

# Imersão Dev - Aula 10

**Rafaella:** Estamos na nossa última aula da imersão! Eu sei que ainda vamos nos ver por aqui, mas queria já deixar os parabéns por você ter feito tudo durante essas duas semanas. Não é fácil aprender programação, e o fato de você ter chegado até aqui já é muita coisa.

Caso algum conteúdo tenha ficado atrasado, não se preocupe, pois você pode desenvolver aquilo que conseguir. Não é necessário fazer tudo que propomos em todas as aulas, e nós, inclusive, estamos no Discord para ajudar em qualquer dúvida que tiver.

Estou muito ansiosa para mostrar para você o projeto dessa aula, que é muito especial, e que você precisa postar nas redes sociais (e logo descobrirá porque).

Pegue seu café, abra o Codepen e vamos à nossa décima aula!

Eu gostaria de dar as boas-vindas à última aula da Imersão Dev. Eu nem acredito que chegamos até aqui. Foram tantos conteúdos, desde variáveis no Javascript até estilização com CSS, muita coisa que vocês aprenderam aqui e que tenho certeza que levarão para o resto da vida, conseguindo fazer os seus próprios projetos - claro, além de todos os que já fizemos até agora.

O projeto que vamos terminar hoje é justamente adicionar esses trabalhos em seu portfólio, para você compartilhar tudo que fez até então com todas as pessoas, adicionando cada vez mais coisas.

**Paulo:** E eu queria lembrar que na Alura temos trabalhado cada vez mais dessa forma, construindo portfólios e terminando projetos para que você implemente as ideias passadas nos cursos ou imersões, seja em eventos gratuitos ou dentro da nossa plataforma. Acho que é uma forma de você conhecer nosso trabalho e fazer parte dessa comunidade que não é só nossa, ela vai muito além da Alura.

Estou muito contente de chegarmos nesse último dia, de ver você compartilhando e conversando no Discord, fazendo streaming do que está fazendo e assim por diante. Nesse momento da estilização, cada pessoa tem uma ideia muito maluca, muito particular, e é isso mesmo que nós queremos: que você dê a sua cara para seus projetos, para seu portfólio, e que demonstre no LinkedIn e no Instagram a sua capacidade de aprendizado.

Mais do que mostrar o que você aprendeu com a Alura, o importante é mostrar a sua capacidade de refletir, analisar, aprender e chegar aos resultados, pois é isso que buscamos nos profissionais.

**Rafaella:** Hoje vamos terminar o projeto que fizemos ontem, que foi nosso portfólio. Enquanto fazemos isso, ainda vamos incluir outras coisas bem legais. Gui, quer mostrar para o pessoal como ficará o projeto final dessa aula?

**Guilherme:** A ideia dessa aula é arriscada, como vocês podem ver nessa tela do Figma. Teremos um botão para alterar o tema para claro ou escuro (\*dark mode\*) e uma aba de projetos (que vamos evoluir ainda mais. Além disso, nessa parte do projeto, conseguiremos visualizar trechos do código e executar o projeto feito, como a calculadora, o mentalista e assim por diante.

Também teremos uma aba de conquistas, que por enquanto não sei se vai ficar como desafio, vamos descobrir ao longo da aula. Ao final de tudo, vamos colocar todo esse projeto em um espaço onde será possível gerar um link e enviar para que qualquer pessoa consiga acessar a nossa página, incluindo esses projetos, os links do seu próprio Github, LinkedIn e assim por diante.

Vamos começar, Rafa?

**Rafaella:** Vamos recapitular. Tínhamos feito todo o esqueleto HTML da nossa página, incluindo um `<body>`, que é a tag principal da página. Nela nós estilizamos o fundo, que tem um gradiente do azul escuro para azul claro, além da fonte que vamos utilizar.

Temos também uma `<div>` da classe `container`, que tem desde a parte do cabeçalho até o conteúdo principal da nossa página. Temos o `<header>`, que é o nosso cabeçalho de fato, e o `<main>`, a parte dos projetos que estamos construindo agora.

Vamos começar a estilizar a nossa lista de projetos, algo que ainda não terminamos de fazer. Quer trocar algumas configurações, Gui?

**Guilherme:** Nas configurações do CodePen, vou colocar as tags da ImersãoDev e da Alura para as pessoas encontrarem projetos similares. Em "Behavior" eu vou desativar o salvamento automático e o Preview, e ativarei a formatação automática.

Nossa lista atualmente não está bonita, ela está bem simples. Como fazemos para tirar os pontos que estão ao lado e deixar com uma cara melhor?

**Paulo:** Portfólio é algo para ser bonito, não é? Precisa ser convidativo, além de ter links para seu e-mail, seu LinkedIn, algumas coisas que passei no desafio.

**Rafaella:** Perfeito. Inclusive, se você quiser estilizar com outras cores, fique à vontade. No CSS, temos uma classe `.projetos` que engloba toda a parte azul do Figma, onde temos os nossos projetos.

Atualmente temos nela um `display: flex`, que posiciona os elementos filhos dentro dele (no caso a nossa lista), e o `flex-direction: column`, que define a direção como colunas.

```
.projetos {  
    display: flex;  
    flex-direction: column;  
}
```

Além de posicionar os elementos, precisamos colocar nosso background. No Figma, vemos que a cor dessa parte é o azul (**linear-gradient**), o mesmo que tínhamos usado no **<body>**.

```
.projetos {  
    display: flex;  
    flex-direction: column;  
    background: linear-gradient(230.65deg, #499cfe 27.49%, #9cc8fc  
83.19%);  
}
```

Vamos salvar e rodar para verificarmos como está ficando. Com essa alteração, a seção "Projetos" ganhará um fundo com um gradiente azul. Uma coisa que está incomodando um pouco é que nossa foto de perfil está muito "grudada" nessa caixa, e seria interessante distanciarmos um pouco.

Como vimos anteriormente, isso é possível adicionando uma margem, nesse caso **margin-top** para distanciarmos da parte de cima.

```
.projetos {  
    display: flex;  
    flex-direction: column;  
    background: linear-gradient(230.65deg, #499cfe 27.49%, #9cc8fc  
83.19%);  
    margin-top: 32px;  
}
```

Legal, bem melhor. Também podemos ajustar os elementos dentro de `.projetos` um pouco mais espaçados, algo que faremos com o `padding`.

```
.projetos {
  display: flex;
  flex-direction: column;
  background: linear-gradient(230.65deg, #499cfe 27.49%, #9cc8fc
83.19%);
  margin-top: 32px;
  padding: 32px;
}
```

Quando utilizamos apenas `padding` ou apenas `margin`, estamos adicionando essa propriedade a todos os cantos. Já quando usamos uma especificação, como `margin-top`, estamos aplicando a um canto específico (cima, nesse caso).

Vamos analisar o projeto no Figma para entendermos o que mais podemos estilizar. Ainda temos um link de redirecionamento para os projetos, mais tarde faremos com que as telas do CodePen apareçam. Por enquanto vamos focar na caixa azul dos nossos projetos.

Ainda falta incluirmos um `border-radius`, que deixa o elemento arredondado, e o `box-shadow`, que faz uma sombra na parte inferior. Vamos copiar esses elementos e incluir em nosso `.projetos`.

```
.projetos {
  display: flex;
  flex-direction: column;
  background: linear-gradient(230.65deg, #499cfe 27.49%, #9cc8fc
83.19%);
  margin-top: 32px;
  padding: 32px;
  box-shadow: 2px 2px 4px rgba(16, 16, 16, 0.42);
  border-radius: 20px;
}

.projetos-titulo {
  list-style: none;
  font-weight: 700;
  font-size: 36px;
}
```

Já temos as bordas arredondadas e a sombra, e agora precisamos estilizar os textos, tanto o título da lista quanto os elementos dela. Primeiramente, criaremos uma classe para o título da lista, que é a nossa `<ul>`. No HTML, criaremos uma `class="projetos-titulo"`

```
<main class="projetos">
  <ul class="projetos-titulo">Projetos</ul>
```

No CSS, incluiremos um `.projetos-titulo` e abriremos chaves para começarmos a estilizar.

```
.projetos-titulo {
}
```

Primeiramente, removeremos o estilo dessa lista, o que inclui o ponto e a indentação. Quando queremos estilizar e personalizar um elemento, é interessante removermos a decoração padrão que vem do HTML e do CSS, algo que fazemos com `list-style: none` (de "nenhum").

```
.projetos-titulo {
  list-style: none;
}
```

Em seguida, alteraremos o peso da fonte (`font-weight`) para `700`, seguindo o nosso projeto do Figma. Além disso, alteraremos o tamanho da fonte (`font-size`) para `36px`.

```
.projetos-titulo {
  list-style: none;
  font-weight: 700;
  font-size: 36px;
}
```

**Guilherme:** Ficou mais ou menos, né? O título tá legal, mas os outros elementos estão estranhos.

**Rafaella:** Os outros ainda precisamos arrumar. Agora adicionaremos uma classe para os itens da lista, que são nossos `<li>`, chamada `projetos-item`.

**Guilherme:** Eu poderia colocar uma classe que engloba todas as `<li>`?

**Rafaella:** Não, você vai criar a mesma classe para todas, mas precisa adicionar essa classe em cada uma.

**Guilherme:** Entendi, a ideia é criarmos uma estrutura que será reutilizada em outras partes.

**Rafaella:** Isso mesmo. Nós poderíamos, assim como fizemos ali em cima, colocar o `<li>` dentro de todos os `projetos`, por exemplo, como fizemos o `<h1>` dentro de todas as `<div>` de `perfil`, mas ambos os jeitos funcionam. Aqui vamos mostrar como utilizar uma classe para mais de um elemento.

```
<main class="projetos">
  <ul class="projetos-titulo">Projetos<ul>
    <li class="projetos-item"><a
href="https://codepen.io/guilhermeonrails/pen/poPZGov?editors=01111">
Conversor de Moeda</a></li>
    <li class="projetos-item"><a
href="https://codepen.io/guilhermeonrails/pen/yLbxpwm">Aluraflix</a><
/li>
  </li>
```

**Rafaella:** Então vamos estilizar o nosso item da lista. No CSS, criaremos um `.projetos-item` e abriremos chaves. Para removermos os pontos antes de cada elemento da lista, usaremos o `list-style-type: none`.

```
.projetos-item {
  list-style-type: none;
}
```

Além disso, alteraremos o tamanho da fonte (`font-size`) e o peso (`font-weight`). Por fim, temos também uma propriedade `line-height`, que é a altura da linha, que é interessante adicionarmos. Dessa forma, teremos mais distanciamento entre cada linha da nossa lista.

```
.projetos-item {  
  list-style-type: none;  
  font-size: 24px;  
  font-weight: 400;  
  line-height: 48px;  
}
```

**Rafaella:** Falta estilizar as "âncoras" dos nossos hiperlinks, que deixam um sublinhado no texto. Para alterarmos isso, usaremos uma classe `.projetos-item a`. Assim, toda tag `<a>` que esteja dentro de um elemento `projetos-item` terá essa estilização.

Adicionaremos um `color: #1c1c1c` para alterarmos a cor do texto para a mesma dos outros elementos e removeremos a decoração do texto com `text-decoration: none`.

```
.projetos-item a {  
  color: #1c1c1c;  
  text-decoration: none;  
}
```

**Rafaella:** Com isso, nosso projeto estará padronizado e poderemos clicar sobre o link do "Conversor de Moeda" da nossa página.

**Guilherme:** Imediatamente já identifica como um link, e direciona para a nossa página do Conversor de Moedas.

**Rafaella:** Exatamente, e poderíamos colocar links do **LinkedIn**, **Facebook** e outras páginas, basta mudar o `href` que colocamos o link do CodePen.

**Guilherme:** Ficou bem bacana este link sem o texto padrão que vem do HTML. Mas para nosso projeto principal, abriremos uma lista com caixinhas contendo os projetos que fizemos para visualizarmos.

Ou seja, queremos abrir apenas um pedaço do projeto que estamos pegando do nosso projeto no CodePen para exibir. Para isso, clicaremos com o botão direito sobre o retângulo que aparece com os projetos no Figma e abriremos o Conversor de Moedas em uma nova aba.

Neste projeto que fizemos, encontraremos um botão chamado "Embed" na barra inferior, ao lado de "Fork". Clicando em "Embed", abriremos uma caixa chamada "Embed This Pen" que apresenta algumas opções, como a de "Theme" para escolhermos o tema, que pode ser "Default", "Light", "Dark" ou "User Default".

Na aba de "HTML (Recommended)", encontraremos uma tag `<p>` com várias coisas, como o usuário, o tamanho que irá aparecer e etc.

Copiaremos todo este conteúdo, que não é CSS nem JavaScript, e sim um parágrafo HTML. Em nosso projeto, abriremos a parte do HTML e tiraremos a tag-âncora `<a>`.

**Rafaella:** Ao invés da tag `<a>`, deixaremos a `<p>` ainda com o título de "Conversor de Moeda" para não o perdermos, pois também temos isso no Figma.

Tiraremos o `href` também e colocaremos a tag `<p>` nas duas linhas de `<li>`.

//código omitido

```
<main class="projetos">
  <ul class="projetos-titulo">Projetos</ul>
    <li class="projetos-item"><p> Conversor de moeda</p></li>
    <li class="projetos-item"><p> Aluraflix</p></li>
  </main>
</div>
</body>
```

**Guilherme:** De volta à caixa "Embed The Pen", copiaremos todo o conteúdo da tag `<p>` com a classe `"codepen"` e todas as propriedades, e depois colaremos embaixo da `<p>` que contém os títulos no nosso projeto.

**Rafaella:** Podemos pular uma linha se quisermos, e depois salvamos.

//código omitido

```
<main class="projetos">
  <ul class="projetos-titulo">Projetos</ul>
```



```

        <li class="projetos-item">
            <p> Conversor de moeda</p>
            <p class="codepen" data-height="300"
data-theme-id="dark" data-default-tab="html,result" data-slug-hash="poPZGov"
data-user="guilhermeonrails" style="height: 300px; box-sizing: border-box; display:
flex; align-items: center; justify-content: center; border: 2px solid; margin: 1em 0;
padding: 1em;">

                <span> See the Pen <a
href="https://codepen.io/guilhermeonrails/pen/poPZGov">
                    Conversor de moedas - ID3</a> by
                    @guilimadev (<a href="https://codepen.io/guilhermeonrails">@guilhermeonrails</a>)
                    on <a
href="https://codepen.io">CodePen</a>.</span>
            </p>
            <script async
src="https://cpwbassets.codepen.io/assets/embed/ei.js"></script>
        </li>

        <li class="projetos-item"><p> Aluraflix</p></li>
    </main>
</div>
</body>

```

**Guilherme:** Na página, veremos que aparece o HTML, o CSS e o JavaScript do projeto Conversor de Moeda.

Ao centro e acima deste espaço com o código, há o botão de "Result" para vermos a página do Conversor de fato.

Se colocarmos um valor "10" no campo de "Insira o valor" por exemplo, e depois clicarmos em "Converter", receberemos a mensagem de que "o resultado em real é R\$50".

Ou seja, estamos rodando o nosso projeto dentro do portfólio direto pelo CodePen. Podemos escolher como queremos deixar o título, o alinhamento, o tamanho, é interessante explorar!

**Rafaella:** Podemos estilizar o quanto quisermos, mas por enquanto está bom.

**Guilherme:** Para vermos funcionando de verdade, abriremos o outro projeto chamado "Aluraflix" feito no quinto dia da Imersão em uma outra aba. Ao lado do botão "Fork", encontraremos o botão "Embed" novamente.

Clicando sobre este, teremos a caixa "Embed This Pen" também com a geração de um texto na tag `<p>` fornecido na aba "HTML (Recommended)".

Copiaremos todo este código da tag `<p>` e colaremos em nosso projeto, da mesma forma que fizemos com o "Embed" do Conversor de Moeda.

**Paulo:** E tudo bem se não entendermos completamente o texto, pois é um código que criaram para colocarmos algo dentro de algo.

Inclusive, esse botão "Embed" existe no **YouTube** e em outros lugares quando quisermos colocar um vídeo, um CodePen mesmo, uma postagem do **Instagram** ou Facebook e etc.

Usamos o termo "embedar" para fazer isso no dia a dia, ou seja, colocamos algo dentro de outra coisa.

**Guilherme:** Então faremos o mesmo processo que fizemos com a tag `<p>` do Conversor de Moeda.

Em seguida, salvaremos e rodaremos para vermos o resultado na página.

//código omitido

```
<main class="projetos">
  <ul class="projetos-titulo">Projetos</ul>
  <li class="projetos-item">
    <p> Conversor de moeda</p>
    <p class="codepen" data-height="300" data-theme-id="dark"
data-default-tab="html,result" data-slug-hash="poPZGov"
data-user="guilhermeonrails" style="height: 300px; box-sizing: border-box; display:
flex; align-items: center; justify-content: center; border: 2px solid; margin: 1em 0;
padding: 1em;">
      <span>See the Pen <a
href="https://codepen.io/guilhermeonrails/pen/poPZGov">
        Conversor de moedas - ID3</a> by @guilimadev (<a
href="https://codepen.io/guilhermeonrails">@guilhermeonrails</a>)
        on <a href="https://codepen.io">CodePen</a>.</span>
    </p>
    <script async
src="https://cpwebassets.codepen.io/assets/embed/ei.js"></script>
  </li>
  <li class="projetos-item">
    <p> Aluraflix</p>
    <p class="codepen" data-height="300" data-theme-id="dark"
```

```

data-default-tab="html,result" data-slug-hash="yLbxpwm"
data-user="guilhermeonrails" style="height: 300px; box-sizing: border-box; display:
flex; align-items: center; justify-content: center; border: 2px solid; margin: 1em 0;
padding: 1em;">
    <span>See the Pen <a
href="https://codepen.io/guilhermeonrails/pen/yLbxpwm">
    Aluraflix - dia 5</a> by @guilimadev (<a
href="https://codepen.io/guilhermeonrails">@guilhermeonrails</a>)
    on <a href="https://codepen.io">CodePen</a>.</span>
    </p>
    <script async
src="https://cpwebassets.codepen.io/assets/embed/ei.js"></script>
    </li>
</main>
</div>
</body>

```

**Guilherme:** Pronto, agora temos os dois projetos sendo carregados na página, o Conversor de Moeda e o Aluraflix.

Não faremos para todos os projetos porque não temos tempo suficiente, mas como o processo é bem tranquilo e repetitivo, fica o desafio para vocês.

Caminhamos um pouco nesta parte do "Embed", e ficou bem legal. Mas ainda queremos saber como fazer um botão de alterar tema para clicarmos e mudarmos a cor de fundo da página para termos o tema "dark" por exemplo, parecido com o que há no GitHub.

**Rafaella:** Para fazermos este botão no nosso cabeçalho, voltaremos ao Figma para vermos como deve ser.

Ele ficará no canto superior direito do container com a legenda "Alterar o tema". De volta ao código, acessaremos a tag `<header class="perfil">` e criaremos uma outra `<div>` após o título.

Portanto serão três elementos principais no nosso cabeçalho: a imagem, o título com o nome e o que fazemos, além do botão que altera o tema.

Essa divisão terá uma classe chamada `"tema"` que iremos estilizar. Dentro dela, colocaremos um botão de fato com a tag `<button>`.

Depois, usaremos o `onClick` que vimos ao longo de todo o curso. Sempre pegávamos o que já vinha no código feito por outra pessoa no HTML e CSS com toda a função.

Desta vez criaremos a função que utilizaremos depois para podermos alterar o tema, a qual pode ser chamada `"mudaTema()"`.

Não faremos isso imediatamente porque vamos estilizar primeiro, mas já deixaremos a função dentro do `onClick` no JavaScript que usaremos.

Fecharemos essa tag `<button>` e depois colocaremos o texto "Alterar tema" para a pessoa saber o que o botão faz.

Podemos salvar e rodar para vermos como está até agora.

```
<body>
  <div class="container">
    <header class="perfil">
      
      <div class="titulo">
        <h1>Rafaella Ballerini</h1>
        <h3>Instrutora e desenvolvedora front-end #imersaodev</h3>
      </div>
      <div class="tema">
        <button onclick="mudaTema()">Alterar Tema</button>
      </div>
    </header>

    //código omitido

  </div>
</body>
```

**Guilherme:** Vamos deixar ele bem legal na posição certa.

**Rafaella:** Vamos lá! A primeira coisa que faremos é estilizar de fato. Vamos ao arquivo com o código CSS no CodePen, e o ideal seria seguirmos uma linearidade dos elementos que vamos escrever para sabermos onde podemos achá-los quando entrarmos no HTML.

Então é interessante irmos até a parte do cabeçalho e alterar de lá mesmo. Temos o `.perfil`, o `.perfil-foto`, `.titulo`, `.titulo h1` e o `.titulo h3`.

Antes de `.projetos`, colocaremos a estilização da nossa classe de `.tema`. Entre as chaves, colocaremos o `button` para estilizarmos diretamente de dentro da classe, a qual é a `<div>` que engloba esse botão.

Precisaremos usar algumas propriedades novas, pois se trata da estilização de um botão. Primeiro, vamos alinhá-lo de uma forma diferente.

Os itens que já alinhamos estão todos à esquerda, e queremos que só o botão fique sozinho à direita. Como estamos estilizando com Flexbox, existe também a propriedade `align-self` para isso, sendo `flex-end` para deixarmos ao final no canto direito do conjunto.

Vamos salvar e executar para vermos onde ficou.

```
//código anterior omitido

.titulo h1 {
  font-weight: 700;
  font-size: 36px;
}

.titulo h3 {
  font-weight: 400;
  font-size: 24px;
}

.tema button {
  align-self: flex-end;
}

.projetos {
  display: flex;
  flex-direction: column;
  background: linear-gradient(230.65deg, #499cfe 27.49%, #9cc8fc 83.19%);
  margin-top: 32px;
  padding: 32px;
  box-shadow: 2px 2px 4px rgba(16, 16, 16, 0.42);
  border-radius: 20px;
}

//código posterior omitido
```

**Guilherme:** ficou bem no canto direito mesmo.

**Rafaella:** Exato, ele foi para o canto que queríamos, mais afastado dos outros elementos de texto do cabeçalho.

O próximo passo é irmos ao Figma para vermos como o botão está no projeto. Clicando sobre este, iremos até as configurações de "Typography" na aba lateral e veremos qual é o tamanho da fonte usada.

**Guilherme:** O tamanho é vinte e quatro pixels com peso de quatrocentos.

**Rafaella:** Então vamos estilizar isso. podemos copiar do Figma e colar no nosso arquivo CSS também. Portanto o `font-size`: fica `24px` e o `font-weight` é `400`.

Vamos rodar e ver o resultado na página.

```
//código anterior omitido

.titulo h1 {
  font-weight: 700;
  font-size: 36px;
}

.titulo h3 {
  font-weight: 400;
  font-size: 24px;
}

.tema button {
  align-self: flex-end;
  font-size: 24px;
  font-weight: 400;
}

.projetos {
  display: flex;
  flex-direction: column;
  background: linear-gradient(230.65deg, #499cfe 27.49%, #9cc8fc 83.19%);
  margin-top: 32px;
  padding: 32px;
  box-shadow: 2px 2px 4px rgba(16, 16, 16, 0.42);
  border-radius: 20px;
}

//código posterior omitido
```

**Rafaella:** Ficou bem grande no *zoom*! Mas já melhora no tamanho normal.

Outra estilização que podemos fazer é darmos uma distância um pouco melhor do escrito para as bordas do botão, para que o texto fique um pouco mais centralizado.

Isso é o `padding`, pois estamos alterando o elemento que queremos e estamos colocando o que está dentro dele mais para dentro.

Colocaremos esta propriedade no `.tema button`, e colocaremos `8px` seguido de `16px`. Estamos colocando duas informações nos dois números, em que a primeira diz respeito à distância que há entre o topo e o elemento interno, e entre a base e o elemento interno.

Já a segunda informação é relativa às distâncias laterais entre os limites laterais do botão e o elemento interno do texto. A mesma coisa acontece para o `margin`, então se quisermos dar uma margem acima e embaixo e outra de um lado e de outro, colocamos dois valores seguidos para cada um respectivamente.

Outra coisa é que o botão está com o fundo cinza, diferente do projeto no Figma, então precisamos encontrar a cor certa para usarmos. Ao selecionarmos o botão de "Alterar tema", teremos o valor na parte "Colors" para copiarmos.

**Guilherme:** Portanto o `background` será de `#ECF4FF`.

O próximo passo é transformarmos as bordas do botão em arredondadas com o já conhecido `border-radius`. De volta ao Figma, copiaremos o valor da propriedade de `100px` e colaremos no nosso `.tema button`.

Outra coisa que faremos é passar a nossa borda, pois ela ainda está um pouco simples demais, e podemos deixá-la melhor definida. Podemos escolher sua espessura, seu tamanho e a cor que queremos.

```
//código anterior omitido
```

```
.titulo h1 {  
  font-weight: 700;  
  font-size: 36px;  
}
```

```
.titulo h3 {  
  font-weight: 400;  
  font-size: 24px;  
}
```

```
.tema button {  
  align-self: flex-end;
```

```

font-size: 24px;
font-weight: 400;
padding: 8px 16px;
background: #ECF4FF;
border-radius: 100px;
}

.projetos {
display: flex;
flex-direction: column;
background: linear-gradient(230.65deg, #499cfe 27.49%, #9cc8fc 83.19%);
margin-top: 32px;
padding: 32px;
box-shadow: 2px 2px 4px rgba(16, 16, 16, 0.42);
border-radius: 20px;
}

//código posterior omitido

```

**Rafaella:** Porém, ainda não conseguimos colocar o botão no exato canto superior direito que precisamos, pois por enquanto está apenas alinhado à direita e centralizado horizontalmente com os demais elementos do cabeçalho.

Então vamos separar em mais uma divisão, e em uma teremos o nome e o título com o subtítulo, e em outra teremos o botão com suas configurações.

De volta ao nosso arquivo HTML, englobaremos a foto com o texto do cabeçalho dentro de uma tag `<div>` abaixo do `<header>`, cuja classe chamaremos de `"perfil"` com as informações do perfil.

Já o `<header>` será chamado de `"cabeçalho"`, pois englobará os elementos de perfil e o botão de alterar o tema.

```

<body>
  <div class="container">
    <header class="cabecalho">
      <div class="perfil">
        
        <div class="titulo">
          <h1>Rafaella Ballerini</h1>
          <h3>Instrutora e desenvolvedora front-end #imersaodev</h3>
        </div>
      </div>
    </header>
  </div>

```



```

    </div>
    <div class="tema">
      <button onclick="mudaTema()">Alterar Tema</button>
    </div>
  </header>

  //código omitido
    </div>
</body>

```

**Rafaella:** Perfeito. Vamos ver como o nosso CSS está. O `perfil` agora vai ser a nossa imagem junto com as nossas informações. Vamos precisar mudar a disposição dos elementos, podemos apagar o `align-items: center`; e acima desse `perfil` vamos estilizar o nosso cabeçalho.

Lembrando que nosso cabeçalho tem a tag `perfil` e a tag `tema`, que é o botão e o conjunto de foto com os títulos. Nesse `.cabecalho` do CSS colocaremos `display: flex`, para posicionar os elementos, e `justify-content: space-between`, para justificar os elementos. Queremos que tenha espaço entre os elementos, com o `space-between` colocaremos um espaço entre o perfil, com a foto e o título, e o nosso tema.

```

.cabecalho {
  display: flex;
  justify-content: space-between;
}

```

**Rafaella:** Vamos salvar e rodar para ver como vai ficar.

**Guilherme:** Já mudou. Comparado com o projeto do Figma está correto.

**Rafaella:** Agora queremos fazer o botão funcionar. Como fazemos a parte do JavaScript para ele funcionar?

**Paulo:** Faz uma aula e meia que não mexemos no JS. Agora vai ter uma `function mudaTema`, não é?

**Rafaella:** Exato.

**Guilherme:** Por enquanto, ao clicar no botão só aparece no console que `"mudaTema is not defined"`. A primeira coisa a fazer é criar essa função `function mudaTema()`. Vamos fazer alguma coisa dentro dessa função que vai manipular o CSS. Como fazemos isso, Rafa?

```
function mudaTema() {  
  
}
```

**Rafaella:** Assim como temos aquele `document.getElementById()`, `document.getElementsByName()`, entre outras coisas que usamos no JavaScript para manipular o HTML e o CSS, temos o `document.body()`. Esse `body` é a tag `<body>` do HTML, que engloba todos os elementos que são visíveis na nossa página.

E é justamente aí que queremos atacar. Queremos pegar o nosso `body` e atribuir uma classe para ele. Ao atribuir essa classe, teremos no nosso CSS um CSS para essa classe. Vamos criar uma "classe dark", por exemplo. Vamos escrever no nosso CSS a estilização para essa classe, mas é o JavaScript que vai decidir quando essa classe entra e quando ela sai. Nesse caso, será no `onClick`.

No `onClick` do nosso botão de tema queremos alterar a classe do nosso `body`. Então colocaremos `document.body.classList.toggle()`, `toggle` é o comando que faz isso, traduzido para o português significa "alternância". Vamos alternar a `classList`, a classe do nosso elemento `body`, e colocar `dark` como parâmetro para o nosso elemento.

```
function mudaTema() {  
  document.body.classList.toggle("dark");  
}
```

**Rafaella:** Vamos salvar. Porém, ainda não vai mudar nada em relação ao nosso tema. Porque ainda não fizemos a estilização da classe `dark`. Ao clicar no botão, ele vai mudar para a classe `dark`, mas ainda não temos a estilização desse estilo no CSS.

**Guilherme:** É aquilo que o Paulo comentou em uma das aulas. Esse é um cenário complexo: ao clicar no botão "Alterar Tema" não exibe uma mensagem de erro e não tem o comportamento que esperamos. Vamos inserir alguma coisa no CSS para resolver isso.

**Rafaella:** Vamos escrever esse código no CSS abaixo do código que já temos. Vamos inserir o `.dark`. Ao estilizar esse `.dark`, o que estamos estilizando de fato? A nossa tag `body` quando ela tiver a classe `dark`. O que normalmente estilizamos na tag `body` é o `background`, a fonte - a fonte vai ficar a mesma, não precisa trocar - mas o `background` vamos trocar. Vamos verificar no Figma qual é a cor do `background` do tema escuro.

**Guilherme:** Vou clicar na cor do `dark mode` e copiar o código CSS da cor e do linear-gradient do `background` para colar no nosso CSS.

```
.dark {  
  background: linear-gradient(236.85deg, #375b86 27.26%, #6b87a9  
96.03%);  
}
```

**Rafaella:** Vamos salvar e rodar para ver se o botão está funcionando.

O fundo já está sendo alterado. Vamos, agora, estilizar o resto das coisas. Abre o Figma para ver o que mais precisamos alterar no nosso CSS. Precisamos trocar o nosso `container`, que antes era o espaço branco, e trocar a cor da fonte e a cor do botão. Já pode copiar o código CSS da cor do `background` do `container` e vamos voltar para o CodePen.

Agora temos que lembrar que cada elemento será estilizado de uma forma específica. Podemos colocar o `.container` da classe `dark`, então `.dark .container`. Vamos colocar o código da cor que você copiou do `background`. Além disso, também trocaremos a cor da fonte, vamos pegar lá no Figma qual é a cor da fonte no modo `dark`.

```
.dark {  
  background: linear-gradient(236.85deg, #375b86 27.26%, #6b87a9  
96.03%);  
}  
  
.dark .container {  
  background: #333439;  
  color: #f6f6f6;  
}
```

**Guilherme:** Vamos testar. Agora ficou legal. A fonte do texto de "Projetos" também ficou branca no modo `dark`, mas ficou estranho. O pessoal que fez o layout no Figma também percebeu que era melhor deixar essa fonte dos projetos e os nomes dos projetos em uma fonte escura.

**Rafaella:** Vamos ver no HTML qual era a tag dessa parte de projetos. Podemos pegar essa `classe="projetos"`, porque queremos trocar a cor de todos os elementos. Então vamos escrever no CSS, `.dark .projetos` e colocar color escura igual a que está no Figma.

```
.dark .container {  
  background: #333439;  
  color: #f6f6f6;  
}
```

```
.dark .projetos {  
  color: #1c1c1c;  
}
```

**Rafaella:** Vamos salvar e rodar. Agora não está mais mudando a cor da fonte dos projetos. Outra coisa que devemos fazer é alterar o botão, porque no *dark mode* ele também fica escuro.

**Guilherme:** Já vou copiar o código da cor do botão no Figma.

**Rafaella:** Vamos criar o `.dark .tema button`, e dentro dele a cor do *background*. Além disso, temos que mudar a cor da fonte do botão. Além disso, podemos mudar também a borda do botão com o `border` de 2px, que é o tamanho da borda; `solid`, para ser uma borda sólida e a cor pode ser um pouco mais clara.

```
.dark .tema button {  
  background: #1c1c1c;  
  color: #ffffff;  
  border: 2px solid #f7f7f7;  
}
```

**Rafaella:** Vamos salvar e rodar. Clica em "Alterar Tema". Está ótimo.

**Guilherme:** Ficou bem legal. Vamos colocar esse projeto no ar para que qualquer pessoa, com o link que vamos passar, possa visualizar esse layout.

**Paulo:** Quando temos um projeto web, que é o caso do projeto que estamos desenvolvendo. Aqui está dentro do CodePen, que é uma ferramenta web e você só vai ver no CodePen. Às vezes queremos colocar isso no nosso site ou tem outros sistemas que permitem que você faça isso.

Então, quando queremos hospedar, existem vários serviços de cloud (na nuvem), tem ferramenta da Amazon, da Oracle, do Azure, do Google, tem esses grandes. Mas tem serviços menores de host de website. Tem algumas ferramentas para devs, como o GitHub, que até dá um espaço gratuito para fazermos isso. Não foi à toa que mostramos o GitHub na aula passada e estamos te provocando para usar o tal do GitHub porque, querendo ou não, você vai acabar usando GitHub na sua vida.

Ele não é feito para colocar sites no ar. Mas ele é feito para guardar, chamamos de repositório, de sistema de controle de versão. Estamos criando essa página, tudo o que fizemos de mudança; se estou trabalhando com a Rafa, o que a Rafa mexeu, o que eu mexi. Um dos objetivos do GitHub é te ajudar a organizar todo o *workflow* que um time precisa ter.

Tem uma série de complexidades, estamos deixando alguns links sobre o GitHub, alguns vídeos do Mario Souto, da série de GitHub para sobrevivência. E agora vamos usar mais esse mecanismo de colocar o site no ar usando o GitHub. Dá para fazer em diversos outros lugares, vocês estão convidados a colocarem onde preferirem.

Agora o ideal é pegar esse site que fizemos e colocar no ar. Já está no ar no CodePen, mas vamos fazer de um jeito um pouco mais tradicional, mais profissional, vamos chamar assim.

Lembrando que também vamos colocar nos desafios finais que você deixe esse projeto mais completo. No Figma tem até um espaço para as conquistas, mas dá para fazer um monte de coisas para deixar esse projeto incrível e melhor que o seu LinkedIn. Esse é o nosso objetivo, que você saia da Imersão Dev com isso no ar, linkando com seus projetos, mostrando o que você fez e a sua capacidade.

**Guilherme:** E o que precisamos para conseguir vincular esse projeto que fizemos no CodePen lá no LinkedIn e ter um link para mostrar para várias outras pessoas?

A primeira coisa a fazer é acessar o [GitHub](#). Eu já estou logado. Mas se você não estiver, no canto superior direito vai ter um botão "Sign up", clica nele e faz um cadastro para logar no GitHub.

Para conseguirmos pegar o código no CodePen e passar para o GitHub, vamos criar uma pasta, também chamada de repositório, para manter todos os códigos do CodePen. Para criar essa pasta, vou clicar no ícone de mais, no canto superior direito, e clicar em "New Repository". Vamos ter que escolher um nome para esse repositório. Qual vai ser o nome, Rafa?

**Rafaella:** Pode ser "certificard", é um bom nome para o nosso certificado.

**Guilherme:** Boa. Lembrando que não podemos ter dois repositórios com o mesmo nome, o GitHub faz essa verificação. Agora, não vou me preocupar com os outros campos, a descrição do projeto, se ele é público ou privado, readme, gitignore, não vamos nos preocupar com isso. Vamos clicar no botão "Create Repository".

Em seguida, aparecerá uma página com algumas linhas de código. É possível enviar os arquivos do CodePen para o GitHub só com linhas de código no terminal. Não é o que vamos fazer. Nós vamos fazer upload de arquivos existentes, clicando em "uploading an existing file". Vai aparecer um quadro para passarmos nossos códigos para o GitHub.

Mas antes, vamos precisar fazer download do nosso código no CodePen. No canto inferior direito do CodePen existe um botão "Export", ao clicar nele temos as opções de salvar como Gist do GitHub, não é o que queremos, e tem o "Export.zip" que é a opção que escolheremos. Quando eu clicar em "Export.zip", ele vai preparar o ZIP e avisar que já podemos fazer o download dos arquivos.

**Paulo:** Aposto que é um arquivo para cada linguagem: CSS, HTML e JavaScript.

**Guilherme:** Isso. Vou abrir esse arquivo que baixamos, "certificardaula-10.zip". Ele vai extrair. Se você estiver usando Windows, pode usar o WinRar ou algo do tipo para extrair os arquivos do ZIP. Dentro dele tem uma série de arquivos, tem o "license.txt"; "README.markdown"; uma pasta "src" com index, script e style; e a pasta "dist", que é a pasta que vamos usar.

Vamos entrar na pasta "dist". Lembra que a Rafa comentou, na aula anterior, que o HTML consegue conversar com o CSS e com o JavaScript? Esse arquivo `dist` já tem isso, vou abrir o arquivo "index.html" da pasta "dist" no Visual Studio Code para visualizarmos essa parte no código.

Quando abrimos o arquivo `index.html` do `dist`, na linha 6, temos, `<link rel="stylesheet" href="./style.css">`, ele já tem um link. Se abrimos o `source` direto, ele não tem. Vamos abrir para verificarmos, e também ao link, como o VSCode. Esse link traz exatamente o código que estávamos vendo anteriormente, não é o que queremos. Nós queremos o código do `dist` já com CSS e JavaScript funcionando.

Então, vamos selecionar, não do `src`, mas do `dist`, os três arquivos e arrastar para o VSCode. Selecionaremos os três links do `dist` e arrastaremos para dentro do VSCode. Assim, ele pegará os três arquivos e mandará para o GitHub. Mais abaixo, há uma palavra que nos depararemos sempre como Devs que é *Commit*.

O **Commit** é a linha do tempo. No "Commit changes" eu "adicionei os arquivos do certificard". Isso nos permite perceber as alterações que estamos realizando no GitHub, temos o *commit* em uma foto - que nós colocamos - dos arquivos que fizemos e precisamos alterar um script, o `index.html`, por isso faremos um novo *commit*, para ficar registrado.

Ao fazer o *commit* ele processa os arquivos e conseguiremos visualizar, mas não a página, como a Rafa e o Paulo esperam. Quando selecionamos o `index.html`, aparece o `certificard` com todo o código que desenvolvemos. O mesmo para o `script.js` e para o `style.css`. Algo interessante do GitHub é que ele mostra porcentagem de HTML que há no repositório de JavaScript.

Para que seja possível ter, de fato, um link que abre o nosso projeto, acessaremos "Settings" ou configurações. Se "scrollarmos" a página, encontraremos uma série informações, sendo, uma delas, a "GitHub Pages" que nós apertaremos. Na próxima página, selecionaremos o local onde desejamos inicializar o nosso projeto.

Porém, antes disso, precisamos saber que, no "certificard", implicitamente está dito: tudo que você está fazendo aqui, está na "Branch > main". Quando criarmos e ativarmos o GitHub Pages, precisamos indicar: o meu código principal está na "main".

Portanto, retornando agora ao GitHub Pages e falamos: o GitHub Pages está desabilitado. Qual é a origem das páginas do projeto?

No primeiro campo, vamos selecionar "main" e salvar. Logo, ele apresentará uma mensagem avisando que o nosso site está pronto para ser publicado, "certificard". Vamos selecionar o link com o botão direito, acessá-lo e receberemos uma mensagem de erro, "404". Retornando ao "GitHub Pages" repare que estamos usando uma palavra chamada *root*. Outros *frameworks* compreendem melhor o que é esse arquivo *root*.

O *root* conseguirá identificar por onde o projeto começa. No nosso caso, como só temos três, note que quando selecionamos o *root*, ele mostra: o *root*, que é para *frameworks* mais complexos; e o *docs*. Nós escolheremos esta segunda opção, *docs*, apertaremos o botão "Save", ele apresenta "certificard". Agora, atualizaremos a nossa página e está tudo no ar.

Para isso acontecer, nós passamos por dois pontos importantes: A questão da "Branch", que é sobre o local em que estamos fazendo, e nós indicamos que é no "main", no principal; e não utilizamos o *root*, isto é, não estamos usando um *framework* JavaScript (em que uma parte do *root* aponta para a página índice). Optamos pelo *docs* e, a partir desse momento, já temos o nosso "certificard" funcionando corretamente.

Caso você queira colocar outro jogo, assim como a Rafa sugeriu, por exemplo, o Super Trunfo ou o Aluraflix, o processo é o mesmo.

**Paulo:** O que você quer dizer, Guilherme, é que é possível criar mais um, isto é, outro projeto, que não se chama certificado, mas, sim, Aluraflix e fazer o download, o export, só do Aluraflix. Não estamos falando de "embedar" dentro do projeto, mas de deixar o Aluraflix em um endereço na internet para as pessoas acessarem apenas ele.

Podemos até fazer de forma mais elaborada, por exemplo, "linkar" do "certificard" para o Aluraflix, que também está no GitHub e fazer uma mistura de sites e projetos. Nós passamos por alguns detalhes do GitHub e, às vezes, os menus mudam de lugar. Esses detalhes podem te barrar, ficar dando "4004". Se isso acontecer, não deixe de nos mandar no Discord.

Nós também deixaremos alguns links sempre atualizados para se alguma questão, no GitHub, acabar mudando onde está o seu site. Repare que, agora, apesar de estar meio estranho, já que o site da Rafaela está dentro da URL do Guilherme, aprendemos a como ter um endereço próprio para colocarmos no rodapé dos e-mails. Quem acessar esse e-mail pode consultar o seu site e descobrir o que você já fez, o que está estudando e visitar o seu LinkedIn.

Inclusive, seu LinkedIn, Instagram, ou o que mais você quiser pode estar do lado, no topo da página, do seu nome, como, no nosso exemplo, "Rafaella Ballerine" com os logos ou no footer. Enfim, é possível deixar o site muito completo e bonito.

**Guilherme:** Isso que você comentou, Paulo, de deixar o site aberto, ele está, literalmente, aberto. Qualquer pessoa no planeta que digitar `guilhermeonrails.github.io/certificard/` terá acesso aos projetos que a Rafa fez durante a Imersão.

**Rafaella:** Então, é isso, pessoal! Estamos no final e esperamos que vocês tenham gostado de tudo que fizemos até então. Foram vários projetos e muitos deles vocês podem estilizar da maneira que preferirem. Tenho certeza que, a partir daí, surgirão várias ideias, e que vocês criarão muitas coisas para o *portfolio* de vocês.

Não há forma melhor para aprender do que praticar e ensinar. Por isso, aproveitem a semana para rever alguns pontos, pesquisar e entender o que está acontecendo. Tenho certeza que vocês aproveitarão bastante! Me contem também o que vocês gostaram mais, da parte de estilização ou de lógica? Isso é interessante para você se conhecer e escolher qual caminho seguir na programação.

Dentro da tecnologia existem várias áreas, desde conteúdos que não tem a ver com programação, por exemplo, o design. As pessoas do design que desenvolveram o Figma. Em outras áreas, você pode trabalhar com projetos e também existe a área da programação que é dividida entre: Front-End, relacionado com interface, HTML, CSS e JavaScript; e Back-end, relacionado à lógica e às funções, enfim, ao que acontece na plataforma.

Tente analisar isso para que você possa começar, a partir de agora, a traçar o seu caminho. Esperamos que você tenha gostado!

**Guilherme:** Minha mensagem pessoal é: não desistam da carreira que vocês almejam para a sua vida profissional. Corram atrás! Não é fácil, a jornada é complexa. Nós já estamos cientes dessa trajetória difícil pelos filmes do Star Wars e outros filmes geeks que tanto gostamos. Conhecemos a jornada do herói, composta de vitórias e derrotas. Há dias em que o projeto não dá certo, não funciona, e você acha que "não é para você". Não desista!

Toda grande caminhada começa com pequenos passos, então, que a Imersão seja o primeiro passo da sua jornada para uma nova profissão, uma nova carreira, para um novo momento da sua vida. Me sinto muito honrado por esse momento com a Rafa e o Paulo e por você ter nos acompanhado durante a Imersão e ter se dedicado.

A Imersão começa com muitos estudantes, pessoas de todas as partes do Brasil e do mundo, e nós temos respostas tão positivas, de pessoas que realmente mudaram, conseguiram, e é possível! A minha mensagem é essa. Minha motivação será, para sempre, essa: é possível mudar e mudar para melhor! Ter coisas boas e um futuro melhor na programação. Não é fácil e nem será "do dia para a noite".

Não sei quantos dias ou anos levará, mas é possível! Acredite no seu sonho e na sua capacidade. Não seja uma pessoa tão empolgada, seja uma pessoa dedicada, que



sentar para estudar, como nós fizemos, aprender determinado conteúdo, absorver, praticar, treinar, refazer, pegar uma aula da Imersão e refazer a aula sozinha, assim, tenho certeza que você terá muito sucesso na carreira profissional.

**Paulo:** Guilherme, você poderia abrir o site da Alura? Já que muitas pessoas estão tendo agora o primeiro contato com a Alura e gostariam de entender melhor o que é a nossa escola. Tem um artigo que eu escrevi e se chama "Aprender a programar dez anos". Na verdade, eu traduzi um pedaço de artigo muito famoso da internet do professor **Peter Norvig**, "Teach yourself programming in 10 years".

Ele está relacionado com o que o Guilherme colocou, porque, a jornada não dura os dez dias. Os dez dias da Imersão Dev não te tornarão um Dev. O professor Peter Norvig, pesquisador do Google e de Inteligência Artificial fala que para você dominar uma linguagem, precisa de muitos anos, dez anos, talvez. Ele faz uma provocação para que não nos empolgarmos pensando que já sabemos tudo ou que aprendemos uma linguagem muito rápido.

De fato, algumas pessoas aprendem muito rápido, mas, quando falamos em experiência, pensamos em construir uma carreira. Daqui a dez anos, você estará trabalhando em uma carreira, se souber investir no seus estudos, na prática. Intercalando momentos de mais estudos e outros de menos, temos grandes chances de melhorar enquanto profissionais.

Sei que algumas pessoas não tiveram os mesmos privilégios que eu tive na vida ou que o Guilherme e a Rafaella tiveram. Também sei que existem injustiças muito grandes e que as pessoas não têm as mesmas oportunidades, mas, acho que a tecnologia é um pouco melhor para estudar, por poucas coisas serem necessárias.

Existe o problema do acesso. Eu sei que muitas pessoas precisam, infelizmente, assistir à Imersão pelo celular. Até dá! Com certeza é um obstáculo gigante, mas dá. Se nos programarmos para saber, nos próximos dez anos, onde queremos estar na nossa carreira e o que desejamos fazer, não sairá exatamente como esperamos, mas ter esse planejamento é muito importante.

Eu queria dar as **boas-vindas na carreira de Dev** e agradecer o mergulho na Imersão. Por mais uma semana, estaremos no Discord conversando com vocês, tirando dúvidas, explicando sobre a carreira, realizando encontros ao vivo, streamings, enfim, são muitas atividades. Sem contar que você já está nos seguindo no canal do YouTube da Alura e no Podcast do Hipsters, no canal da Rafaella e no Instagram do Guilherme.

São várias pessoas no Aluraverso! Nós temos uma lista e vamos mandando para você nos e-mails. Queremos que você se aproxime e respire! Esteja você em um momento de mergulhar de vez e encabeçar tudo na carreira ou ainda "tateando", entendendo melhor o que gostaria de fazer. Nós temos dado cada vez mais atenção à Imersão Dev, porque percebemos que tem sido o primeiro passo de muitas pessoas, não só na Alura, mas para a carreira de tecnologia.

**Parabéns a você!!** Agradeço a todas as pessoas que participaram. São muitas! Na edição, no time do Scuba, pessoas desenvolvendo o roteiro, fazendo imagens, editando, gravando, enfim, são muitas pessoas envolvidas mesmo! Não é a toa que a Imersão alcançou esse tamanho. Última coisa, você pensou que a Imersão era de graça? Não, você precisa dar algo em troca!

Estou pedindo para que você faça uma avaliação do que você aprendeu, faça um vídeo ou um texto explicando algum dos projetos, o que você mais gostou, a parte de JavaScript ou de CSS e que você reproduza o conhecimento que adquiriu e mostre o seu portfolio, conte onde você quer chegar.

Falar e mostrar o seu código, facilitará o seu aprendizado, reforçará seu conhecimento e, quem sabe, até ajudará a expor seu portfolio para as empresas, para que você comece a encontrar as vagas de emprego. Sem promessa nenhuma, mas esse é um formato que também nos visibiliza. Você falará da Alura, da Imersão Dev e trará outras pessoas para que, no futuro, nas próximas edições da Imersão Dev, elas façam esse mesmo caminho que você trilhou.

Muito obrigado!! Espero você nessa carreira, nessa comunidade e no mercado de tecnologia.

**Rafaella:** Obrigada, pessoal!!

**Guilherme:** Tchau!!