Parte superior do formulário

Parte inferior do formulário

**Front-end**

**Projeto de conclusão**

**Faça esse curso de HTML e CSS e:**

* Pratique o que aprendeu na formação Front-end
* Crie do zero um layout proposto
* Utilize recursos do CSS3 para aplicar um visual fidedigno
* Pratique conceitos de design responsivo
* Utilize também JavaScript e jQuery

**Aulas**

[Introdução e construção do cabeçalho](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5257/tasks)

[0 / 1242min](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5257/tasks)

* + Introdução
  + Download do projeto (Obrigatório)
  + Orientação geral
  + Estrutura do menu
  + Gabarito
  + Estilo do menu
  + Gabarito
  + Estrutura do painel principal
  + Gabarito
  + Estilo do painel principal
  + Gabarito - parte1
  + Gabarito - parte2

[Painel sobre](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5259/tasks)

[0 / 413min](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5259/tasks)

* + Estrutura do painel
  + Gabarito
  + Estilo do painel
  + Gabarito

[Painel palestrantes](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5260/tasks)

[0 / 527min](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5260/tasks)

* + Estrutura do painel
  + Gabarito
  + Estilo do painel
  + Gabarito - parte 1
  + Gabarito - parte 2

[Formulário de inscrição](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5317/tasks)

[0 / 620min](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5317/tasks)

* + Estrutura do formulário
  + Gabarito
  + Estilo do formulário
  + Gabarito - parte 1
  + Gabarito - parte 2
  + Gabarito - parte 3

[Dotando a página de comportamento](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5262/tasks)

[0 / 411min](https://cursos.alura.com.br/course/projeto-carreira-front-end/section/5262/tasks)

* + Comportamento
  + Gabarito
  + Considerações finais
  + Download do projeto final

# 01Introdução

Seja bem-vindo! Meu nome é Flávio Almeida (@flaviohalmeida) e muito provavelmente já nos encontramos em outros cursos da Alura. Desta vez, estou aqui para propor um desafio. Você gosta de desafio?

O desafio é a construção de uma página do zero utilizando todo o seu conhecimento adquirido na carreira de [desenvolvedor front-end](https://www.alura.com.br/carreira-desenvolvedor-front-end).

**Um desafio**

Para que o desafio realmente exista, adotaremos um fluxo um tanto diferente. Você primeiro receberá as instruções do que deve ser feito e o layout que deve atingir para só depois consultar opcionalmente a solução do instrutor. Caberá a você decidir o quanto se dedicará ao problema a ser resolvido antes de consultar o gabarito. Aliás, a solução do instrutor é uma entre as dezenas de soluções possíveis e a melhor solução será aquela desenvolvida por você! Vai encarar?

**Colocando em prática seu conhecimento adquirido**

A boa notícia é que você pode recorrer a qualquer momento aos cursos que compões a carreira [desenvolvedor front-end](https://www.alura.com.br/carreira-desenvolvedor-front-end). Na data de criação deste treinamento, são eles:

* [Curso HTML5 e CSS3 I: Suas primeiras páginas da Web](https://www.alura.com.br/curso-online-introducao-html-css)
* [HTML5 e CSS3 II: Turbinando as suas páginas](https://www.alura.com.br/curso-online-avancando-html-css)
* [Curso JavaScript: Programando na linguagem da web](https://www.alura.com.br/curso-online-javascript-programando-na-linguagem-web)
* [Curso jQuery: Domine a biblioteca mais popular do mercado](https://www.alura.com.br/curso-online-jquery-a-biblioteca-do-mercado)
* [Curso jQuery: Avance na biblioteca mais popular do mercado parte 2](https://www.alura.com.br/curso-online-jquery-a-biblioteca-do-mercado-parte-2)
* [Web Design Responsivo: Páginas que se adaptam do mobile ao desk](https://www.alura.com.br/curso-online-web-design-responsivo)

Nossas carreiras evoluem e não se preocupe se novos cursos forem introduzidos na carreira, você só precisará focar nos que foram listados anteriormente para concluir o desafio. E falando em desafio, que desafio é esse?

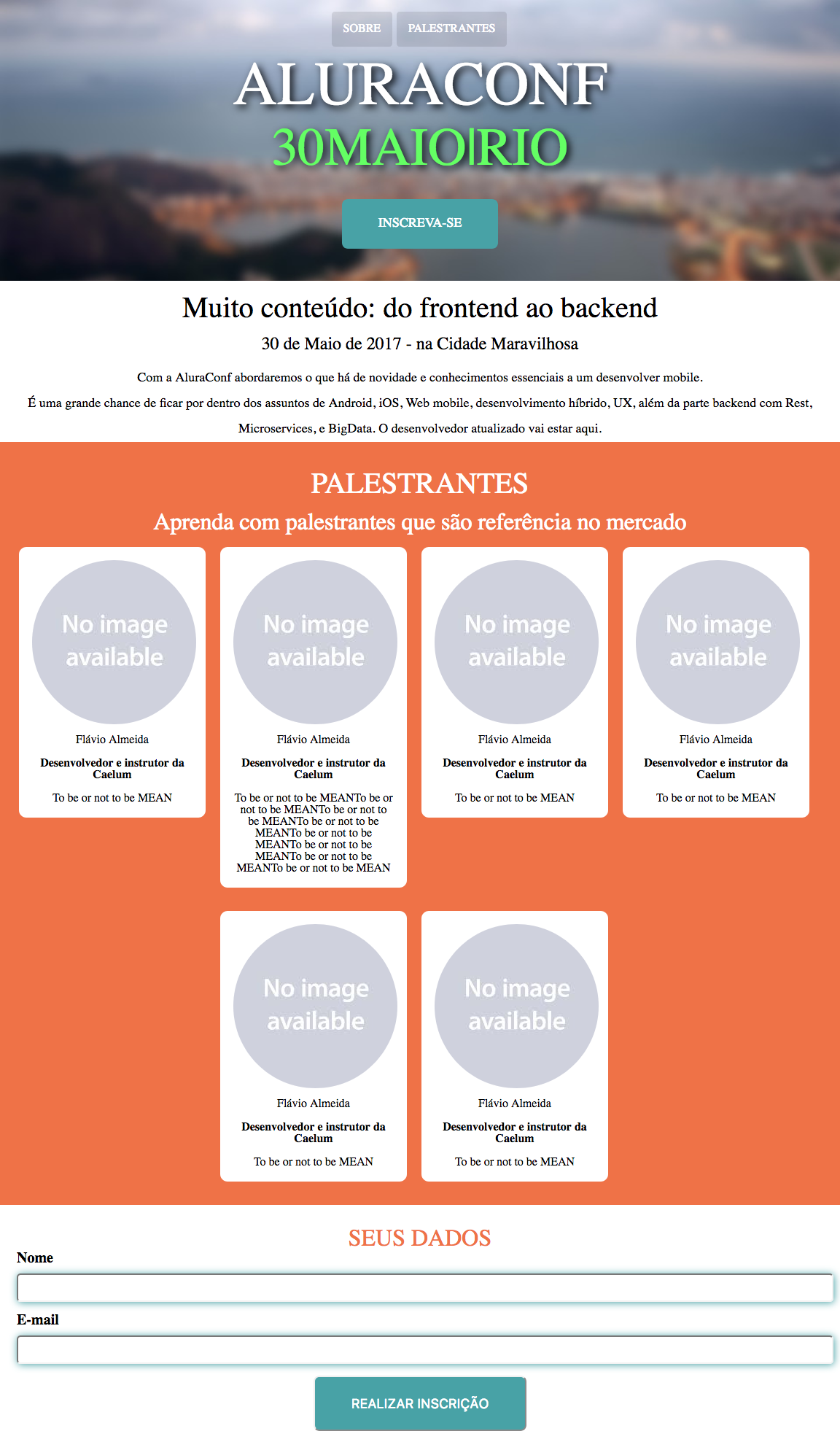
**O projeto**

Nós criaremos uma página de conferência, aquela que exibe informações do evento, palestrantes, formulário de contato entre outras informações. Em um primeiro momento, pode lhe parecer que o escopo do projeto é pequeno, mas você verá rapidamente sentirá na pele que precisará da bagagem adquirida na carreira [desenvolvedor front-end](https://www.alura.com.br/carreira-desenvolvedor-front-end). Além disso, sites de conferência precisam ser bem-feitos como qualquer outro site, ainda mais tendo um tempo de vida curto no qual a primeira visita é importante.

Por fim, é importante deixar claro que o projeto não entra em detalhes de implementação de API's ou lado do servidor, pois esses conhecimentos fazem parte de outras trilhas da Alura e, como vimos, a proposta é você aplicar seu arsenal de conhecimento adquirido dentro da carreira [desenvolvedor front-end](https://www.alura.com.br/carreira-desenvolvedor-front-end).

**Visualizando o resultado final**

Temos aqui o produto final do desafio.



Agora que você já sabe até onde chegaremos, faça o primeiro exercício obrigatório do capítulo. Nele você baixará o projeto com as imagens e os scripts necessários que serão utilizados durante o curso. Uma maneira cômoda de você ter que baixá-los.

Sucesso e bom estudo, meu alunos.

# 02Download do projeto (Obrigatório)

Baixe o projeto que utilizaremos [neste](https://s3.amazonaws.com/caelum-online-public/carreira-front/stages/01-aluraconf.zip) link. Ele possui apenas as imagens que utilizaremos, um CSS Reset e o jQuery para podermos dar início ao nosso projeto.

# 03Orientação geral

Você está prestes a dar início com a construção do projeto. Há liberdade suficiente para que você siga o caminho que achar mais adequado para atingir o layout alvo. Contudo, **há um padrão que você deve seguir** para estar em conformidade com os capítulos deste treinamento.

Nosso layout será dividido em grandes partes, são elas:

* 1 - Cabeçalho
* 2 - Painel Sobre
* 3 - Painel Palestrantes
* 4 - Formulário de inscrição

Cada capítulo tratará de cada parte individualmente, iniciando com a estrutura HTML para em seguida definir o estilo da parte na qual estamos trabalhando.

Em cada capítulo, haverá um briefing da estrutura e do estilo a serem atingidos, com orientações gerais do instrutor. Por fim, o aluno terá acesso ao gabarito, tanto da estrutura quando do CSS. É importante deixar claro que o gabarito é opcional e pode ser consultado a qualquer momento se assim o aluno desejar. Porém, é importante que ele tente resolver primeiro o problema e só depois compare seu resultado com o do instrutor, aliás, não será incomum que haja divergência entre os layouts, porém o importante é que a implementação do aluno esteja próxima ao layout esperado.

Por fim, a parte comportamental da página responsável pelas interações ficará por último, ou seja, toda a parte que envolve JavaScript. A razão disso é que alguns efeitos que aplicaremos precisam do HTML como um todo.

No próximo vídeo teremos um briefing da estrutura HTML do cabeçalho a ser construída.

# 04Estrutura do menu

A estrutura do cabeçalho da página é constituída de um menu e um painel principal que possui a chamada do evento. É importante estar atento para:

* Questão de semântica
* O menu é de navegação primária do site

Mãos na massa! Depois de terminar, você pode conferir a gabarito com o instrutor. O importante é verificar se a semântica da sua marcação está bem feita, pois o layout só conseguiremos verificar depois da aplicação dos estilos.

# 05Gabarito

Nosso primeiro passo será criarmos o arquivo aluraconf/index.html com a estrutura mínima HTML para que possamos dar início a construção do cabeçalho da página:

<!-- aluraconf/index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

<title>Aluraconf</title>

</head>

<body>

</body>

</html>COPIAR CÓDIGO

Não podemos nos esquecer do <meta charset="UTF-8">. Lembre-se que essa TAG dá uma pista para o navegador de qual enconding foi usado para criar nosso arquivo no sistema de arquivos do sistema operacional. Geralmente, os arquivos são criados em UTF-8. Dessa forma, deve haver uma paridade entre o enconding usado pelo arquivo e o atributo charset da tag <meta>. Algunas navegadores conseguem já inferir o charset correto, contudo para garantirmos maior compatibilidade, vamos deixar explícito o charset utilizado. A tag <meta name="viewport"> também é importante, para que nossas medias queries levem em consideração o viewport do dispositivo e não a sua resolução.

O HTML5 introduziu a tag <header>. Esta tag tem a mesma funcionalidade de uma <div>, contudo ela é uma tag semântica indicando para leitores de acessibilidade, inclusive para buscadores que todo o conteúdo que existir dentro dela faz parte do cabeçalho da página.

<!-- aluraconf/index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

<title>Aluraconf</title>

</head>

<body>

<header>

</header>

</body>

</html>COPIAR CÓDIGO

Vamos agora criar o menu da aplicação. Aliás, é uma lista de itens de menu e os itens dessa lista não parecem ter uma ordem evidente. É por isso que usaremos uma <ul> com duas tags <li>. Cada uma dessas tags terá um âncora de navegação interna que aponte para os painéis "about" e "speakers" respectivamente. Todavia, como não temos esses painéis ainda, vamos por enquanto utilizar um#`, que sempre fará o scroll para o topo da página quando clicado. que por enquanto não apontará para lugar nenhum, mas uma coisa é certa, elas não serão links externos, serão links internos que apontarão para a própria página:

<!-- aluraconf/index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

<title>Aluraconf</title>

</head>

<body>

<header>

<ul>

<li>

<a href="#">Sobre</a>

</li>

<li>

<a href="#">Palestrantes</a>

</li>

</ul>

</header>

</body>

</html>COPIAR CÓDIGO

Podemos ainda inserir ul como filha da tag nav. Esta tag do HTML indica um grupo de links de navegação primária do site, isto é, aqueles que navegam para dentro do próprio site ou dentro da própria página.

<!-- aluraconf/index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

<title>Aluraconf</title>

</head>

<body>

<header>

<nav>

<ul>

<li>

<a href="#">Sobre</a>

</li>

<li>

<a href="#">Palestrantes</a>

</li>

</ul>

</nav>

</header>

</body>

</html>COPIAR CÓDIGO

# 06Estilo do menu

Em dispositivos mobile, o menu da nossa aplicação deverá ter um visual parecido como este:



Já em telas maiores, Desktop, seu visual deverá ser assim:



Não se preocupe em ser milimetricamente fiel ao modelo apresentado, o importante é que o conceito se mantenha. Porém, há alguns requisitos que você deve implementar.

**Atenção, detalhes na implementação**

1) Quando o mouse passar por cima dos itens de menu, o fundo deve mudar e seguir a cor conforme imagem abaixo:



2) A mudança da cor do nemu deve ser feita através de uma transição de meio segundo, inclusive quando o mouse for removido do item.

**Dica para encontrar as cores de um layout já existente**

Quando recebemos um PDF ou qualquer outro documento que tenha o layout que precisamos implementar geralmente recebemos a paleta de cores utilizadas. Porém, quando isso não acontece, precisamos arrumar uma forma de identificar as cores do layout.

Se você não tem Photoshop ou outra ferramenta, você pode usar o serviço gratuito http://html-color-codes.info/colors-from-image/. **Sendo um serviço de terceiros, verifique se ele continua existindo na data na qual você está iniciando o curso**. Nesse serviço podemos fazer upload de uma imagem. Com a imagem carregada, podemos descobrir em hexadecimal a cor de qualquer área bastando clicar na área desejada.

Experimente esse serviço para tentar descobrir a cor de fundo do item de menu com o mouse sobre ele. A cor que utilizarei é #48A3A6. Nem sempre a cor é exatamente a que foi usada. Faça um teste, verifique o quanto próxima ela é de #48A3A6.

# 07Gabarito

Aprendemos no curso que é uma boa prática utilizar um arquivo de CSS RESET para garantirmos que todos os navegadores utilizarão os mesmos valores base para as propriedades resetadas. É por isso que já importaremos o arquivo css/reset.css como primeiro CSS da nossa página.

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Aluraconf</title>

<link rel="stylesheet" href="css/reset.css">

</head>

<!-- código posterior omitido>COPIAR CÓDIGO

O CSS Reset utilizado foi <https://meyerweb.com/eric/tools/css/reset/>

Para cada grande área da nossa estrutura, teremos seu respectivo CSS. Essa estratégia ajuda na manutenção do nosso código. É por isso que criaremos o arquivo aluraconf/css/header.css.

Como arquivo criado, sem pestanejarmos, vamos importá-lo imediatamente no head de index.html:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Aluraconf</title>

<link rel="stylesheet" href="css/reset.css">

<link rel="stylesheet" href="css/header.css">

</head>

<!-- código posterior omitido -->COPIAR CÓDIGO

A primeira coisa que faremos é criar a classe header. Essa classe aplicada a imagem destaque.jpeg como fundo de todo o header.

/\* css/header.css \*/

.header {

background: url('../img/destaque.jpeg') no-repeat;

background-size: cover;

padding-top: 32px;

}COPIAR CÓDIGO

Usamos background-size: cover para que a imagem cresça o máximo possível para que toda a área do background seja preenchida. Quando estivermos em um dispositivo mobile, de menor largura, a imagem não será totalmente exibida e não há problema algum nisso, queremos apenas um fundo. Se aumentarmos a largura da tela gradativamente a imagem será grande o suficiente para ocupar toda 100% da área, isto porque o padrão do header, um elelmento block, é ocupar 100% da área disponível na tela.

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

<link rel="stylesheet" href="css/reset.css">

<link rel="stylesheet" href="css/header.css">

<title>Aluraconf</title>

</head>

<body>

<header class="header">

<nav>

<ul>

<li>

<a href="#">Sobre</a>

</li>

<li>

<a href="#">Palestrantes</a>

</li>

</ul>

</nav>

</header>

</body>

</html>COPIAR CÓDIGO

Agora vamos centralizar o menu da aplicação através da classe main-meu. Desta vez, antes de criar a classe, já vamos posicioná-la no elemento correto dentro do HTML:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Aluraconf</title>

<link rel="stylesheet" href="css/reset.css">

<link rel="stylesheet" href="css/header.css">

</head>

<body>

<header class="header">

<nav class="main-menu">

<ul>

<li>

<a href="#">Sobre</a>

</li>

<li>

<a href="#">Palestrantes</a>

</li>

</ul>

</nav>

<!-- código posterior omitido --> COPIAR CÓDIGO

Agora nosso CSS:

/\* css/header.css \*/

/\* CSS anterior omitido \*/

.header .main-menu {

text-align: center;

}COPIAR CÓDIGO

Nosso menu principal está centralizado, mas como as tags <li> possuem display block uma fica abaixo da outra. Podemos resolver isso facilmente mudando seu display para inline-block. Não podemos usar inline, pois ainda precisaremos alterar mudar as margens de cada <li> e apenas elementos block ou inline-block permitem isso:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Aluraconf</title>

<link rel="stylesheet" href="css/reset.css">

<link rel="stylesheet" href="css/header.css">

</head>

<body>

<header class="header">

<nav class="main-menu">

<ul>

<li class="menu-item">

<a href="#">Sobre</a>

</li>

<li class="menu-item">

<a href="#">Palestrantes</a>

</li>

</ul>

</nav>

<!-- código posterior omitido --> COPIAR CÓDIGO

Nosso CSS até agora fica assim:

/\* css/header.css \*/

.header {

background: url('../img/destaque.jpeg') no-repeat;

background-size: cover;

padding-top: 32px;

}

.header .main-menu {

text-align: center;

}

.header .main-menu .menu-item {

display: inline-block;

margin: 0 2px 32px 0;

}COPIAR CÓDIGO

Quando passamos 4 parâmetros estamos passando os valores de margin-top, margin-right, margin-botton e margin-left respectivamente.

Agora, vamos estilar os itens de menu. A cor da letra será branca e o fundo, preto praticamente transparente. Podemos usar rgba para conseguir a transparência do fundo apenas. Ele também terá a borda um pouquinho aredondada e 16px de espaçamento entre o conteúdo e a borda. Não queremos cara de link por isso usamos text-decoration: none. Por fim, queremos garantir que o item do menu esteja sempre em caixa alta, sem depender que conteúdo tenha sido digitado neste formato. Fazemos isso através de text-transform: uppercase:

/\* código anterior omitido \*/

.header .main-menu .menu-item a {

color: white;

padding: 16px;

background: rgba(255, 255, 255, 0.1);

border-radius: 4px;

text-decoration: none;

text-transform: uppercase;

}COPIAR CÓDIGO

Queremos ainda que ao passar o mouse sobre o item do menu (um efeito mais voltado para Destkop e não mobile), o menu fique como fundo na cor #48A3A6 e a cor da letra branca:

/\* código anterior omitido \*/

.header .main-menu .menu-item a:hover {

background-color: #48A3A6;

}COPIAR CÓDIGO

Excelente, mas para deixar ainda melhor a experiência do usuário, vamos aplicar uma transição para a cor do fundo aplicada pelo efeito hover. Queremos uma transição também na volta, por isso pasta adicionarmos a propriedade transition: background-color 0.5s no seletor que afeta a âncora.

/\* código anterior omitido \*/

/\* modificando o seletor já existente \*/

.header .main-menu .menu-item a {

color: white;

padding: 16px;

background: rgba(0, 0, 0, 0.1);

border-radius: 4px;

text-transform: uppercase;

text-decoration: none;

transition: background-color 0.5s;

}

.header .main-menu .menu-item a:hover {

background-color: #48A3A6;

}COPIAR CÓDIGO

Excelente. Agora podemos ir para o painel!

# 08Estrutura do painel principal

Vejamos a estrutura com estilo aplicado.



Apesar de estarmos visualizando a estrutura com estilo, nosso foco agora será única e exclusivamente a estrutura. Seguem o briefing.

O painel principal deve fazer parte do header da página. Como o nome do evento é importante, ele deve ter grande relevância semântica, já o subtítulo do evento uma relevância menor.

**Uma exigência** é que o botão que exibido no layout seja uma âncora. Isso será necessário, porque mais tarde ela fará um link interno para um formulário de cadastro. O restante é com vocês.

# 09Gabarito

Ainda dentro do header da página, temos um painel de destaque e o de inscrição. No caso, os dois ficarão dentro do cabeçalho da página porque a finalidade da tag <header> é conter um conteúdo introdutório e uma série de links de navegação:

<!-- aluraconf/index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

<title>Aluraconf</title>

</head>

<body>

<header class="header">

<nav class="main-menu">

<ul>

<li class="menu-item">

<a href="#">Sobre</a>

</li>

<li class="menu-item">

<a href="#">Palestrantes</a>

</li>

</ul>

</nav>

<div>

<section>

<h1>ALURACONF</h1>

<h2>30MAIO|RIO</h2>

</section>

<section>

<a href="#" class="button">Inscreva-se</a>

</section>

</div>

</header>

</body>

</html>COPIAR CÓDIGO

Como dentro do header temos duas seções distintas, usamos a tag section. Ela serve para dar um contexto a um conjunto de elementos e pode ser uma tag de destaque ou não. Na primeira seção, usamos h1 para destacar o nome do evento, já na segunda temos apenas uma lista de navegação.

Com isso temos a estrutura do header pronta.

# 10Estilo do painel principal

Em dispositivos mobile, nosso header completo deverá estar parecido com:



Já em tela maiores como Desktop:



Não se preocupe em ser milimetricamente fiel ao modelo apresentado, o importante é que o conceito se mantenha.

**Há uma exigência aqui**. Apenas em telas maiores (o feeling do tamanho é com vocês!), o tamanho da fonte dos elementos do painel principal deve aumentar um pouquinho para que ocupem um espaço maior na tela.

# 11Gabarito - parte1

Agora que já temos nosso menu, vamos para a área principal. Queremos que tanto painel principal quanto o painel de inscrição fiquem em destaque. Podemos inclusive até homenagear o Bootstrap criando nossa própria classe jumbotron.

Primeiro, vamos adicionar a classe na div que possuem dos dois painéis:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Aluraconf</title>

<link rel="stylesheet" href="css/reset.css">

<link rel="stylesheet" href="css/header.css">

</head>

<body>

<header class="header">

<nav class="main-menu">

<ul>

<li class="menu-item">

<a href="#">Sobre</a>

</li>

<li class="menu-item">

<a href="#">Palestrantes</a>

</li>

</ul>

</nav>

<!-- nova classe aqui -->

<div class="jumbotron">

<section>

<h1>ALURACONF</h1>

<h2>30MAIO|RIO</h2>

</section>

<section>

<a href="#" class="button">Inscreva-se</a>

</section>

</div>

</header>

</body>

</html>COPIAR CÓDIGO

Nossa classe removerá todos os paddings da div, deixando apenas um padding-botton de 44px, alinhará todos os seus elementos pelo centro e garantindo que todos estejam em maiúcsula:

/\* css/header.css \*/

/\* código anterior omitido \*/

.header .jumbotron {

padding: 0 0 44px 0;

text-align: center;

text-transform: uppercase;

}COPIAR CÓDIGO

Agora, cada painel dentro do nosso jumbotron terá um estilo diferente. Vamos criar primeiro a classe painel-main.

Adicionando-a no HTML:

<!-- código anterior omitido -->

<div class="jumbotron">

<section class="panel-main">

<h1>ALURACONF</h1>

<h2>30MAIO|RIO</h2>

</section>

<!-- código posterior omitido -->COPIAR CÓDIGO

A cor da letra dos elementos do nosso painel será white, inclusive os elementos terão uma sobra de texto posicionada à 4px 4px, blur de 10px e cor black:

/\* código anterior omitido \*/

.header .panel-main {

color: white;

text-shadow: 4px 4px 10px black;

}COPIAR CÓDIGO

Agora precisamos realizar alterações no título e no subtítulo. Para isso, vamos criar a classe .title e .subtitle:

/\* código anterior omitido \*/

.header .panel-main .title {

font-size: 48px;

}

.header .panel-main .subtitle {

color: #64ff64;

font-size: 42px;

margin-top: 8px;

}COPIAR CÓDIGO

# 12Gabarito - parte2

Agora, vamos aplicar um estilo na âncora da inscrição, fazendo-a parecer com um botão:

/\* código anterior omitido \*/

.header .jumbotron .button {

display: inline-block;

padding: 25px 50px;

margin: 5px;

background-color: #48A3A6;

border-radius: 8px;

box-shadow: 1px 1px 5px black;

text-decoration: none;

font-size: 18px;

font-weight: bold;

color: white;

}COPIAR CÓDIGO

O botão ficou muito bolado com o painel-main. Podemos resolver isso facilmente criando a classe panel-subscribe, que fará um margin-top de 28px:

/\* código anterior omitido \*/

.header .panel-subscribe {

margin-top: 28px;

}COPIAR CÓDIGO

Não podemos nos esquecer de adicionar no HTML:

<!-- código anterior omitido -->

<div class="jumbotron">

<section class="panel-main">

<h1 class="title">ALURACONF</h1>

<h2 class="subtitle">30MAIO|RIO</h2>

</section>

<section class="panel-subscribe">

<a href="#" class="button">Inscreva-se</a>

</section>

</div>

<!-- código posterior omitido -->COPIAR CÓDIGO

Tudo muito bonito, mas ainda não resolvemos a questão de aumentar o tamanho da fonte dos elementos do painel de destaque. Podemos resolver isso através de uma media query levando em consideração o breakpoint 650 para realizar o ajuste no tamanho das fontes. Pode ser que o bom senso de outro programador entenda que esse valor do breakpoint seja maior ou menor, mas esse valor é bom tamanho que na hora deste instrutor criar o design achou adequado.

No final de header.css temos a media query que aumenta o tamanho da fonte do jumbrotron para 28px:

/\* header.css \*/

/\* código anterior omitido \*/

@media(min-width: 650px) {

.header .jumbotron {

font-size: 28px;

}

}COPIAR CÓDIGO

Porém, essa solução apenas não resolve. Primeiro, o font-size não é passado através de cascata para os elementos filhos. Segundo, mesmo que fosse passado em cascata, não faria sentido aplicar esse valor porque o title e o subtitle possuem tamanhos diferentes. Então, qual a razão dessa manobra no CSS? Para que ela seja completa, precisaremos trabalhar com um tamanho de fonte relativa em title e subtitle, trabalharemos com em.

O valor de 1em é o valor do font-size do elemento pai. Aliás, o padrão do tamanho do font-size de um elemento que não modificamos é 16px. Sendo assim, hoje nosso title possui 48px, que é a mesma coisa que 3em. Já o subtitle equivale a 2.6em.

Modificando de pixel para em:

/\* header.css \*/

/\* código anterior omitido \*/

.header .jumbotron .title {

font-size: 3em; /\* alteração aqui \*/

}

.header .jumbotron .subtitle {

font-size: 2.6em; /\* alteração aqui \*/

color: #64FF64;

margin-top: 8px;

}

/\* código posterior omitido \*/COPIAR CÓDIGO

Perfeito. Quando nossa media query não está ativa, os elementos se basearão no tamanho do elemento pai que possui 16px por padrão, mas quando a media query for aplicada, o elemento pai terá um acrescimento ao seu tamanho, fazendo com que o title e o subtitle aumentem.

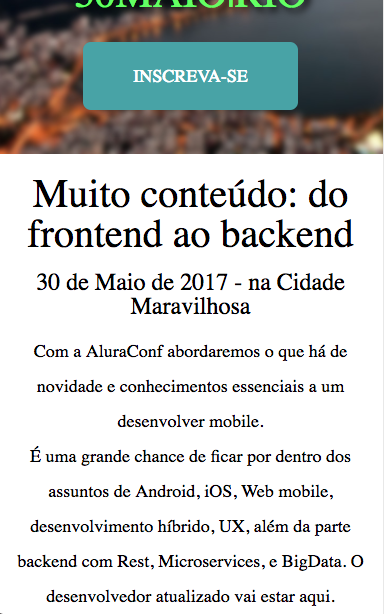
Outro solução definir o tamanho do title e do subtitle diretamente na media query, porém o uso da media relativa nos fez escrever menos CSS e ainda facilitou a manutenção. Se mais tarde não acharmos que o valor base de 28px não é mais suficiente, aumentamos em um único lugar que o tamanho do título e subtítulo respeitarão.

# 01Estrutura do painel

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/carreira-front/stages/02-aluraconf.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Vejamos a estrutura já estilizada:

**versão mobile**



**versão desktop**



Não há muito o que dizer aqui. Use o bom senso e pense na semântica ao criar a estrutura do painel.

# 02Gabarito

Vamos começar adicionando a estrutura do painel. Vou aproveitar e já adicionar as classes que utilizaremos no estilo:

<!-- código anterior omitido

<section class="panel panel-about">

</section>COPIAR CÓDIGO

O painel terá um título e um subtítulo. Como já usamos um <h1> em nossa página, utilizaremos um h2:

<section class="panel panel-about">

<h2 class="title">Muito conteúdo: do frontend ao backend</h2>

<h3 class="subtitle">30 de Maio de 2017 - na Cidade Maravilhosa</h3>

</section>COPIAR CÓDIGO

Por fim, temos dois parâmetros do painel com informações complementares.

<section class="panel panel-about">

<h2 class="title">Muito conteúdo: do frontend ao backend</h2>

<h3 class="subtitle">30 de Maio de 2017 - na Cidade Maravilhosa</h3>

<div class="about-description">

<p>

Com a AluraConf abordaremos o que há de novidade e conhecimentos essenciais a um desenvolver mobile.

</p>

<p>

É uma grande chance de ficar por dentro dos assuntos de Android, iOS, Web mobile, desenvolvimento híbrido, UX, além da

parte backend com Rest, Microservices, e BigData. O desenvolvedor atualizado vai estar aqui.

</p>

</div>

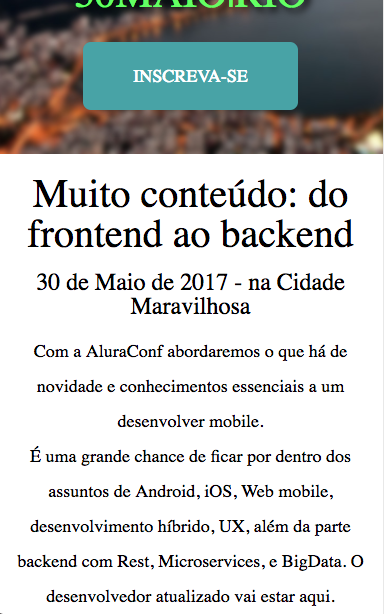
</section>COPIAR CÓDIGO

Excelente. Agora que temos a estrutura, podemos dar início a criação do estilo.

# 03Estilo do painel

Um ponto a destacar é o espaçamento entre as linhas dos parágrafos e a margem esquerda e direita do conteúdo do painel. Do restante, siga seu feeling e tente e aproximar ao máximo do layout apresentado.

**versão mobile**



**versão desktop**



Mãos na massa!

# 04Gabarito

Nosso painel precisa ficar um pouco afastado do cabeçalho da página e seu conteúdo deve estar centralizado. Para isso usaremos as propriedades margin-top e text-align:

.panel-about {

margin-top: 20px;

text-align: center;

}COPIAR CÓDIGO

Precisamos aumentar a fonte do título e do subtítulo, pois como usamos CSS Reset seus valores base do navegador foram resetados:

.panel-about .title {

font-size: 40px;

margin-bottom: 16px;

}

.panel-about .subtitle {

font-size: 24px;

margin-bottom: 16px;

}COPIAR CÓDIGO

Por fim, os parágrafos dentro de .about-description precisam ter suas fontes aumentadas, inclusive o espaçamento entre linhas. Conseguimos o último através da propriedade line-height:

.panel-about .about-description {

font-size: 18px;

line-height: 35px;

}COPIAR CÓDIGO

Excelente. Porém, o conteúdo do painel é limitado, possuindo a mesma margin direita e esquerda, tanto na versão mobile quanto Desktop. Para isso, vamos criar a classe container. Mas como ela será utilizada por outras partes da nossa página (elas também são limitadas), nada mais justo do que criá-la no arquivo base.css, aquele que será comum por toda a página.

Então, dentro de css/base.css temos a classe container:

.container {

width: 96%;

margin: 0 auto;

}COPIAR CÓDIGO

É claro, precisamos importá-lo em index.html:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

<title>Aluraconf</title>

<link rel="stylesheet" href="css/reset.css">

<link rel="stylesheet" href="css/base.css">

<link rel="stylesheet" href="css/header.css">

<link rel="stylesheet" href="css/about.css">

</head>

<!-- código posterior omitido -->COPIAR CÓDIGO

Outro ponto é que não podemos simplesmente adicionar a classe em <section class="panel panel-about"> porque como o seletor .container tem a mesma especificidade de .panel-about, sendo o último a ser carregado, será aplicado zerando o valor de margin-topde .painel-about. Em suma, não queremos limitar o painel, mas seu conteúdo.

Para isso, podemos adicionar uma <div> que não possui qualquer valor semântico apenas como elemento estruturante e movermos para dentro dela todo o conteúdo do painel.

<section class="panel panel-about">

<div class="container">

<h2 class="title">Muito conteúdo: do frontend ao backend</h2>

<h3 class="subtitle">30 de Maio de 2017 - na Cidade Maravilhosa</h3>

<div class="about-description">

<p>

Com a AluraConf abordaremos o que há de novidade e conhecimentos essenciais a um desenvolver mobile.

</p>

<p>

É uma grande chance de ficar por dentro dos assuntos de Android, iOS, Web mobile, desenvolvimento híbrido, UX, além da

parte backend com Rest, Microservices, e BigData. O desenvolvedor atualizado vai estar aqui.

</p>

</div>

</div>

</section>COPIAR CÓDIGO

Com isso, nosso panel-about continuará com seu margin-top e o conteúdo do painel será limitado na tela.

# 01Estrutura do painel

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/carreira-front/stages/03-aluraconf.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Não há muito o que falar sobre a estrutura do painel. O importante é que você esteja sempre direcionado a uma estrutura semântica e que evite pensar no estilo durante a criação. As imagens a seguir apresentam a estrutura já estilizada, apenas para que vocês tenham uma visão geral





# 02Gabarito

Criaremos também uma seção dedicada aos palestrantes. Assim como as outras seções que representam painéis, teremos um título e um subtítulo. Aliás, a medida que eu for construindo a estrutura, já adicionarei classes para deixar claro o que cada parte da estrutura é e também preparar o terreno para a estilização. Lembre-se que é uma boa evitarmos de utilizar seletores de TAG para não ficarmos amarrados na estrutura do documento:

<section class="panel panel-speakers">

<h2 class="title">PALESTRANTES</h2>

<h3 class="subtitle">

Aprenda com palestrantes que são referências na comunidade

</h3>

</section>COPIAR CÓDIGO

Teremos uma lista não ordenada de palestrantes, porque a ordem não nos importa:

<section class="panel panel-speakers">

<h2 class="title">PALESTRANTES</h2>

<h3 class="subtitle">

Aprenda com palestrantes que são referências na comunidade

</h3>

<ul class="speakers">

</ul>

</section>COPIAR CÓDIGO

Cada item da lista terá uma figure que representará o retrato do palestrante. Já a descrição da palestra ficará fora da figure, mas ainda dentro da <li> pois faz parte da mesma informação, a do palestrante e sua palestra:

<section class="panel panel-speakers">

<h2 class="title">PALESTRANTES</h2>

<h3 class="subtitle">

Aprenda com palestrantes que são referências na comunidade

</h3>

<ul class="speakers">

<li class="speaker">

<figure class="speaker-portrait">

<img src="img/palestrantes/placeholder.jpeg" alt="foto do palestrante">

<figcaption class="speaker-name">

Flávio Almeida

</figcaption>

<p class="speaker-bio">

Desenvolvedor e instrutor - Alura/Caelum

</p>

</figure>

<p class="speaker-speach">To be or not to be MEAN</p>

</li>

</ul>

</section>COPIAR CÓDIGO

# 03Estilo do painel

Uma dica: você não precisa usar media queries muito menos outro artifícios de design responsivo mais sofisticados aqui para que o painel de palestrantes tenha um visual como das imagens a seguir:

**Versão Desktop**



**Versão Mobile**



# 04Gabarito - parte 1

Vamos começar afastando o painel do painel anterior e definindo uma cor de fundo para ele.

.panel-speakers {

padding-top: 40px;

background: #EF7247;

}COPIAR CÓDIGO

O título e subtítulos do painel devem estar centralizados, inclusive o tamanho da fonte também precisa ser ajustado. Só fique atento que o subtítulo deve estar um pouco separado do título, ou seja, há um valor de margem aí. Inclusive o subtítulo também está um pouco afastado da lista de palestrantes:

.panel-speakers .about-speakers {

text-align: center;

}

.panel-speakers .title {

color: white;

font-size: 40px;

margin-bottom: 16px;

text-transform: uppercase;

}

.panel-speakers .subtitle {

color: white;

font-size: 32px;

margin-bottom: 16px;

}COPIAR CÓDIGO

# 05Gabarito - parte 2

Vamos começar centralizando a lista de palestrantes:

.panel-speakers .speakers {

text-align: center;

}COPIAR CÓDIGO

Cada li que representa um palestrante precisa estar lado a lado e ainda permitir a alteração da sua largura. É por isso que utilizaremos o display: inline-block e definiremos o tamanho de cada painel que será 250px. Não há problema em usar essa medida absoluta para design responsivo, pois o viewport dos smartphones são superiores a esse valor. Uma atenção especial para o vertical-align. Vamos alinhar pelo topo, caso cada li de palestrante seja maior do que a outra:

.panel-speakers .speaker {

display: inline-block;

width: 250px;

background-color: white;

margin: 0 16px 32px 0;

padding: 18px;

border-radius: 10px;

vertical-align: top;

}COPIAR CÓDIGO

A imagem do palestrante esta arredonda, nada que o border-radius não resolva. Também foi utilizado uma altura e uma largura de 120px que deve ser o padrão de cada imagem de palestrante, no caso, usamos esse valor que é do nosso placeholder.

.panel-speakers .speaker-image {

border-radius: 50%;

height: 120px;

width: 120px;

}COPIAR CÓDIGO

Agora basta ajustarmos os espaçamento, margem e tamanho dos demais elementos:

.panel-speakers .speaker-name {

padding-top: 10px;

}

.panel-speakers .speaker-bio {

font-weight: bold;

margin-top: 16px;

}

.panel-speakers .speaker-speach {

margin-top: 16px;

padding-top: 8px;

font-size: 16px;

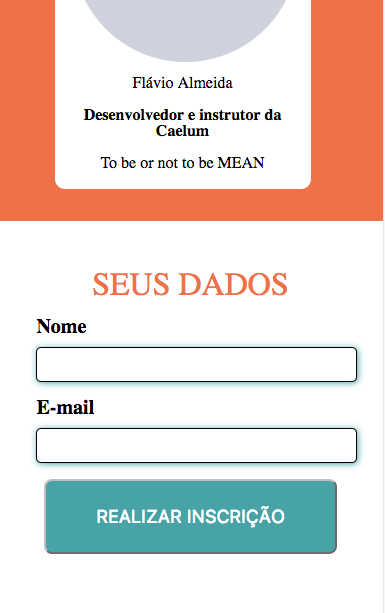
}COPIAR CÓDIGO

# 01Estrutura do formulário

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/carreira-front/stages/04-aluraconf.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

O mesmo formulário criado para a versão mobile first será usado para a verão desktop. Trabalhe com medidas relativas para que ele se adapte em telas maiores.

Segue duas imagens do formulário já estilizado. Não esqueça que nosso foco é a estrutura por enquanto.





# 02Gabarito

A estrutura do formulário é bem simples. É um form dentro de uma section que possui um título. Dois campos e dois labels. Por fim, há um botão:

<section>

<h2>Seus dados</h2>

<form>

<label>Nome</label>

<input>

<label>E-mail</label>

<input>

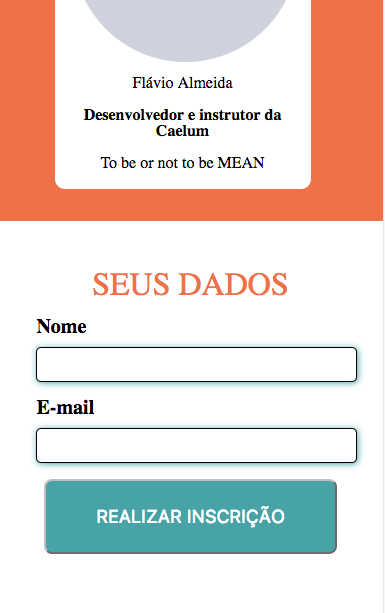
<button>REALIZAR INSCRIÇÃO</button>

</form>

</section>COPIAR CÓDIGO

# 03Estilo do formulário

Chegou a hora de estilizar nosso formulário. Temos duas imagens que ilustram sua estilização em dispositivos menores e maiores:





# 04Gabarito - parte 1

Nessa primeira etapa, diferente do que fizemos antes, vamos adicionar classes nos elementos da página para só depois iniciarmos a sua estilização:

<section class="panel-form">

<div class="container">

<h2 class="title">Seus dados</h2>

<form>

<div class="field">

<label CLASS="field-label">Nome</label>

<input class="field-input">

</div>

<div class="field">

<label class="field-label">E-mail</label>

<input class="field-input">

</div>

<div class="button-place">

<button class="button">REALIZAR INSCRIÇÃO</button>

</div>

</form>

</div>

</section>COPIAR CÓDIGO

Criar o arquivo css/form.css e importá-lo em index.html.

# 05Gabarito - parte 2

Vamos começar afastando o painel do painel anterior e estilizando o seu título:

.panel-form {

margin-top: 32px;

}

.panel-form .title {

font-size: 32px;

color: #EF7247;

text-align: center;

text-transform: uppercase;

}

Um field é caracterizado por um label e um input, praticamente amigos inseparáveis. Nesse sentido, cada field é um pouco afastado do anterior. Vamos aplicar uma margin nele:

.field {

margin-botton: 16px;

}COPIAR CÓDIGO

Por padrão um label é do tipo inline, assim como um input. Todavia, precisamos de um label acima do input. Conseguimos isso mudando seu modo de display de inline para block. Os demais ajuste dizem respeito ao tamanho da fonte, margin e negrito:

.field label {

display: block;

font-size: 20px;

margin-bottom: 10px;

font-weight: bold;

}COPIAR CÓDIGO

Precisamos que cada input tenha 100% do tamanho do seu elemento pai para que estiquem para qualquer tamanho de tela. Para que possamos fazer isso, também transformaremos o elemento em block para que seja possível alterar a propriedade width:

.field .field-input {

display: block;

width: 100%;

font-size: 20px;

padding: 5px;

}COPIAR CÓDIGO

# 06Gabarito - parte 3

As classes dos nossos fields podem ser aproveitadas em outras páginas da nossa aplicação. Dessa forma, vamos extrair o CSS que fizemos para dentro do arquivo css/base.css:

/\* css/base.css \*/

.container {

width: 96%;

margin: 0 auto;

}

.field {

margin-bottom: 16px;

}

.field label {

display: block;

font-size: 20px;

margin-bottom: 10px;

font-weight: bold;

}

.field .field-input {

display: block;

width: 100%;

box-sizing: border-box;

font-size: 20px;

padding: 5px;

border-radius: 5px;

box-shadow: 1px 1px 10px #48A2A6;

}COPIAR CÓDIGO

Precisamos arredondar a borda do input.

.field .field-input {

display: block;

width: 100%;

font-size: 20px;

padding: 5px;

border-radius: 5px;

}COPIAR CÓDIGO

Haverá também uma sobra no input sutil, com a mesma cor do botão:

.field .field-input {

display: block;

width: 100%;

font-size: 20px;

padding: 5px;

border-radius: 5px;

box-shadow: 1px 1px 10px #48A2A6;

} COPIAR CÓDIGO

Aliás, o estilo do button que definimos em header.css será o mesmo estilo do button do nosso formulário. Ele é mais um candidato a ser removido e adicionando dentro de css/base.css.

Primeiro, vamos remover a definição da seguinte classe em css/header.css:

/\* css/header.css \*/

/\* remover \*/

.header .jumbotron .button {

display: inline-block;

color: white;

text-decoration: none;

font-size: 18px;

padding: 25px 50px;

background-color: #48A2A6;

border-radius: 8px;

}COPIAR CÓDIGO

Adicionar em css/base.css:

.button {

display: inline-block;

color: white;

text-decoration: none;

font-size: 18px;

padding: 25px 50px;

background-color: #48A2A6;

border-radius: 8px;

}COPIAR CÓDIGO

<div class="button-place">

<button class="button">REALIZAR INSCRIÇÃO</button>

</div>COPIAR CÓDIGO

Para que seja possível centralizar o botão, é o seu elemento PAI que precisa receber text-align:

.button-place {

text-align: center;

}COPIAR CÓDIGO

# 01Comportamento

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/carreira-front/stages/05-aluraconf.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Há apenas um padrão de comportamento em nossa página. Quando clicarmos nos itens de menu "Sobre" e "Palestrantes", faremos um scroll suave até a respectiva área da tela. Sinta-se livre para usar jQuery. Aliás, a ideia é que você o utilize para colocar em prática o que aprendeu até agora.

# 02Gabarito

Antes de mais nada, vamos adicionar ID's mos painéis Sobre, Palestrantes, Inscreva-se. É através desses ID's que faremos uma navegação interna passando esses ID's para as âncoras do menu da aplicação. Quando o usuário clicar, seremos direcionados instantaneamente para essas áreas:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

<title>Aluraconf</title>

<link rel="stylesheet" href="css/reset.css">

<link rel="stylesheet" href="css/base.css">

<link rel="stylesheet" href="css/header.css">

<link rel="stylesheet" href="css/about.css">

<link rel="stylesheet" href="css/speakers.css">

<link rel="stylesheet" href="css/form.css">

</head>

<body>

<header class="header">

<nav class="main-menu">

<ul>

<li class="menu-item"><a href="#panel-about">Sobre</a></li>

<li class="menu-item"><a href="#panel-speakers">Palestrantes</a></li>

</ul>

</nav>

<div class="jumbotron">

<section class="panel-main">

<h1 class="title">ALURACONF</h1>

<h2 class="subtitle">30MAIO|RIO</h2>

</section>

<section class="panel-subscribe">

<a href="#panel-form" class="button">Inscreva-se</a>

</section>

</div>

</header>

<section class="panel panel-about" id="panel-about">

<div class="container">

<h2 class="title">Muito conteúdo: do frontend ao backend</h2>

<h3 class="subtitle">30 de Maio de 2017 - na Cidade Maravilhosa</h3>

<div class="about-description">

<p>Com a AluraConf abordaremos o que há de novidade e conhecimentos essenciais a um desenvolver mobile.</p>

<p>É uma grande chance de ficar por dentro dos assuntos de Android, iOS, Web mobile, desenvolvimento híbrido, UX, além da parte backend com Rest, Microservices, e BigData. O desenvolvedor atualizado vai estar aqui.</p>

</div>

</div>

</section>

<section class="panel panel-speakers" id="panel-speakers">

<div class="container">

<div class="about-speakers">

<h2 class="title">PALESTRANTES</h2>

<h3 class="subtitle">Aprenda com palestrantes que são referência no mercado</h3>

</div>

<ul class="speakers">

<li class="speaker">

<figure class="speaker-portrait">

<img class="speaker-image" src="img/palestrantes/placeholder.jpeg" alt="Foto do palestrante">

<figcaption class="speaker-name">Flávio Almeida</figcaption>

<p class="speaker-bio">Desenvolvedor e instrutor da Caelum</p>

</figure>

<p class="speaker-speach">To be or not to be MEAN</p>

</li>

<li class="speaker">

<figure class="speaker-portrait">

<img class="speaker-image" src="img/palestrantes/placeholder.jpeg" alt="Foto do palestrante">

<figcaption class="speaker-name">Flávio Almeida</figcaption>

<p class="speaker-bio">Desenvolvedor e instrutor da Caelum</p>

</figure>

<p class="speaker-speach">To be or not to be MEANTo be or not to be MEANTo be or not to be MEANTo be or not to be MEANTo be or not to be MEANTo be or not to be MEANTo be or not to be MEANTo be or not to be MEAN</p>

</li>

<li class="speaker">

<figure class="speaker-portrait">

<img class="speaker-image" src="img/palestrantes/placeholder.jpeg" alt="Foto do palestrante">

<figcaption class="speaker-name">Flávio Almeida</figcaption>

<p class="speaker-bio">Desenvolvedor e instrutor da Caelum</p>

</figure>

<p class="speaker-speach">To be or not to be MEAN</p>

</li>

<li class="speaker">

<figure class="speaker-portrait">

<img class="speaker-image" src="img/palestrantes/placeholder.jpeg" alt="Foto do palestrante">

<figcaption class="speaker-name">Flávio Almeida</figcaption>

<p class="speaker-bio">Desenvolvedor e instrutor da Caelum</p>

</figure>

<p class="speaker-speach">To be or not to be MEAN</p>

</li>

<li class="speaker">

<figure class="speaker-portrait">

<img class="speaker-image" src="img/palestrantes/placeholder.jpeg" alt="Foto do palestrante">

<figcaption class="speaker-name">Flávio Almeida</figcaption>

<p class="speaker-bio">Desenvolvedor e instrutor da Caelum</p>

</figure>

<p class="speaker-speach">To be or not to be MEAN</p>

</li>

<li class="speaker">

<figure class="speaker-portrait">

<img class="speaker-image" src="img/palestrantes/placeholder.jpeg" alt="Foto do palestrante">

<figcaption class="speaker-name">Flávio Almeida</figcaption>

<p class="speaker-bio">Desenvolvedor e instrutor da Caelum</p>

</figure>

<p class="speaker-speach">To be or not to be MEAN</p>

</li>

</ul>

</div>

</section>

<section class="panel-form" id="panel-form">

<div class="container">

<h2 class="title">Seus dados</h2>

<form>

<div class="field">

<label class="field-label">Nome</label>

<input class="field-input">

</div>

<div class="field">

<label class="field-label">E-mail</label>

<input class="field-input">

</div>

<div class="button-place">

<button class="button">REALIZAR INSCRIÇÃO</button>

</div>

</form>

</div>

</section>

<script src="js/jquery.js"></script>

<script src="js/index.js"></script>

</body>

</html>

Veja também que já importamos o jquery inclusive o arquivo index.js, aquele no qual escreveremos nosso código.

O comportamento que queremos para os itens de menu e para o botão Inscreva-se é o mesmo. Para evitarmos repetição de código vamos criar a função ativaScrollSuave. Ela receberá o seletor do elemento alvo para nosso efeito de transição suave.

function ativaScrollSuave(selector) {

}

Para o seletor que receberemos, vamos selecionar o elemento e adicionar o código para o evento clique:

function ativaScrollSuave(selector) {

$(selector).click(function(event){

event.preventDefault();

});

}

Precisamos fazer event.preventDefault() para cancelar o evento padrão das nossas âncoras, que é seguir o valor do atributo href. Queremos que transição seja feita por JavaScript.

Para a âncora que foi clicada, precisamos extrair o valor do seu href. Esse valor guarda o ID do elemento no qual queremos fazer o scroll suave.

function ativaScrollSuave(selector) {

$(selector).click(function(event){

event.preventDefault();

var target = $(this).attr('href');

});

}COPIAR CÓDIGO

Em nosso código, target é o valor do href do elemento que clicamos. Fizemos $(this) para promovermos o this, que é um elemento do DOM para um jQuery Object para então extrairmos o valor do atributo href através de $(this).attr('href').

Agora que já sabemos para a âncora clicada até onde ela deve ir, faremos uma animação através da função animate do jQuery. Ela será responsável em fazer nosso scroll suave:

function ativaScrollSuave(selector) {

$(selector).click(function(event){

event.preventDefault();

var target = $(this).attr('href');

$('html, body').animate({

scrollTop: $(target).offset().top

}, 1000);

});

}COPIAR CÓDIGO

Agora só precisamos passar como parâmetro para a função ativaScrollSuave o seletor CSS de cada âncora que desejamos fazer o scroll suave. Vamos usar um seletor de atributo para selecionar cada uma delas:

ativaScrollSuave('a[href\*=panel-about]');

ativaScrollSuave('a[href\*=panel-speakers]');

ativaScrollSuave('a[href\*=panel-form]');COPIAR CÓDIGO

Excelente. Temos uma solução agora genérica que funcionará para qualquer âncora que faça uma navegação interna em nosso site.

# 03Considerações finais

Chegamos mais uma vez ao final de mais um curso. Com um grau de dificuldade de básico para intermediário, a ideia deste treinamento sempre foi tornar quem esta começando ainda mais seguro praticando em um projeto o que aprendeu. Para aqueles que já tem um bom conhecimento, o desafio de terminar o mais rapidamente o layout também é válido.

Dependendo do feedback deste curso, outros projetos poderão ser propostos, com graus de dificuldade diferentes para contemplar um amplo espectro de alunos.

Sucesso e bom estudo para todos!

# 04Download do projeto final

Perdeu o projeto final? Não se preocupe, você pode baixá-lo [aqui](https://s3.amazonaws.com/caelum-online-public/carreira-front/stages/06-aluraconf.zip).