

Universidade Paulista

Ciência da Computação

Relatório referente a atividade solicitada para
entrega até o dia
02/06

DESENVOLVIMENTO DA PLAYLIST

Diogo Sousa Nogueira - N539CF-0
Enrico Brosque Palma - F271EI-0

Bauru
2020-05-31

Introdução

Relatório sobre o desenvolvimento do website Bellpepper (nome ilustrativo). O website tem um modelo de playlist onde o usuário poderia incluir suas músicas favoritas e curtir um som enquanto exerce outras atividades.

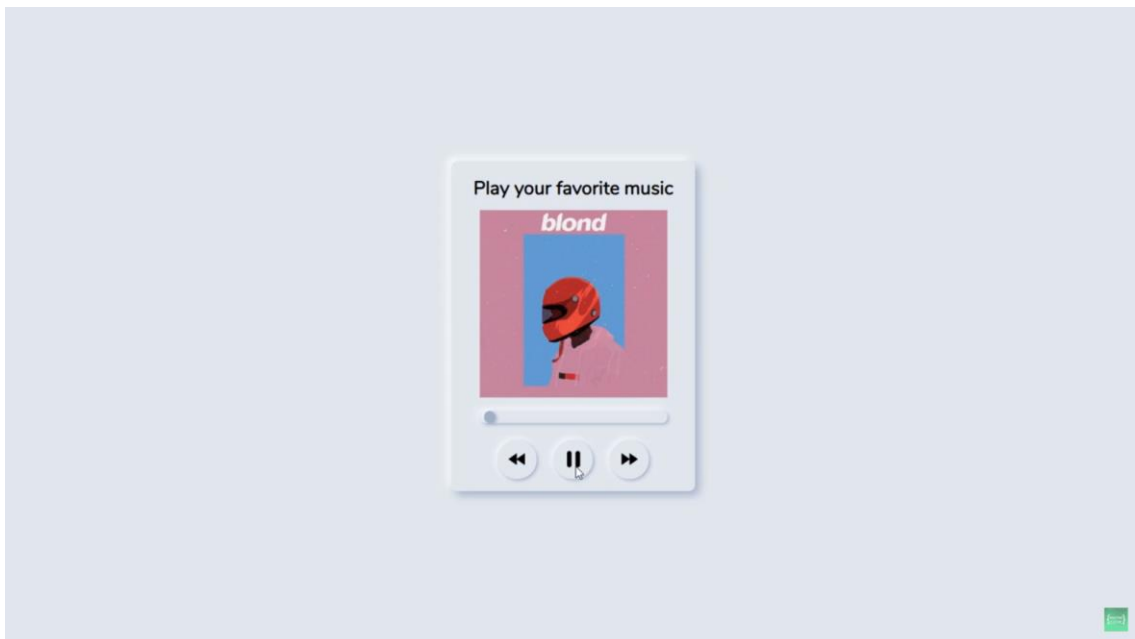
No trabalho foi utilizado:

- Figma
- Visual Studio Code
- HTML5
- CSS3
- JavaScript

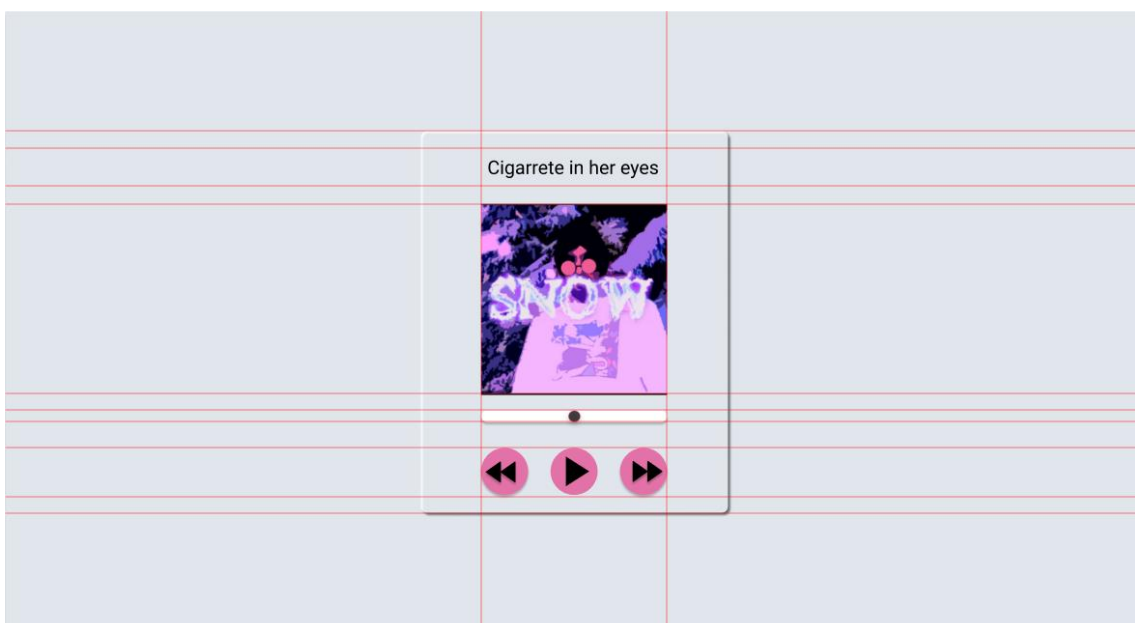
Protótipos

Para o protótipo, tínhamos em mente um design similar ao do Spotify, porém acabamos achando um vídeo cujo design nos agradava mais.

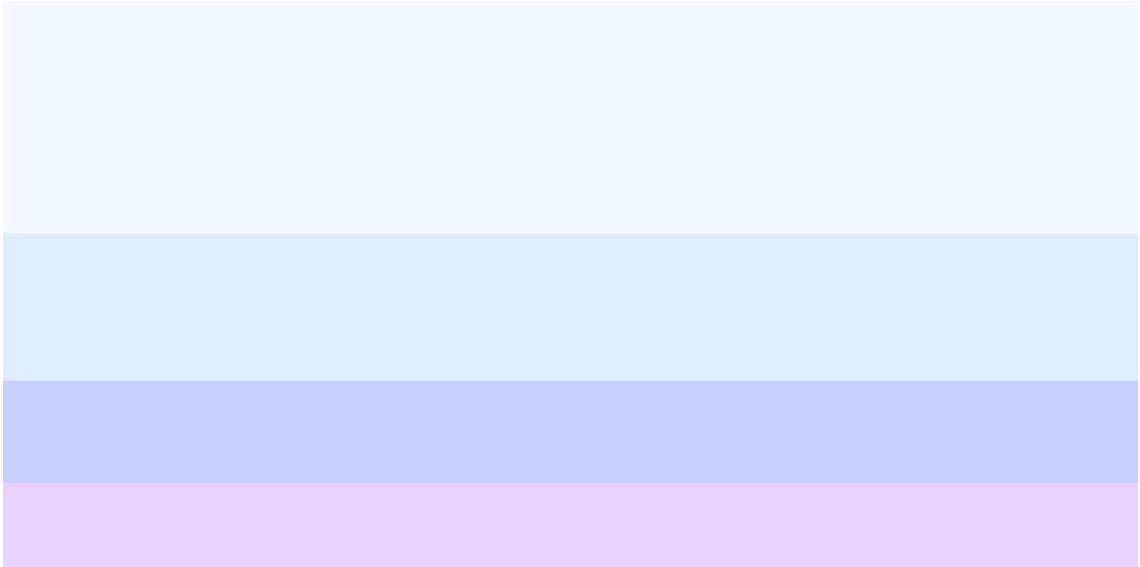
Passamos o protótipo para o Figma visando testar algumas cores, mantivemos o tom de azul porém o alteramos no projeto final, para uma paleta de cores pasteis.



Design usado de base.



Design feito no Figma.



[Paleta de cores](#) pasteis utilizadas no projeto final.

Conteúdo

O site foi desenvolvido no software Visual Studio Code, o plano era ter uma playlist com duas ou mais músicas, quando clicasse no botão “próximo” ou “anterior” a música e seu título seriam trocados junto com a capa do álbum correspondente.

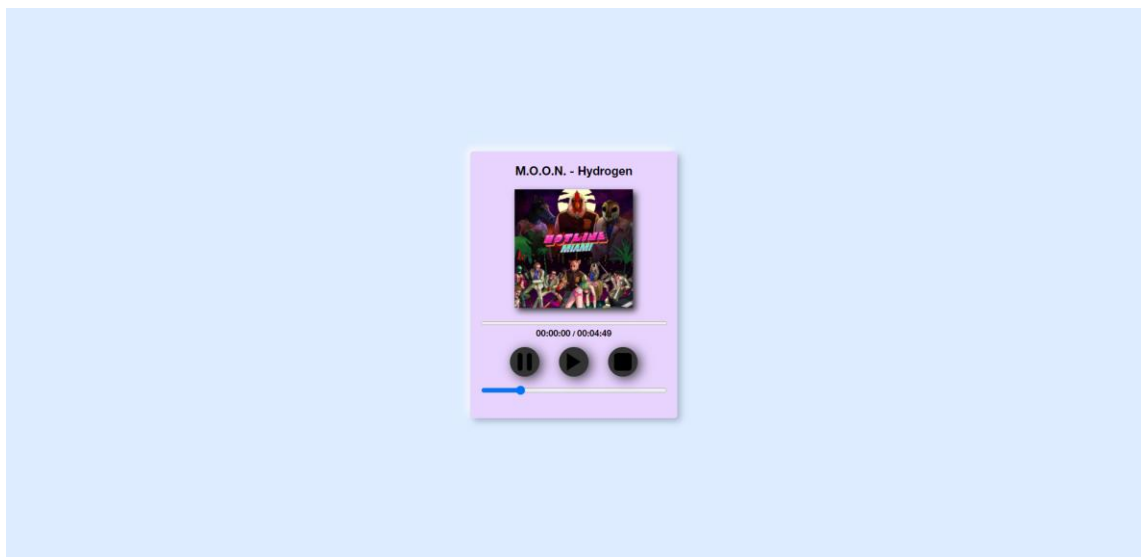
Como citado em Protótipos, foi escolhido um site com cores pasteis e de tonalidades fracas, para acrescentar um contraste e um efeito de Material Design nós deixamos os elementos com efeitos de sombra, dando a impressão de profundidade e luz. Efeito pode ser percebido no player e nos botões dentro do mesmo.

```
box-shadow: 5px 5px 11px #a2b1c6, -5px -5px 11px #ffffff;
```

Efeito de sombra presente na box do player.

```
box-shadow: 5px 4px 20px rgba(26, 25, 25, 0.87);
```

Efeito de sombra presente nos botões do player.



Site com efeitos aplicados.

M.O.O.N. - Hydrogen



00:00:46 / 00:04:49



Player detalhado de perto.

CSS

Todo o design final do site foi trabalhado em cima de tendências atuais, onde as sombras dão o efeito de realidade aumentada e as cores são claras/pasteis para trazer uma paz ao usuário.

```
.botaoPlayer {  
  width: 50px;  
  height: 50px;  
  border-radius: 50%;  
  background-color: #333333;  
  box-shadow: 5px 4px 20px rgba(26, 25, 25, 0.87);  
  position: relative;  
  margin: 0 15px 10px;  
  display: inline-block
```

Nesta tela foi utilizado um background de cor neutra escura, porém não totalmente preta para existir um contraste entre a cor do botão e da sombra aplicada.

Os círculos dos botões são feitos através de DIVS, com uma cor de fundo e um box-shadow aplicando o efeito de sombra, também aplicado alguns espaçamentos para criar uma harmonia e um alinhamento melhor entre os componentes.

Para a aplicação principal também foi usado uma classe com as seguintes propriedades:

```
.containerPlayer{  
  width: 350px;  
  height: 450px;  
  border-radius: 5px;  
  text-align: center;  
  background : #e8d3ff;  
  left: 50%;  
  top: 50%;  
  transform: translate(-50%, -50%);  
  position: absolute;  
  box-shadow: 5px 5px 11px #a2b1c6, -5px -5px 11px #ffffff;
```

Um background com uma das cores pasteis, posições escaladas em 50% de sua posição para a centralização total, tanto com a classe left, top, transform com um translate(-50%, -50%) que seria a definição em 2d das transições no caso, transições em direções opostas, tudo para manter o contêiner sempre no centro. Da mesma maneira que nos botões foi aplicado o box-shadow com suas posições e intensidade e logo depois as mesmas posições porém com símbolos negativos, para aplicar uma sombra idêntica porém do outro lado, assim mantendo a simetria.

JavaScript

Uma consideração que tivemos ao produzir o site foi a questão do volume muito alto por default, foi possível alteração visual pelo HTML e o som em si pelo JavaScript.

```
<input type="range" min="0.0" max="1" step="0.1" class="volume" value="0.2" onchange="volume(this)">
```

value="0.2" corresponde a 20% visualmente.

```
//Página carregada
function onLoadPage() {
    //Acessar controle Audio via JavaScript
    audio = document.getElementById("audio");

    audio.addEventListener("play", onPlayMusic, false);
    audio.addEventListener("timeupdate", onAtualizarTempo, false);

    audio.volume = 0.2

    document.getElementById("titulo").innerHTML = audios[0];
}
```

Dentro da função onLoadPage para quando carregada a música começar com 20% de seu volume máximo e puxa de uma lista criada em JS o nome do arquivo.

A ideia de criar uma lista foi para que em uma próxima atualização já seja possível adicionar mais arquivos e automaticamente com o JS, o player puxar o nome do mesmo e mostrar como título da música.

```
var audios = ["M.O.O.N. - Hydrogen", "LSD - Audio"]
```

```
<p id="titulo"></p>

<div>
```