

Documento de Requisitos do Sistema (Revelations)
Versão 1.0

Cronograma do Projeto

Data	Descrição
25/05/2023	Definição da equipe, tema do projeto e criação do repositório.
07/06/2023	Primeira reunião da equipe e entrega da base da landing page.
08/06/2023	Adicionado footer ao site.
09/06/2023	Adicionado javascript para carregamento de notícias e quizzes.
12/06/2023	Adicionado página sobre nós ao site.
13/06/2023	Correções finais, entrega e gravação do vídeo.

Equipe de desenvolvimento

Nome	Função
Arthur Xavier	Desenvolvedor Front-End
Augusto Medeiros	Desenvolvedor Front-End e Back-End
Pedro Morais	Desenvolvedor Front-End
Rafaella Veiga	Desenvolvedora Front-End

1. Introdução

Este documento especifica os requisitos do sistema *Revelations*, fornecendo aos desenvolvedores e stakeholders as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

1.1.1 O Projeto consiste na criação de um sistema web voltado para o público jovem e curioso sobre as notícias das personalidades do momento e questionários com intuito de entreter.

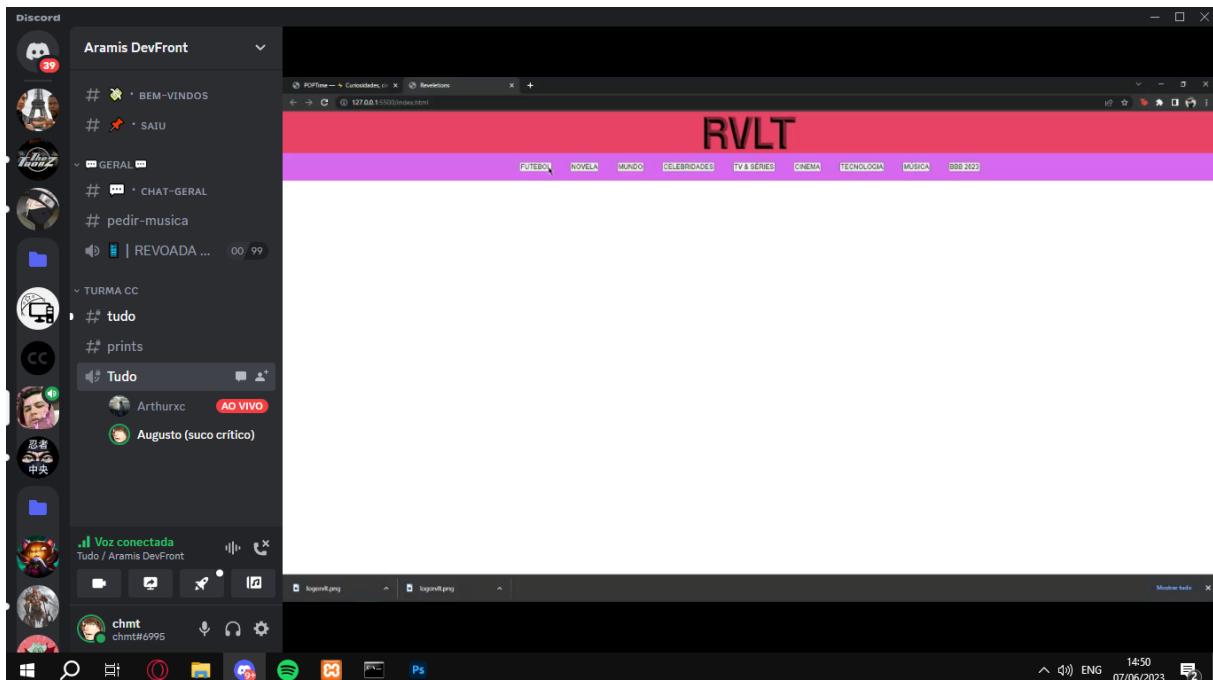
Link do repositório: <https://github.com/augustovmedeiros/Revelations>

Link do vídeo: <https://www.youtube.com/watch?v=StXMD6SbBBw>

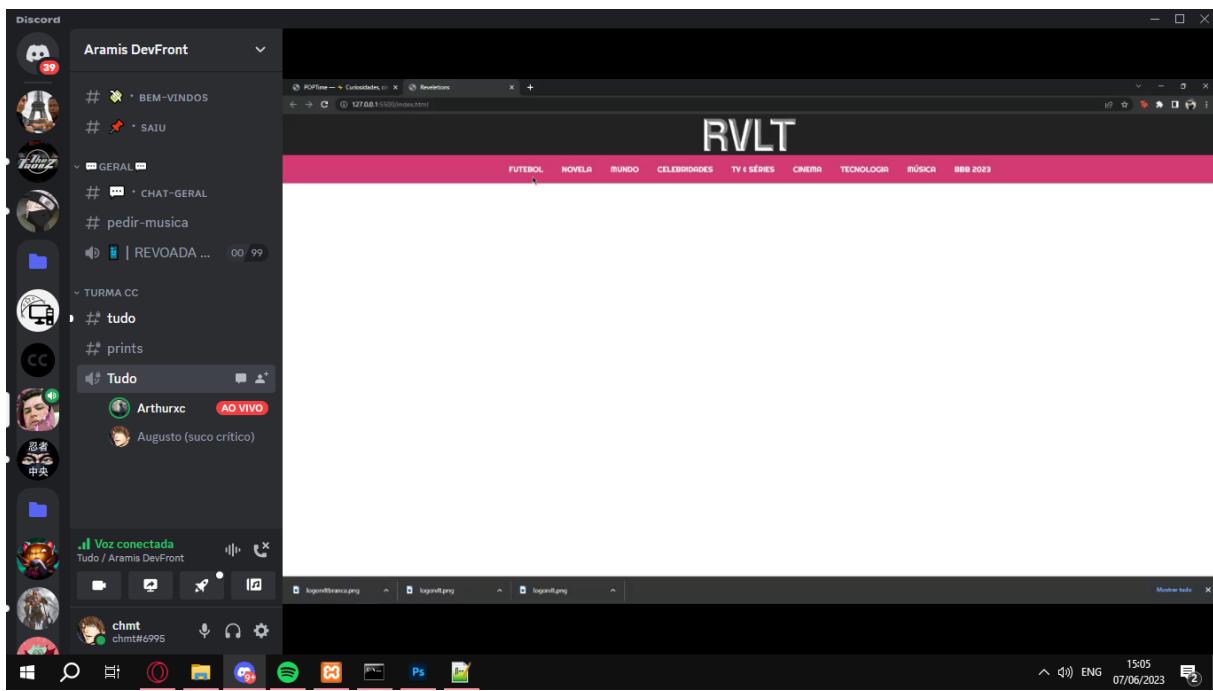
2. Descrição geral do sistema

O Sistema consiste em uma landing page indexadora de notícias e questionários, que estarão espalhados pela página e pelas laterais, os questionários estarão nas laterais enquanto as notícias serão apenas manchete, imagem e descrição. Também teremos uma página sobre os desenvolvedores da equipe.

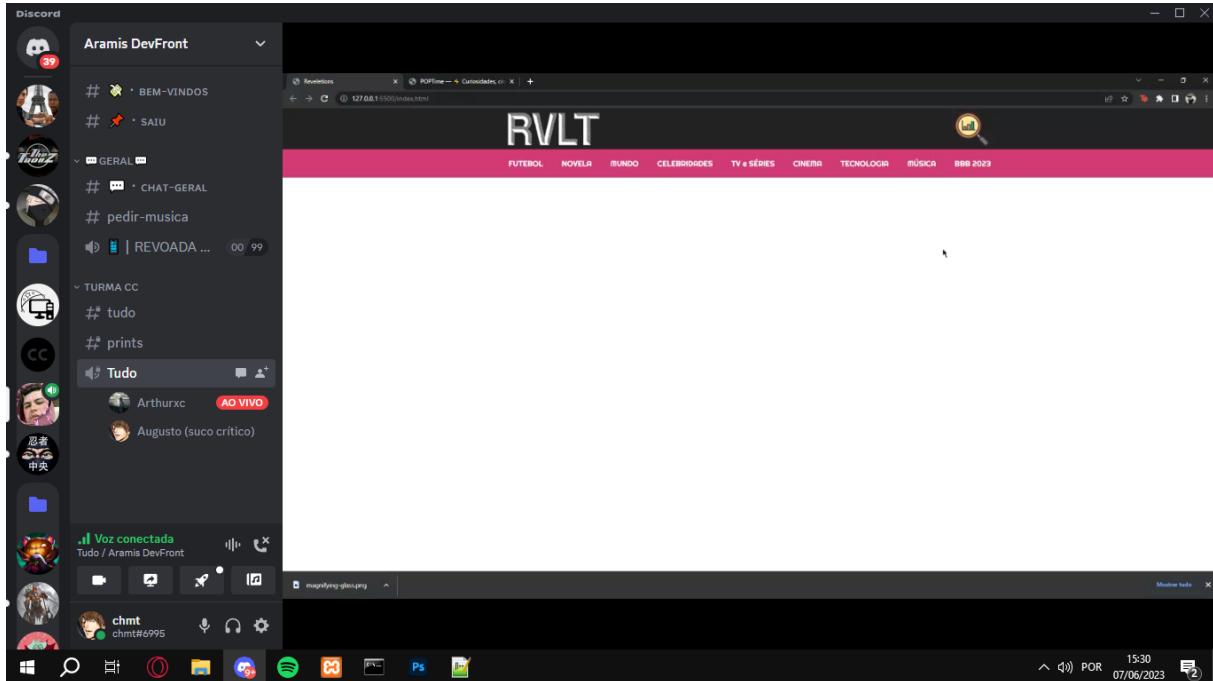
3. Requisitos Funcionais



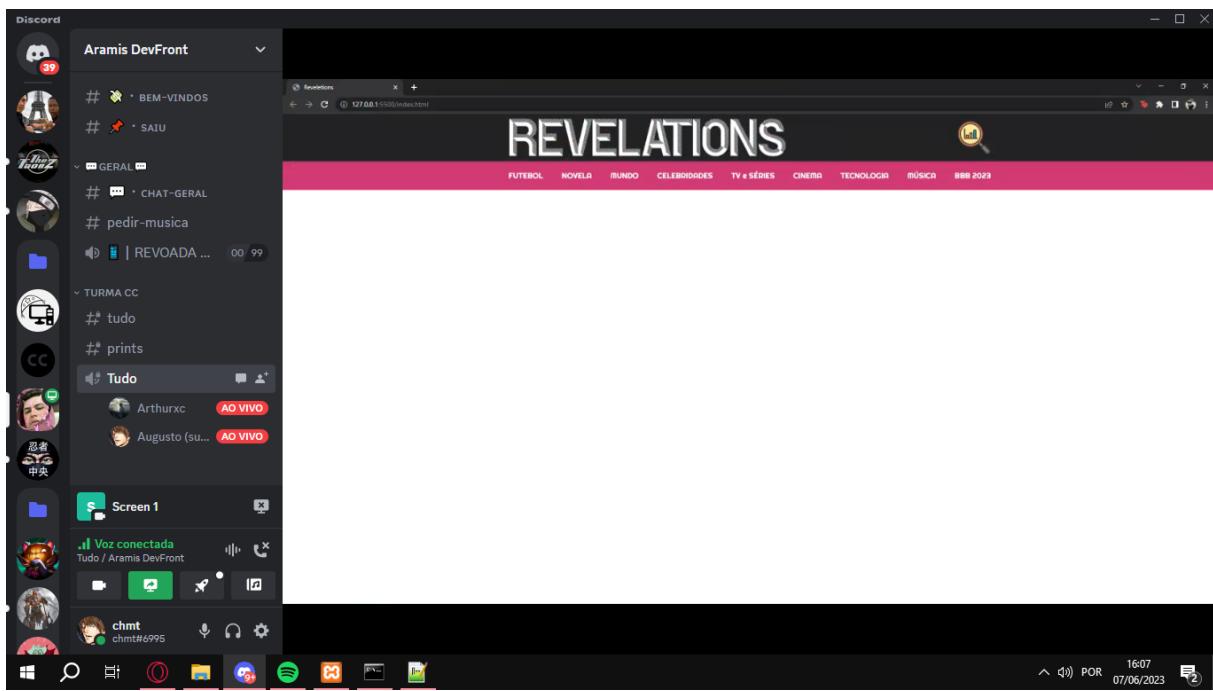
Um primeiro esboço do header e da navbar.



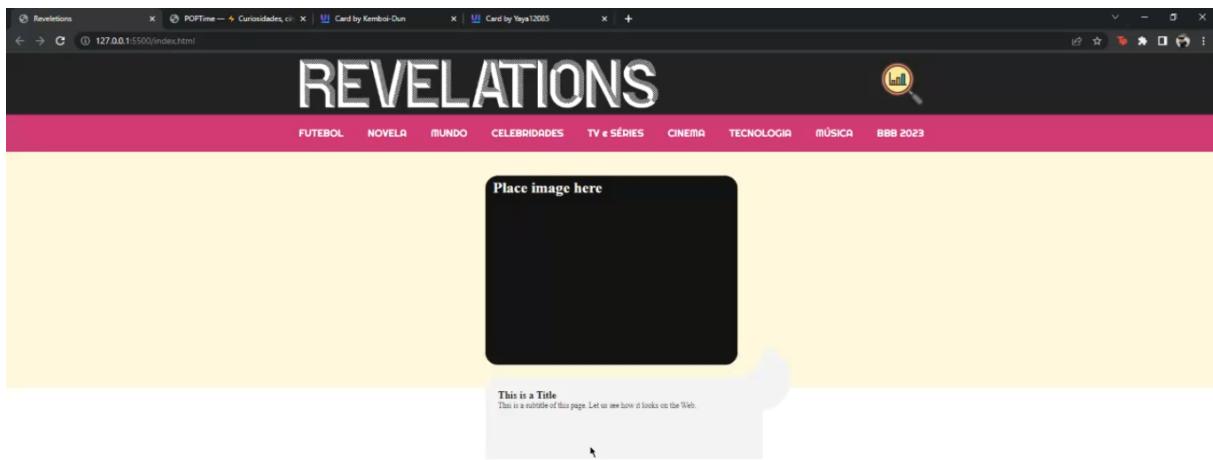
Aqui estávamos chegando na ideia que queríamos, porém ainda parecia vazio ao nosso ver.



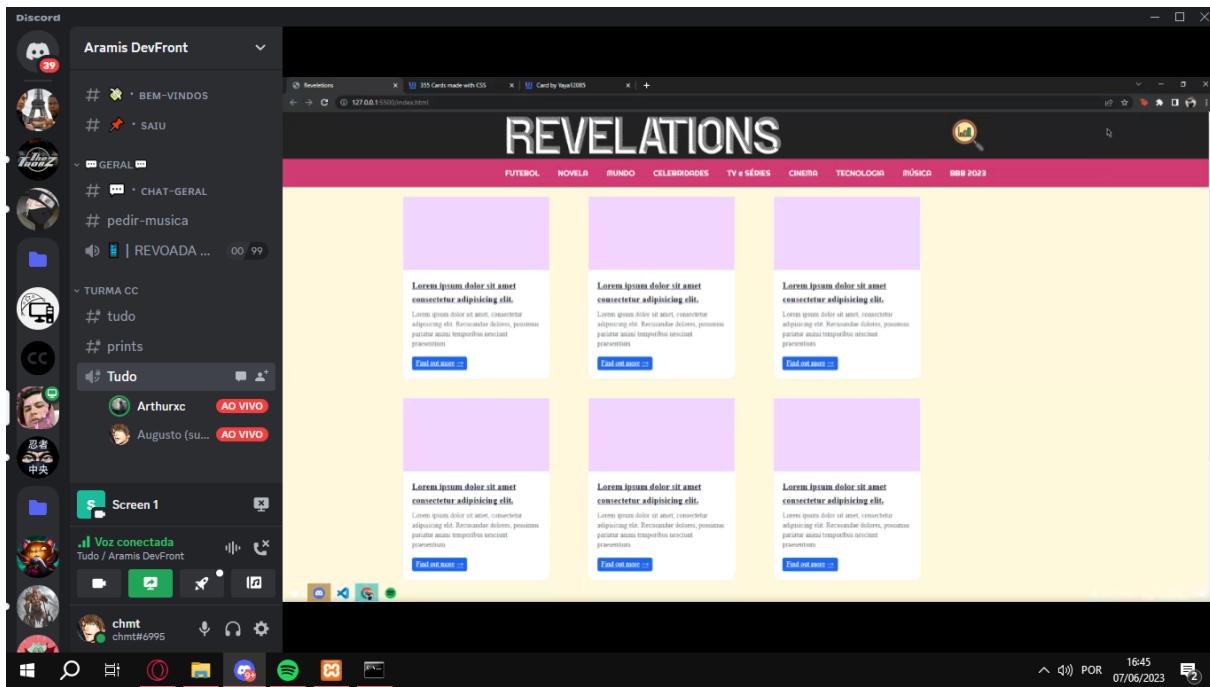
Trocamos a posição da logo e adicionamos um botão para acessar a futura página sobre nós.



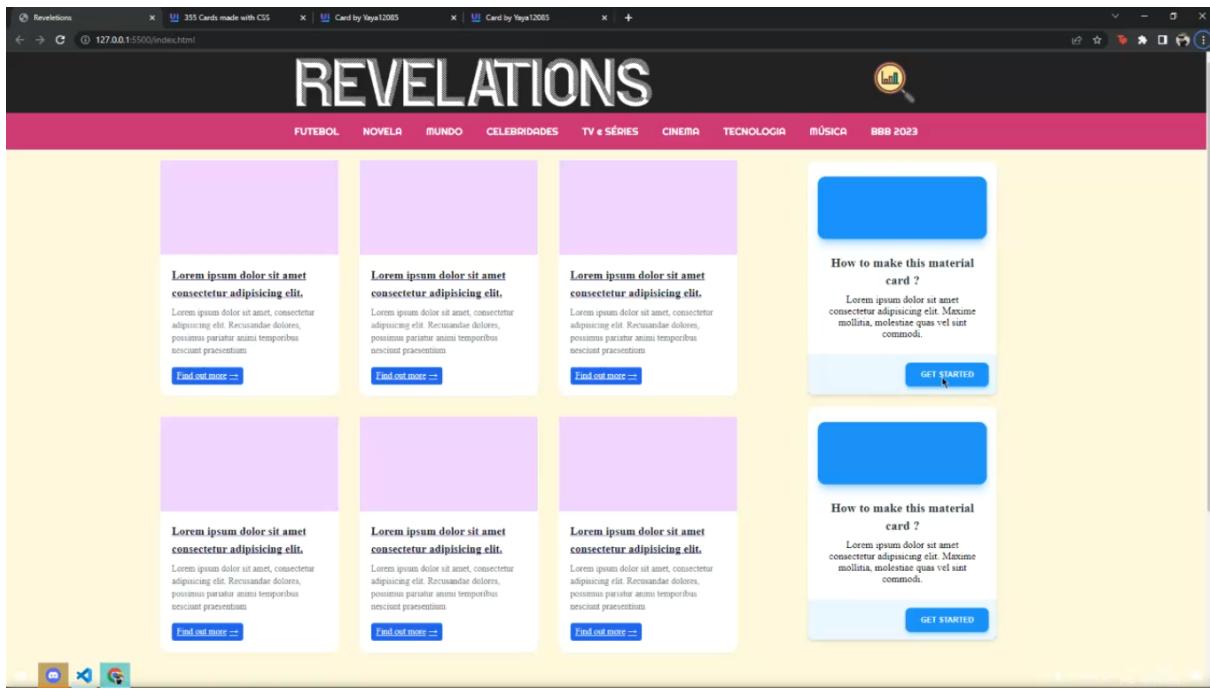
Trocamos a logo no header.



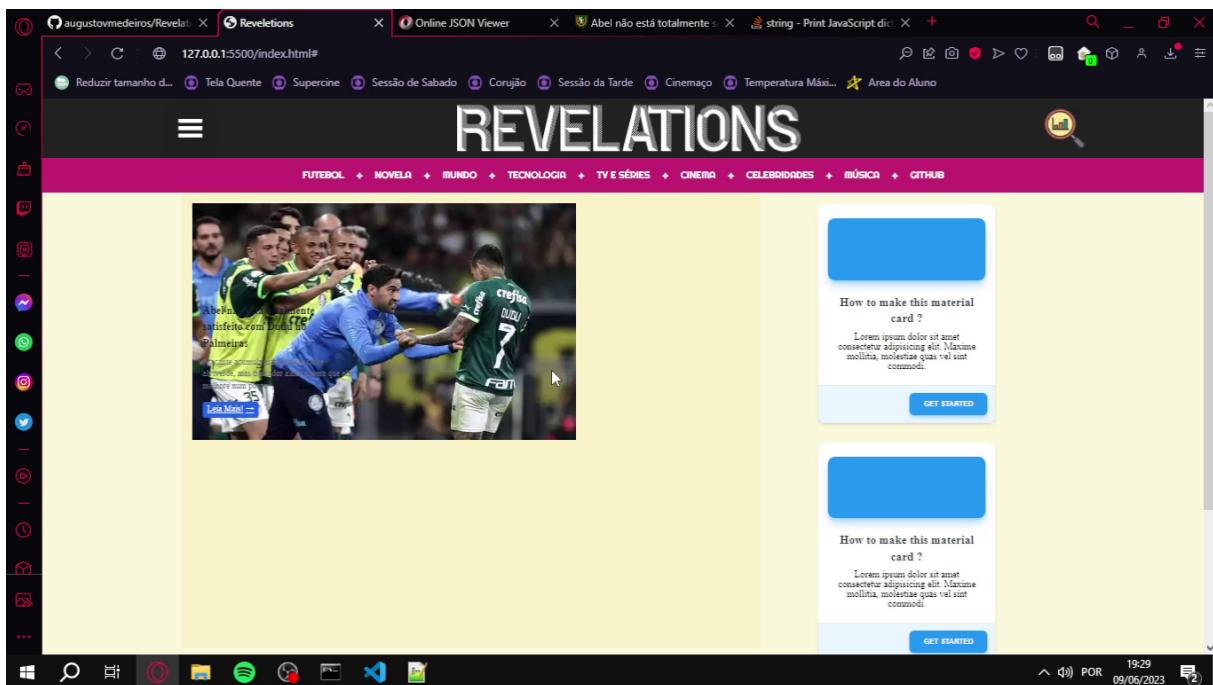
Começamos a testar alguns cards para as notícias mas eles não agradavam.



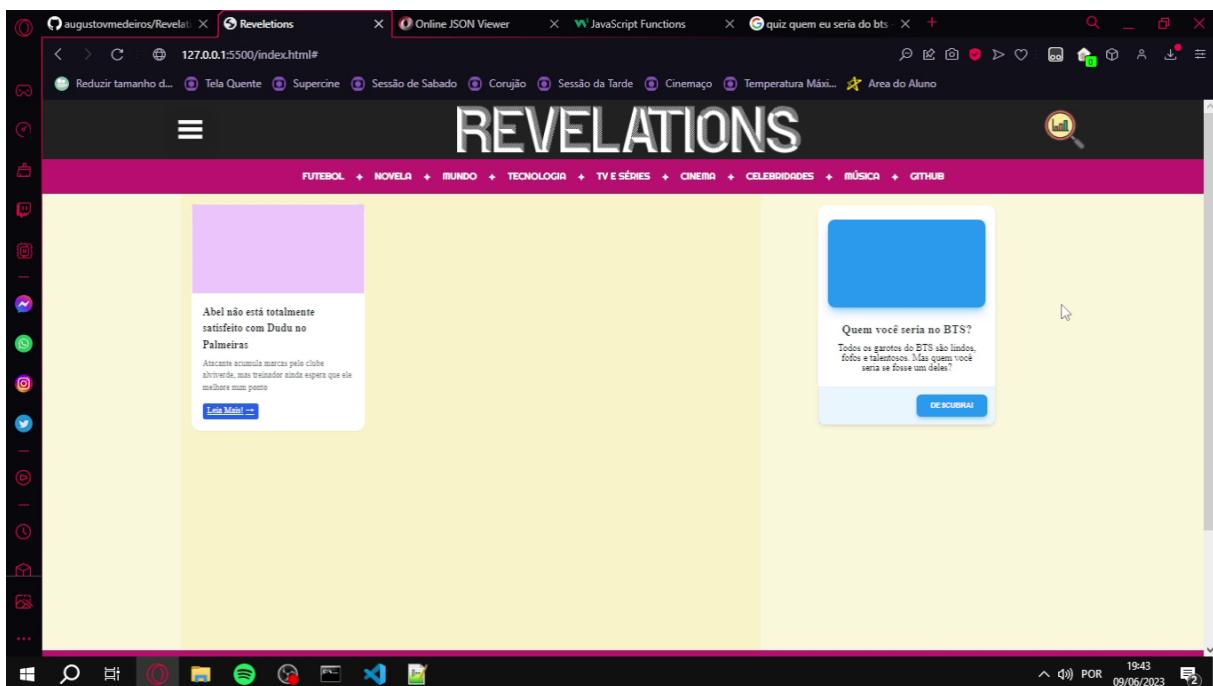
Encontramos o card ideal para os cards de notícia.



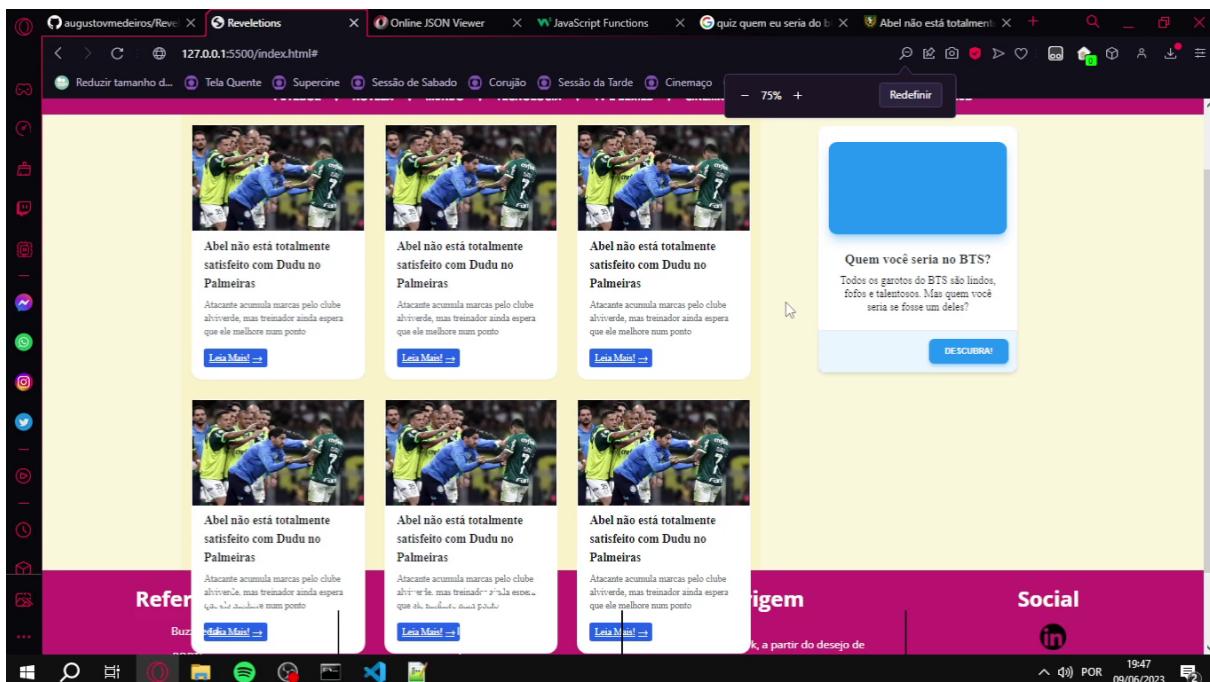
Escolhemos também um card para os quizzes.



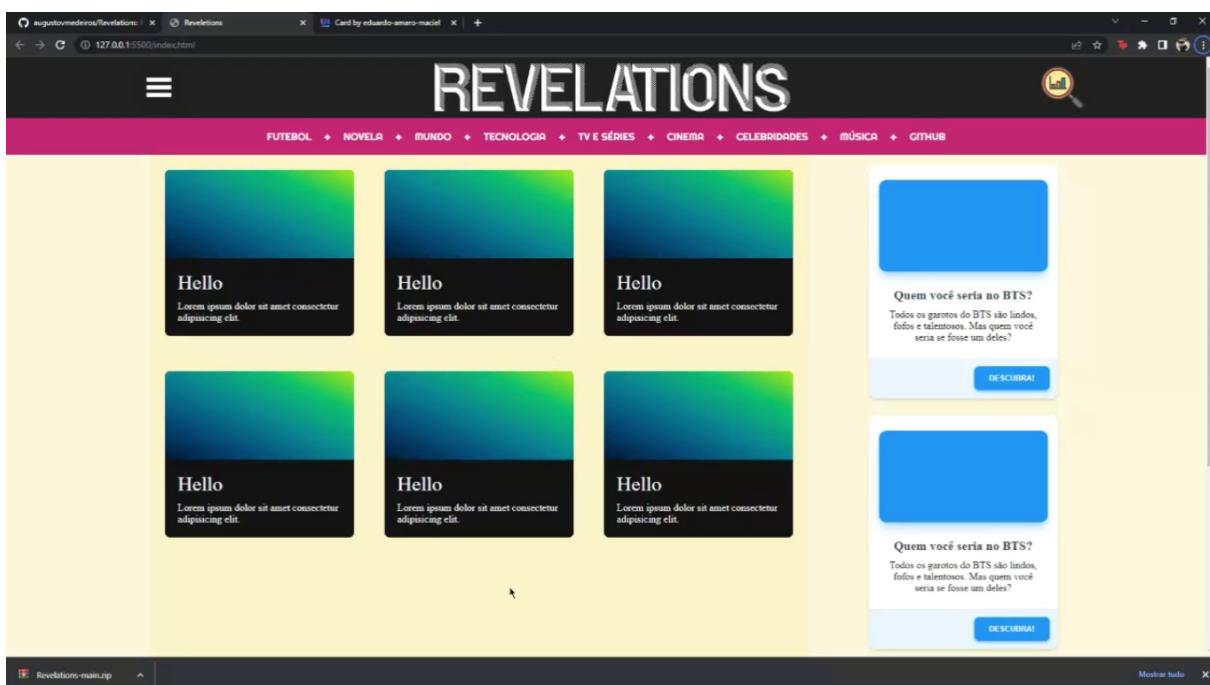
Testamos o JS com os cards e eles não funcionaram como esperado.



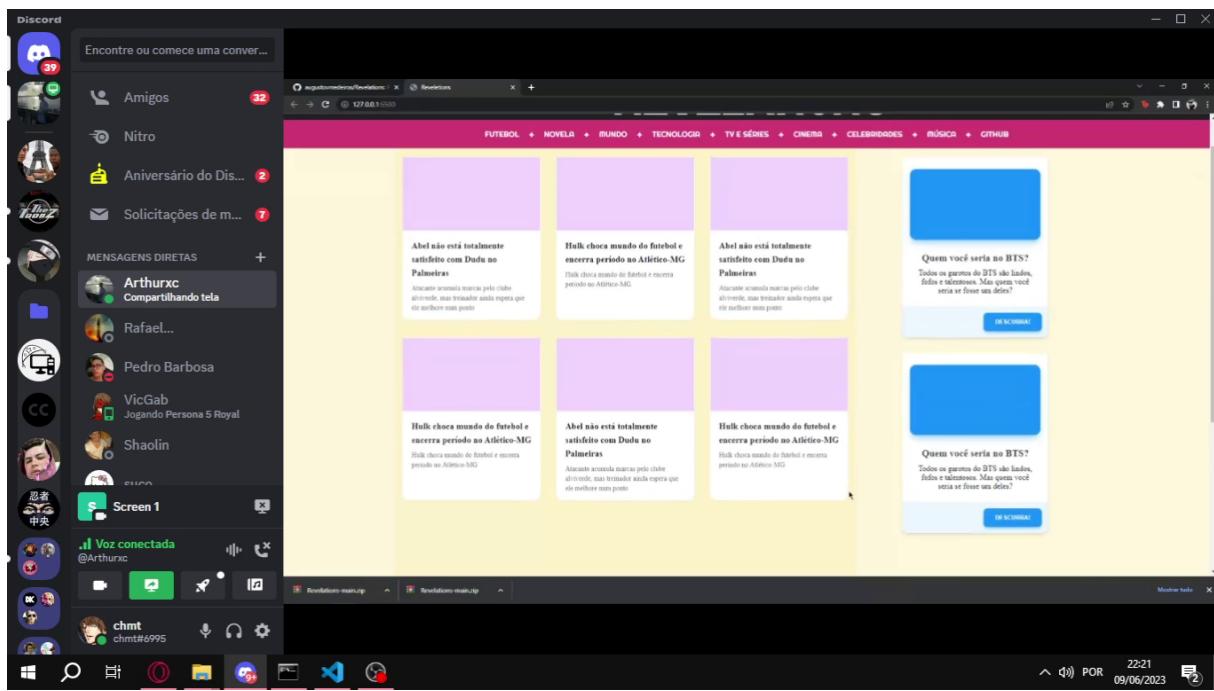
Configuramos a width da imagem no card.



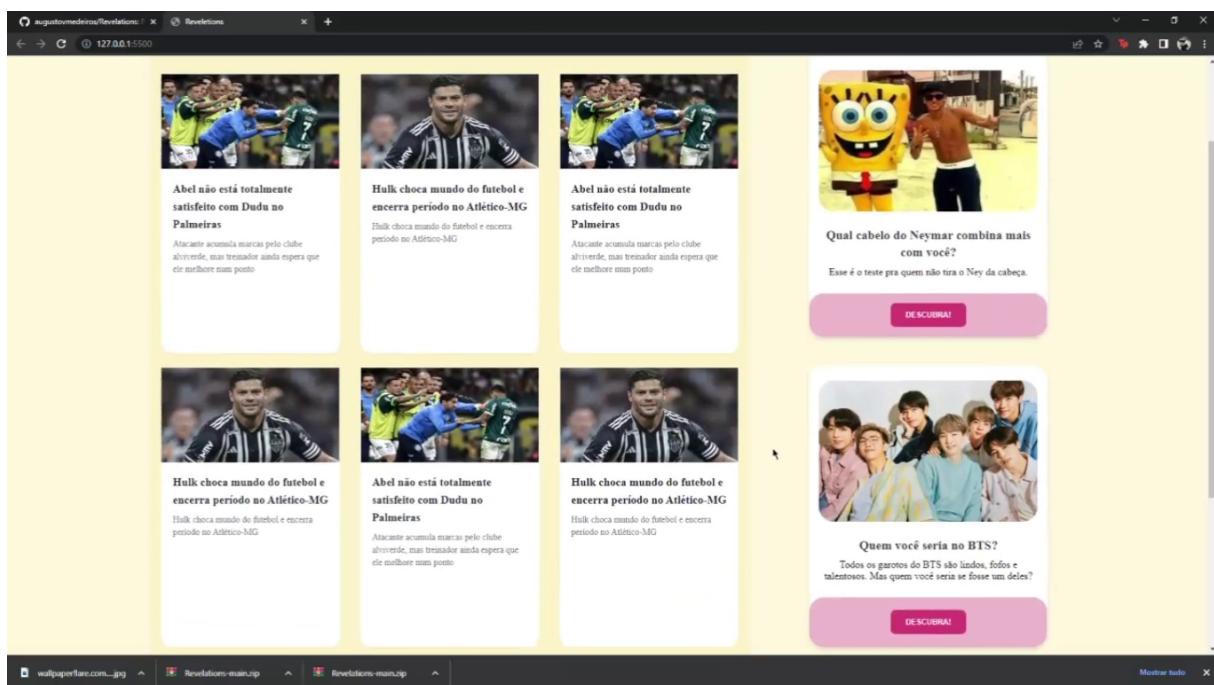
Com muitos cards eles começaram a invadir o footer.



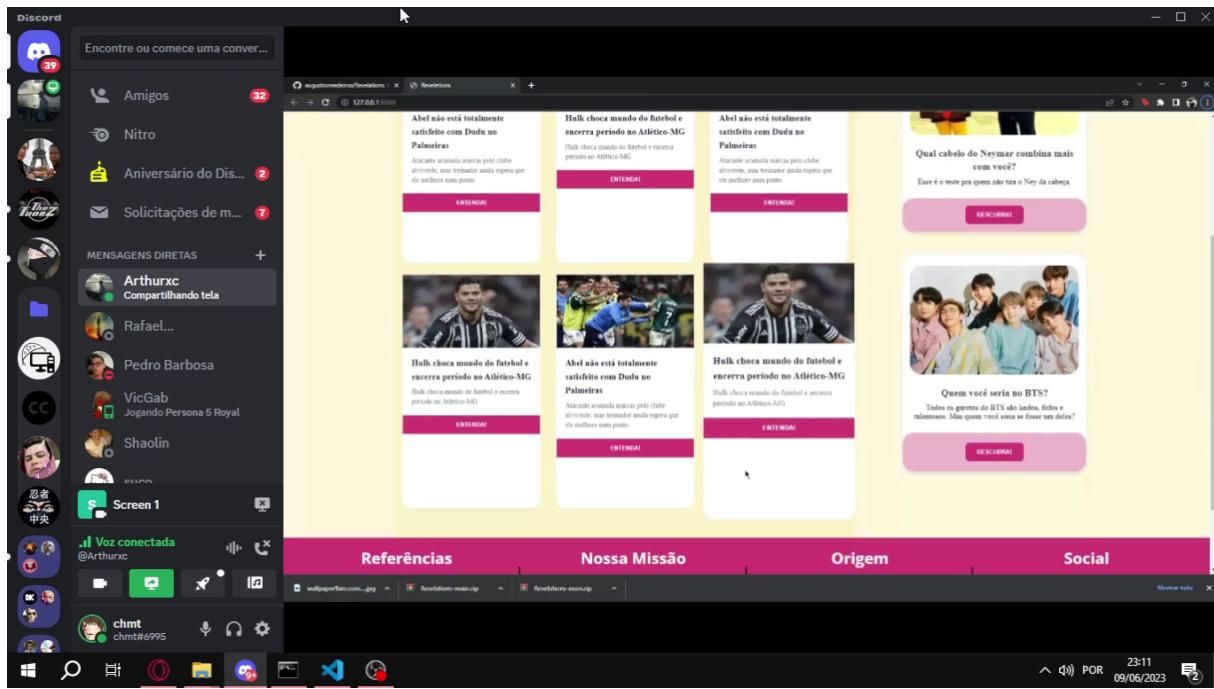
Testamos outro card pra ver como ficaria, mas preferimos corrigir o anterior pois combinava muito com a página.



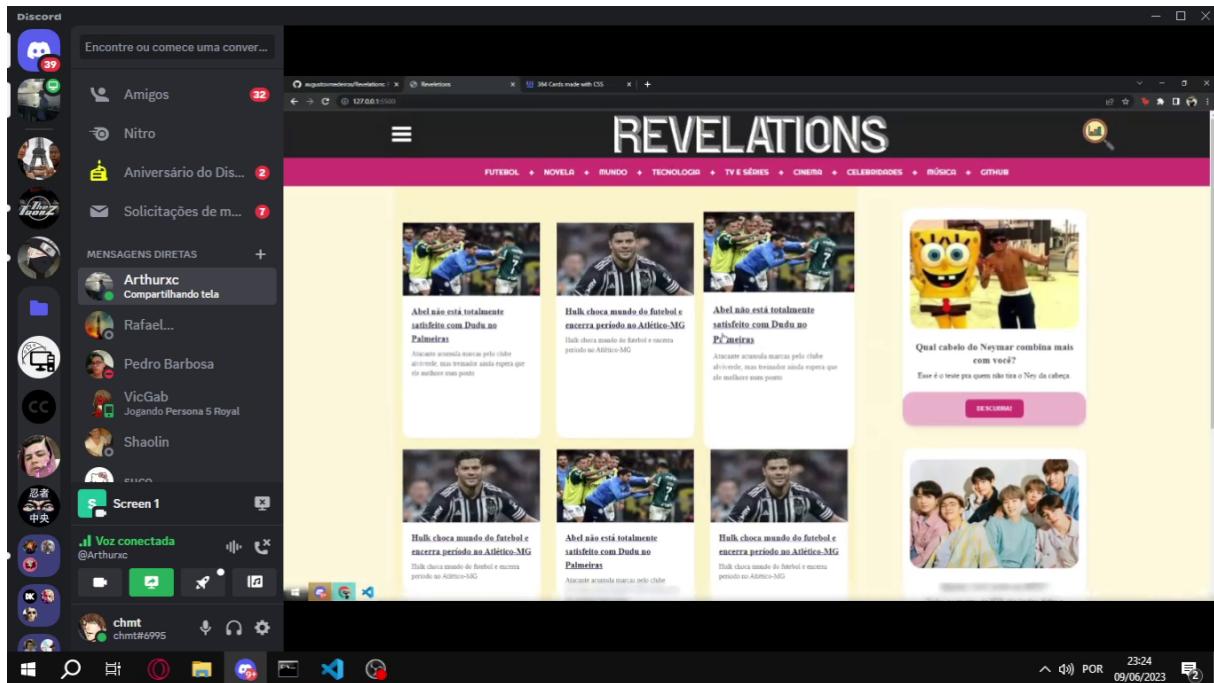
Card corrigido e agora mantém o mesmo tamanho para qualquer notícia e a div não invadia o footer.



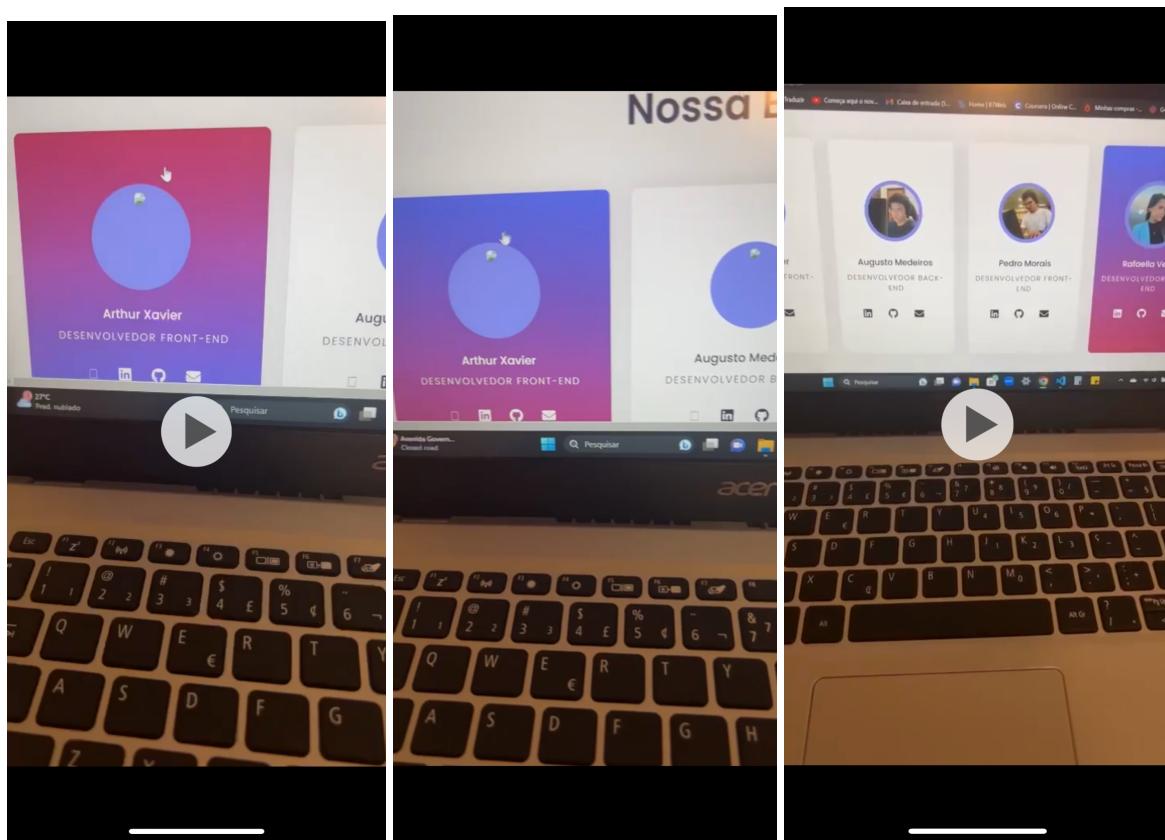
Testamos os cards com imagem.



Testamos colocar um botão pra ir pra notícia mas não conseguimos fixar ele no footer do card sem quebrar.

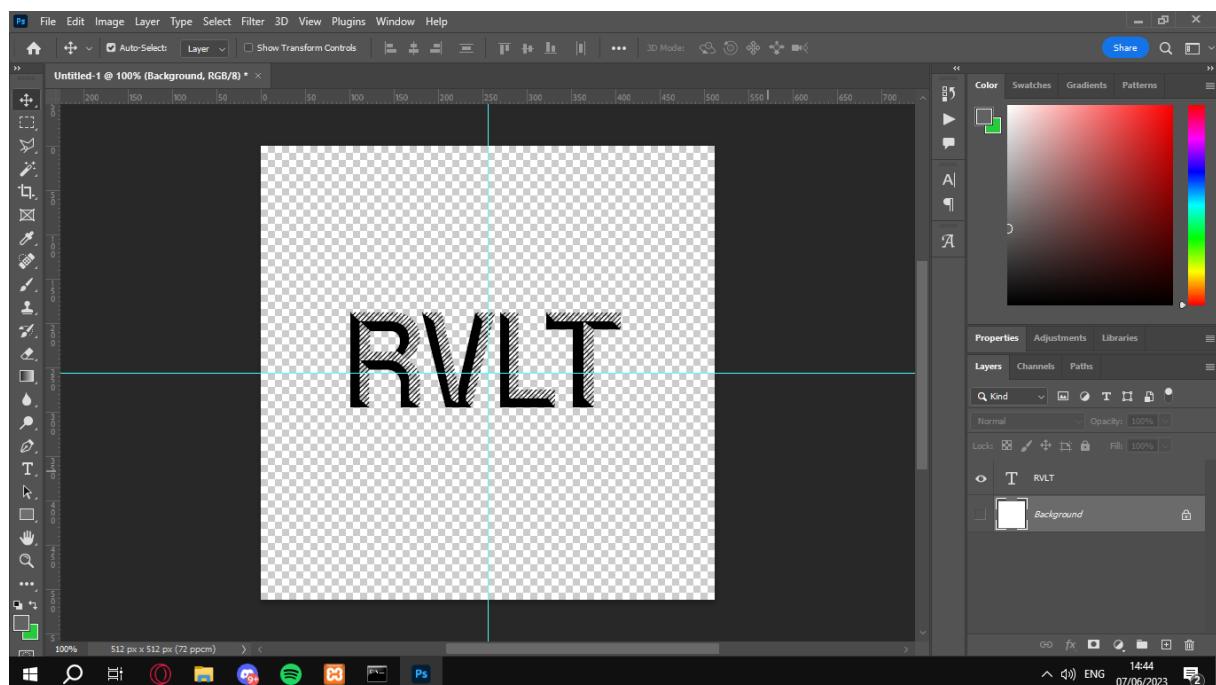


E por fim removemos o botão, resultando no layout final do nosso site.

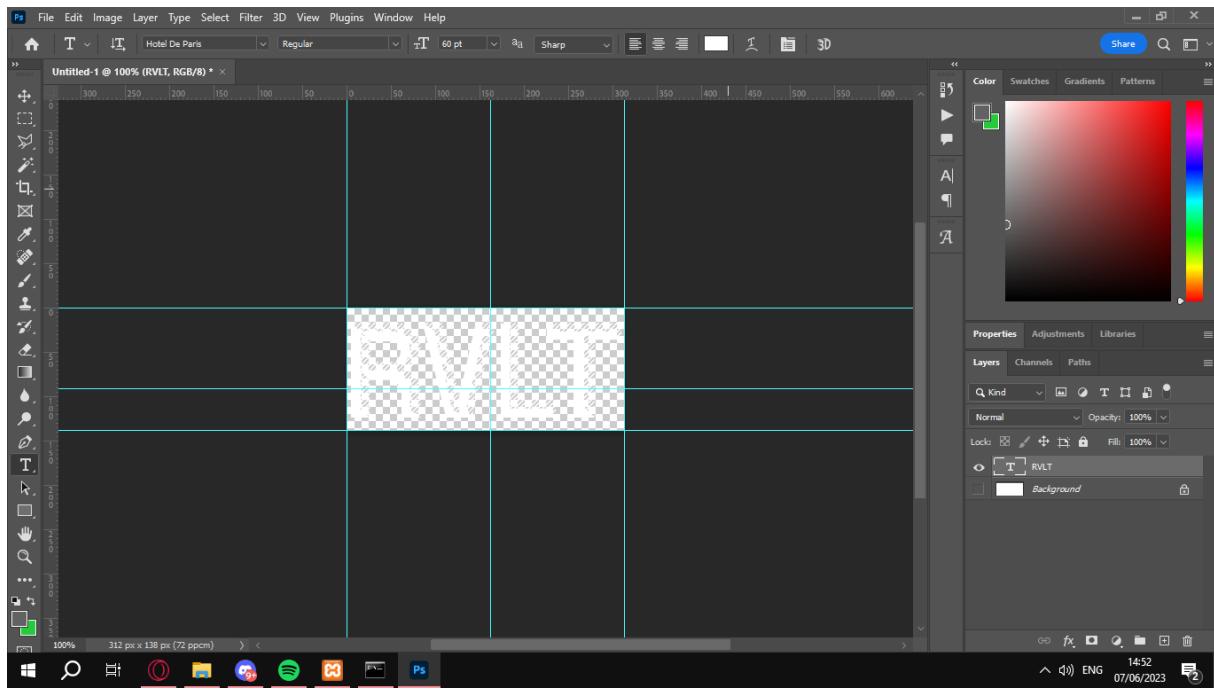


Testando algumas opções de cores para o card do sobre nós e mostrando a versão final.

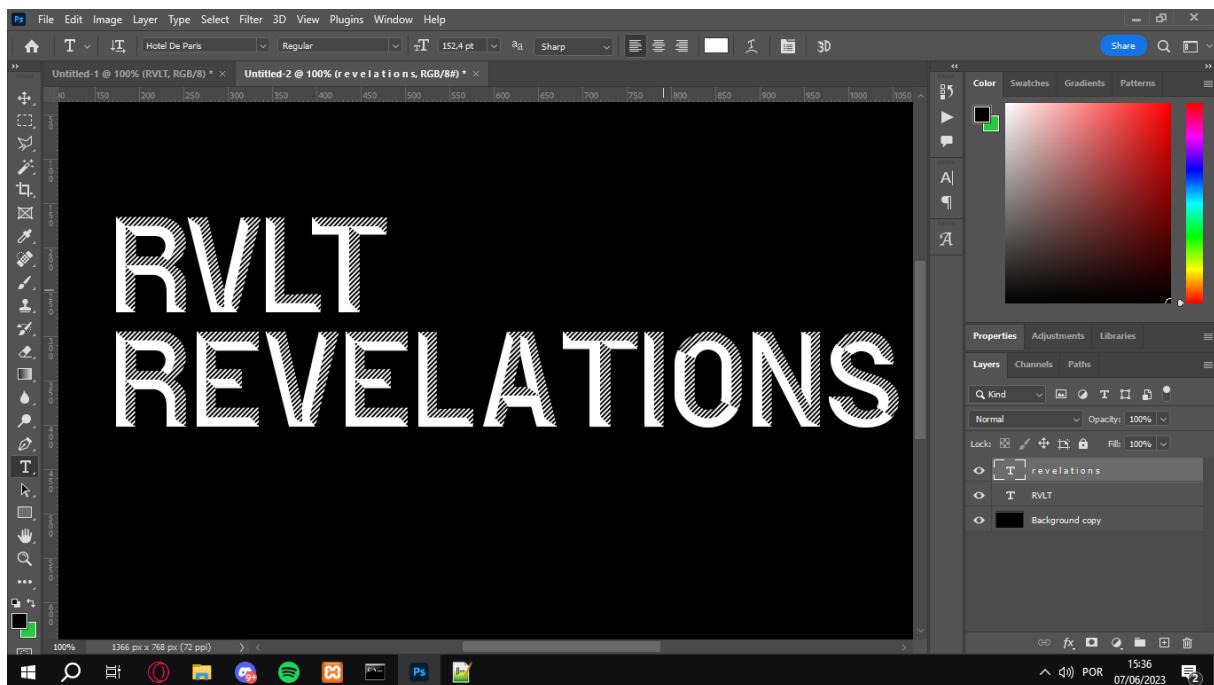
4. Requisitos Não Funcionais



Fizemos uma logo em preto no photoshop com os dizeres “RVLT” que funcionaria como uma sigla.



Alteramos a cor da logo.



Tomando base na logo anterior que não foi aprovada pela equipe refizemos com o nome inteiro do site.

A screenshot of the Visual Studio Code interface. The title bar reads "rvltjs - web - Visual Studio Code". The left sidebar shows a project structure under "OPEN EDITORS": "index.html" (orange icon), "JS rvltjs" (green icon, currently selected), and "src" (grey icon). Inside "src", there are "css" and "img" folders, and a "js" folder which contains "rvltjs.js" (green icon) and "index.html" (orange icon). The main editor area displays the following code:

```
src > js > JS rvltjs > noticias  
1 let noticias = [ ];
```

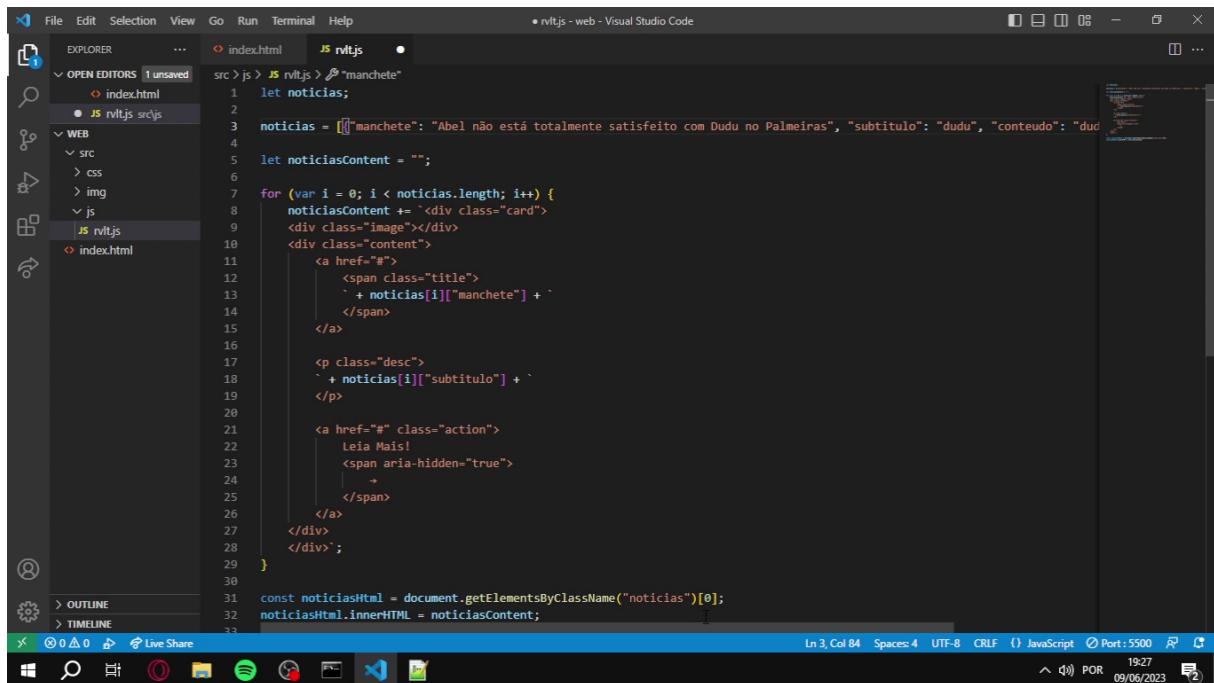
The status bar at the bottom shows "Ln 1, Col 17" and "JavaScript".

Iniciada a criação do javascript.

A screenshot of the Visual Studio Code interface, identical to the previous one except for the code in the editor:

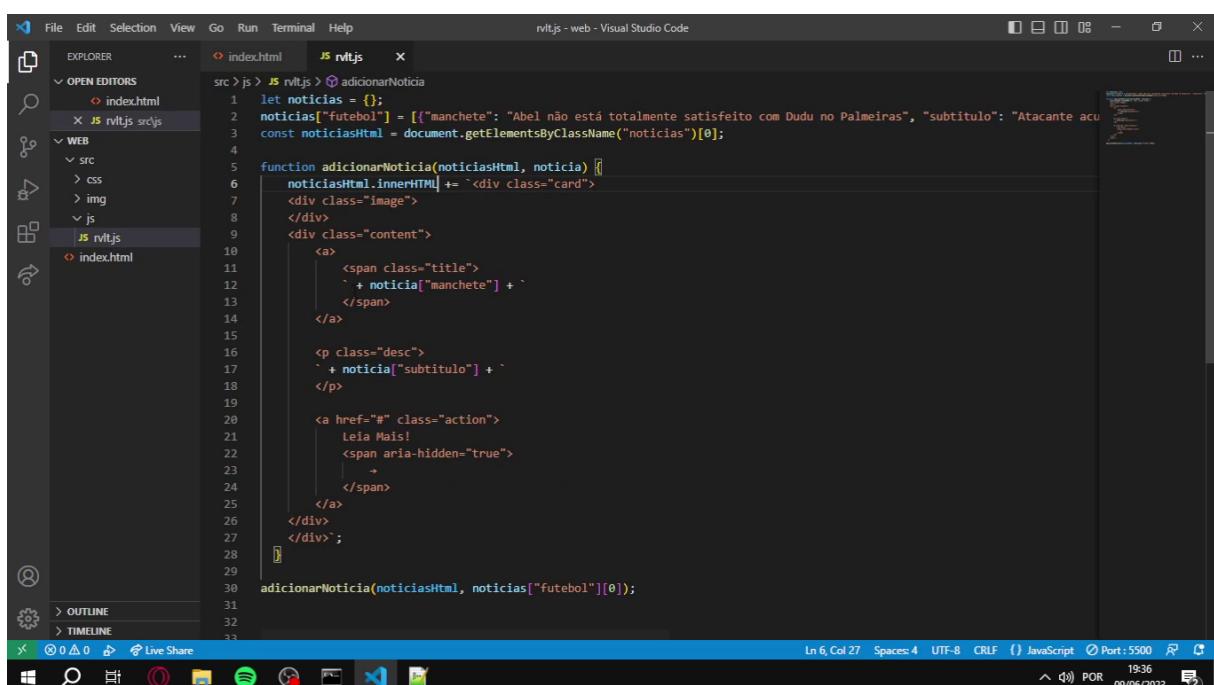
```
src > js > JS rvltjs > ...  
1 let noticias = [ {"manchete": "dudu", "subtitulo": "dudu", "conteudo": "dudu", "imagem": "jpg"}];
```

Começamos a definir a estrutura do nosso objeto.



```
src > js > JS rvltjs > "manchete"
1 let noticias;
2
3 noticias = [{"manchete": "Abel não está totalmente satisfeito com Dudu no Palmeiras", "subtitulo": "dudu", "conteudo": "dudu", "image": "https://www.rvltjs.com.br/assets/images/noticias/1.jpg"}, {"manchete": "Atacante acusa Dudu de querer sair do Palmeiras", "subtitulo": "atacante", "conteudo": "atacante", "image": "https://www.rvltjs.com.br/assets/images/noticias/2.jpg"}];
4
5 let noticiasContent = "";
6
7 for (var i = 0; i < noticias.length; i++) {
8   noticiasContent += `<div class="card">
9     <div class="image"></div>
10    <div class="content">
11      <a href="#">
12        <span class="title">
13          ${noticias[i].manchete}
14        </span>
15      </a>
16
17      <p class="desc">
18        ${noticias[i].subtitulo}
19      </p>
20
21      <a href="#" class="action">
22        Leia Mais!
23        <span aria-hidden="true">
24          +
25        </span>
26      </a>
27    </div>
28  </div>`;
29 }
30
31 const noticiasHTML = document.getElementsByClassName("noticias")[0];
32 noticiasHTML.innerHTML = noticiasContent;
```

Criamos um laço recepcional para colocar o conteúdo do nosso objeto na div e adicionar a mesma na seção.



```
src > js > JS rvltjs > adicionarNoticia
1 let noticias = {};
2 noticias["futebol"] = [{"manchete": "Abel não está totalmente satisfeito com Dudu no Palmeiras", "subtitulo": "dudu", "conteudo": "dudu", "image": "https://www.rvltjs.com.br/assets/images/noticias/1.jpg"}, {"manchete": "Atacante acusa Dudu de querer sair do Palmeiras", "subtitulo": "atacante", "conteudo": "atacante", "image": "https://www.rvltjs.com.br/assets/images/noticias/2.jpg"}];
3 const noticiasHTML = document.getElementsByClassName("noticias")[0];
4
5 function adicionarNoticia(noticiasHTML, noticia) {
6   noticiasHTML.innerHTML += `<div class="card">
7     <div class="image"></div>
8     <div class="content">
9       <a href="#">
10         <span class="title">
11           ${noticia.manchete}
12         </span>
13       </a>
14
15       <p class="desc">
16         ${noticia.subtitulo}
17       </p>
18
19       <a href="#" class="action">
20         Leia Mais!
21         <span aria-hidden="true">
22           +
23         </span>
24       </a>
25     </div>
26   </div>`;
27 }
28
29 adicionarNoticia(noticiasHTML, noticias["futebol"][0]);
```

Transformamos o laço repeticional em uma função.

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a tree view of files and folders. The 'data.json' file is selected. Other visible files include '# style.css', 'index.html', and 'rvlt.js'. A 'NOTIFICATIONS' icon is also present.
- Editor:** The main editor area displays the 'data.json' file content. The code defines a 'noticias' object with properties like 'futebol', 'manchete', 'subtitulo', 'conteudo', and 'imagem'.
- Status Bar:** Shows the current file is 'data.json', the line and column are 'Ln 8, Col 11', and the workspace is 'Visual Studio Live Share'.

```
1  {
2      "noticias": {
3          "futebol": [
4              {
5                  "manchete": "Abel no est\u00e1 totalmente satisfeito com Dudu no Palmeiras",
6                  "subtitulo": "Atacante acumula marcas pelo clube alviverde, mas treinador ainda espera que ele melhore num po",
7                  "conteudo": "dudu",
8                  "imagem": "https://images-ext-2.discordapp.net/external/XU1_I0bRahrEx8ZQ38ApHFHJkD9hdpyQMWr6s0iFAuQ/https://i.s"
9              }
10         ],
11         "manchete": "Hulk choca mundo do futebol e encerra per\u00f3dio no Atl\u00e9tico-MG",
12         "subtitulo": "Hulk demonstra poder de decis\u00e3o no Galo.",
13         "conteudo": "\u25aa",
14         "imagem": "https://i.superesportes.com.br/bD80Hdcp0uB5Q7uUE_4fNvGfwg-/650x0/smart/imgsapp.mg.superesportes.co"
15     },
16     "games": {},
17     "mundo": {},
18     "tech": {},
19     "tv": {},
20     "cinema": {}
21 }
```

Removemos o objeto do JS e o colocamos num arquivo JSON (Arquivo de notação de objetos do javascript).

File Edit Selection View Go Run Terminal Help rvltjs - Visual Studio Live Share (Workspace) - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this workspace to enable all features. Manage Learn More

OPEN EDITORS

- data.json # style.css index.html JS rvltjs
- datajson src/js # style.css src/css index.html
- JS rvltjs src/js

VISUAL STUDIO LIVE SHARE

- web
- src
 - css
 - # style.css
 - js
 - data.json
 - JS rvltjs
 - index.html

data.json

```
1 import data from './data.json' assert { type: 'json' };
2 let quizzes = {};
3
4 quizzes['celebridades'] = [{"titulo": "Qual cabelo do Neymar combina mais com você?", "subtitulo": "Esse é o teste pra quem", "musica": ["Quem votaria no BTS?", "Todos os garotos do BTS são lindos, fofos e talentosos"]};
5 const noticiasHTML = document.getElementsByClassName("noticias")[0];
6 const quizzesHTML = document.getElementsByClassName("quizzes")[0];
7
8 function adicionarNoticia(noticiasHTML, noticia) {
9     noticiasHTML.innerHTML += `<div class="card">
10 <div class="image">
11     <img class="fotonocticia" src="" + noticia["imagem"] + ">
12 </div>
13 <div class="content">
14     <a href="#">
15         <span class="title">
16             ` + noticia["manchete"] + `
17         </span>
18     </a>
19
20     <p class="desc">
21         ` + noticia["subtitulo"] + `
22     </p>
23 </div>
24 </div>;
25
26 }
27
28 function adicionarQuiz(quizzesHTML, quiz) {
29     quizzesHTML.innerHTML += `
30 <div class="cardq">
31 <div class="headerq">
```

OUTLINE

TIMELINE

LIVE SHARE

Live Share Restricted Mode 0 ▲ 0 Joined 1 Ln 4, Col 1 Spaces: 4 UTF-8 LF JavaScript 15:15 12/06/2023

Importamos o JSON no arquivo JS.

```

{
  "noticias": [
    {
      "manchete": "Hulk choca mundo do futebol e encerra período no Atlético-MG",
      "subtitulo": "Hulk demonstra poder de decisão no galão.",
      "conteudo": [
        {
          "gênero": "mundo"
        },
        {
          "gênero": "tech"
        },
        {
          "gênero": "tv"
        },
        {
          "gênero": "cinema"
        }
      ],
      "imagem": "https://i.superesportes.com.br/b0804dc085070e_6fNvG1fw-/650x0/smart/imgsapp.mg.superesportes.com.br/ap"
    },
    {
      "celebridades": [
        {
          "titulo": "Qual cabelo do Neymar combina mais com você?",
          "subtitulo": "Esse é o teste pra quem não tira o Ney da cabeça.",
          "imagem": "https://res.cloudinary.com/buzzfeed-brasil/image/upload/q_auto:best,f_auto,dpr_2.0,c_fill,w_320,h_230/imagens"
        },
        {
          "titulo": "Quem você seria no BTS?",
          "subtitulo": "Todos os garotos do BTS são lindos, fofos e talentosos. Mas quem você seria se fosse um deles?",
          "imagem": "https://popnow.com.br/wp-content/uploads/2020/09/BTS.jpeg"
        }
      ]
    }
  ],
  "quizzes": [
    {
      "quiz": {
        "title": "Qual cabelo do Neymar combina mais com você?",
        "subtitle": "Esse é o teste pra quem não tira o Ney da cabeça.",
        "image": "https://res.cloudinary.com/buzzfeed-brasil/image/upload/q_auto:best,f_auto,dpr_2.0,c_fill,w_320,h_230/imagens"
      }
    }
  ]
}

```

E por fim preenchemos todos os valores das notícias, quizzes e os tipos de cada um dos dois.

```

export function adicionarNoticia(noticiasHtml, noticia) {
  noticiasHtml.innerHTML += `<div class="card">
<div class="image">
  
</div>
<div class="content">
  <a href="#" + noticia["conteudo"] + ">
    <span class="title">
      ${noticia["manchete"]}
    </span>
  </a>

  <p class="desc">
    ${noticia["subtitulo"]}
  </p>
</div>
</div>`;
}

export function adicionarQuiz(quizesHtml, quiz) {
  quizesHtml.innerHTML += `<div class="cardq">
<div class="headerq">
  
</div>
<div class="infoq">
  <p class="titleq">${quiz["titulo"]} </p>
  <p>${quiz["subtitulo"]} </p>
</div>
<div class="footerq">
  <p class="tagq"></p>
  <a href="#" + quiz["conteudo"] + "><button type="button" class="actionq">Descubra!</button></a>
</div>
</div>`;
}

```

Transformamos as funções em funções de exportação.

```
import data from './data.json' assert { type: 'json' };
import { adicionarNoticia, adicionarQuiz } from './rvlt.js';

const noticiasHtml = document.getElementsByClassName("noticias")[0];
const quizzesHtml = document.getElementsByClassName("quizes")[0];

let i = 0;
let tipos = Object.keys(data["noticias"]);
while (i < tipos.length) {
    let randomEscolha = Math.floor(Math.random() * data["noticias"][tipos[i]].length);
    adicionarNoticia(noticiasHtml, data["noticias"][tipos[i]][randomEscolha]);
    i++;
}

adicionarQuiz(quizzesHtml, data["quizes"]["celebridades"][0]);
adicionarQuiz(quizzesHtml, data["quizes"]["musica"][0]);
```

Criado JS para o index com laço repeticional para cada tipo de notícia definida no nosso JSON.

```
import data from './data.json' assert { type: 'json' };
import { adicionarNoticia, adicionarQuiz } from './rvlt.js';

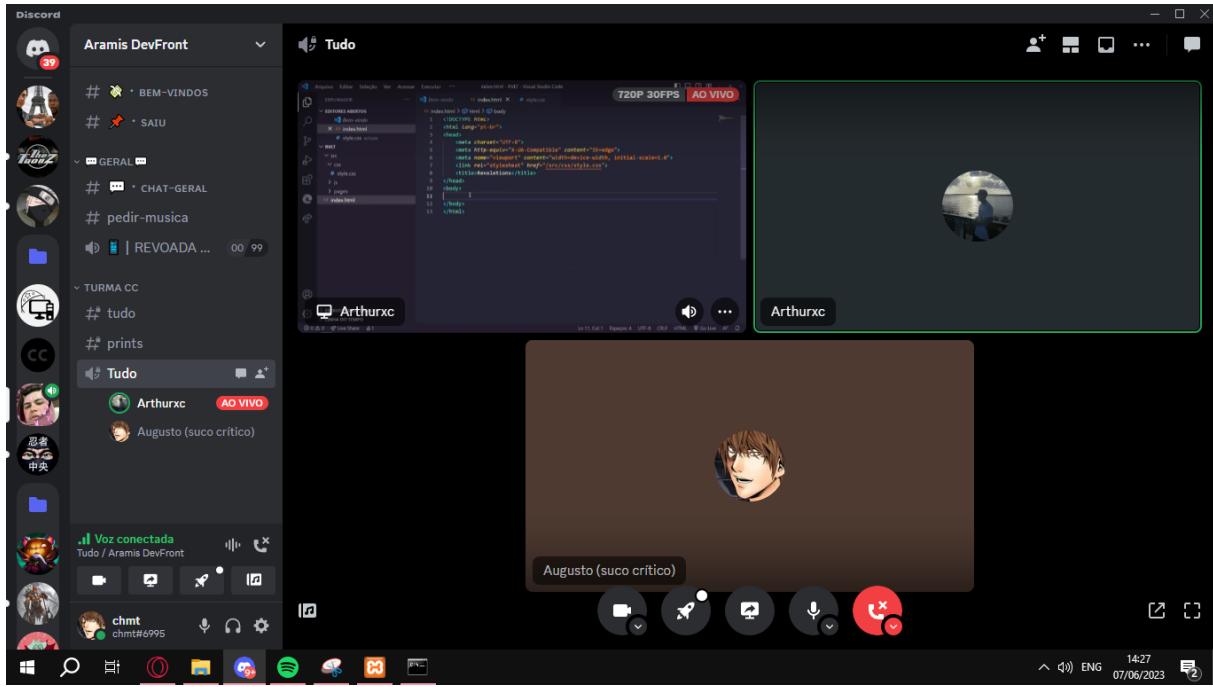
const noticiasHtml = document.getElementsByClassName("noticias")[0];
const quizzesHtml = document.getElementsByClassName("quizes")[0];
const params = new URLSearchParams(location.search);
let tipo = params.get("tipo");

let i = 0;
while (i < data["noticias"][tipo].length) {
    adicionarNoticia(noticiasHtml, data["noticias"][tipo][i]);
    i++;
}

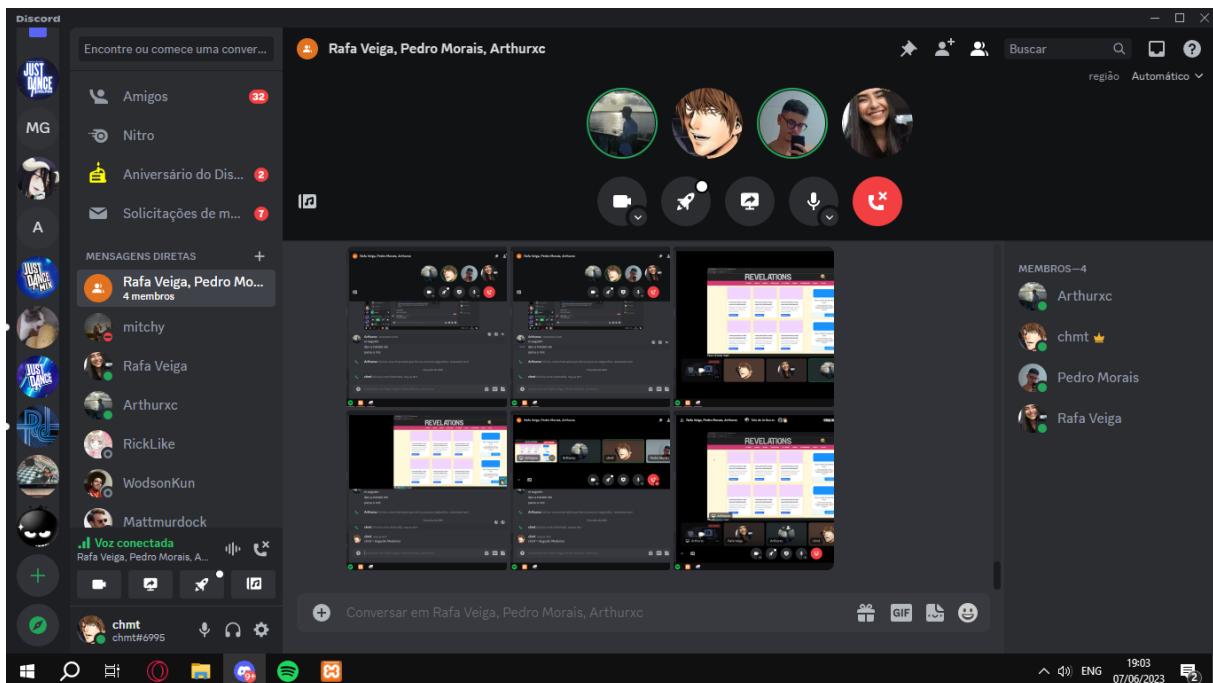
adicionarQuiz(quizzesHtml, data["quizes"]["celebridades"][0]);
adicionarQuiz(quizzesHtml, data["quizes"]["musica"][0]);
```

Criado JS para a página de listagem de notícias por tipo.

5. Reuniões



Reunião entre Arthur Xavier e Augusto Medeiros no dia 07/06/2023 para iniciarmos a programação do site.



Reunião no Discord no dia 07/06/2023 para estabelecer prazos e metas e mostrar os avanços. (Todos estavam presentes: Arthur Xavier, Augusto Medeiros, Pedro Morais e Rafaella Veiga)

The screenshot shows a Discord group chat interface. At the top, there's a header with user icons and a search bar. Below the header, a code editor window displays a snippet of JavaScript. To the right of the code editor, a sidebar titled "MEMBROS-3" lists three members: Arthurxc, chmt (with a Spotify icon), and Rafa Veiga. The main chat area contains messages from users chmt, Arthurxc, and Rafa Veiga. A message from chmt includes a link to a Google Fonts CSS file. Another message from chmt links to a Visual Studio Code for the Web page. The bottom of the screen shows a message from Rafa Veiga and a message input field.

```
import data from './data.json' assert { type: 'json' };
const noticiasHtml = document.getElementsByClassName("noticias")[0];
const quizzesHtml = document.getElementsByClassName("quizzes")[0];

function adicionarNoticia(noticiasHtml, noticia) {
```

chmt 12/06/2023 16:42
./src/paginas/noticia.html?tipo=futebol&id=1

→ chmt adicionou Rafa Veiga ao grupo. 12/06/2023 18:22

📞 chmt iniciou uma chamada que durou 2 horas. 12/06/2023 18:22

chmt 12/06/2023 18:27
@import 'https://fonts.googleapis.com/css?family=Righteous';
font-family: Righteous;
<https://prod.liveshare.vsengsaas.visualstudio.com/join?B4EAD6D7CF10CC510886CD9B488C5DC71762>

Visual Studio Code for the Web
Build with Visual Studio Code, anywhere, anytime, entirely in your browser.

Rafa Veiga 12/06/2023 18:42

Conversar em Arthurxc, Rafa Veiga

Tivemos uma reunião no dia 12/06/2023 no Discord com Arthur Xavier, Augusto Medeiros e Rafaella Veiga para integrarmos a página sobre nós feita por Rafaella.

The screenshot shows a direct message between two users on Discord. The header indicates the recipient is Arthurxc. The message list shows a message from chmt about a 4-hour call and a message from Arthurxc with a link to a Visual Studio Code for the Web page. The bottom of the screen shows a large image of a man with a beard wearing a white esports jersey with "FALLEN" and "twitch" logos. The image is set against a dark background with the word "FALLEN" in large, metallic letters. The message input field at the bottom shows a message from chmt and the timestamp 14/06/2023 17:44.

Arthurxc | VULGO Arthur Front-End Jr.

chmt iniciou uma chamada que durou 4 horas. Ontem às 18:34

chmt Ontem às 18:46
alou

Arthurxc Ontem às 19:07
<https://prod.liveshare.vsengsaas.visualstudio.com/join?D961EF827EE9D20D2CD633261E5458DA2A65>

Visual Studio Code for the Web
Build with Visual Studio Code, anywhere, anytime, entirely in your browser.

chmt Ontem às 19:27

Conversar em @Arthurxc

17:44 14/06/2023

E por fim uma reunião no dia 14/06/2023 entre Arthur Xavier e Augusto Medeiros para finalizar todos os detalhes do site e gravar o vídeo.