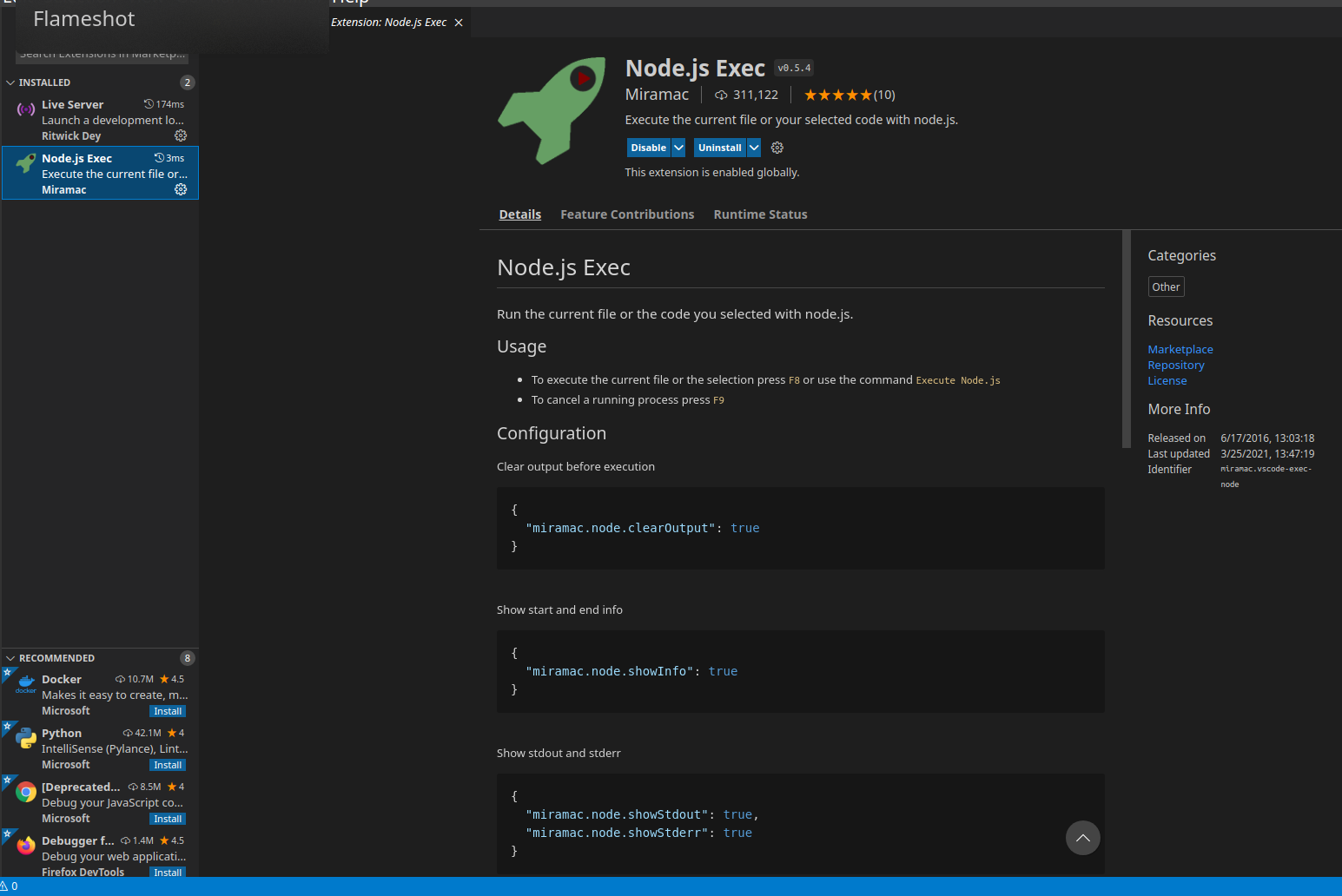
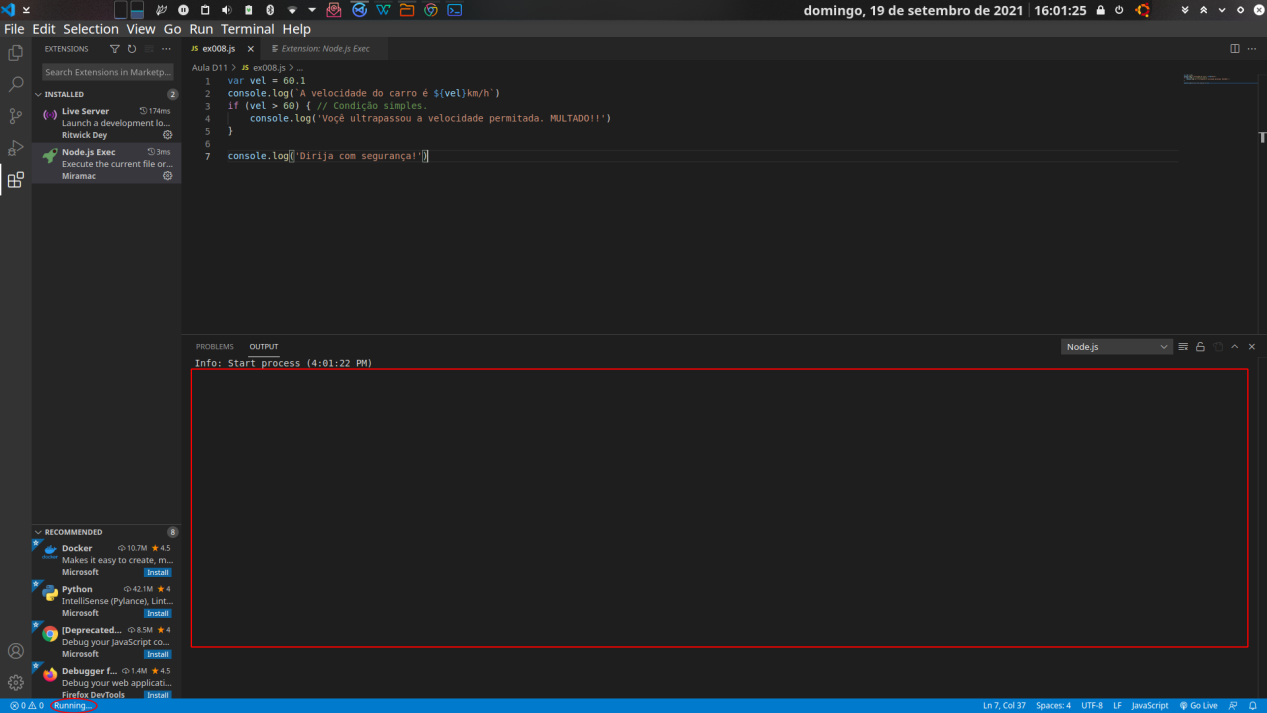
**Problemas que surgiram**

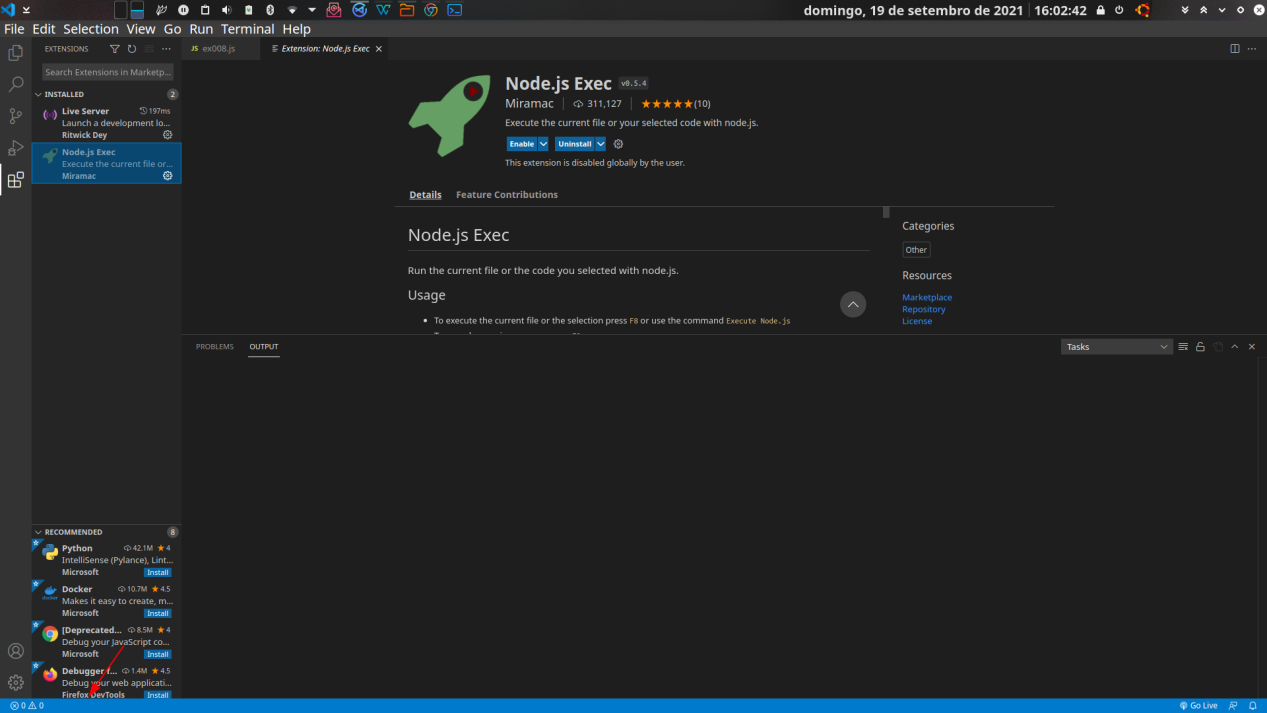
**Não estava a conseguir usar o Nodejs.**

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.

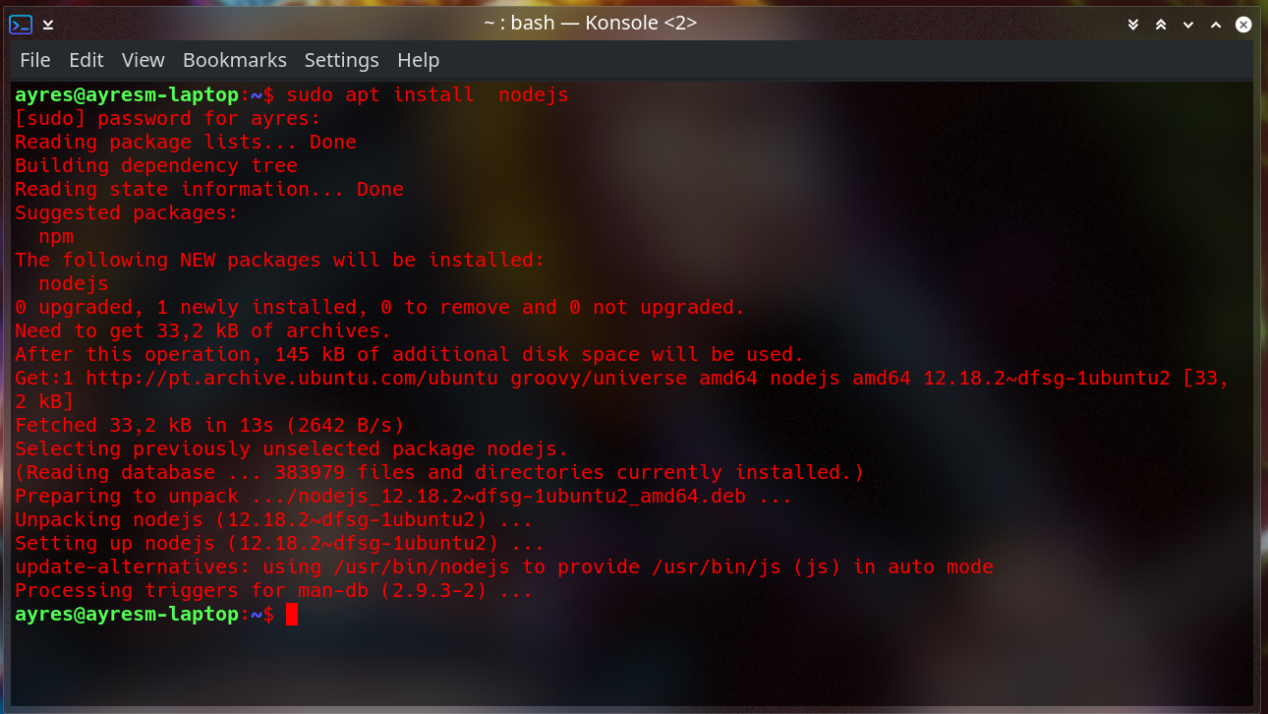


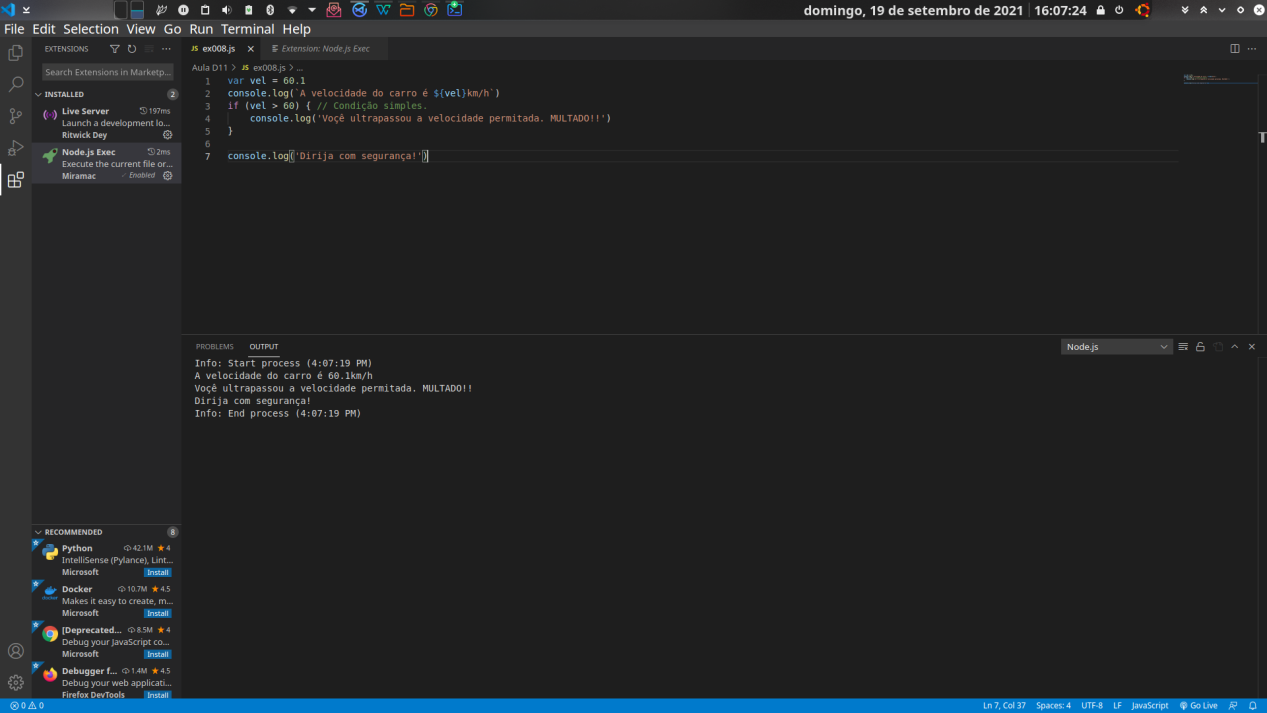
porem quando dava Run nu programa, o programa não aparecia na consola e o node continuava em running permanentemente até se desabilitado





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.



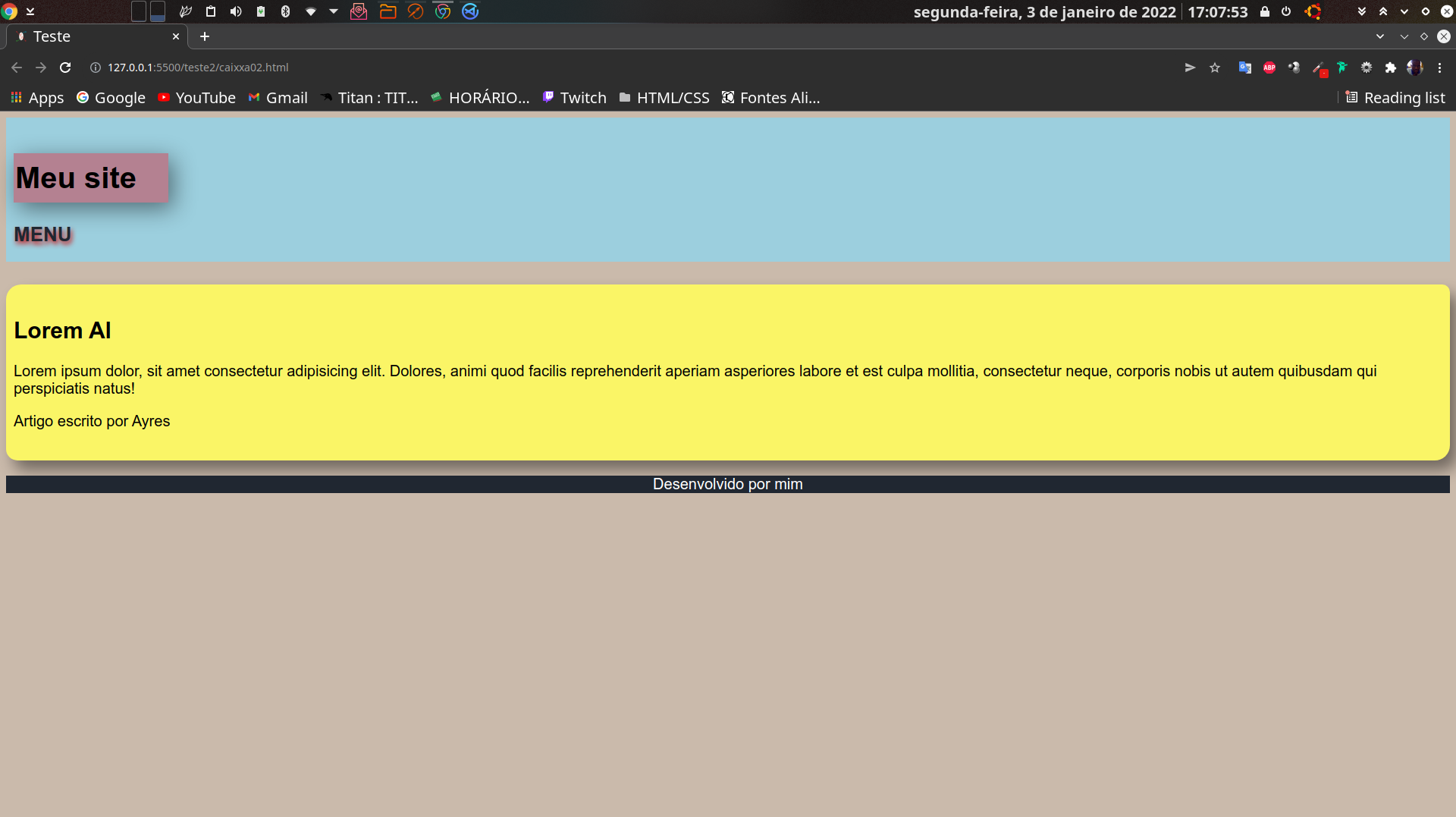
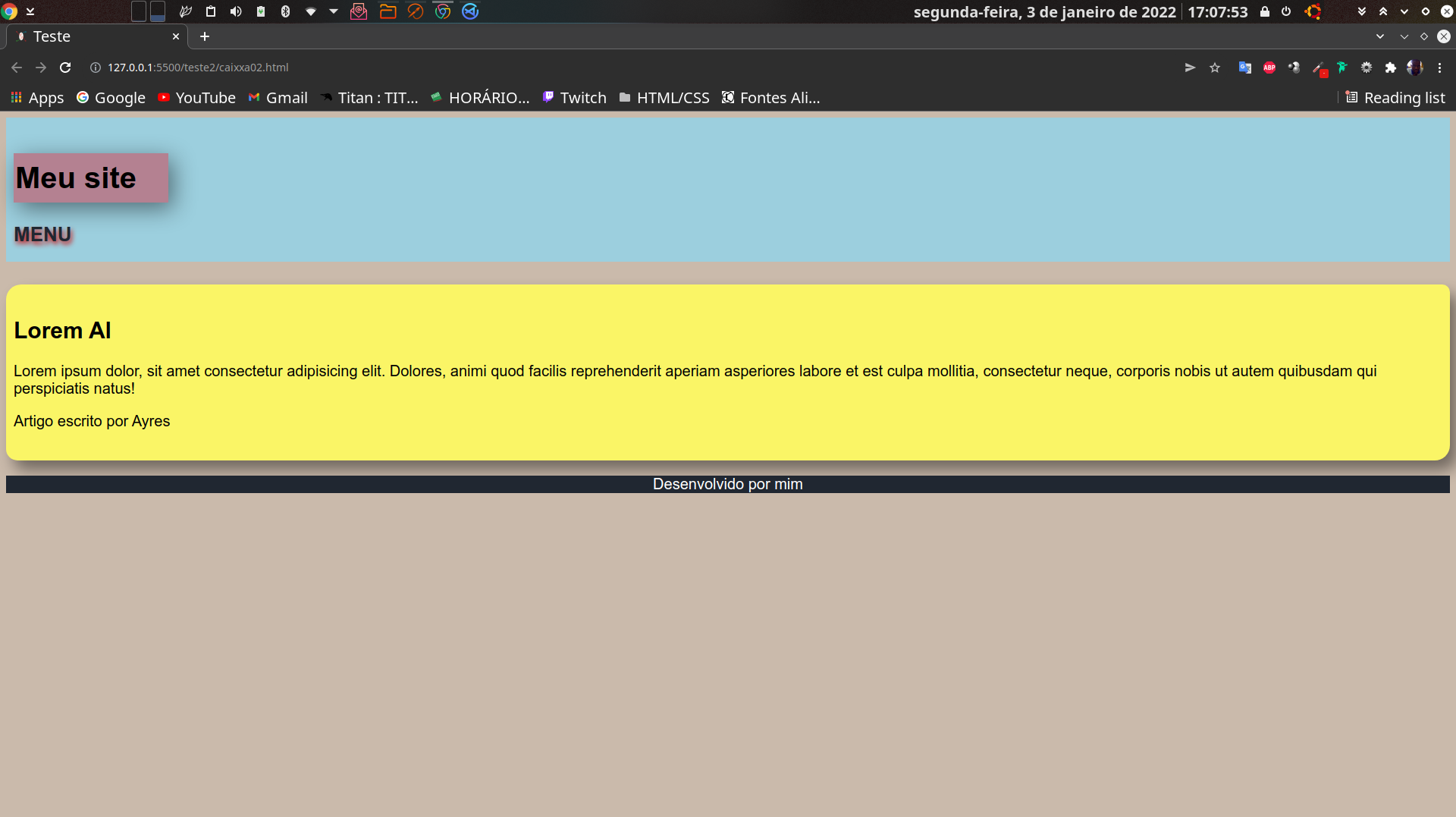


**Logo do site.**

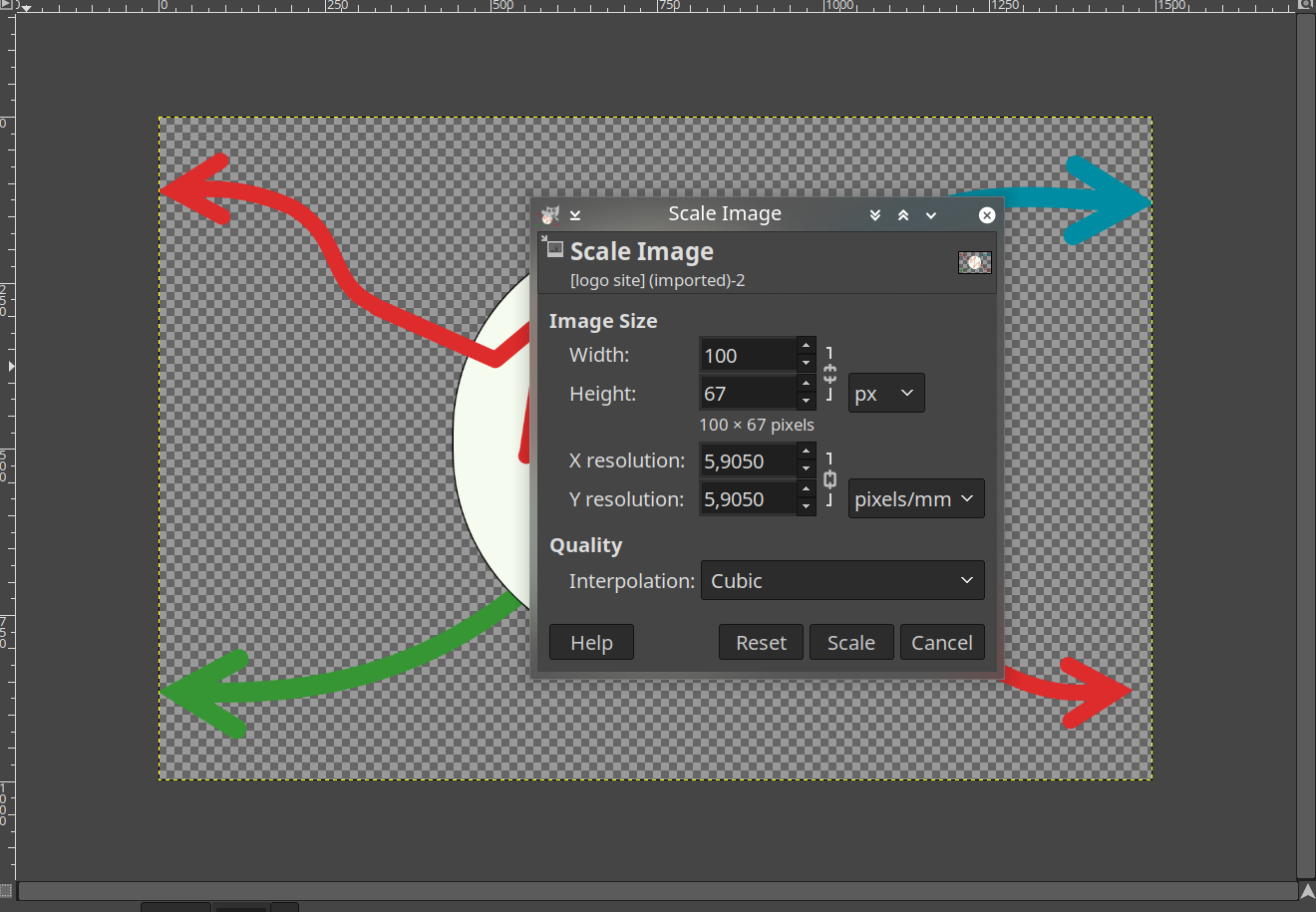
Após eu ter feito o logotípo do site baseado com as cores da paleta que escolhi ficou assim

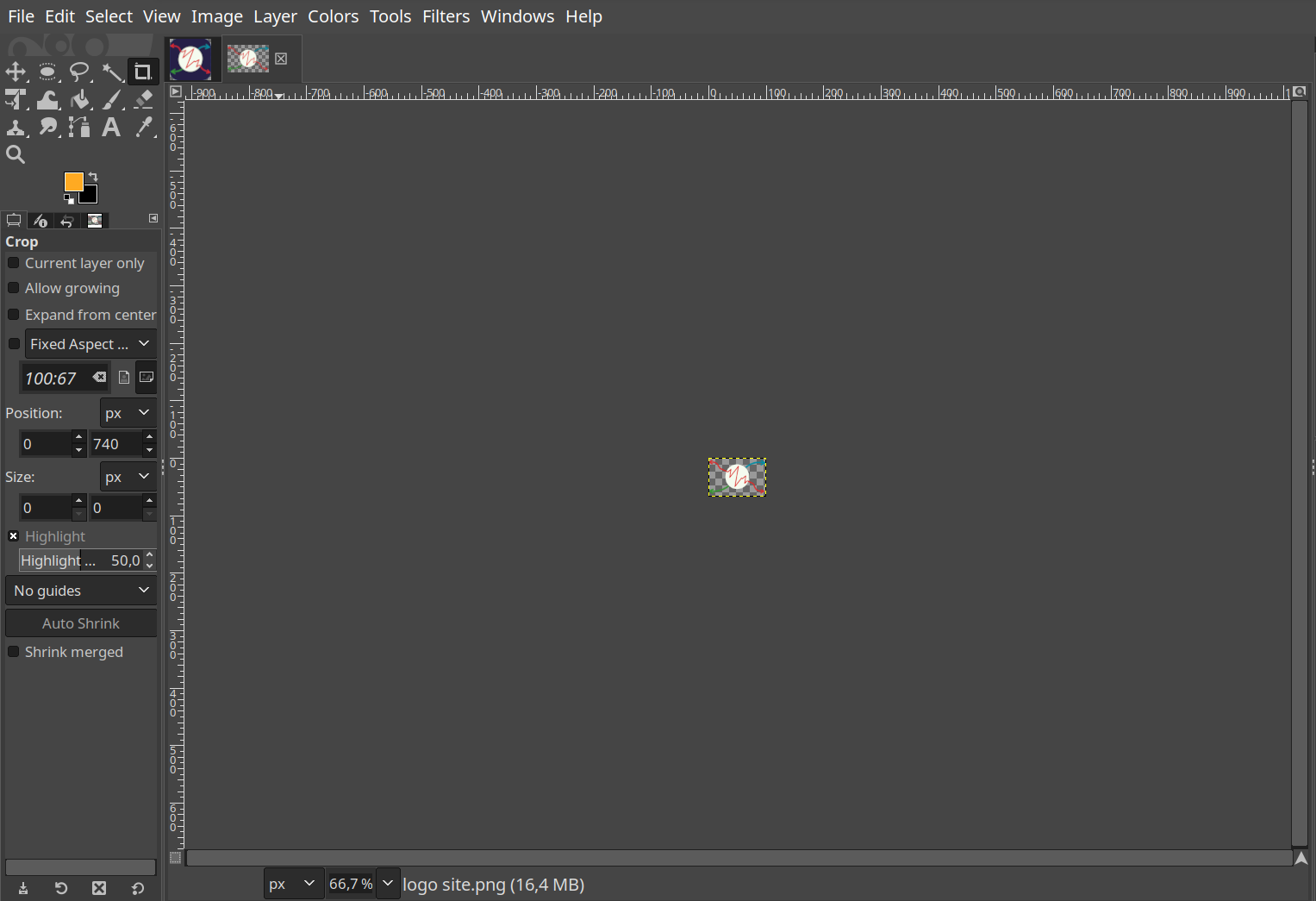


De seguida fui testar para ver se era possível usa-lo com favicon e este foi o resultado:



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:



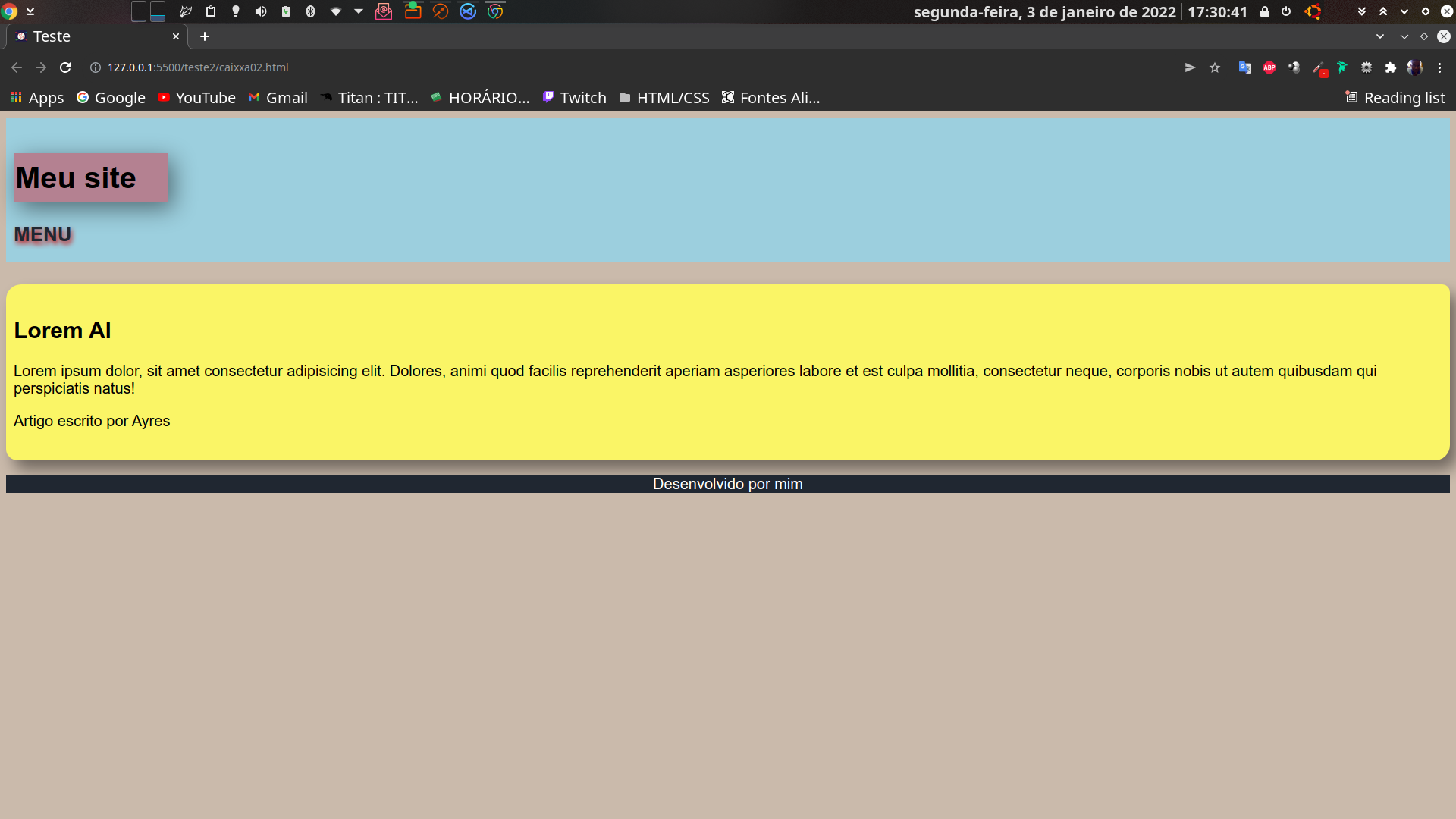


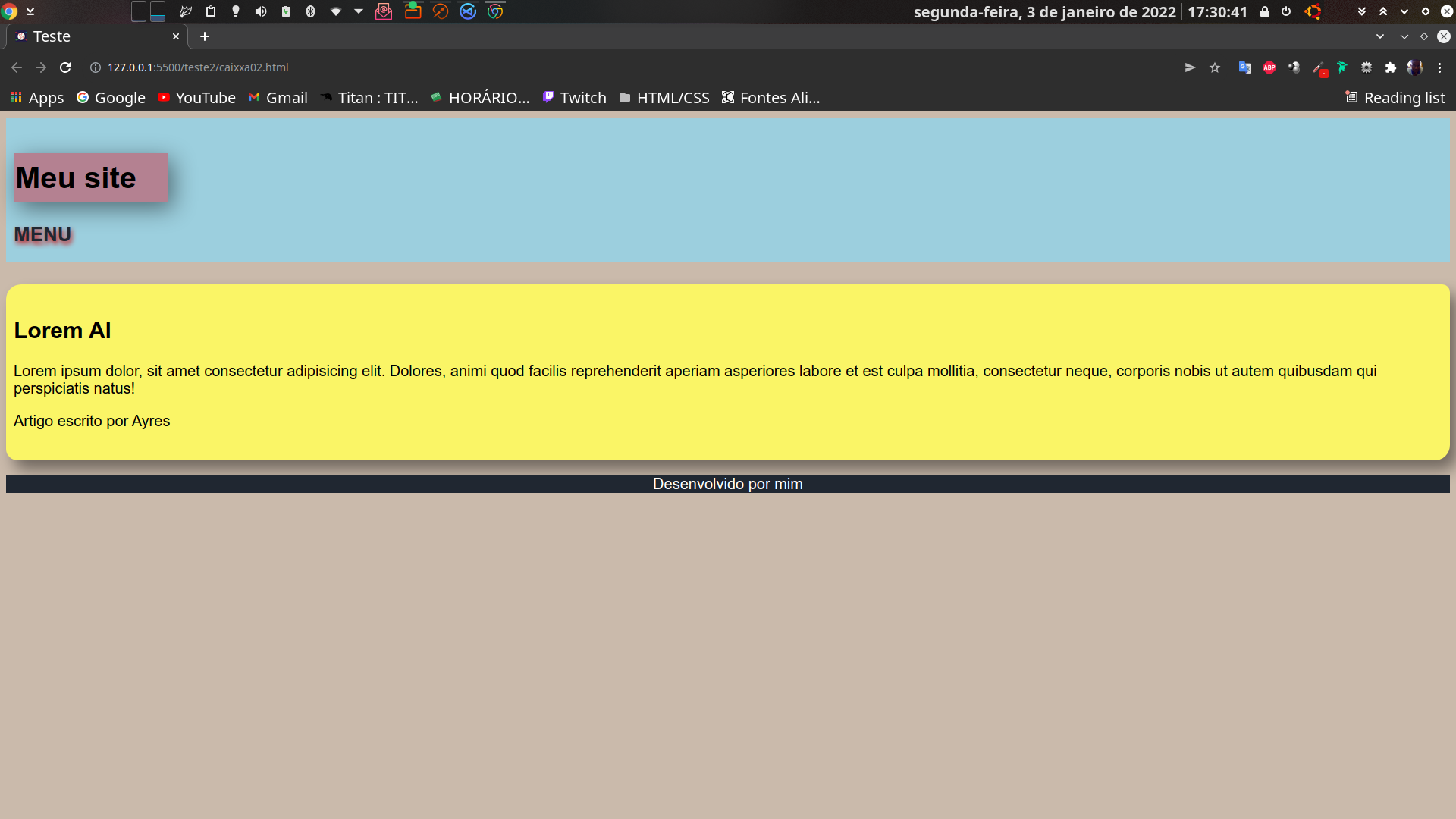
Porém como podem ver o logo ficou com um formato retangular e para os favicon é necessário ter um formato mais quadrangular, ou seja a largura e altura na resolução tem que ser iguais ou números muito proximo, no meu caso seria 100px X 100px p ideal.

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



Com o logo retificado ficou assim:

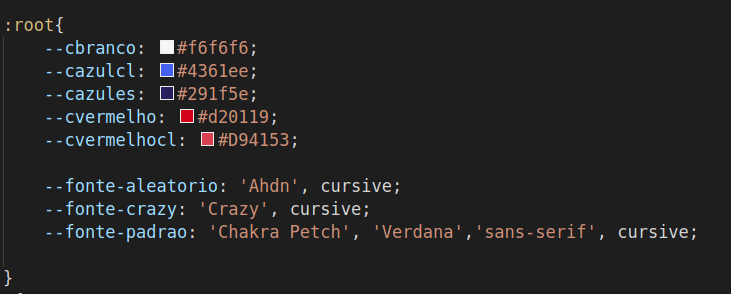




O que ficou bem mais nítido e não deformado.

**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:

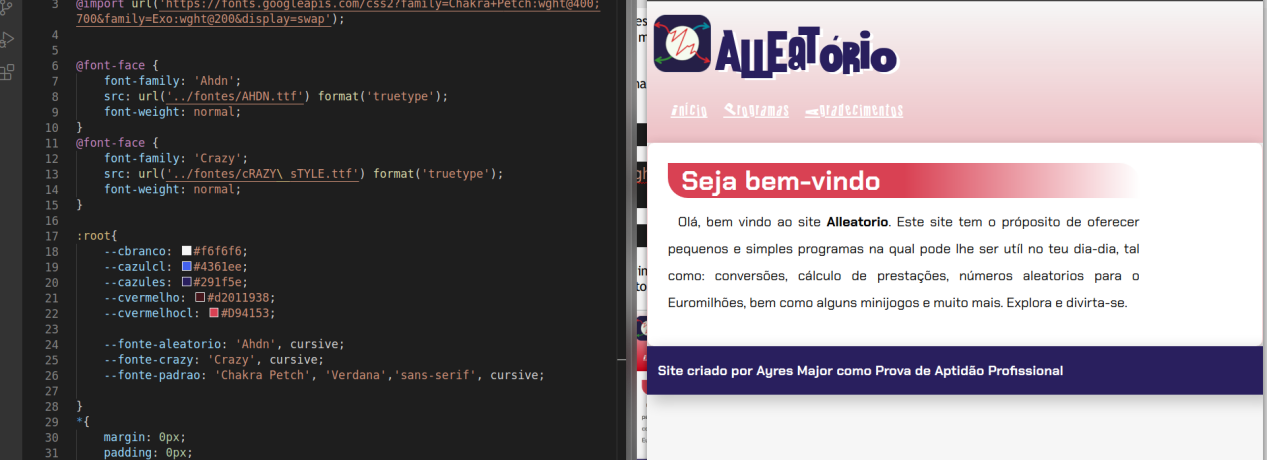
background-color: var(--cbranco);

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{

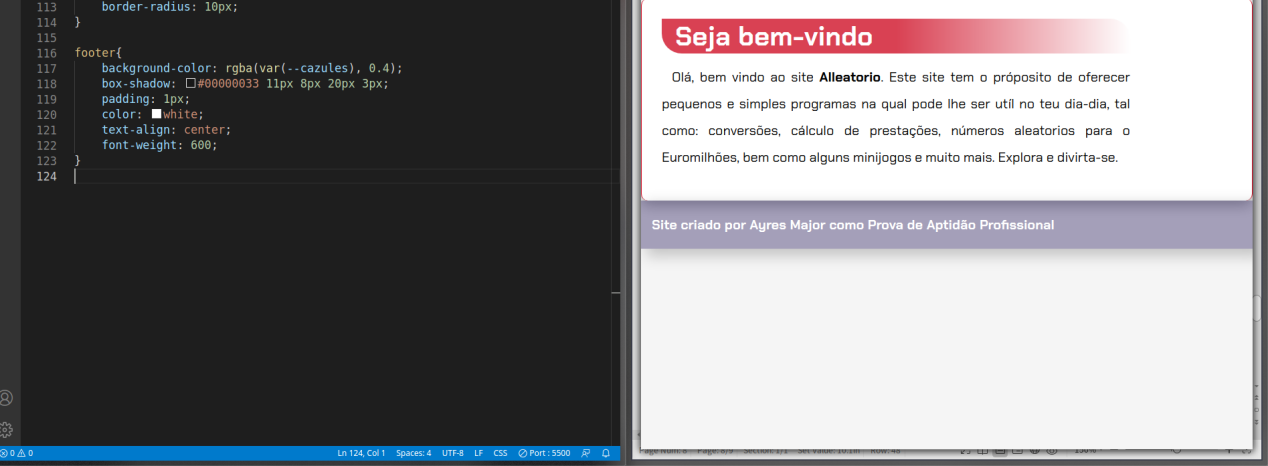
background-color: rgba(rgb(255, 255, 255), 0.4);

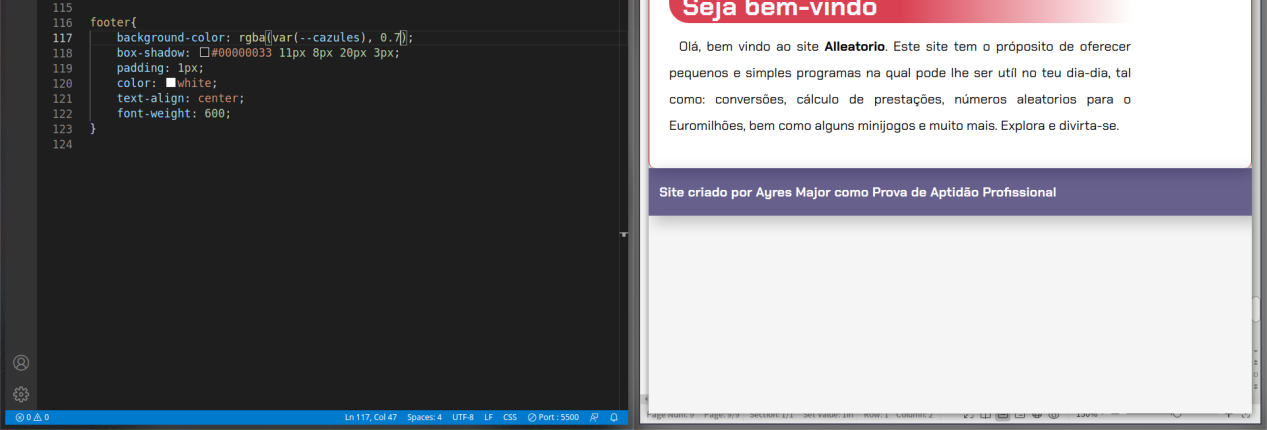
}

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 :

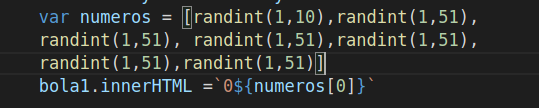
<input type="number">

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).

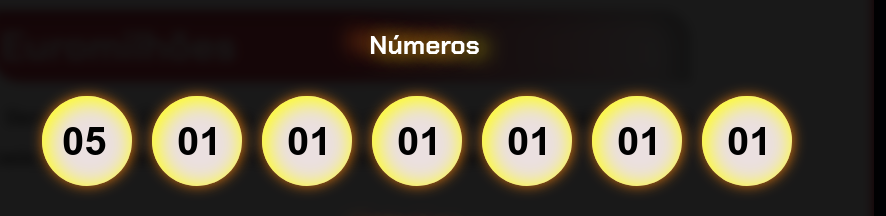
**Numeros nas bolas:**

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.



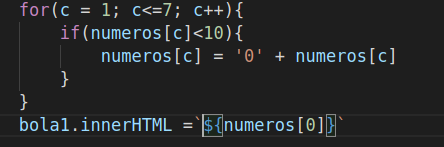
O resultado:



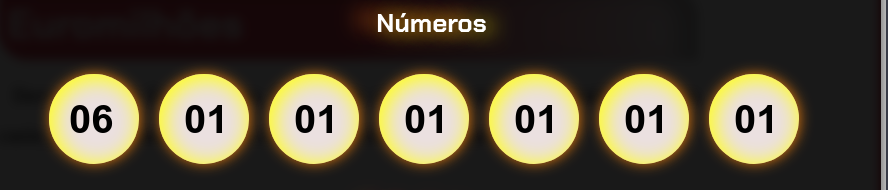
Como ainda estava na fase de teste estava a usar somente 1 bola a primeira no caso. E aparentemente deu certo, porém isso porque como estava a testar coloquei na função randint desta bola para aparecer somente números de 1-9, quando testei com todos os números deu isso:

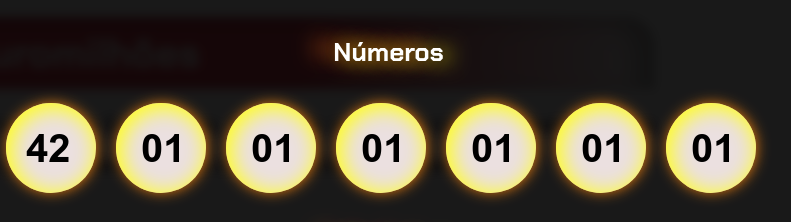


Ou seja o 0 virou um problema quando tratava se de numeros de 2 digitos. então optei por criar um condição para isso:



resultado:





deu certo