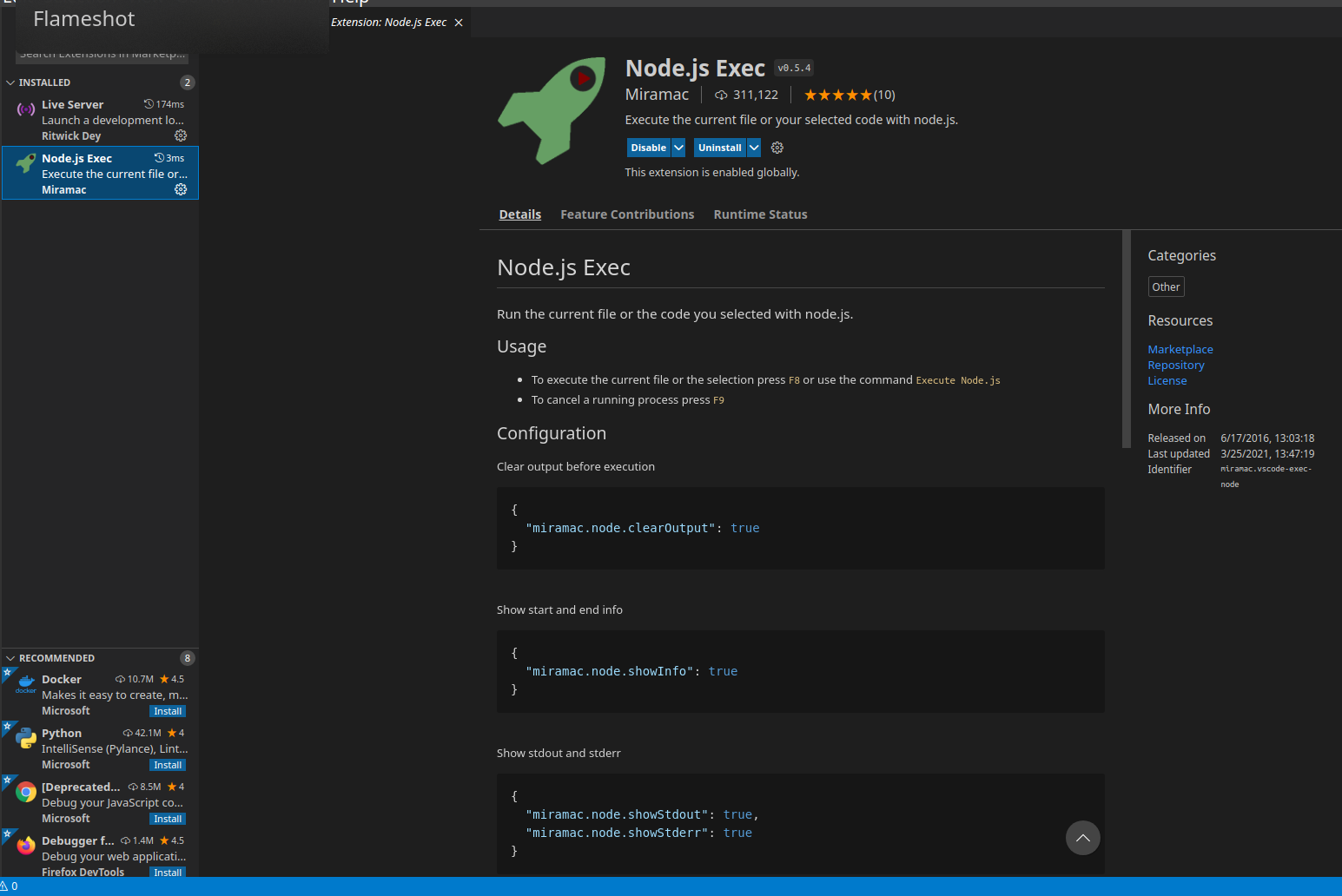
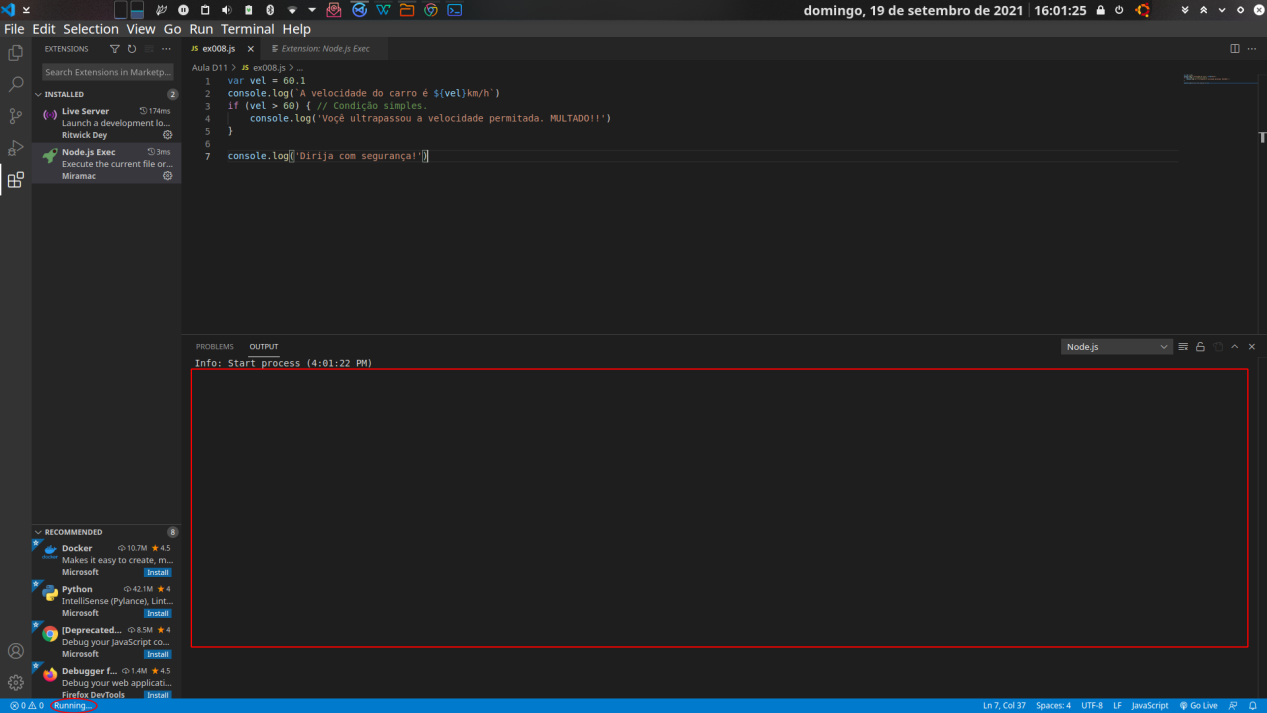
**Problemas que surgiram**

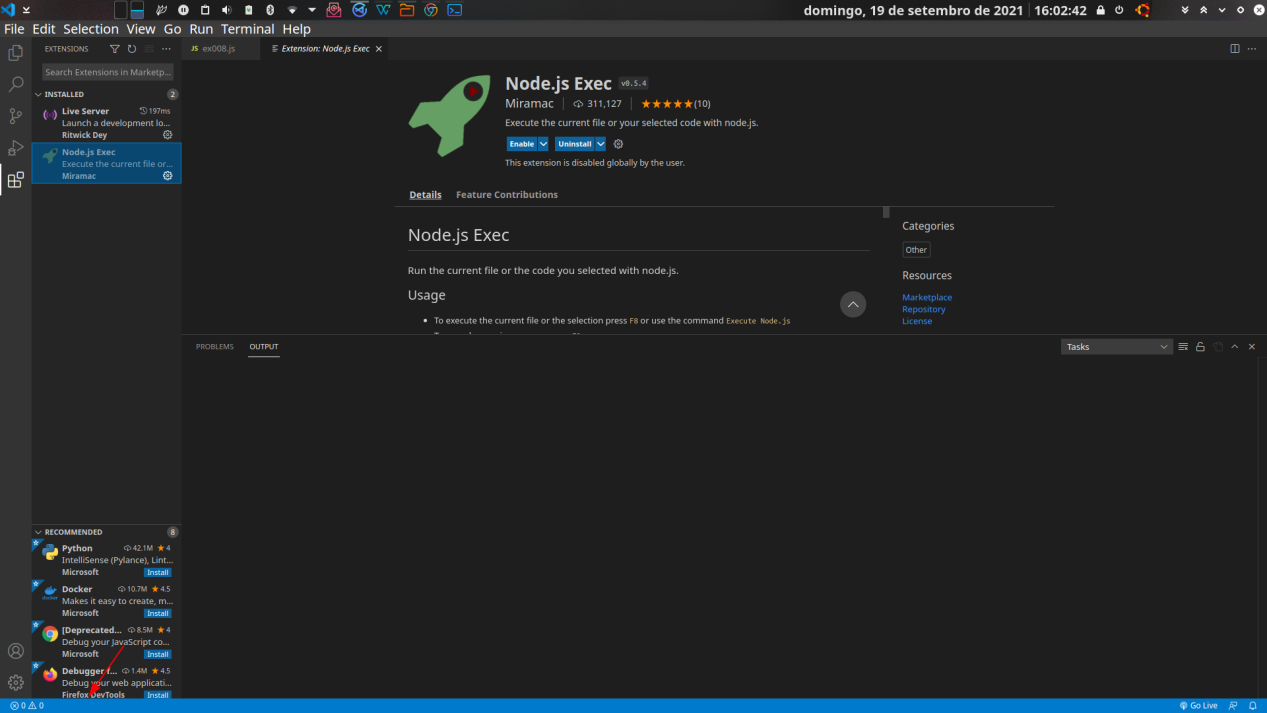
**Não estava a conseguir usar o Nodejs.**

Nodejs permitir rodar programas js fora de um browser, como estou a usar o VScode para o projeto então estava a usar a extensão do node js disponível.



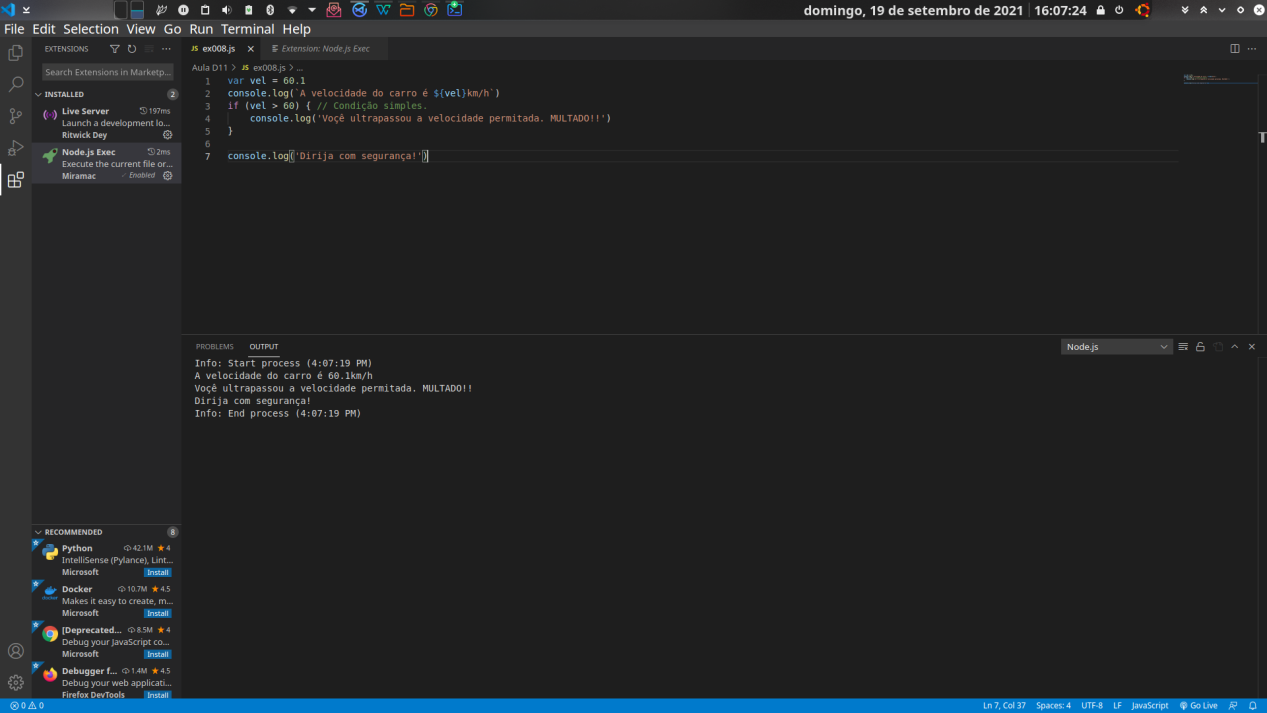
porem quando dava Run nu programa, o programa não aparecia na consola e o node continuava em running permanentemente até se desabilitado





Depois de algumas pesquisas em comunidades da VScode e alguns teste descobrir é necessario ter instalado o próprio Nodejs no computador para conseguir usa-lo no VScode.



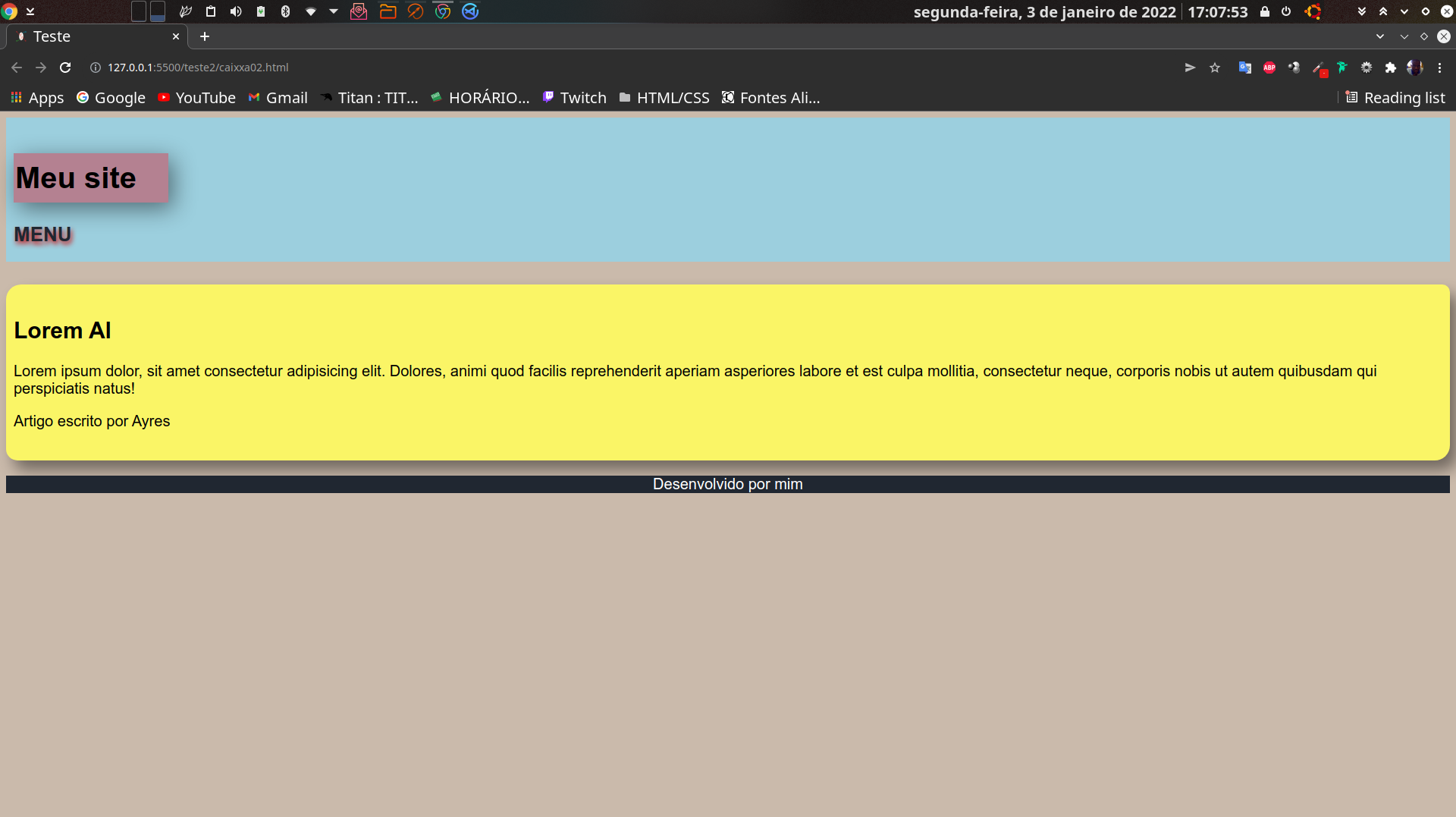
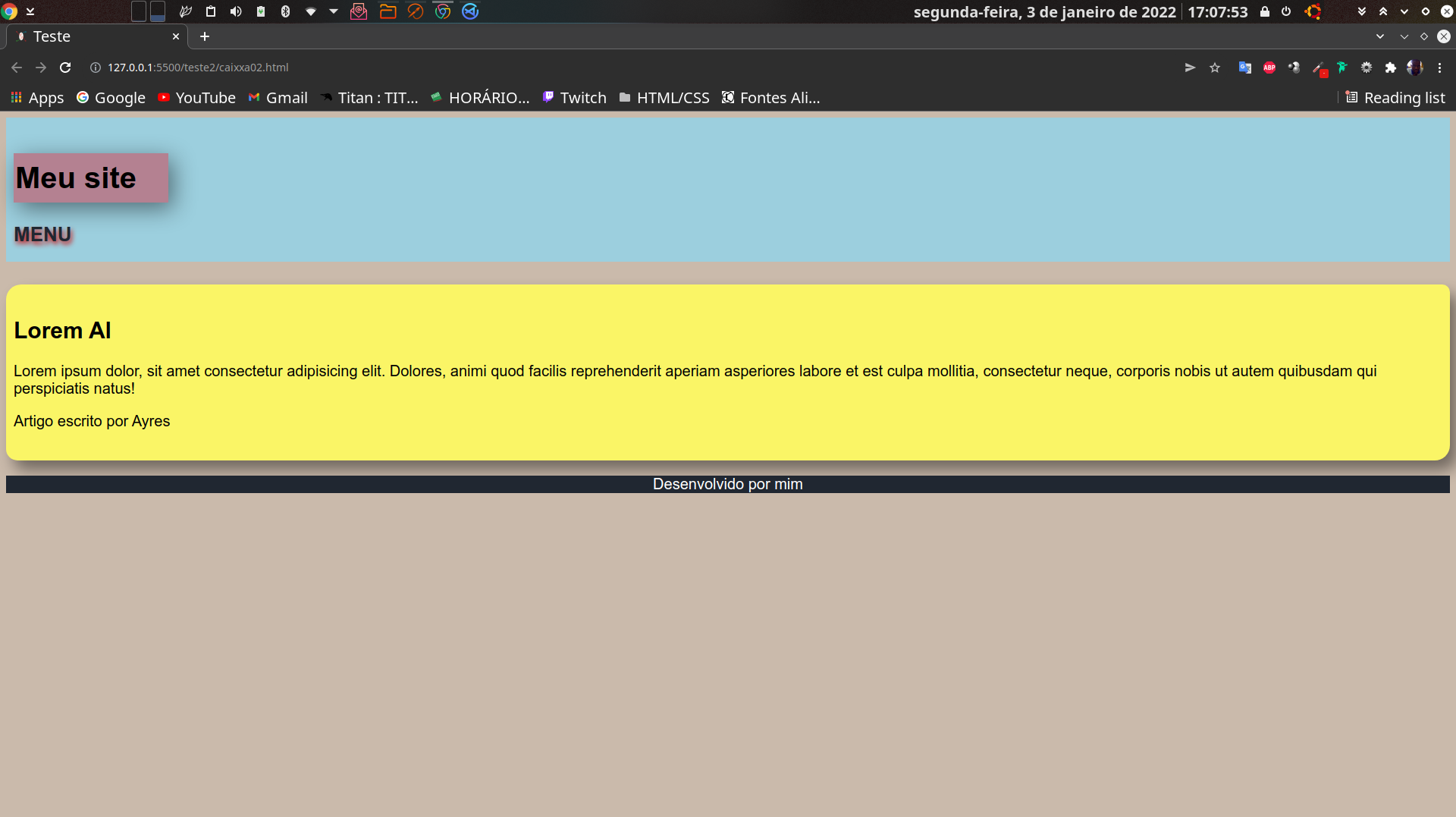


**Logo do site.**

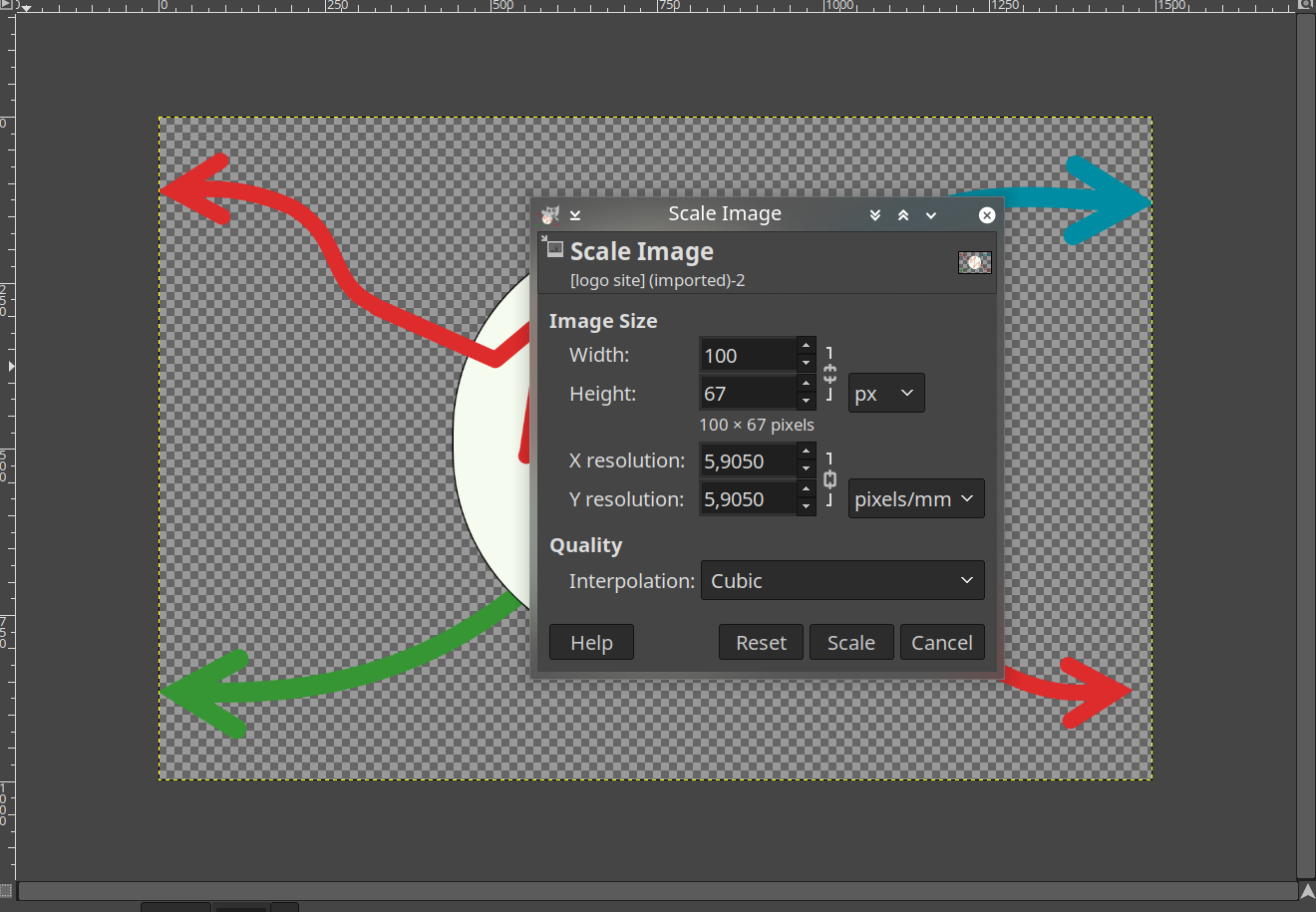
Após eu ter feito o logotípo do site baseado com as cores da paleta que escolhi ficou assim

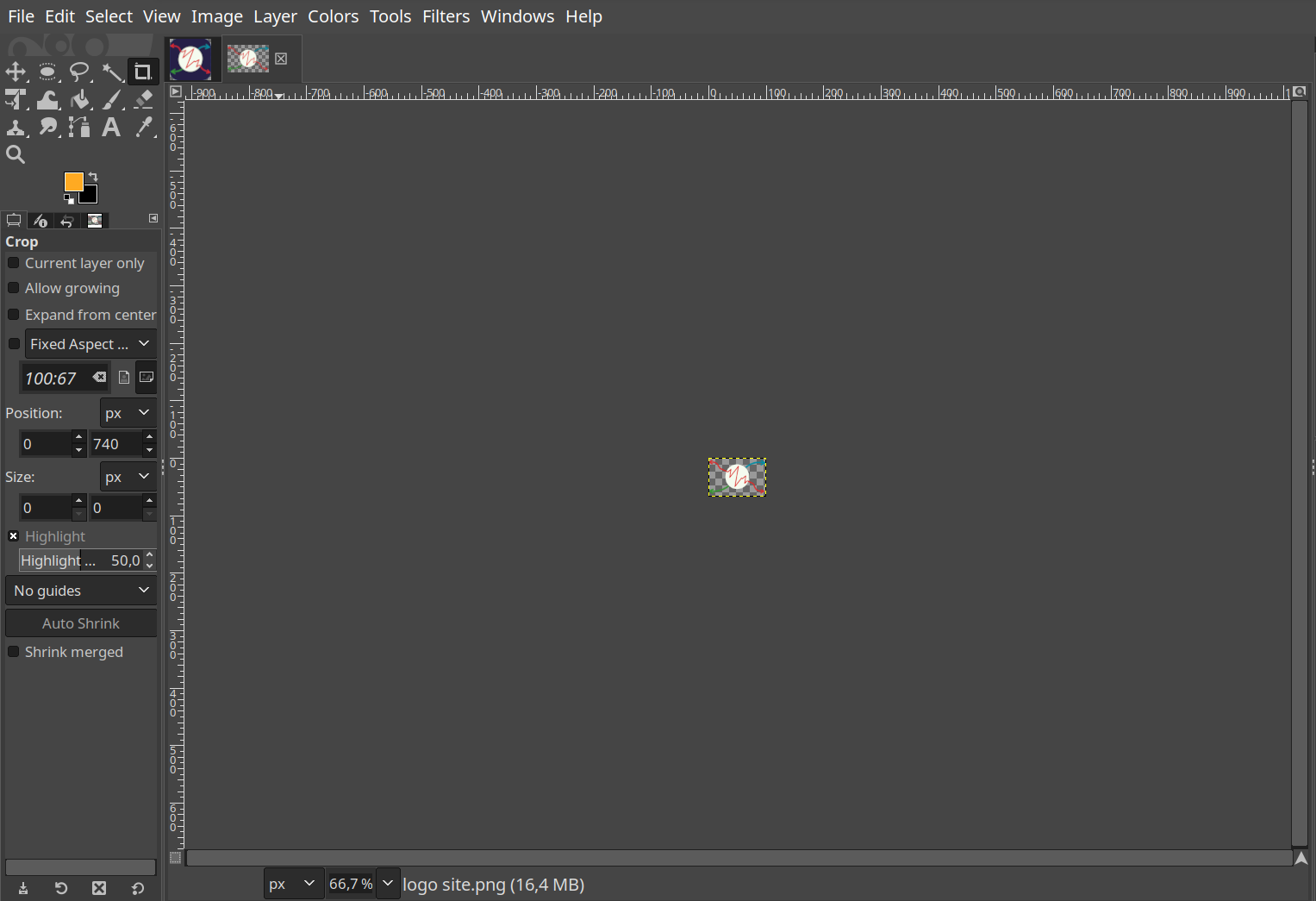


De seguida fui testar para ver se era possível usa-lo com favicon e este foi o resultado:



Foi então que percebi o problema com a resolução do meu logotipo. De forma a diminuir a resolução da imagem e consequentemente o tamanho da imagem eu baixei a resolução para 100px de largura e altura foi proporcional (67px) de forma a não destorcer a imagem:



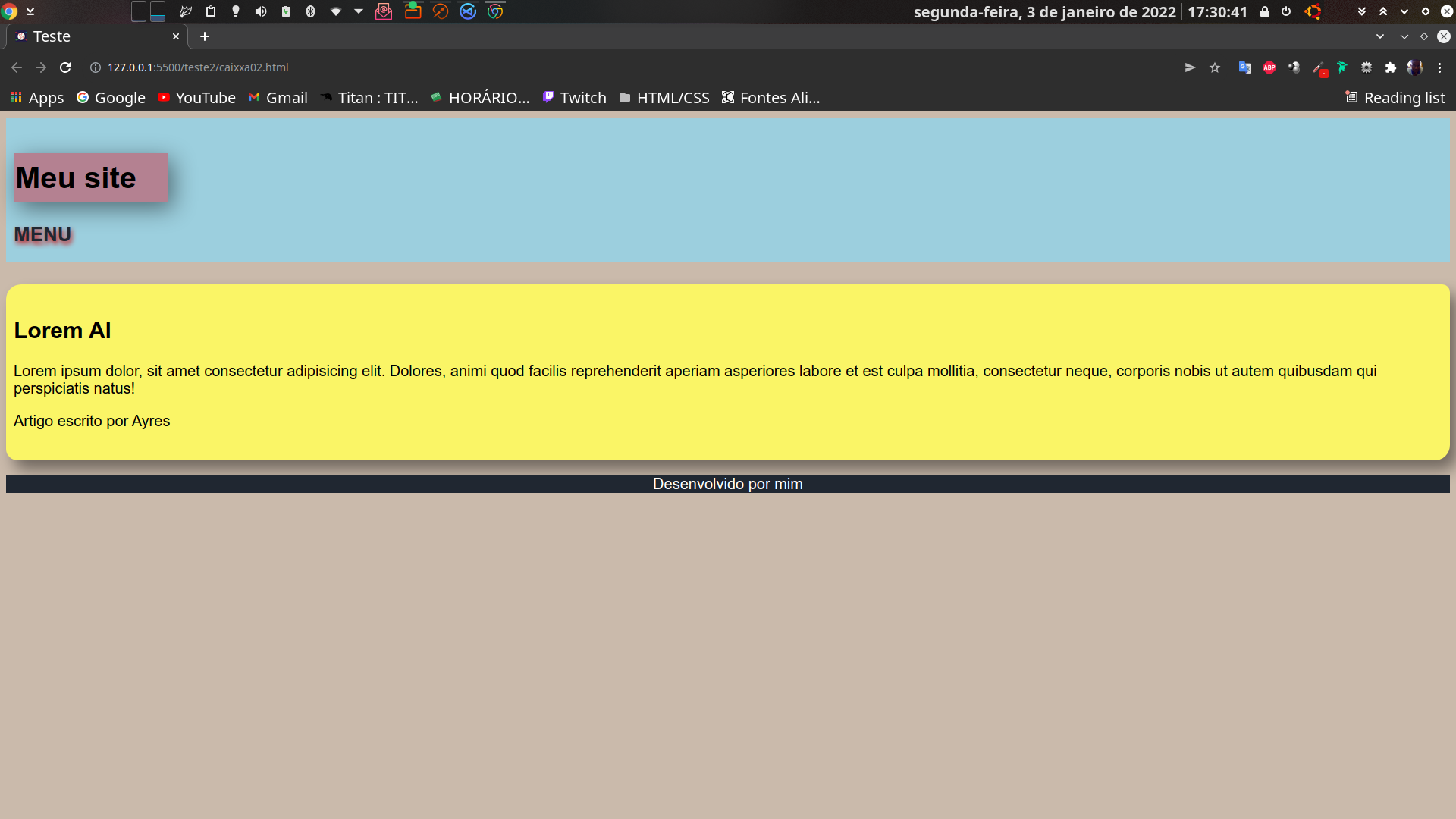


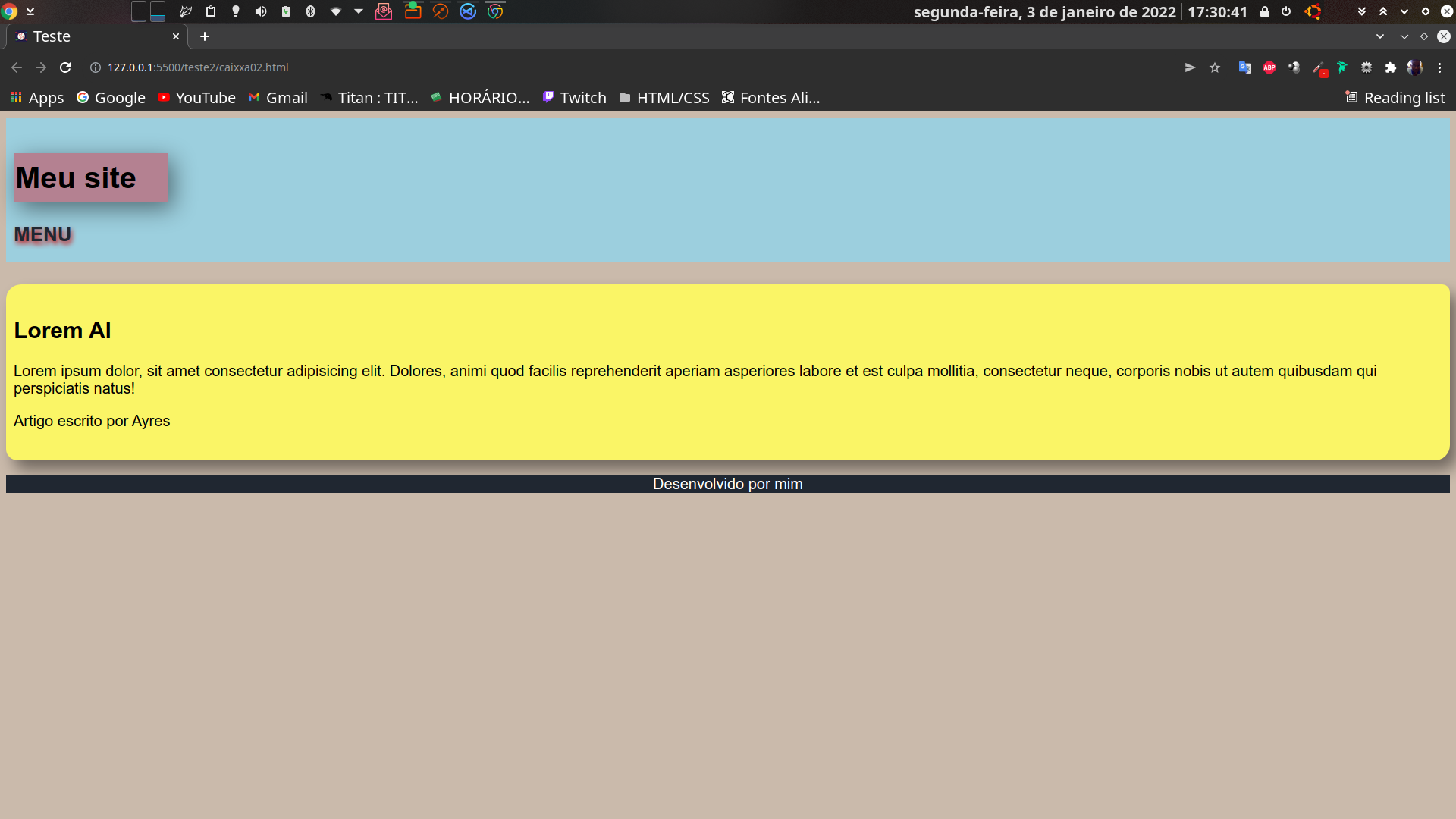
Porém como podem ver o logo ficou com um formato retangular e para os favicon é necessário ter um formato mais quadrangular, ou seja a largura e altura na resolução tem que ser iguais ou números muito proximo, no meu caso seria 100px X 100px p ideal.

Então para resolver esse problema resolvi encurtar as setas e colocar um quadrado com bordas arredondadas de fundo:



Com o logo retificado ficou assim:

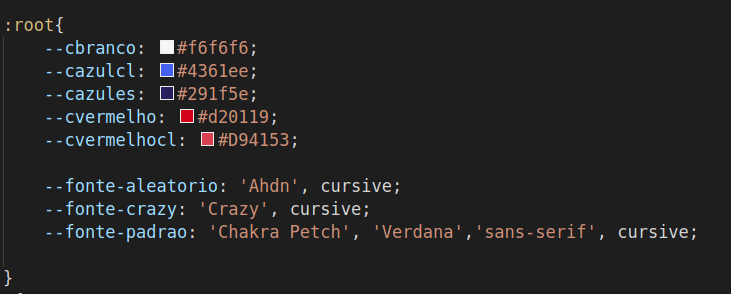




O que ficou bem mais nítido e não deformado.

**Diminuir a opacidade da cor em com variáveis globais (CSS).**

Na personalização do site com o uso do CSS, fiz a utilização de variáveis globais para as cores e as fontes do site:



Fiz isso pois as cores e as fontes ao longo dos sites serão as mesmas e isso facilita na hora em que seja necessário mudar, não será necessário mudar em cada tag mas sim somente mudar no root.

As variáveis estão com os valores em Hexadecimais das cores assim sendo é só colocar a variavel nas funções que permitem cores:

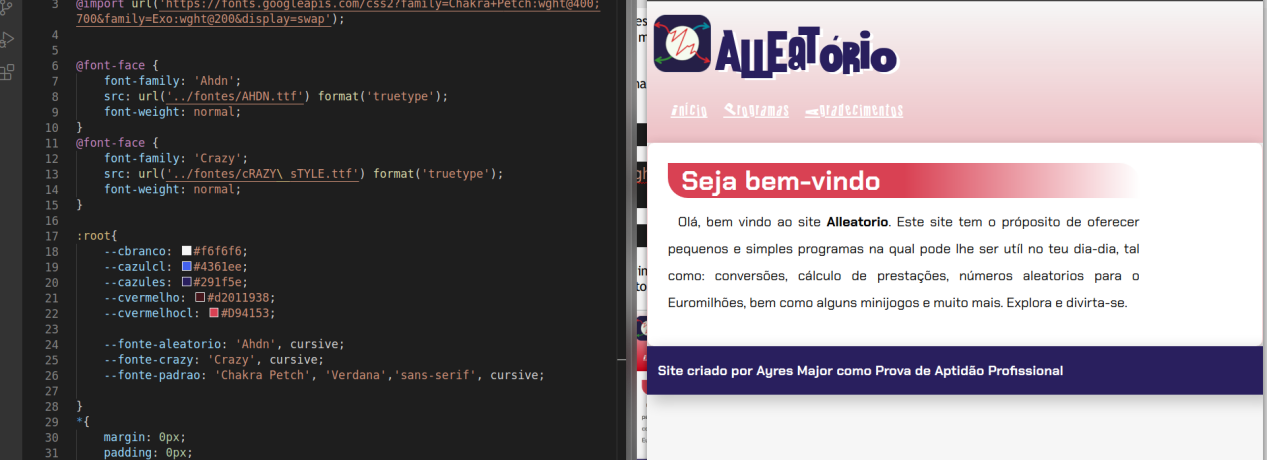
background-color: var(--cbranco);

background-image: linear-gradient(to right, var(--cvermelhocl) 50%, transparent);

color: var(--cazules);

O problema é que dessa forma não me permite diminuir livremente a opacidade, já que como é global se eu mudar o valor da cor afetara em todos os sitíos em que ela esta presente:





E se eu mudar o valor da cor no local, não estará ‘conectada’com o root oq que quer dizer que sempre que quiser mudar de cor terei que procurar e trocar.

As cores podem ser reprentadas no VS code de três formas em Hexadecimal, hsl e rgb:

:root{

--brancohex: #ffffff;

--brancohsl: hsl(0, 0%, 100%);

--brancorgb: rgb(255, 255, 255);

}

e quando a adcionados os valores da opacidade:

:root{

--brancohex: #ffffff4b;

--brancohsl: hsla(0, 0%, 100%, 0.336);

--brancorgb: rgba(255, 255, 255, 0.24);

}

Então de forma em adicionar os valores da opacidade pensei em usar os valores todos em rgb, (porque é mais fácil mudar os valores do que em hexa e estou mais familiarizado do que hsl),

e depois acrescentar o valor da opacidade (que varia de 0-1 em rgb ou hsl) quando fosse necessário.

Para isso como estou a adicionar a opacidade então deixa de ser rgb e passa a ser rgba, então testei desta forma:

div{

background-color: rgba(var(--brancorgb), 0.4);

}

Porém não deu certo, pois a variável está a receber todos o valores e funcões que lhe atribuí logo a é como se na realidade estivesse assim:

div{

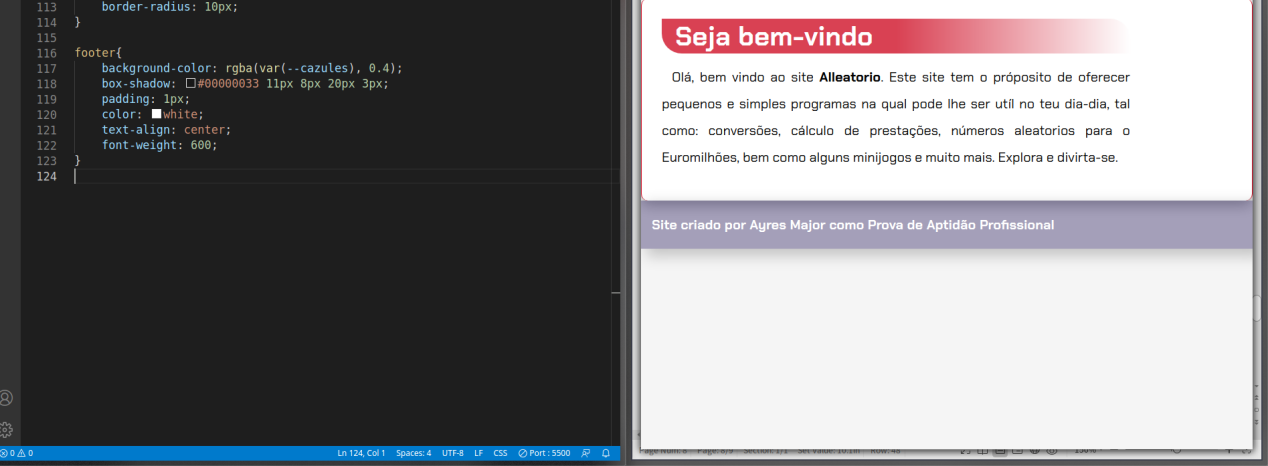
background-color: rgba(rgb(255, 255, 255), 0.4);

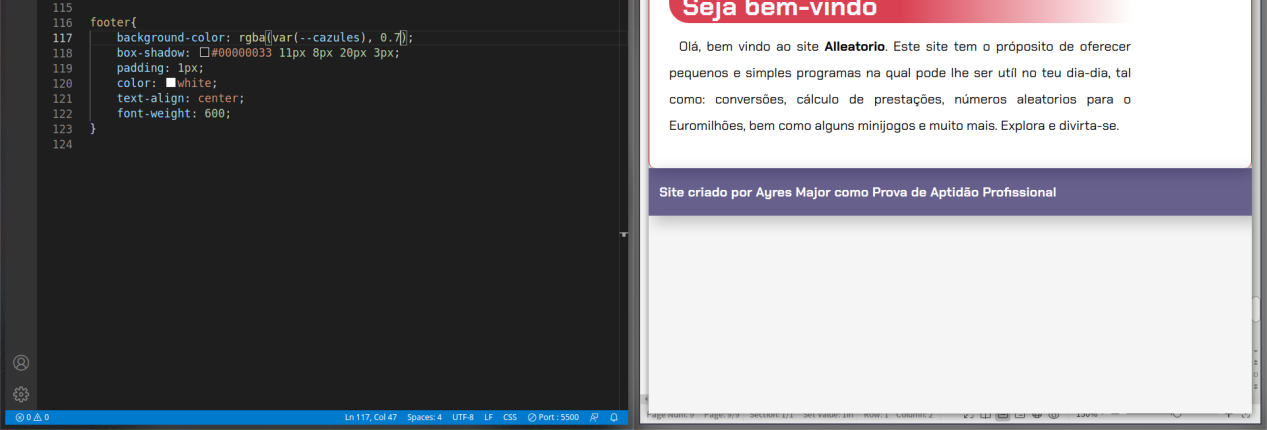
}

O que obviamente não permitira que a função rgba faça a leitura dos vaores, já que está dentro de outra função. Para resolver isso decidi então atribuir para as variaveis somente os valores se nenhuma função.

--brancorgb: 255, 255, 255;

Esse é o resultado:





A desvantagem dessa solução é que agora sou obrigado a colocar a função rgb, ou rgba sempre que utilizar alguma das cores do root.

**Tecla E e e num imput de number.**

Nos programas que requerem uma resposta númerica do usuário, uso o :

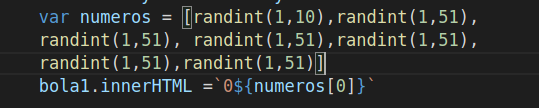
<input type="number">

isso faz com que na caixa que aparece aceite somente números. Porém também aceita “e/E” pois representa os números expoentes 10e2 = 102, porém no meu caso como não preciso fiz uma condição em que somente aceite números de 1 a 5 (no caso do programa euro milhões).

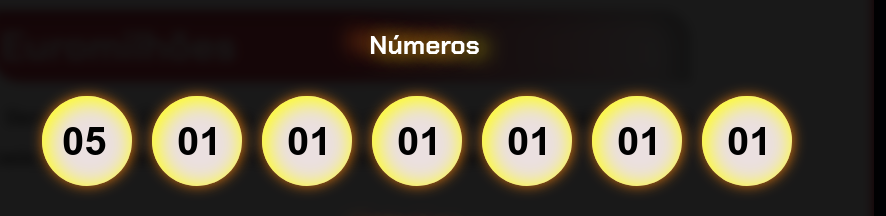
**Numeros nas bolas:**

Na aplicação por conta da estética queria que sempre os números tivessem 2 casas decimais.

Como a função randint() dá sempre números aleatorios de 1-51 os números de 1-9 seriam um problema. para tentar resolver isso primeiramente coloquei sempre um 0 presente na bola.



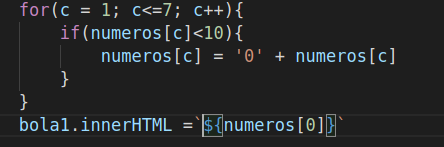
O resultado:



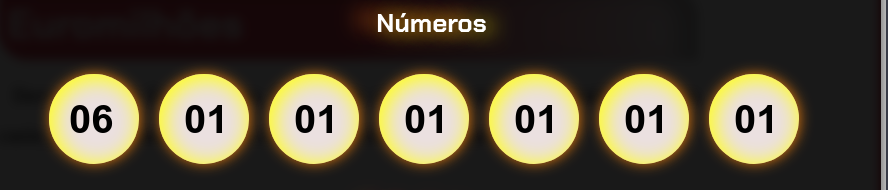
Como ainda estava na fase de teste estava a usar somente 1 bola a primeira no caso. E aparentemente deu certo, porém isso porque como estava a testar coloquei na função randint desta bola para aparecer somente números de 1-9, quando testei com todos os números deu isso:



Ou seja o 0 virou um problema quando tratava se de numeros de 2 digitos. então optei por criar um condição para isso:



resultado:





deu certo

**Altura com as bolas:**

Basicamente fiz um switch case para resolver esse problema:

if(fundo.style.width <= '900px'){

switch(palpite){

case 1:

fundo.style.height = '133%';

break;

case 2:

fundo.style.height = '193%';

break;

case 3:

fundo.style.height = '253%';

break;

case 4:

fundo.style.height = '313%';

break;

case 5:

fundo.style.height = '373%';

break;

default:

fundo.style.height = '100vh';

}

}else{

switch(palpite){

case 1:

fundo.style.height = '133%';

break;

case 2:

fundo.style.height = '193%';;

break;

case 3:

fundo.style.height = '253%';

break;

case 4:

fundo.style.height = '313%';

break;

case 5:

fundo.style.height = '373%';

break;

default:

fundo.style.height = '100vh';

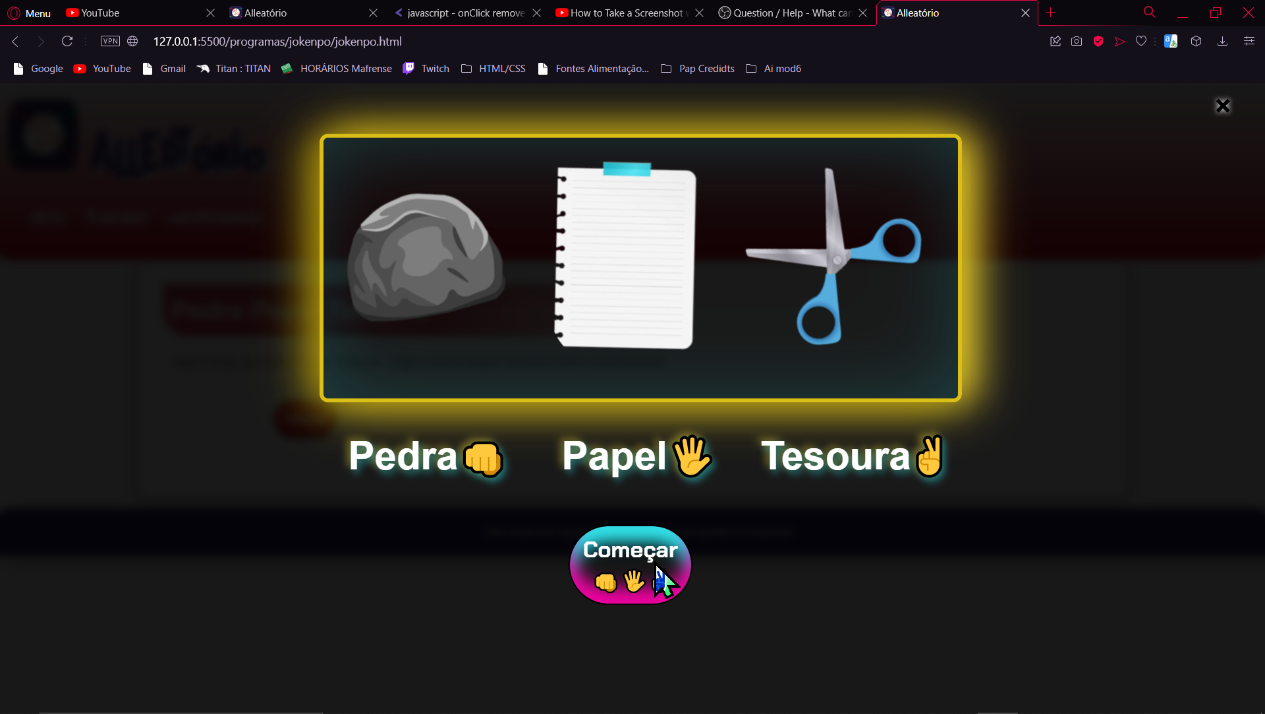
}

}

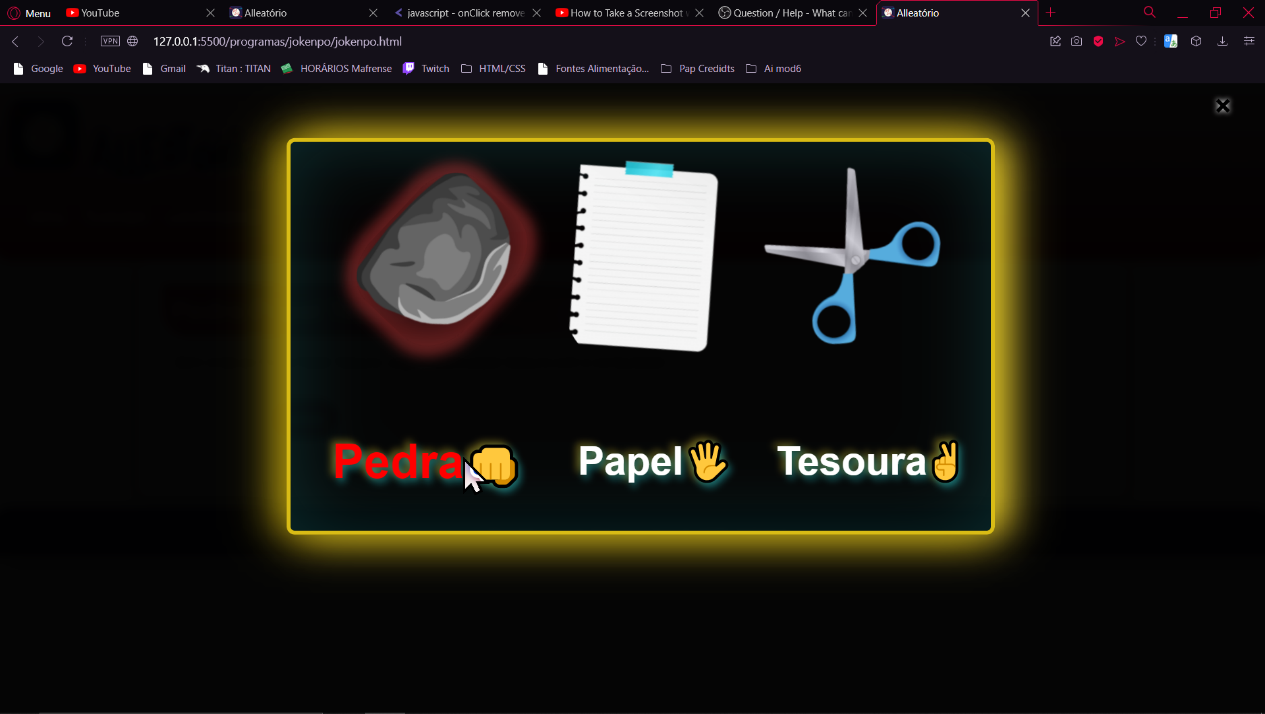
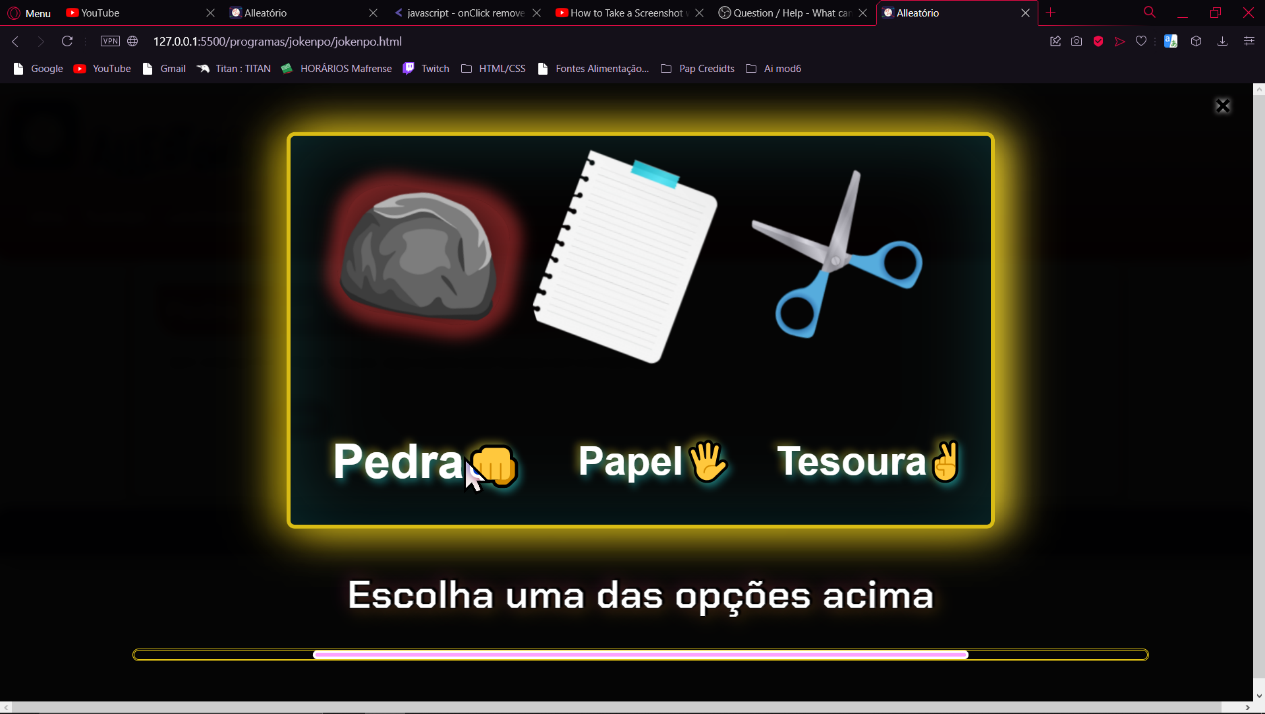
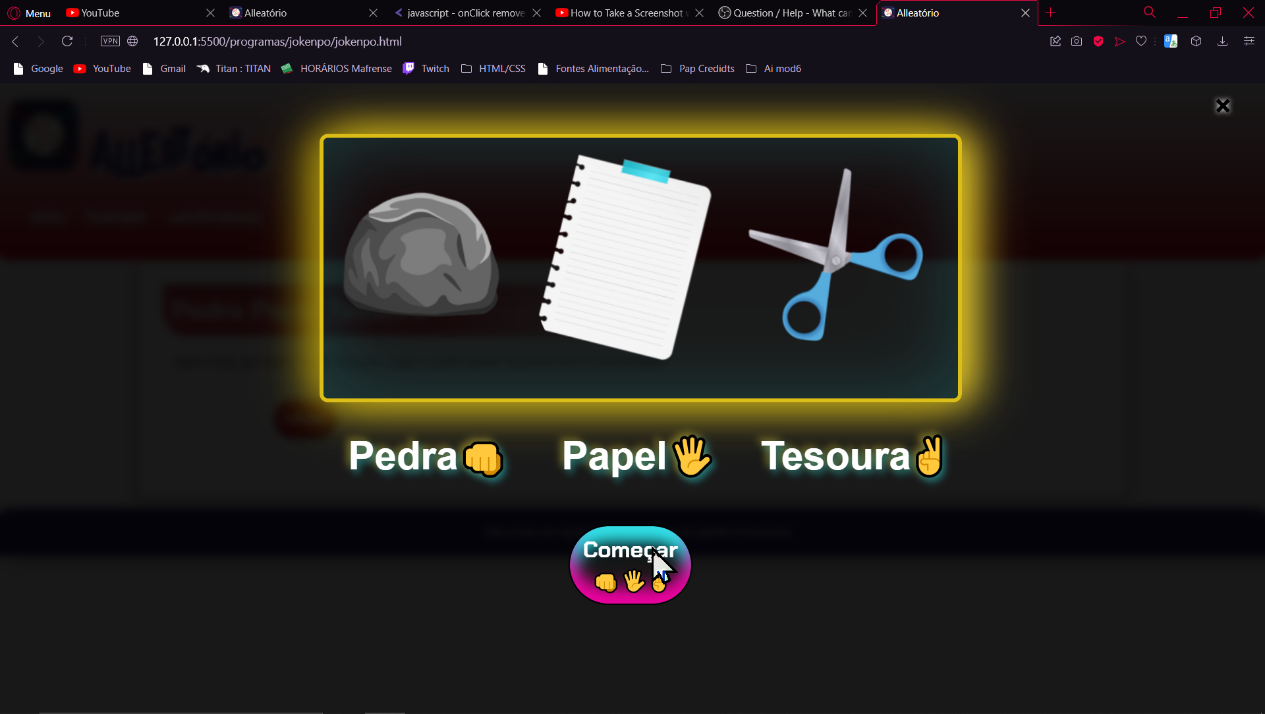
Porém percebi depois que bastava colocar aultura no auto:

fundo.style.height = 'auto';

seleção no jokenpo

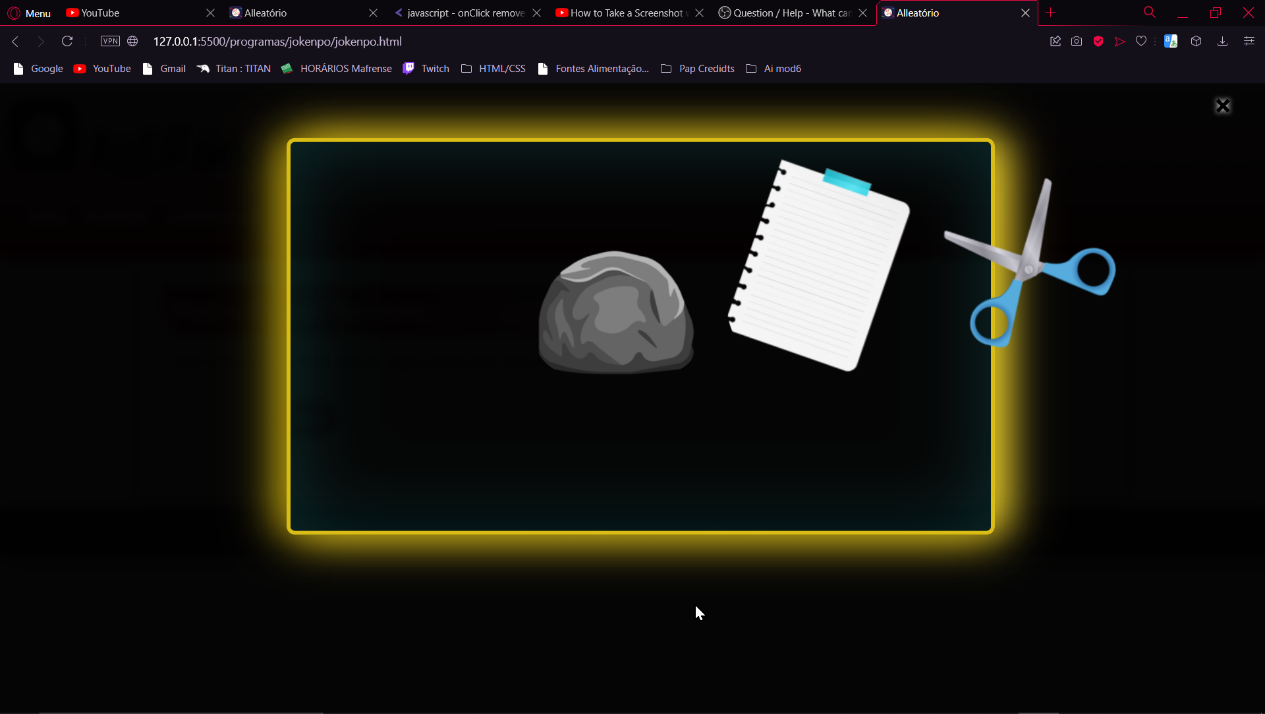






Tava a dar confilito as funções onclick e click, já que a “pedra” saía ativava a função mouseout e votava a posição incial, para resolver isso basta colocar a função mouse out vazia

algo.setAttribute("onmouseout", "")

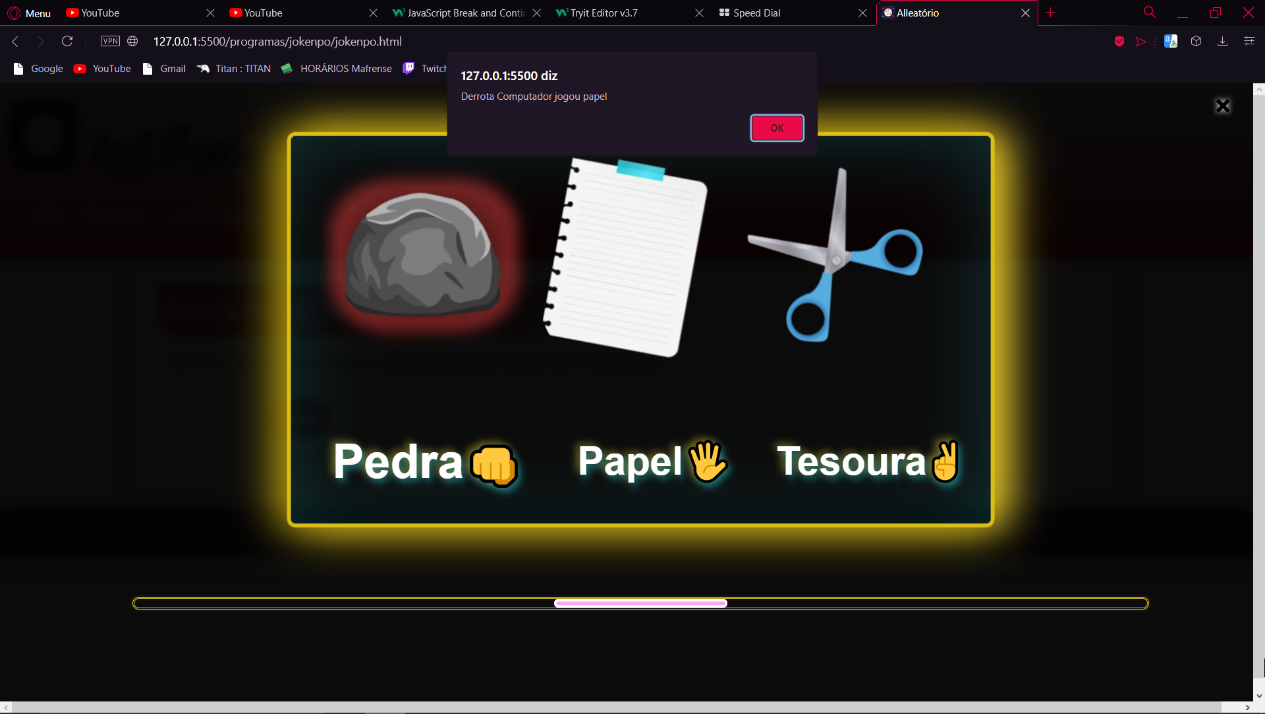


Problemas para o game over depois do tempo acabar.

Durante a seleção de umas das opções tem um tempo limitado de 15 segundos, então quando desse os 15 segundos o computador seleciona umas das opções aleatoriamente para o usuário. Então para isso tive que criar diversas estratégias.

Pensando futuramente no programa.

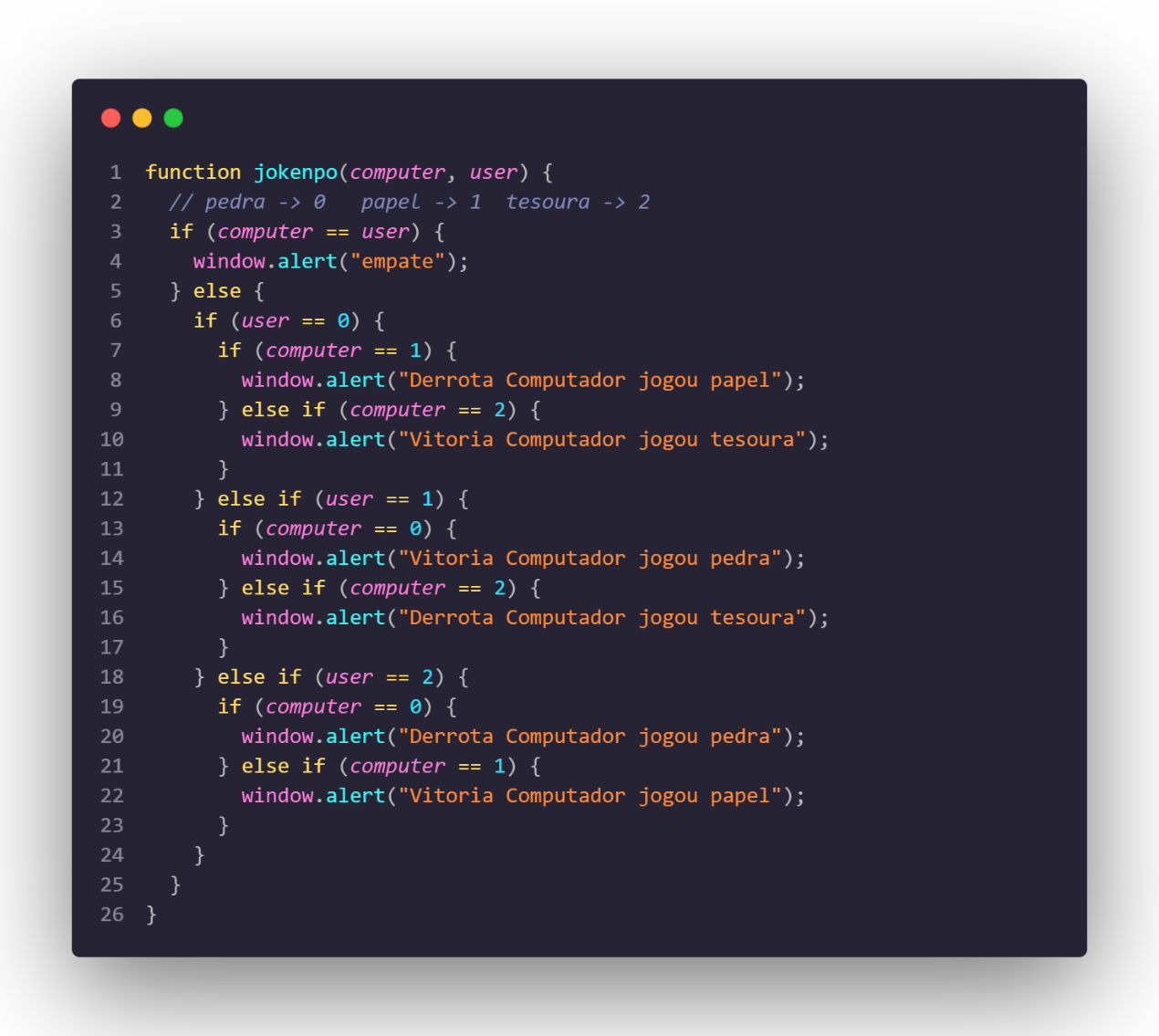
Antes de pensar nesse problema já tinha feito o programa “jogável”, ao selecionar umas das opções dava uma mensagem a anunciar o resultado, claro isso somente para testes:



E isso tava dentro da função selected



O problema é que a barra de loading estava numa outra função, que era a função start, e era dentro dessa função eu dava para eu resolver esse problema já que essa é a única função ativada depois de começar o jogo, logo o jogo parava sem a necessidade do usuário fazer algo. O problema é que como que teria que fazer basicamente a mesma coisa nas duas funções, já que a única diferença seria como o valor do “user ” seria atribuído (seja pelo computador ou pelo usuário) o restante seria igual pra as duas funções oque faria com que tivesse dois trechos de grandes de códigos iguais, logo para resolver esse problema criei uma função jokenpô, onde lá estaria esses trechos de códigos .

E 

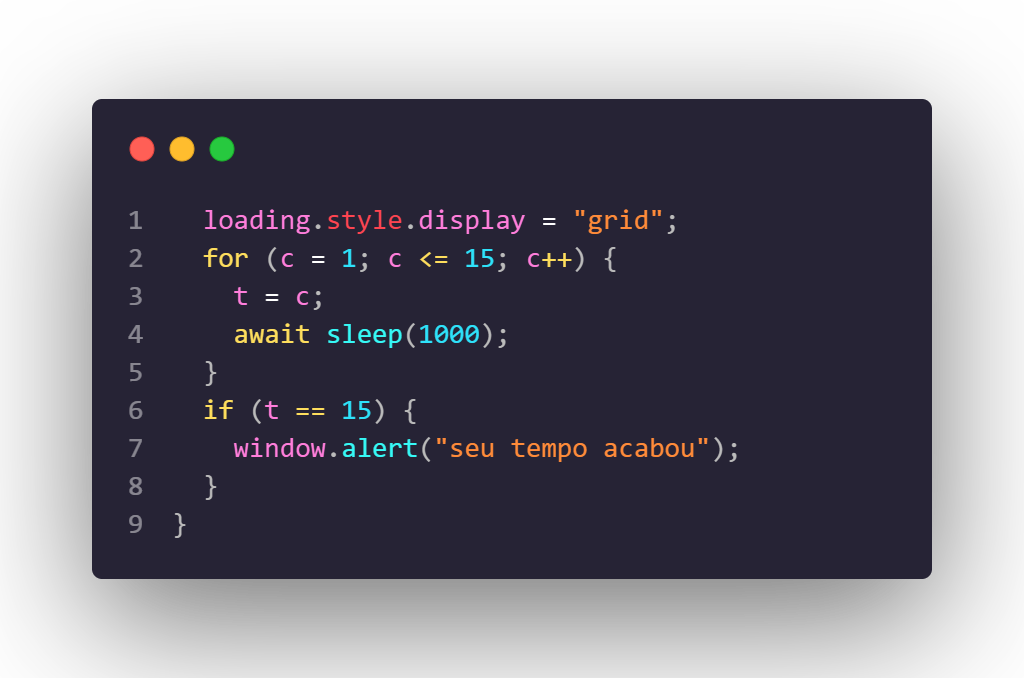
E para acessar essa função, criei duas novas variáveis computador e jogador onde o computador fica com o resultado do computador e o jogador com o do usuário, sendo assim era só colocar essas duas variaveies na função jokenpo e feito:

jokenpo(computador, jogador);

como isso teria acesso a essa função em qualquer parte do código.

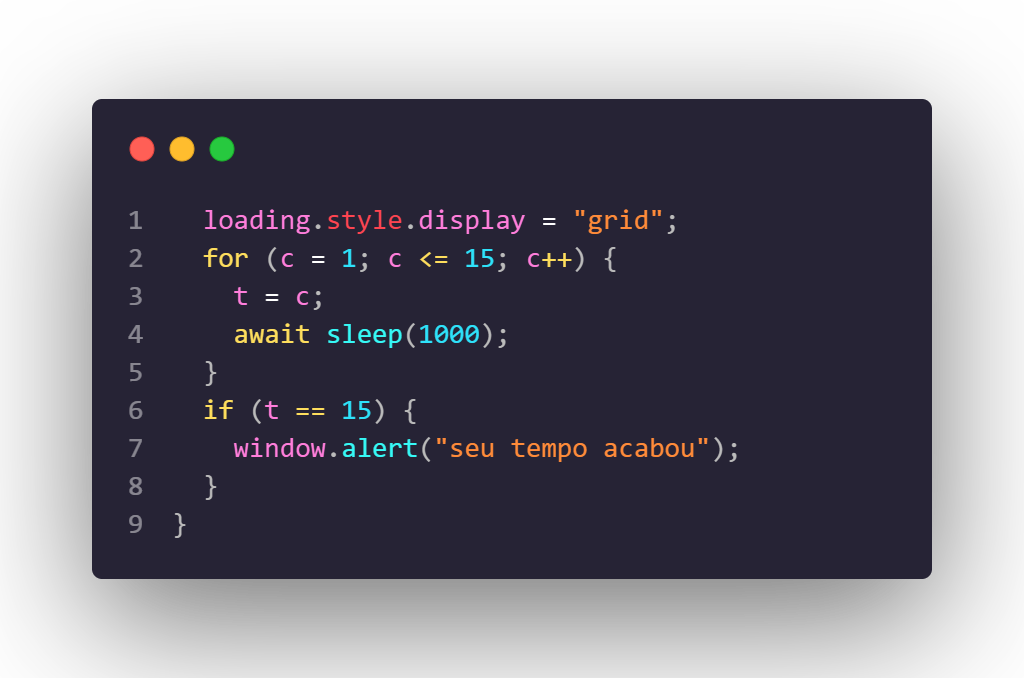
Cronometragem

Partindo agora para o problema em si, um dos passos para resolver esse problema era eu ter uma variável com o tempo real da animação. A animação foi feita no CSS no JS só colocava essa animação a agir na barra loading, então eu não tinha uma variável com tempo decorrer da animação, e segundo minhas pesquisas pra pegar esse “tempo a decorrer” seria algo complexo e que pderia criar conflitos em outras linhas de códigos logo não era viável. Então para eu ter esse tempo criei um temporizador através da função for:



Atráves da função for , faço com que a variável t mude de valores entre 1 e 15 de um em um e crescente num intervalo de 1 segundo, criando assim um temporizador de 15 segundos. (15 segundos, porque é o tempo da animação estabelecido por mim.

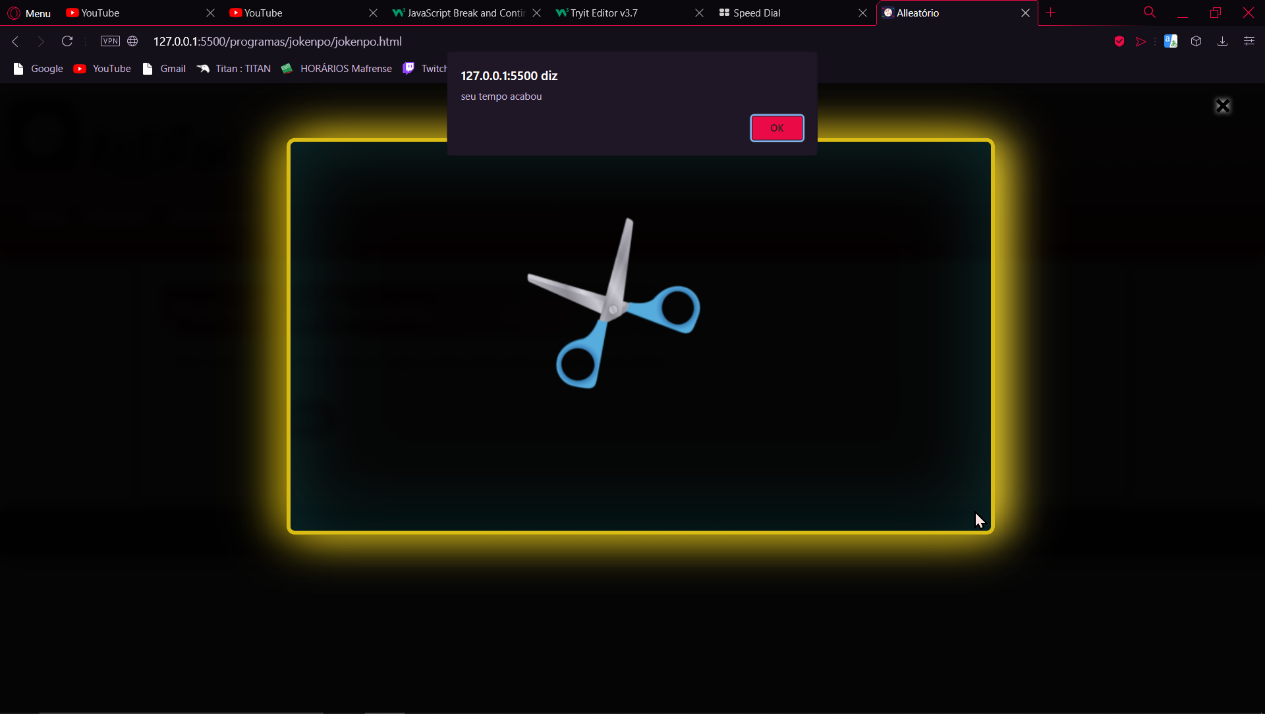
Com tendo a variável com o tempo exato da animação, essa só colocar uma condição “ se t for igual a 15 faz x coisas”:



Com isso garanti o valor automático caso o usuário não seleciona—se a tempo.

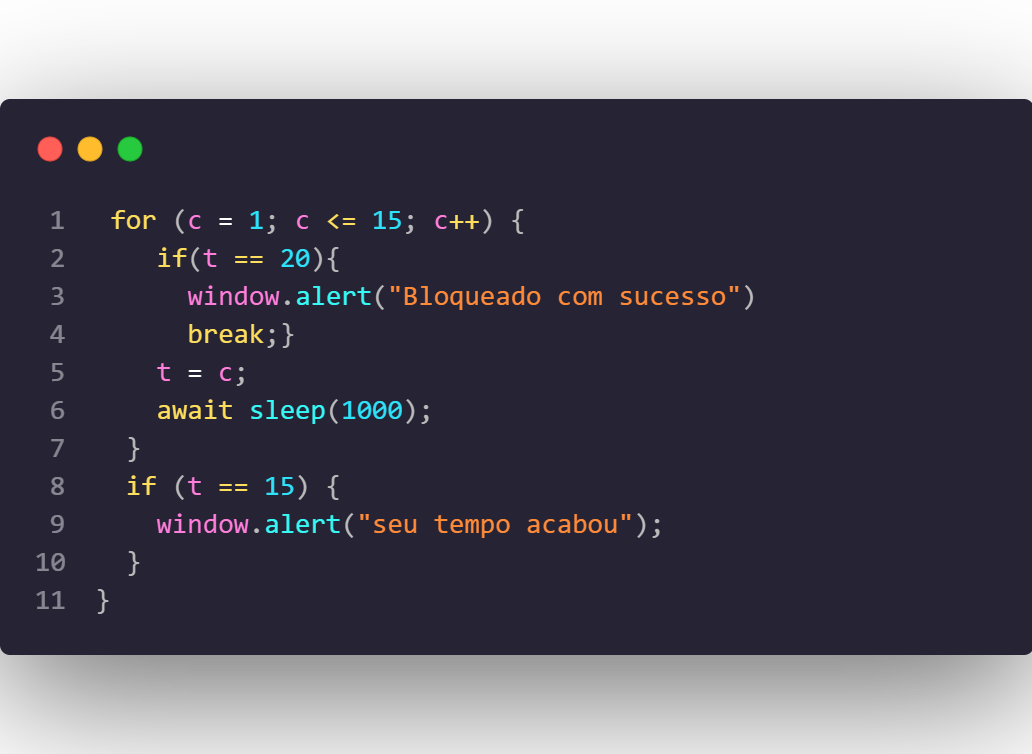
Conflito com um bom usuário:

Esse mecanismo era caso não conseguisse selecionar uma opção dentro de 15 segundos, porém quando o usuário selecionasse esse mecanismo não deveria ser acionado, mas quando fiz uns testes não foi esse o resultado:



Mesmo depois de ter selecionado dava a mensagem sobre tempo ter terminado, claro isso sendo só uma mensagem teste não teve muito impacto, mas caso tivesse as coisas completas seria bem problemático.

Isso ocorria porque apesar de ter selecionado uma das opções a o temporizador ainda tava ativo logo quando t fosse igual a 15, dava a mensagem. Logo o objetivo era parar o temporrizador quando selecionasse as opções. Para isso tentei usar o temporizador dentro de um while com diferentes condições sem sucessso, tentei desativar a a função start quando selecessiona-sse e sem efeito. Foi então que lebmbrei da função break, que é normalmente usado pra quebrar esses loops feitos pelo for ou while. Então decide atrubuir a variável a t um valor diferentes dos numeros de 1 a 15, no caso escolhi o 20, no momento que selecionasse uma das opções. E coloquei uma condição relativamente a essse novo valor com o break no temporizador:



Resolvendo assim esse problema.