1. HTML 5

HTML5 (Hypertext Markup Language, versão 5) é uma linguagem para estruturação e apresentação de conteúdo para a World Wide Web e é uma tecnologia chave da Internet originalmente proposto por Opera Software.[1] É a quinta versão da linguagem HTML. Esta nova versão traz consigo importantes mudanças quanto ao papel do HTML no mundo da Web, através de novas funcionalidades como semântica e acessibilidade. Possibilita o uso de novos recursos antes possíveis apenas com a aplicação de outras tecnologias. Sua essência tem sido melhorar a linguagem com o suporte para as mais recentes multimídias, enquanto a mantém facilmente legível por seres humanos e consistentemente compreendida por computadores e outros dispositivos (navegadores, parsers etc).

2. CSS

Cascading Style Sheets (CSS) é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. O seu principal benefício é a separação entre o formato e o conteúdo de um documento.

Em vez de colocar a formatação dentro do documento, o desenvolvedor cria um link (ligação) para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um portal. Quando quiser alterar a aparência do portal basta portanto modificar apenas um arquivo.

3. O conceito de Tableless

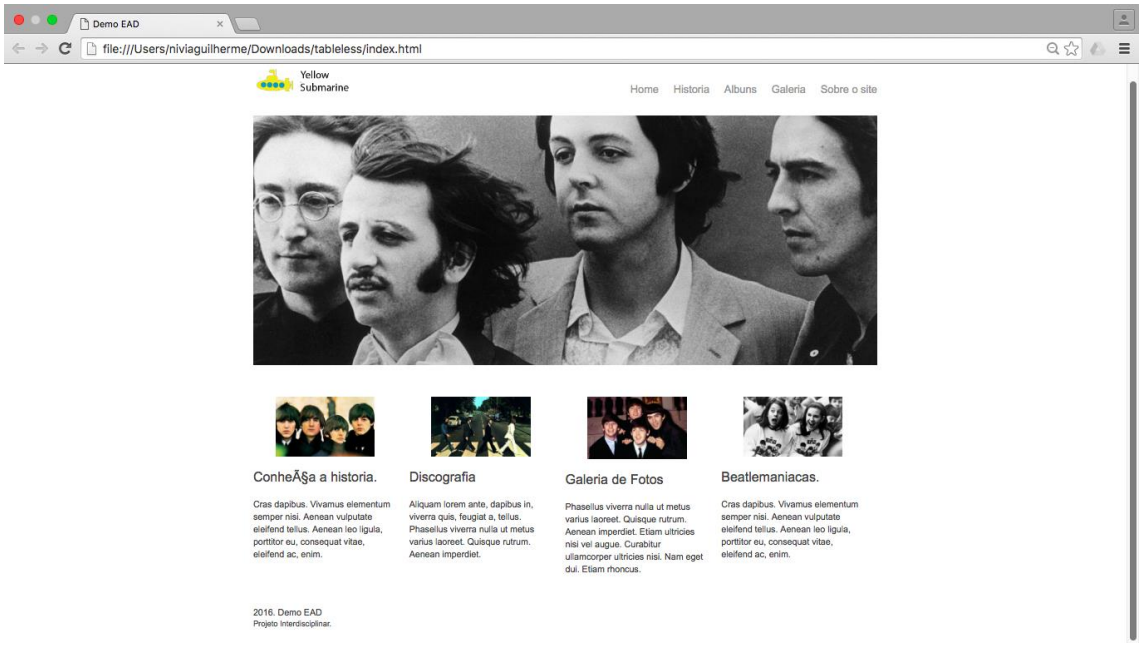
Tableless é uma forma de desenvolvimento de sites que não utiliza tabelas para disposição de conteúdo na página.

3.1. Vantagens

Adotar este padrão de desenvolvimento também facilita a separação da camada de apresentação da aplicação para o arquivo de estilo (CSS).
Diminuição de banda. Os navegadores modernos armazenam arquivos de CSS e de JavaScript em cache, se a maneira que o site será visualizado é guardado em um CSS (padrão tableless) então o arquivo será cacheado após o primeiro acesso e todos os acessos seguintes não carregarão este arquivo, carregarão apenas o conteúdo (texto) do site. Quando se usa tabela, a apresentação das tabelas (tags como "tr" e "td", gifs vazios, atributos como "cellspacing" e "border") são carregados todas as vezes que o usuário acessar o site.
Manutenção. Estando o estilo separado do arquivo html, facilita quando o desenvolvedor deseja mudar algo relacionado ao conteúdo, pois o conteúdo do site deixa de ficar oculto dentro de tabelas e subtabelas para estarem dentro de divs (caixas de conteúdo) que não trazem nenhuma informação de aparência. Caso ele deseje alterar o estilo, basta ir no arquivo CSS e não precisará procurá-lo entre códigos e conteúdos que não tem relação com a aparência e apresentação do site em geral.

4. O Projeto para esse tutorial

A ideia para esse tutorial é utilizar conceitos de HTML 5 e CSS para desenvolver a pagina inicial de do site abaixo.



5. Desenvolvendo o projeto

5.1. TAGS HTML utilizadas

As tags HTML que serão utilizadas para desenvolvimento são:

 - O nome div vem de divisão, e esta tag vai lhe dar o simples poder de dividir qualquer trecho de seu código. Isso mesmo, você pode criar um bloco, uma divisão, e dentro deste bloco ter uma imagem, links, textos e o que mais você quiser.

<header></header> - Como o próprio nome pode sugerir, ela que vai encabeçar uma região de seu site.

<nav></nav> - O elemento nav representa uma seção da página que contém link para outras páginas ou partes do mesmo website. Resumindo: nav é uma seção de links de navegação.

 - determina o inicio / fim de uma lista não numerada.

 - delimitando os elementos de uma lista não numerada.

<section></section> - Define um bloco ou um grupo de um assunto específico. É importante entender que a section agrupa diversos elementos que tenham relação entre si. Por exemplo, se há uma área no site que há links, conteúdo, imagens e etc de um assunto em comum, você agrupará esses elementos com uma section.

<figure></figure> - representa o conteúdo independente e é normalmente referido como uma única unidade. Enquanto ela está relacionada com o fluxo principal, sua posição é independente do fluxo principal. Normalmente, isso é uma imagem, uma ilustração, um diagrama, um trecho de código ou uma esquema que é referenciado no texto principal, mas que pode ser movido para outra página ou para um apêndice, sem afetar o fluxo principal.

<h2></h2> - sub-títulos de uma página.

 - delimitam parágrafos.

<footer></footer> - O elemento semântico <footer>, do HTML5, é análogo ao elemento <header>, mas este se refere ao rodapé, que geralmente fica abaixo, em seu site.

<small></small> - Define textos pequenos.

5.2. Iniciando o desenvolvimento.

O primeiro passo para o desenvolvimento é criar dois arquivos diferentes. O primeiro arquivo deve ser criado e salvo como index.html, o segundo arquivo deve ser salvo dentro de uma pasta chama css com o nome estilos.css.

O arquivo CSS serve para armazenar todas as configurações e formatações referentes à página HTML que foi criada primeiro.

Neste inicio vamos montar o CSS básico da página para ter uma ideia de como ele é prático. Podemos dizer que o CSS é a melhor metade do HTML. Codificando, não há melhor parceria: HTML é responsável pelo trabalho pesado (a estrutura), enquanto CSS dá o toque de elegância (layout).

O CSS pode ser utilizado de varias maneira: aplicado diretamente em uma tag HTML, na criação de uma classe de estilos ou na criação de um id de um elemento HTML. As normatizações básicas acontecerão em tags HTML

Abra o arquivo estilos.css. Nesse arquivo, começaremos definindo padrões para as tags <body> e <a>:

```
body {
    font-size: 62.5%;
    line-height: 1.5;
    font-family: "Arial";
    color: #444;
}
```

Para referenciar uma tag html basta cita-la da maneira original. Toda classe ou tag que receberá o CSS deverá ser aberta e fechada por chaves ({ ... }), onde ficará contida toda formatação.
Para a tag body ficou definido:

- font-size: o tamanho da fonte. Como medida pode ser usados: pixels (px), porcentagem (%), em (medida relativa) entre outros. A grande diferença de em e px é que em é uma medida relativa. O valor é calculado levando sempre em consideração o font-size do pai. Isso quer dizer que um elemento com font-size: 2em; vai ter o dobro do tamanho da fonte do pai, seja ele qual for.
A principio adotaremos a porcentagem, pensando na possibilidade do site ser acessado por dispositivos móveis.
- line-height: espaçamento entre linhas.
- font-family: a fonte escolhida como padrão para o site;
- color: cor da fonte, em hexadecimal, para atingir a cor exata;

```
a {
    color: #00ccff;
    text-decoration: none;
    font-size: 1.6em;
}

a:hover {
    text-decoration: underline;
}
```

O próximo passo é definir padrões para a tag a (links):
- color: cor da fonte, em hexadecimal, para atingir a cor exata;
- text-decoration: é a propriedade que permite ou não a linha abaixo do link. Nesse caso, none, quer dizer que a linha foi retirada.
- fonte-size: com medida em, quer dizer que em relação ao tamanho da fonte definida na tag body a tag a assumiu 1.6

A propriedade hover da tag a é acionada quando passamos o mouse em cima do link. Nesse caso, quando o mouse passar por cima do link o mesmo passará a exibir a linha abaixo do texto.

A sequência do projeto se dá com a preparação do arquivo html. Vamos, inicialmente, dividir o html da seguinte maneira:

A Tag inicial é uma div para definir o tamanho padrão máximo que o html atingirá na tela do computador. Dentro dessa div, encontraremos a tag header, onde estarão o menu e a logomarca, uma tag div para um banner, a tag section que demarcará o conteúdo inicial e a tag footer que demarca o rodapé do site.

Já no código abaixo, também é possível observar a tag link, colocada no <head> do html para referenciar o arquivo estilos.css, já criado, determinando assim, os padrões iniciais para a index.html.

```
<html>

<head>

  <title>Demo EAD</title>
        <link rel="stylesheet" href="css/estilos.css">

</head>

<body>

<div> <!—- div inicial -->

  <header >
     <!—- logomarca e menu  -->
  </header>

  <div >
    <!—- div para o banner -->
  </div>

  <section >
     <!—- conteúdo  -->
  </section>

  <footer >
     <!—- rodape -->
  </footer>

</div> <!—- fim da div inicial -->

</body>
</html>
```

5.3. Criando classes em CSS para as tags html

Para criar uma classe em CSS, basta inicia-la com um ponto “.” e batiza-la com o nome que desejar. No caso das tags principais foram montadas 4 classes: container, header, banner e footer. Todos esses estilos devem ser colocados no arquivos estilos.css, definido anteriormente.

A classe container está definida para div inicial, nela, podemos encontrar a propriedade max-width, que defini o tamanho máximo que o documento vai atingir em um navegador e a propriedade margin, que defini as margens que essa div assumirá. Da maneira como está configurada na

classe a propriedade assume margem para o topo como 0 pixel e o restante como automático para centralizar a div na tela quando necessário.
A classe header defini o tamanho que o cabeçalho do html assumirá (altura e largura), bem como suas margens superior de inferior. Também fica definido que para toda tag img a largura máxima é de 100%.

A classe banner define a imagem que será usada de fundo para div com a propriedade background, a altura que a div assumirá sua margem inferior e seu posicionamento em relação à do navegador. Posicionamento como relative, significa que a div irá se adequar de acordo com a resolução do dispositivo.

```
.container {
    max-width: 1128px;
    margin: 0 auto;
}

img{
    max-width: 100%;
}

.header {
    width: 100%;
    height: 48px;
    margin-top: 3.6em;
    margin-bottom: 3.6em;
}


.banner {
    background: url('../img/banner.jpg');
    height: 450px;
    margin-bottom: 4.8em;
    position: relative;
}

.footer {
    width: 100%;
    margin-top: 2.4em;
    margin-bottom: 2.4em;
    float: left;

}

.footer p {
    margin-bottom: 0;
}

.footer small {
    font-size: 1.4em;
}

.footer a {
    font-size: 1em;
```

}
Também ficou definida uma classe para a tag footer: a classe footer, nela, além das características básicas que a tag assumirá também é possível ver que as tags p, small e a assumirão características especificas. É possível definir características especificas para tags dentro de uma classe para isso bas seguir o padrão : .nomedaclasse tagdesejada { propriedades }. Uma propriedade ainda não citada, a propriedade float, é responsável por fazer flutuar todos os elementos da tag para a esquerda.

Na sequencia, podemos visualizar a maneira como uma classe CSS deve ser chamada. Dentro da tag em questão, pela propriedade class referenciamos a classe criada no arquivo CSS.

```
<div class=”container”> <!—- div inicial -->

  <header class=”header”>
    <!—- logomarca e menu  -->
  </header>

  <div class=”banner”>
    <!—- div para o banner -->
  </div>

  <section >
    <!—- conteúdo  -->
  </section>

  <footer class=”footer”>
    <!—- rodape -->
  </footer>

</div> <!—- fim da div inicial -->
```

5.4. Posicionando a logomarca e o menu

A ideia do menu e da logomarca dentro da tag header é a seguinte:

Para essa ideia, vamos criar uma classe logo e uma classe menu. A classe logo será aplicada na tag img e a classe menu será aplicada na tag nav, responsável por destacar o menu da pagina. Abaixo seguem os estilos aplicados as classes. Para a classe logo ficou definida a propriedade float para a esquerda da tag header e largura de 20%. Para a classe menu assumira 74% da largura da tag header e flutuará para a direita.

```
.logo {
   float: left;
   width: 20%;
}
```

```
.menu {
    width: 74%;
    float: right;
}
```

Para efetivamente montarmos o menu é preciso a ajuda das tag <ul> . A utilização delas é muito comum. A propriedade mais importante a ser destacada no conjunto de propriedades dessas duas tags é a propriedade display elencada na tag <li>. Por padrão uma lista definida por é exibida verticalmente. Na propriedade display, quando inicializada com inline-block exibirá a lista na horizontal, permitindo que o menu seja exibido conforme a figura inicial do projeto.

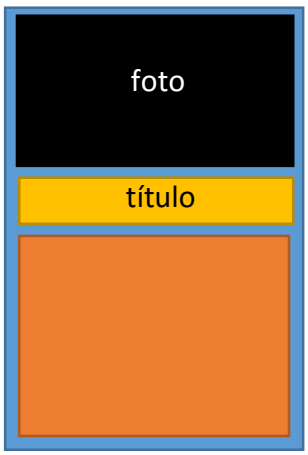
```
.menu ul {
    float: right;
}

.menu li {
    font-size: 1.2em;
    margin-left: 2em;
    margin-top: 1em;
    display: inline-block;
}

.menu li a {
    color: #999;
}
```

Por fim, seguem o código html para a montagem do menu dentro da tag header.

```
<header class="header">
      <img src="img/logo.png" class="logo">
      <nav class="menu">
        <ul>
          <li><a href="#">Home</a></li>
          <li><a href="#">Historia</a></li>
          <li><a href="#">Albuns</a></li>
          <li><a href="#">Galeria</a></li>
          <li><a href="#">Sobre o site</a></li>
        </ul>
      </nav>
  </header>
```

5.5. Criando estilos para o conteúdo

O conteúdo para a tag section ficará da seguinte forma:



Vamos criar uma classe coluna que engloba todos os outros elementos da notícia. Nessa classe coluna, serão configuradas as propriedades: largura (width) com 23%, float (flutuar) como left e margin-right (margem direita) com 2%. Assim para cada nova coluna gerada, teremos largura e espaçamento corretos, além de cada elemento obedecer a orientação de se alinhar à esquerda da tag section.

```
.coluna {
    width: 23%;
    float: left;
    margin-right: 2%;
}

.coluna:last-child {
    margin-right: 0;
}
```

A propriedade last-child diz respeito à ultima div criada em uma cadeia de divs com a classe coluna. Para esse último caso, a margem direita fica definida como 0 para a mesma se alinhe diretamente à tag section.

Na sequência, o CSS será aplicado às tags h2 e p. Essas tags servirão de padrão para o título e texto da notícia que ficará dentro da div de classe coluna

```
h2 {
    font-size: 2.4em;
    font-weight: 200;
}

p {
    font-size: 1.6em;
}
```

O html para essa noticia ficará conforme definido abaixo:

```
<div class="coluna">
        <figure><img src="img/foto-04.jpg"></figure>
        <h2>Beatlemaniacas.</h2>
        <p>Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend
tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim.</p>
      </div>
```

A tag figure entra no contexto para separar o elemento <img> dos outros elementos.

Por fim, para o rodapé temos:

```
  <footer class="footer">
     <p>2016. Demo EAD</p>
     <small class="creditos">Projeto Interdisciplinar. </small>
  </footer>
```