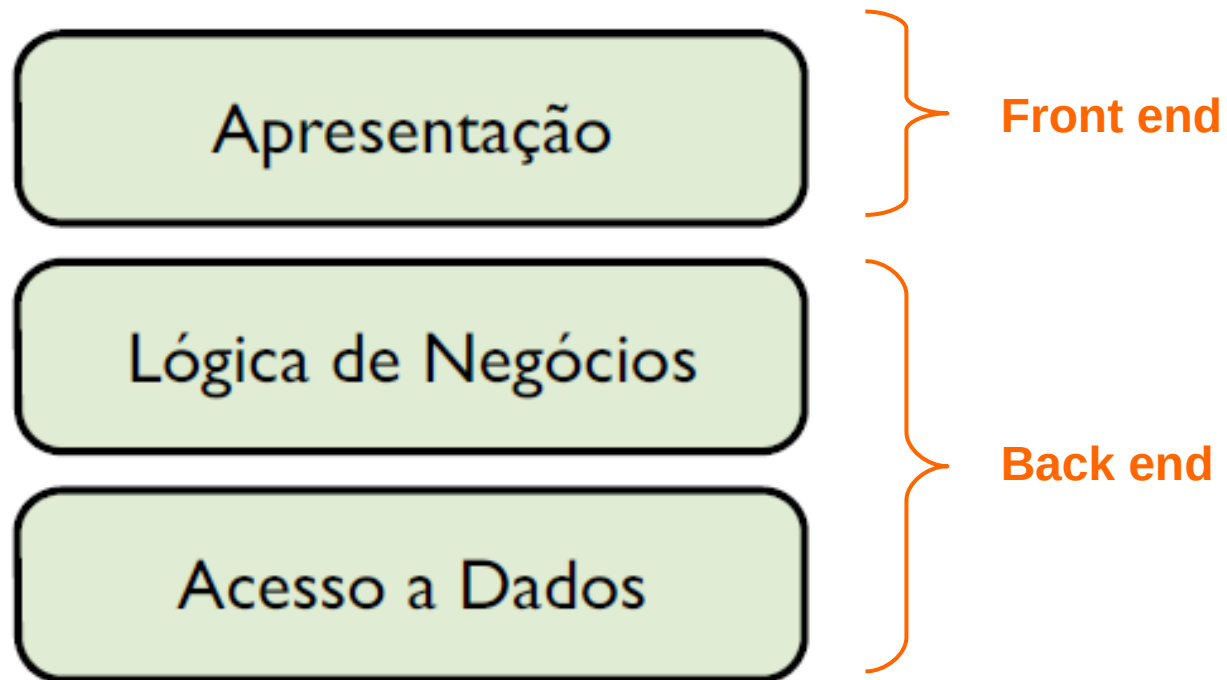
Front end e Back end

- **Front end:**
 - Linguagens/tecnologias interpretadas pelo navegador e que caracterizam as interfaces de um site ou aplicação web.
- **Back end:**
 - Linguagens/tecnologias que executam no servidor
 - As páginas possuem seções que mudam de acordo com os valores das variáveis fornecidas ao servidor, exibindo resultados de buscas, listas de itens cadastrados, etc.
- **Profissional desenvolvedor:**
 - **Pode ser desenvolvedor *front end*, *back end* ou ainda *full stack* (desenvolve tanto a interface quanto a parte no servidor)**



Front end vs. Back end.

Modelo de aplicação WEB



Algumas tecnologias existentes

Front end:

- HTML / HTML5
- CSS
- Javascript
- Bibliotecas e frameworks front end:
 - Bootstrap
 - Materialize
 - jQuery
 - Angular
 - React
 - Ember
 - Vue
 - ...

Back end:

- Linguagens:
 - C#
 - Java
 - PHP
 - Python
 - Perl
- Bibliotecas e frameworks back end:
 - Ruby on Rails
 - Node.JS
 - ASP.NET
 - PHP: Laravel, Zend, Symfony, Phalcon, CakePHP, CodeIgniter
 - ...

Banco de Dados (SGBDs):

- MySQL
- MariaDB
- SQLServer
- PostgreSQL
- MongoDB
- ...

Front end

- **HTML** (*Hypertext Markup Language*): linguagem de marcação utilizada para organizar o conteúdo e estrutura de um documento hipertexto, definindo o significado de cada uma das partes (cabeçalhos, parágrafos, listas de marcadores, etc).
 - Utiliza elementos chamados **tags** para identificar os diferentes componentes de uma página;
- **CSS** (*Cascading Style Sheets* – Folhas de estilos em cascata): controlam como os elementos HTML são estilizados e posicionados.
 - O uso de CSS garante a total separação entre conteúdo e formatação, algo altamente desejável;
- **JavaScript**: linguagem de programação derivada do C para a criação de pequenos programas (scripts).
 - Provê recursos interativos e dinâmicos que serão executados no cliente, sem necessitar de instalação de plugins, visto que é interpretada e executada pelo próprio navegador.

Back end

- **PHP** (Pré-Processador de Hipertexto)
 - Linguagem de programação para geração de documentos HTML dinâmicos, ou seja, o PHP consegue alterar o conteúdo de uma página, antes de enviá-la para o navegador.
 - Também permite capturar entradas de dados do usuário, como formulários e outras formas de interação.
 - Possui recursos nativos de conexão a diversos SGBDs.
 - O interpretador pode ser executado em várias plataformas como: Unix, Linux, Windows.
 - O PHP cruza características provenientes de Java, C++, PERL e C.
 - Possui um conjunto completo de funcionalidades de programação orientada a objetos, incluindo suporte à classes, classes abstratas, interfaces, herança, construtores, exceções e muito mais.

Ferramentas de apoio

- **Editores:**
 - Notepad++, Gedit, Geany
 - Sublime Text
 - Brackets
 - Eclipse (classic + pdt)
 - Netbeans
 - Dreamweaver (Adobe)
 - http://en.wikipedia.org/wiki/List_of_PHP_editors
- **Servidores HTTP:**
 - Apache, IIS (Microsoft), Lighttpd, Nginx, Cherokee...
- **Pacote XAMPP:**
 - Consiste na base de dados MySQL, servidor web Apache e interpretadores PHP e Perl
 - <http://www.apachefriends.org>
- **Cliente FTP (ou SFTP):**
 - WinSCP ou FireFTP
- **Editores gráficos:**
 - Adobe Illustrator
 - Photoshop
 - Gimp
- **Navegador:**
 - Firefox, Safari, Chrome, IE...
 - A maioria possui “Ferramentas do Desenvolvedor”
- **Administração do MySQL:**
 - MySQL Workbench
 - PHPMyAdmin

Leituras complementares

- O que é front-end e back-end:
<https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end/>
- Usabilidade: 10 regras para você desenvolver interfaces agradáveis aos usuários:
<http://thiagonasc.com/usabilidade/usabilidade-10-regras-desenvolver-interfaces-a-gradaveis>
- 11 erros comuns de Design na construção de sites:
<http://tutano.trampos.co/13832-11-erros-design-de-sites/>
- 18 Problemas de Usabilidade que Deixam Seus Leitores Furiosos (e Como Consertá-los): <https://viverdeblog.com/18-problemas-usabilidade/>
- Acessibilidade Legal: <http://acessibilidadelegal.com/>
- 10 dicas de como começar um design responsivo:
<http://designculture.com.br/10-dicas-de-como-comecar-um-design-responsivo>
- Criar um Site Responsivo com HTML5 e CSS3:
<http://www.carloshps.com.br/blog/criar-site-responsivo-com-html5-e-css3-parte-1-de-3/>