Técnico de Eletrónica, Automação e Computadores

Prova de Aptidão Profissional

Título do Trabalho

Triénio de 20**18**/20**21**



Autor:

**Ayres Major Nº4**

Professor Orientador:

**Hugo António**

Escola Técnica e Profissional de Mafra, 31 de maio de 2021

Índice

[Índice de Figuras iii](#_Toc101646642)

[Índice de Tabelas iv](#_Toc101646643)

[Agradecimentos v](#_Toc101646644)

[Resumo vi](#_Toc101646645)

[Abstract vii](#_Toc101646646)

[Simbologia viii](#_Toc101646647)

[1 - Introdução 1](#_Toc101646648)

[2 - Escolha e Fundamentação do Projeto 2](#_Toc101646649)

[2.1. Conceção do Projeto 2](#_Toc101646650)

[3 - Fases de Desenvolvimento 3](#_Toc101646651)

[3.1. As linguagens utilizadas. 3](#_Toc101646652)

[3.1.1. Python 3](#_Toc101646653)

[3.1.2. JavaScript 3](#_Toc101646654)

[3.1.3. HTML 3](#_Toc101646655)

[3.1.4. CSS 3](#_Toc101646656)

[3.2. O Site 3](#_Toc101646657)

[3.2.1. Cores 4](#_Toc101646658)

[3.2.2. Logotipo 4](#_Toc101646659)

[3.2.3. *Layout* 6](#_Toc101646660)

[3.2.4. Atuação do HTML 6](#_Toc101646661)

[3.3. Os programas selecionados. 6](#_Toc101646662)

[3.3.1. Pedra Papel Tesoura 7](#_Toc101646663)

[3.4. Lista de Material 12](#_Toc101646664)

[3.5. Lista de ferramentas 13](#_Toc101646665)

[3.6. Tarefas e Atividades 15](#_Toc101646666)

[3.7. Estimativa Orçamental 15](#_Toc101646667)

[3.8. Orçamento Final 15](#_Toc101646668)

[3.9. Resultados Obtidos 15](#_Toc101646669)

[3.10. Análise dos Resultados 15](#_Toc101646670)

[4 - Considerações Finais 17](#_Toc101646671)

[5 - Referências Bibliográficas 18](#_Toc101646672)

[6 - Anexos (Opcional) 19](#_Toc101646673)

# Índice de Figuras

# Índice de Tabelas

# Agradecimentos

# Resumo

O Resumo deve descrever sumariamente o trabalho e o relatório no seu conjunto. Sintetiza de forma clara, concisa e completa os conteúdos do Relatório, incluindo o problema abordado, a metodologia utilizada, os resultados e as conclusões. Deve ser a última secção a ser escrita, apenas quando tudo o resto já está terminado.

O Resumo é um texto que, não devendo exceder dois parágrafos, consegue captar toda a essência do que é transmitido no Relatório. A decisão de ler ou não um relatório na sua totalidade é muitas vezes decidido com base no Resumo: conforme a impressão com que fica do Resumo, o Leitor vai decidir se vale a pena, ou não, continuar a ler o resto do relatório. Cabe ao Autor fazer um resumo que cative o Leitor e o leve a interessar-se pelo resto do Relatório.

A linguagem utilizada no resumo deve, na medida do possível, evitar o recurso à linguagem técnico, de modo a permitir a sua leitura e compreensão por quem não seja especialista da área.

# Abstract

# Simbologia

px – pixéis

vw – largura da *viewport.*

vh – altura da *viewport.*

em – tamanho da fonte

**Acrónimos**

ETPM – Escola Técnica e Profissional de Mafra

PAP – Prova de Aptidão Profissional

TEAC – Técnico de Eletrónica Automação e Computadores

HTML – *Hypertext Markup Language* (Linguagem de Marcação Hipertexto)

CSS – *Cascading Style Sheets* (Folhas de Estilo em Cascata)

JS – JavaScript

VS – Visual Studio

# Introdução

A Introdução faz a apresentação geral do trabalho descrito no Relatório: qual o problema que se pretende resolver, o seu enquadramento e justificação, a metodologia utilizada. A Introdução dá ao Leitor, para além de uma perspetiva geral sobre o trabalho realizado, o porquê da sua realização e de que forma esse trabalho se relaciona com o problema que se propõe tratar: se se trata de um aspeto particular do problema, se a resposta apresentada é de aplicação universal ou se está sujeita a algumas condicionantes, …

Um relatório é uma ferramenta essencial para qualquer profissional em Ciências ou Engenharia. Para ser eficaz, um relatório deve ser Objetivo, Conciso, completo e Conclusivo. A estrutura de um relatório deve ser adaptada à situação concreta em que é utilizado, havendo um padrão que é comum à grande maioria das situações: uma introdução, a apresentação da metodologia utilizada, os resultados e a análise e discussão dos resultados obtidos. A autenticidade dos que é relatado deve ser escrupulosamente respeitada e todas as alegações sobre a autoria do trabalho apresentado devem ser corretas e verdadeiras; nesse sentido, o plágio ou qualquer outra situação menos correta é inadmissível. Para além das questões de conteúdo, há também que ter em atenção todas as questões de forma, nomeadamente o que se relaciona com a correção da linguagem utilizada, por forma a garantir a legibilidade e objetividade do trabalho apresentado.

# Escolha e Fundamentação do Projeto

## Conceção do Projeto

Apresenta o problema que se pretende resolver ou atacar. Esta secção define os objetivos do trabalho relatado e nela deve ficar clara para o Leitor, qual a pergunta (o ponto de partida) a que o trabalho pretende dar resposta.

Em trabalhos técnicos e científicos, a Descrição do Problema é complementada com o seu enquadramento, que pode incluir, por exemplo, o Estado da Arte, isto é, uma apresentação do conhecimento existente no momento sobre o assunto tratado no texto. Quando aplicável, pode conter também uma análise das soluções potencialmente concorrentes com a analisada no documento, ponderando as suas vantagens e desvantagens.

Esta secção detalha, quando necessário, os aspetos complementares em relação à forma como se abordou o problema: se se está a estudar um aspeto particular do problema, se a resposta encontrada está dependente de hipóteses prévias ou se, pelo contrário, é universal, …

# Fases de Desenvolvimento

Os programas elaborados para projeto, estão presentes de num site com tecnologias cliente-side: HTML5, CSS3 e JavaScript/ECMAScript 6

## As linguagens utilizadas.

Para a realização deste projeto foi utilizado quatro linguagens: Python3, HTML5, CSS3 e JavaScript.

### Python

Python é uma linguagem de programação. Esta linguagem pode ser utilizada em desenvolvimento *web* (servidores), desenvolvimento de softwares, programas matemáticos, entre outros. Neste projeto foi utilizado a versão mais recente o Python 3.

### JavaScript

Tal como Python, JavaScript também é uma linguagem de programação, que geralmente é conhecida como a linguagem script de páginas web (também pode ser utilizada em ambientes fora do browser, como o NodeJs). Está linguagem é a implementação da especificação ECMAScript. Neste projeto foi utilizado a versão mais recente o ECMAScript 2015 ou ECMAScript 6.

### HTML

HTML é uma linguagem de marcação utilizada para definir a estrutura do conteúdo de uma página web, através de marcação, as *tags.* A marcação inclui elementos especiais para cada tipo de conteúdo que possa ter na página (títulos, imagens, figuras e vídeos, entre outros). Neste projeto foi utilizado a versão mais recente a HTML 5.

### CSS

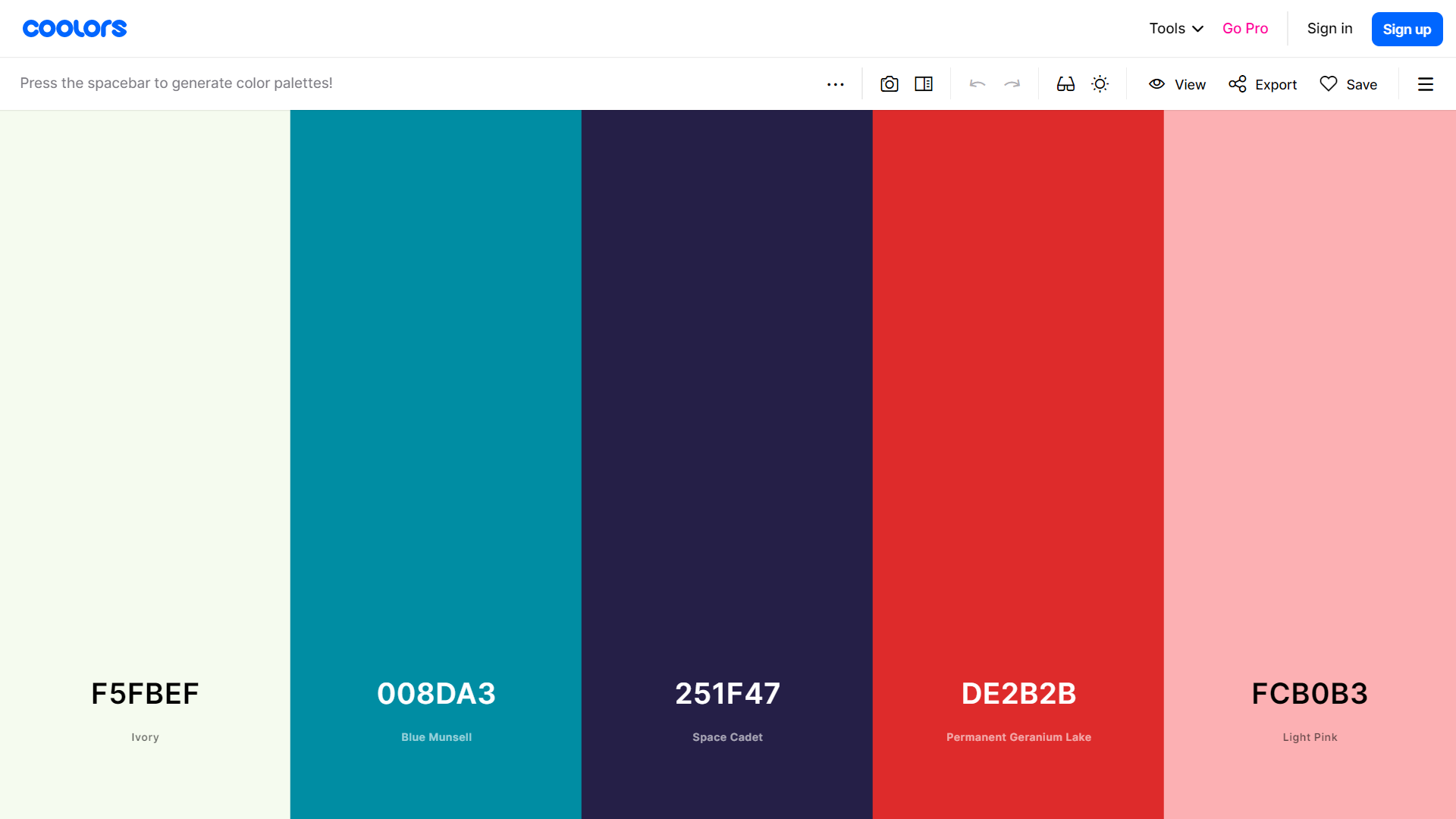
CSS é uma linguagem de estilos, utilizada para personalizar os elementos de um documento HTML, ou outros do tipo. Neste projeto foi utilizado a versão mais recente a CSS 3.

## O Site

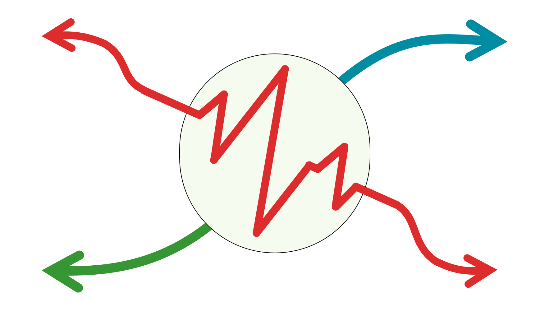
Antes de começar a criar o site antes era necessário ter o logotipo, uma palete de cores e a estrutura geral do site, isto para que o site tivesse uma uniformidade, na página inicial, na página de descrição dos programas e na página de agradecimentos (porém não se aplica aos programas).

### Cores

Para selecionar as cores do site usei o site “coolors”. Neste site cada vez que se pressiona a tecla espaço, o site mostra uma palete com cores que combinam entre si, é possível bloquear as cores, as cores que não estão bloqueadas vão mudando, mas sempre a combinar com a ou as cores bloqueadas. As cores são: duas de tons vermelhos, e duas de tons de azuis e um cinzento quase branco.

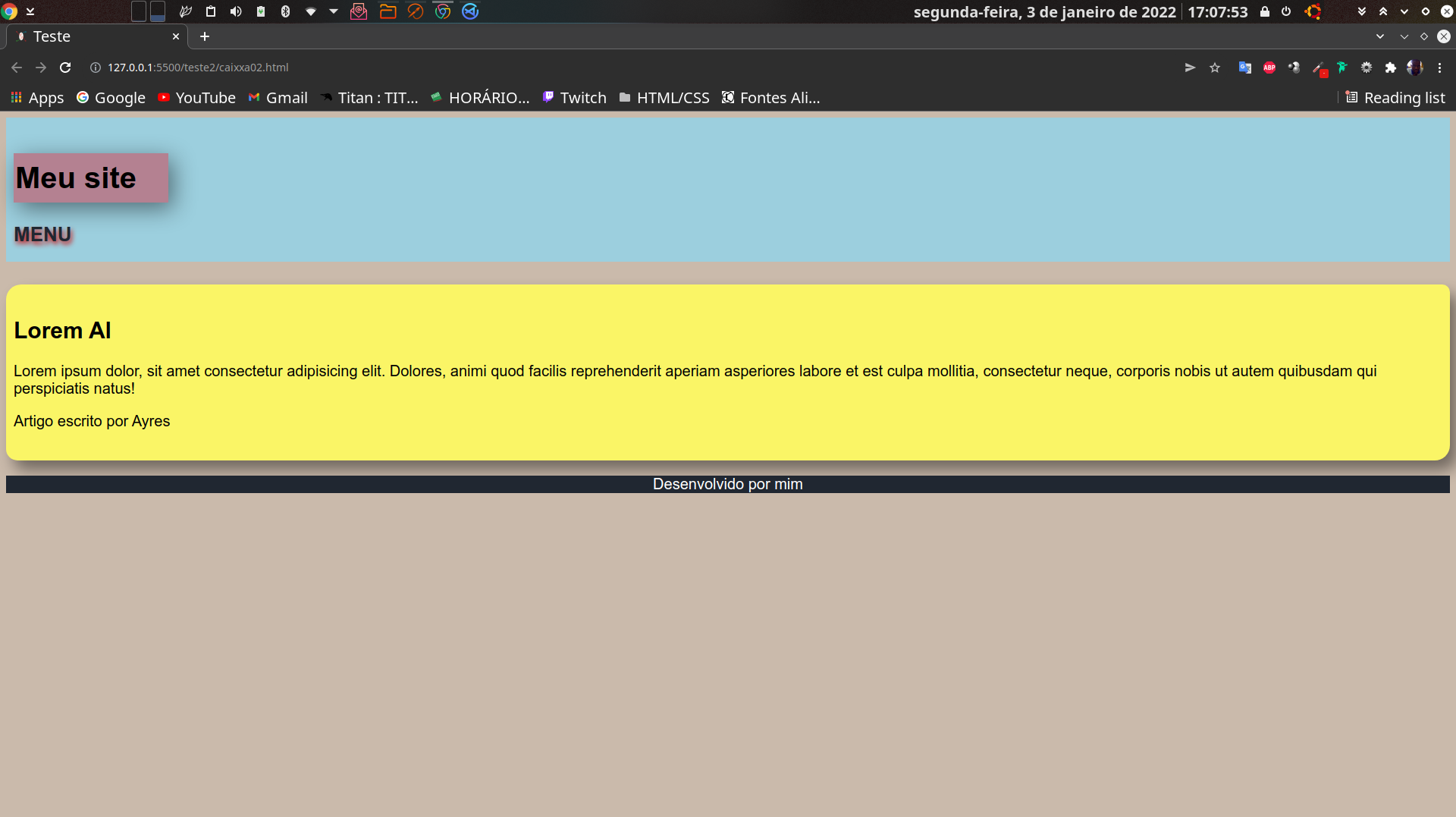


### Logotipo



A proposta inicial do projeto era “site com programas que podem ser úteis no dia-a-dia” o nome era o mesmo “Alleatório”. Seguindo este conceito a primeira logo foi essa:

Este logótipo foi baseado em que as setas formam o símbolo do *random* no centro tem um círculo a representar o planeta e o umas das setas no centro forma uma frequência cardíaca, simbolizando a vida.

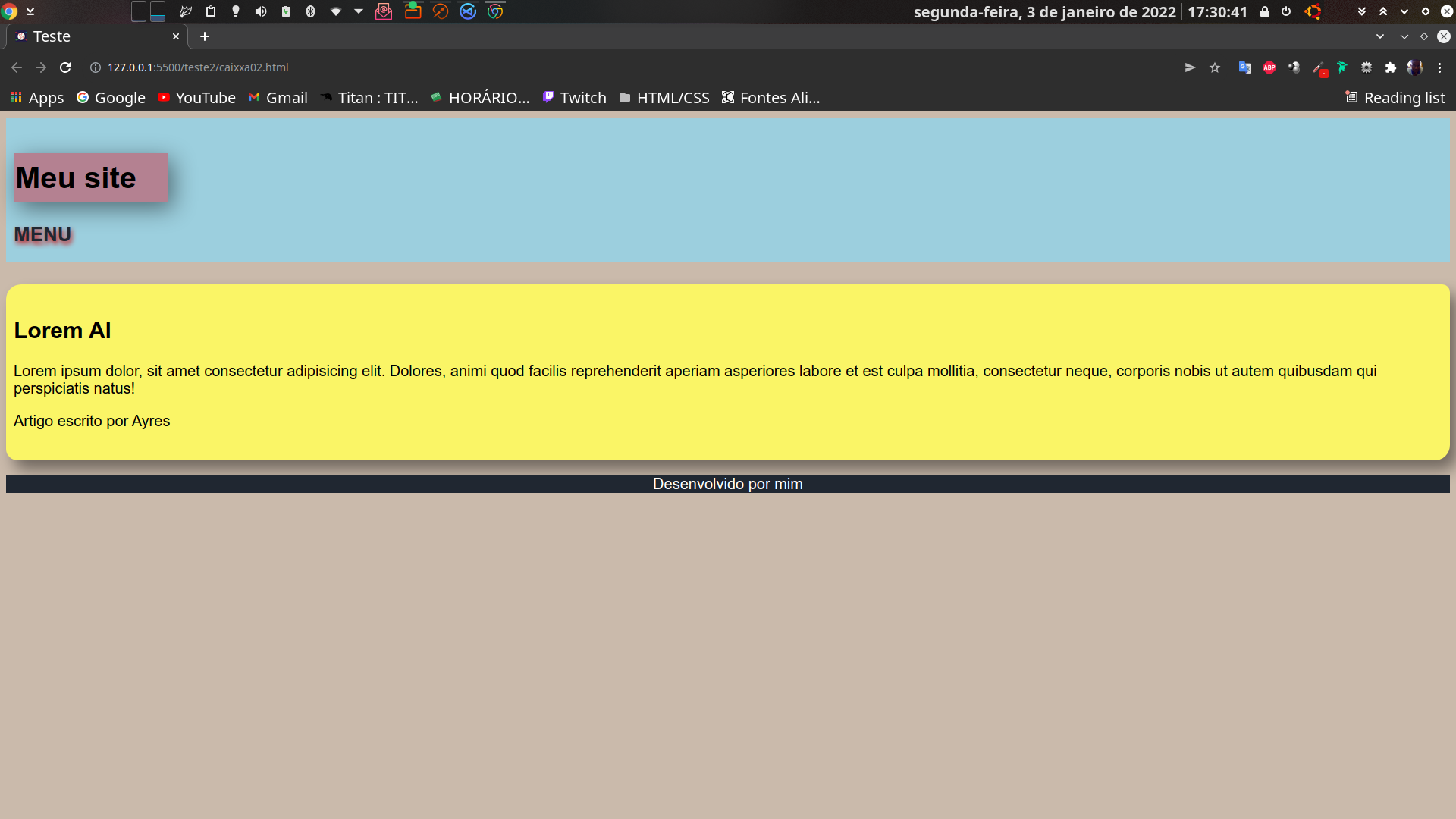


Contudo quando testado como *favicon* do site (ícone mostrado nos separados dos browsers), ficou desformado, isto porque o *favicon* tem o formato quadrangular, e por conta de o logotipo ser retangular foi forçado a forma quadrangular.

Consequentemente o logotipo foi refeito diminuindo o tamanho das setas e adicionando um fundo da cor azul da palete:



Como o formato era quadrangular não disformou como favicon:



Todavia com o avanço do projeto e com a mudança do tema para o atual deste relatório, consequentemente o logotipo anterior perdeu o seu sentido, a solução era fazer um novo. Como o tema atual relaciona-se com conceitos de estética e programação, foi refeito baseando-se em logotipos de spas, joalherias e salões de beleza e logotipos de empresas de programação:

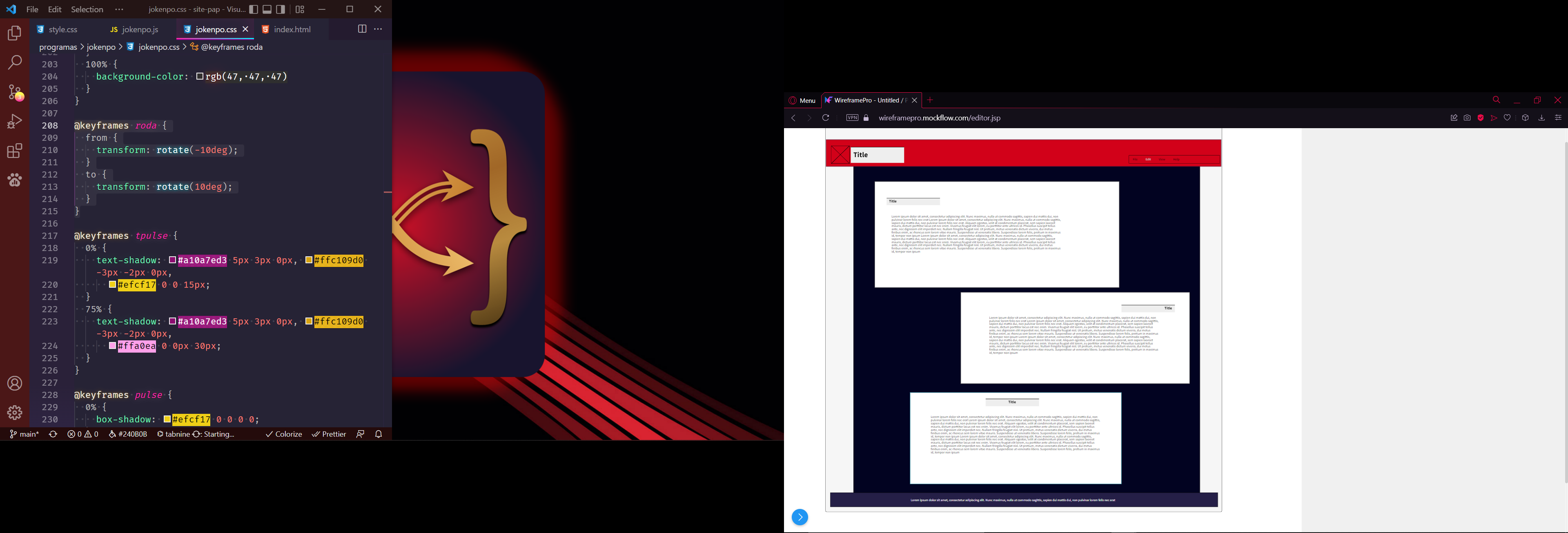


As figuras de cores douradas e sombreadas são inspiradas nos logotipos de estética e as chavetas em logotipos de programação. O símbolo *random* também foi refeito com um visual mais elegante e as cores de fundo são combinações dos tons mais escuros do vermelho e do azul da palete de cores.



### *Layout*

Para definir previamente o *layout* do site foi utilizado o site “MockFlow”, neste site é possível planejar sem a necessidade de programação o *design* de um software, que nesse caso foi uma página web. Com isso foi possível ter uma noção sobre as combinações das cores das paletes e a organização dos blocos de textos e imagens:



### Atuação do HTML

A HTML é a “motherboard” do site, é nela que está contida todo conteúdo do site, por isso é importante estruturar bem, o conteúdo com o uso correto das *tags*. Por exemplo, a usar a *tag* <main> ou a *tag* <div>, não terá diferença visual para o usuário, porém para o navegador e motores de busca como o *google* irão interpretar como o conteúdo principal do site se for <main> e enquanto que <div> como uma secção do site, ou seja, cada *tag* atribui um valor semântico diferente ao conteúdo inserido. Porém como grande parte das tags são para o conteúdo textual, e o site é focado na construção de programas, a maior parte das tags utilizadas foram <div> e <span>, já que são elementos genéricos que não representam nada em concreto e são os ideais para propósitos de estilização, que é o foco do site.

### Atuação do CSS

### Atuação do JS

## Os programas selecionados.

Os programas presentes no site tiveram a sua versão original em Python (exceto a calculadora), que depois foram convertidos para JavaScript. Apesar de ser possível construir o site com Python, converter os programas para JavaScript foi a opção mais viável, isso porque JS como é uma linguagem de programação voltada para páginas web, é mais fácil de tornar os programas interativos com JavaScript do que Python, além de que como são programas básicos, as condições e os operadores utilizados são comuns em ambas as linguagens e mesmo em casos que não são comuns, em JS existe funções ou atributos equivalentes como por exemplo como no caso da listas (*list*) em python são equivalentes aos vetores (*arrays*) em JS, inclusive têm métodos com os mesmos propósitos como por exemplo:

|  |  |  |
| --- | --- | --- |
| Métodos das Listas | Métodos dos Vetores | Propósito |
| lista.append(n) | vetor.push(n); | Adiciona um elemento no final da lista/vetor. |
| lista.index(n) | vetor.indexOf(n) | Procura o elemento especificado dentro da lista/vetor e retornar a sua posição. |
| lista.pop() | vetor.pop() | Elimina o último elemento da lista/vetor |

Porém no caso da variável *randint* (variável importada do módulo *random*, retorna um número inteiro dentro de um intervalo definindo) ou da função *sleep* (função importada do módulo time que causa um *delay* pelo tempo definido dentro da execução do programa), que já são próprios da linguagem Python, JavaScript não tem nativamente algo equivalente, mas é possível criar utilizando funções, métodos e variável nativos da língua:

Para *randint*:

|  |
| --- |
| function randint(min, max) {  return Math.floor(Math.random() \* (max - min)) + min;  }  num = randint(1, 10); |

Para *sleep*:

|  |
| --- |
| const sleep = (ms) => new Promise((resolve) => setTimeout(resolve, ms));  await sleep(1000); |

### Pedra Papel Tesoura

Este programa é a recriação virtual do jogo “pedra papel tesoura”. Neste programa o usuário jogará contra “o seu dispositivo” uma partida de pedra papel tesoura. Ao iniciar o programa após uma pequena animação o usuário terá quinze segundos para escolher umas das 3 opções (“pedra”, “papel” ou “tesoura”) do jogo, o dispositivo do usuário previamente terá escolhido uma das opções e depois aparecerá o resultado: vitoria, derrota ou empate de acordo com as regras originais do jogo. Caso o usuário não escolha dentro dos quinze segundos é escolhido automaticamente uma das opções.

#### **Versão Original (Python)**

|  |
| --- |
| from time import sleep from random import randint print('''Ei, vamos jogar JOKENPO!!! Vou escolher primeiro...''') sleep(1) itens = ('pedra', 'papel', 'tesoura') computador = randint(0, 2) print('''Já escolhi!!! Suas opções: [ 0 ] pedra [ 1 ] papel [ 2 ] tesoura''') joga = int(input('Agora sua vez: ')) if joga > 2:  print('OPÇÃO ERRADA') else:  print('PEDRA')  sleep(1)  print('PAPEL')  sleep(1)  print('OU TESOURA')  print('-=' \* 10)  print('Eu joguei {}'.format(itens[computador]))  print('Você jogou {}'.format(itens[joga]))  print('-=' \* 10)  if joga == computador:  print('EMPATE')  else:  if computador == 0:  if joga == 1:  print('Você ganhou!!')  elif joga == 2:  print('GANHEI')  elif computador == 1:  if joga == 0:  print('GANHEI')  elif joga == 2:  print('Você ganhou!!')  elif computador == 2:  if joga == 0:  print('Você ganhou!!')  elif joga == 1:  print('GANHEI') |

Código:

Terminal:

|  |
| --- |
| Ei, vamos jogar JOKENPO!!!  Vou escolher primeiro...  Já escolhi!!!  Suas opções:  [ 0 ] pedra  [ 1 ] papel  [ 2 ] tesoura  Agora sua vez: 2  PEDRA  PAPEL  OU TESOURA  -=-=-=-=-=-=-=-=-=-=  Eu joguei papel  Você jogou tesoura  -=-=-=-=-=-=-=-=-=-=  Você ganhou!! |

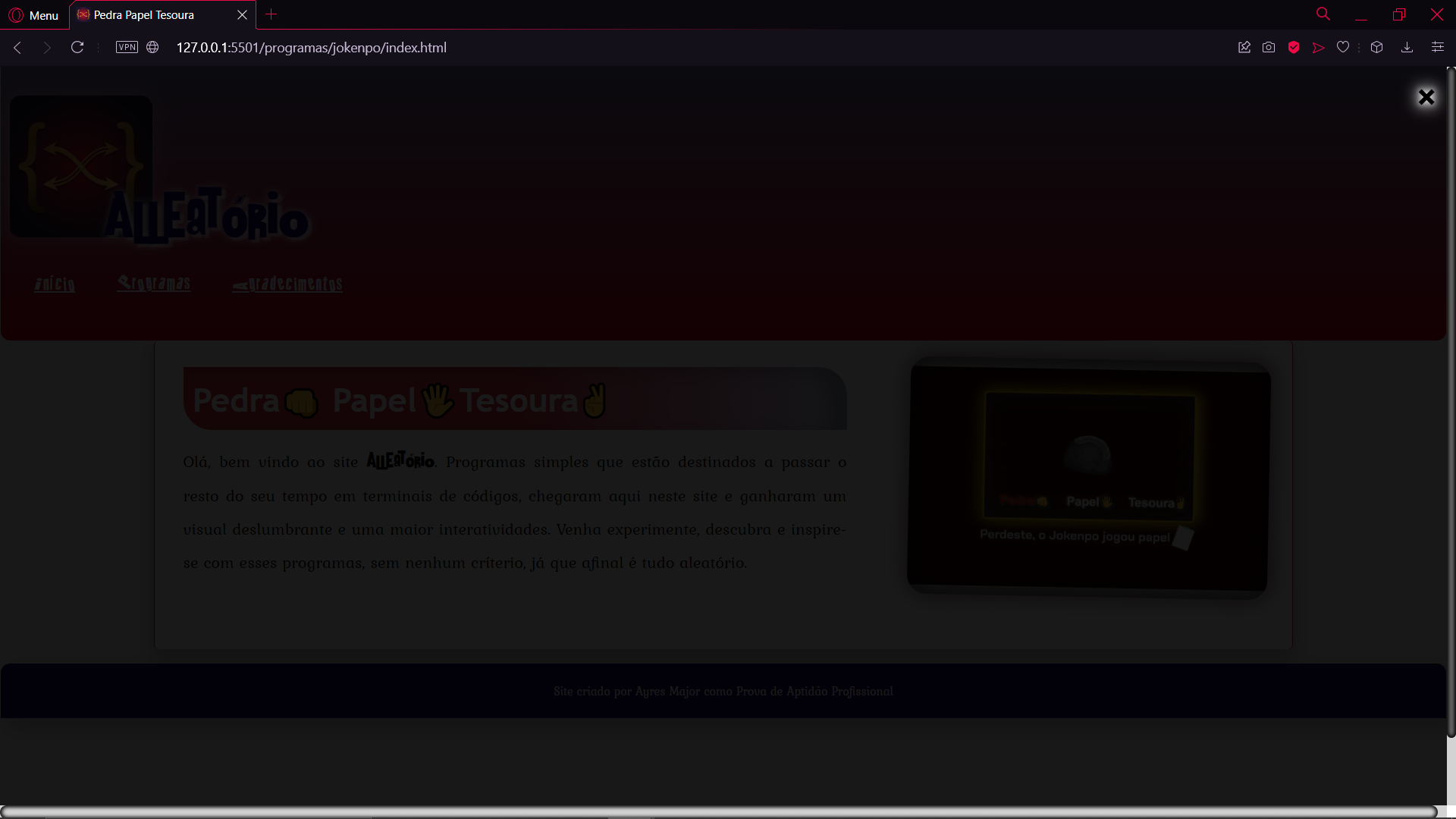
#### **HTML**

Em termos de conteúdo, além da descrição do programa, tem uma <div> que serve de fundo, as imagens respetivas a cada uma das opções, botão para começar o programa e algumas animações do lottie.

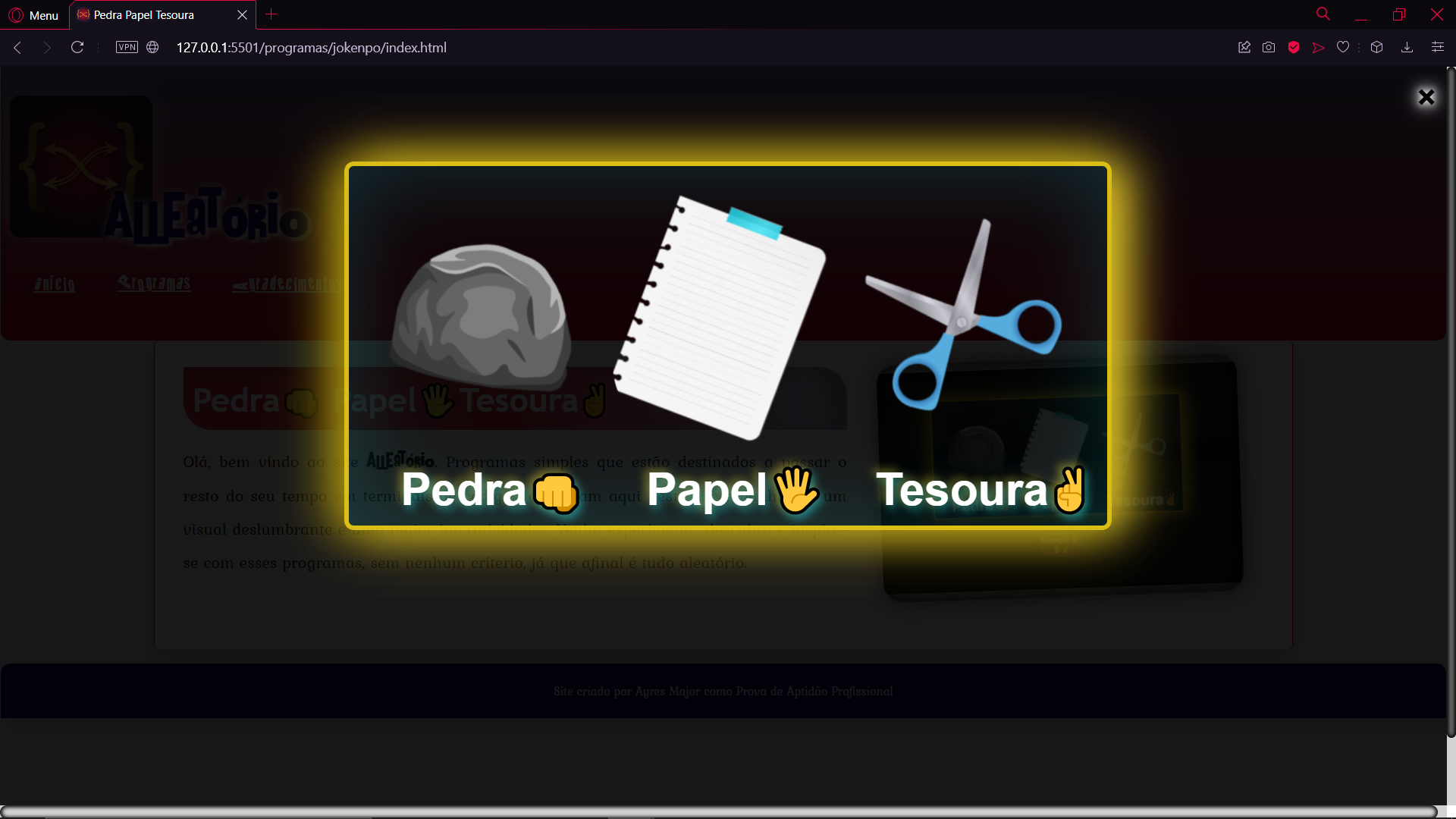
#### **CSS**

Para o *layout* do programa os elementos utilizados estão centralizados com o uso das classes *flex e grid*.

Com o CSS a <div> de fundo fica por cima da página inicial a ocupar o ecrã inteiro, der cor preta com 90% de opacidade, e tem o *aspect-ratio* definido para 16/9, ou seja em qualquer ecrã a altura e largura dessa <div> terá uma proporção de 16/9:



Para o titulo/menu inicial a <div> tem bordas arrendondadas e amarelas e com *box-shadow* amrelo por fora e *box-shadow* azulado com pouca opacidade por dentro, o *background-color* é transparente. Dentro dessa <div> tem imagens correspondentes a cada uma das opções, e o titulo tem um *texto-shadow* azul e amarelo.



O botão “começar” é um <input *type*= “button”> que será depois utilizado no JavaScript para iniciar o programa. O botão é transparente e foi personalizado com bordas arrendodadas, o *background-color* é transparente, as cores apresentadas surgem através do *box-shadow: inset*.



No quesito de animações tem as animações que estão presentes desde o início:

|  |
| --- |
| @keyframes roda {  from {  transform: rotate(-10deg);  }  to {  transform: rotate(10deg);  }  }  div.tituloimg img {  animation: roda 2s alternate infinite;  } |

Bem como animações que somente serão atribuídas através de JavaScript:

|  |
| --- |
| <style>  @keyframes shadowin {  0% {  box-shadow: #ff494900 0px 0px 0px 0px, inset #ff494900 0px 0px 0px 0px;  }  100% {  box-shadow: #ff49496e 1px 1px 17px 15px, inset #ff49498c 1px 1px 78px 18px;  }  }  </style>  <script>  function entrar(clickedid) {  let algo = document.getElementById(clickedid);  algo.style.animation = "aumentartxt 1500ms forwards";  if (clickedid == "pedra" || clickedid == "pedraimg") {  pedraimg.style.animation =  "roda 2s alternate infinite, shadowin 1000ms forwards";  }  }  </script> |

#### JavaScript

## Lista de Material

|  |  |
| --- | --- |
| Portátil ACER Aspire 5 A515-55-59D1 (15.6'' - Intel Core i5-1035G1 - RAM: 8 GB - 512 GB SSD PCIe - Intel UHD Graphics) |  |

## Lista de ferramentas

|  |  |
| --- | --- |
| Visual Studio Code |  |
| GNU Image Manipulation Program | GIMP |
| PowerPoint | Logo Microsoft PowerPoint – Logos PNG |
| GitHub Desktop |  |
| GitBash |  |
| Node.js | Node.js news and podcasts for developers |> Changelog |
| Opera GX |  |
| Google Chrome | Chrome Logo: valor, história, PNG |
| Microsoft Edge |  |
| Firefox | Firefox Logo: valor, história, PNG |

## Tarefas e Atividades

## Estimativa Orçamental

## Orçamento Final

## Resultados Obtidos

Apresentam-se aqui os resultados "em bruto" do trabalho. Não deve haver qualquer interpretação dos resultados (tirar conclusões, dizer se são maus ou bons, atribuir-lhes significados) mas apenas "despejar" (dentro de certos limites) o que se observou.

Tal como se disse antes, os resultados apresentados num relatório devem ser verificáveis e o Relatório deve estar organizado de modo a permitir a sua reprodução.

## Análise dos Resultados

Nesta secção, procede-se à transformação dos resultados "em bruto", apresentados na secção anterior, de forma a que possam ser utilizados: aplicação de fórmulas, extração de médias e desvios padrões, etc. O objetivo final do relatório é, dado um problema, sustentar (isto é, provar que é verdadeira ou correta) a resposta que é apresentada pelo autor. A Análise dos Resultado permite obter informação que seja mais facilmente assimilável pelo leitor. Depois de estruturarmos os dados numa forma que seja conveniente à nossa perceção, vamos começar a fazer a sua análise e, muito especialmente, a sua análise crítica.

O desenvolvimento da fase de análise dos dados depende do enquadramento em que é feito o trabalho de laboratório. Na investigação científica, o trabalho tem geralmente por objetivo demonstrar (servir de suporte de prova a) uma teoria ou hipótese. O tratamento dos dados incidirá, por exemplo, na procura das tendências que relacionam as várias grandezas em análise. “Em que é que o consumo de refrigerantes na população jovem afeta a sua taxa de obesidade?” poderá ser uma questão a que se procura dar resposta.

Outras vezes, o resultado considerado correto já é conhecido: por exemplo, quando queremos verificar a conformidade de determinado equipamento. É o caso do controlo de qualidade, em que se realizam testes laboratoriais em que já se conhece o “bom” resultado e o que se pretende é verificar se as características de determinado equipamento estão ou não conformes a esses resultados considerados “bons”.

Em qualquer caso, o objetivo final é sempre o mesmo: verificar a veracidade ou a falsidade de determinada tese, em que essa tese pode ser uma nova teoria científica, ou pode ser simplesmente a afirmação que um determinado telemóvel aguenta quedas da altura de 1m.

Um dos objetivos da análise crítica é validar os resultados obtidos: são coerentes entre si? São coerentes com o esperado? Aparentam estar isentos de erros significativos? Ou, pelo contrário, são incoerentes? Há resultados que estão claramente em contradição com os restantes? A análise crítica tem a função de ser “advogada e detetive” dos dados apresentados. “Advogada”, quando dá fundamento aos resultados, mostrando que têm todas as razões para serem considerados válidos; “detetive” quando, ao detetar algo que não está bem, não descansa enquanto não identificar a causa da incoerência nos dados. Sempre que os dados, por alguma razão, não estão coerentes entre si, é necessário procurar a causa.

Essa causa pode ser um erro, o que muitas vezes acontece. Mas pode ser também a evidência de uma falha na hipótese inicial; muitas descobertas científicas resultaram da identificação de valores, em experiências, que “não batiam certo” com os restantes.

É essencial que toda a análise dos dados seja feita com honestidade. De nada serve tentar tapar o sol com a peneira (ou, como dizia Eça de Queirós, colocar “sobre a nudez forte da verdade, o manto diáfano da fantasia”). Os resultados obtidos devem ser apresentados sem adulterações e as análises devem ser cruas e objetivas.

# Considerações Finais

A secção de Conclusões encerra a questão que deu origem ao Relatório. Num trabalho de investigação, dar-se-á resposta ao problema que estava a ser investigado: se a hipótese inicial foi verificada ou não, se a resposta é claramente conclusiva ou se os trabalhos fizeram surgir outras questões que necessitam de ser investigadas, … No caso de um trabalho de Engenharia, em que houve o desenvolvimento de um circuito ou equipamento, as Conclusões avaliam se o projeto cumpriu ou não as especificações e os requisitos definidos à partida. Num trabalho escolar, as Conclusões avaliam se os resultados estão de acordo ou não com a teoria que se pretendia demonstrar ou verificar e a contribuição do trabalho para o progresso do aluno.

As conclusões podem também incluir uma síntese da avaliação crítica dos resultados obtidos: se estão de acordo com a teoria, se os erros são elevados, se são credíveis.

# Referências Bibliográficas

Cadorin, J. L. (junho de 2001). *Carga e Descarga de Capacitores.* São José: Escola Técnica Federal de Santa Catarina - Unidade de Ensino de São José.

Felicio, J. C. (2010). Modelo de Relatório técnico-científico baseado na. *NBR 10719: apresentação de relatórios técnico-científicos*, 11.

Fonseca, P. (23 de julho de 2012). Guia para a redação de relatórios. *Universidade de Aveiro - Departamento de Eletrónica, Telecomunicações e Informática*, 13.

Luiz, I. (03 de março de 2011). Como elaborar um Relatório Técnico-Científico. 4.

Lista dos artigos, livros e outra bibliografia consultada e que seja mencionada no texto do trabalho ou relatório. Podem também ser incluídos outros livros e artigos que se debrucem sobre a área do trabalho, devendo, neste caso, constar de uma lista à parte.

Deve haver critério na escolha das referências a apresentar. Se, por um lado, um número muito reduzido de referências é um indicador pouco abonatório da qualidade do trabalho, também é igualmente má a inclusão de referências apenas para fazer volume: referências que em nada contribuem para a compreensão do Relatório e não estão relacionadas com o trabalho desenvolvido. Incluem-se muitas vezes neste caso referências aos “datasheets” dos componentes utilizados, manuais de software ou equipamento, etc., que surgem em muitos relatórios sem qualquer necessidade nem justificação.

# Anexos (Opcional)

Conjunto de documentação diversa, utilizada para fundamentar o texto do relatório. Nos anexos podem estar: esquemas elétricos, documentação de programas, dados, ... Em geral, os anexos contêm o que é informação complementar ou demasiado extensa para constar do texto do relatório.

Os esquemas elétricos de um circuito, desenhos mecânicos do equipamento ou documentação de software são elementos candidatos a serem apresentados como anexos ao Relatório. No entanto, só devem ser anexados se contribuírem de alguma forma para a melhor compreensão do Relatório. Não vale a pena acrescentar Anexos só para fazer volume. A utilidade de anexar as impressões (listagens) do código do software é também bastante discutível.

Outra situação que pode justificar um anexo é a existência de fórmulas ou resultados no texto cuja dedução, embora suportada no Corpo do Relatório, não seja fácil ou evidente. Nessa altura, remete-se essa dedução para os Anexos, sobretudo quando for bastante longa.

Pode ocorrer a necessidade de fazer acompanhar o Relatório de dados originais que sejam bastante extensos. Por exemplo, um estudo sobre uma determinada população pode ser baseado em estatísticas realizadas a partir de um conjunto vasto de indicadores (sociais, económicos, …) recolhidos durante um largo período.

Para alguns leitores, a análise de alguns dados em detalhe poderá ter significado ou importância, pelo que deverão acompanhar o relatório. Mas a sua introdução no corpo do texto pode torná-lo excessivamente grande e difícil de ler. Nessa altura, o corpo do Relatório contém apenas os valores mais relevantes, remetendo-se para o anexo todo o detalhe sobre os dados.