



UNIVERSIDADE  
FERNANDO  
PESSOA

## Relatório: Projeto 2 de Sistemas de informação

Robi Henriquez

Novembro de 2025

# Índice

<b>Introdução.....</b>	<b>3</b>
<b>Apresentação da página.....</b>	<b>4</b>
<b>Desenho mobile first.....</b>	<b>4</b>
<b>Versão web.....</b>	<b>5</b>
<b>Relatório.....</b>	<b>6</b>
<b>1. Armazenamento de dados em variáveis .....</b>	<b>6</b>
<b>2. Estrutura da função principal.....</b>	<b>6</b>
<b>3. Funções secundárias de conversão.....</b>	<b>6</b>
<b>4. Lógica de decisão na função principal.....</b>	<b>6</b>
<b>5. Função auxiliar de identificação da base .....</b>	<b>7</b>
<b>6. Testes durante o desenvolvimento .....</b>	<b>7</b>
<b>7. Integração com o DOM.....</b>	<b>7</b>
<b>Cenários binários .....</b>	<b>8</b>
<b>1. Complemento a Dois .....</b>	<b>8</b>
<b>2. Conversão a hexadecimal .....</b>	<b>8</b>
<b>GIT, GITHUB .....</b>	<b>9</b>
<b>Dificuldades e propostas de melhoras .....</b>	<b>10</b>
<b>Conclusão.....</b>	<b>11</b>
<b>Referências.....</b>	<b>12</b>

# Introdução

A ideia principal deste tipo de ferramentas é resolver operações matemáticas que, quando realizadas manualmente, podem ser bastante complexas e demoradas. Além da sua relevância na tecnologia e na resolução de problemas numéricos, estas ferramentas são essenciais porque os computadores processam informação em linguagem binária, enquanto nós, seres humanos, estamos habituados a trabalhar com o sistema decimal nos nossos cálculos do dia a dia.

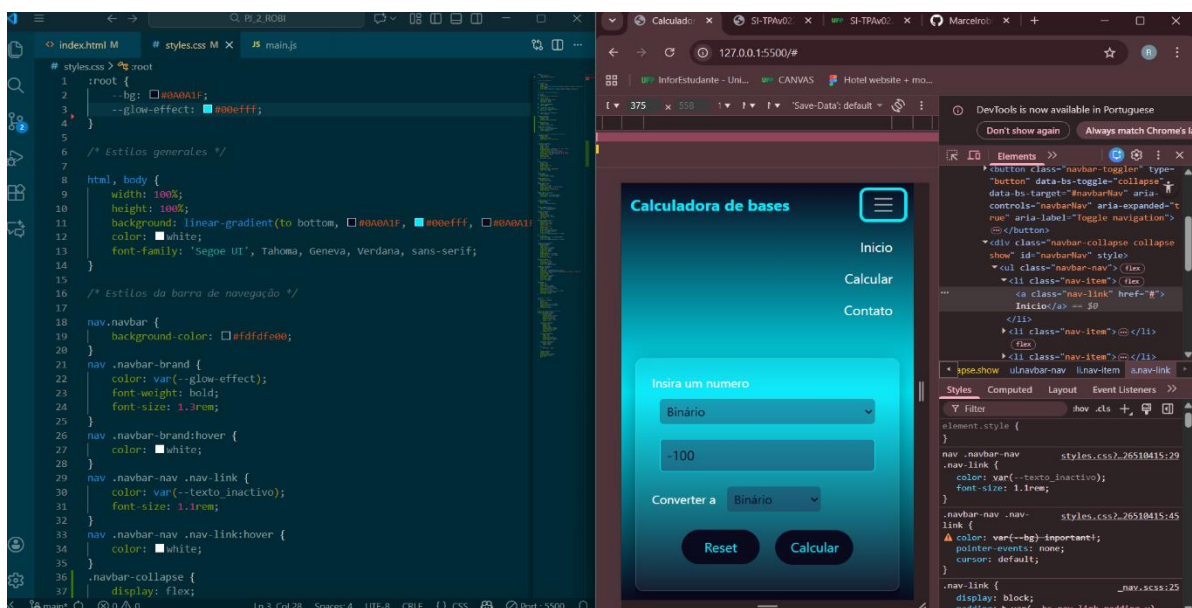
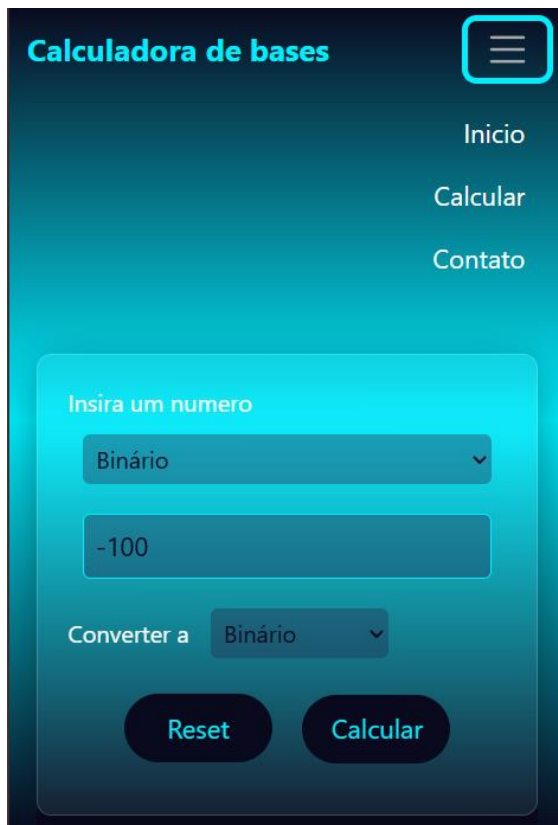
Da mesma forma, as bases octal e hexadecimal desempenham um papel fundamental na programação, pois permitem representar e manipular dados de forma mais eficiente, facilitando a resolução de problemas matemáticos e computacionais avançados.

Durante o desenvolvimento do projeto, percebi que a conversão entre diferentes bases numéricas nem sempre é rápida, intuitiva ou prática quando feita manualmente. Foi precisamente para resolver essa dificuldade que criei a minha calculadora. Ela funciona como um tradutor instantâneo, capaz de converter valores entre várias bases de forma rápida, precisa e simples. Em vez de perder tempo com cálculos extensos — como descobrir o valor decimal do número 1A em hexadecimal — a ferramenta executa essa operação automaticamente, garantindo precisão e eficiência.

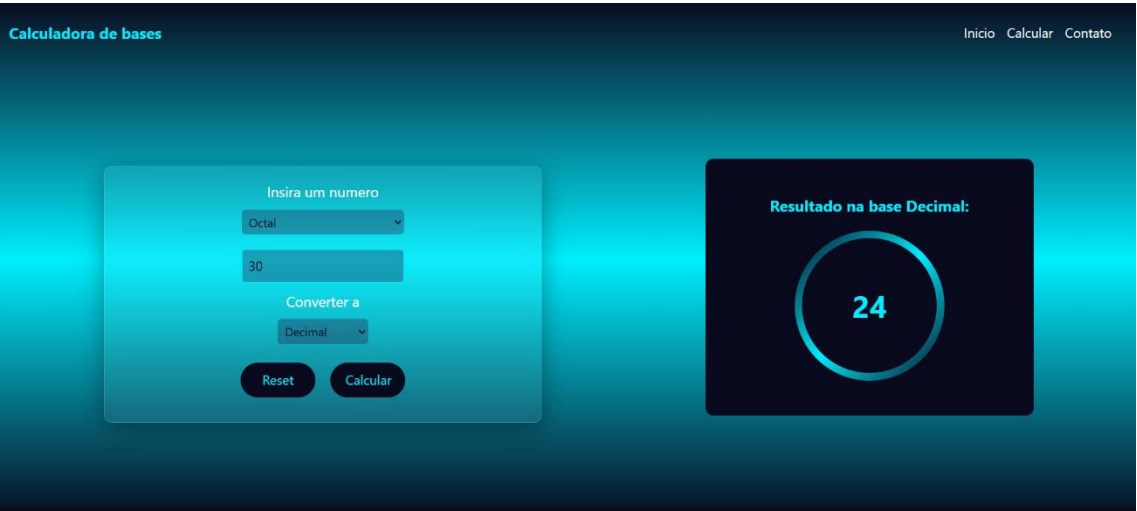
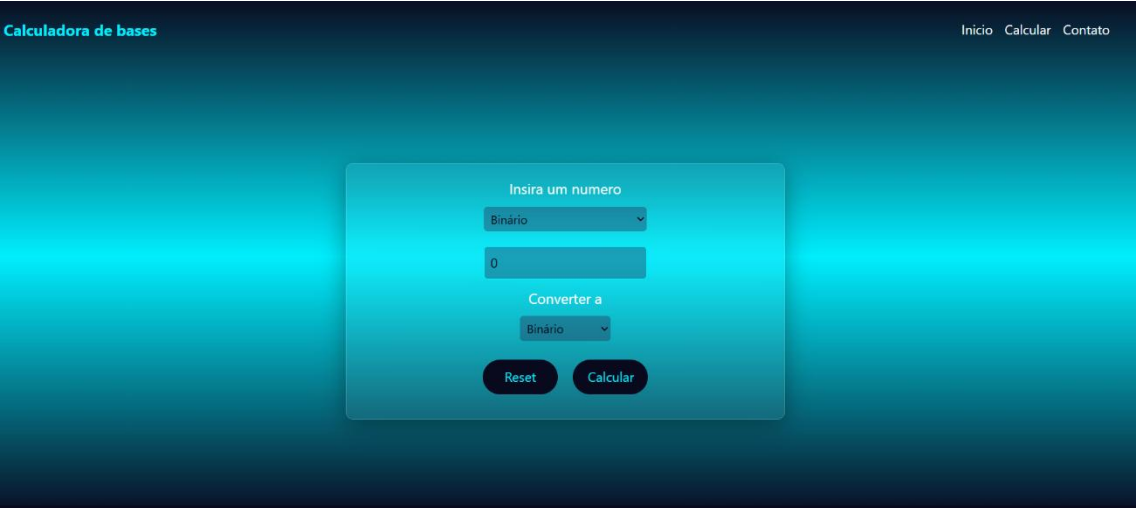
Este relatório apresenta todo o processo de desenvolvimento da calculadora. Nele explico a motivação do projeto, as funcionalidades implementadas, a estrutura lógica utilizada e os algoritmos de conversão aplicados. Por fim, apresento os resultados obtidos e avalio o desempenho da ferramenta. Em suma, trata-se de um registo completo da construção desta pequena, mas poderosa, solução digital.

# Apresentação da página

## Desenho mobile first



# Versão web



# Relatório

A implementação da calculadora de conversão entre bases numéricas foi construída de forma modular, garantindo organização, clareza e facilidade de manutenção do código. Todo o processo segue uma sequência lógica que permite manipular valores em diferentes bases e apresentar o resultado de forma dinâmica no navegador.

## 1. Armazenamento de dados em variáveis

Inicialmente, foram criadas variáveis responsáveis por armazenar os objetos e elementos necessários ao decorrer do programa. Esses valores são manipulados ao longo da execução, permitindo que a aplicação responda às interações do utilizador.

## 2. Estrutura da função principal

A lógica central da aplicação está concentrada numa função principal, responsável por coordenar todas as etapas da conversão. Esta função recebe três parâmetros fundamentais: o número de entrada, a base de origem e a base de destino.

Ela também realiza o chamamento das funções secundárias que executam as operações específicas do processo de conversão.

## 3. Funções secundárias de conversão

Apesar da existência da função nativa `parseInt()`, foi desenvolvida uma implementação própria para converter números de qualquer base para decimal. Esta função analisa a cadeia de caracteres fornecida e calcula manualmente o seu equivalente em base 10.

Além disso, foi criada outra função dedicada à conversão de números negativos para a sua representação em Complemento a Dois. Ambas as funções são utilizadas pela função principal sempre que a base de entrada assim o exigir.

## 4. Lógica de decisão na função principal

Dentro da função principal foram implementados dois blocos principais de condições:

### Primeiro bloco de condições

Responsável por:

- avaliar se a base de entrada é diferente da base decimal,
- verificar se o número se encontra em formato negativo,
- e executar um conjunto alternativo de instruções quando nenhuma dessas situações ocorre.

O objetivo deste bloco é transformar a cadeia de caracteres dada pelo utilizador num número decimal correto, positivo ou negativo, conforme o caso.

## **Segundo bloco de condições**

Após obter o valor em decimal, este segundo conjunto de verificações determina:

- se o número é exatamente zero,
- se se trata de um valor negativo,
- ou se é um número positivo.

Consoante a condição satisfeita, a função procede à conversão do valor decimal para a base de destino, produzindo assim o resultado final pretendido.

## **5. Função auxiliar de identificação da base**

Também foi criada uma função adicional que identifica o nome correspondente à base de saída. Esta função é utilizada posteriormente para exibir informações organizadas no DOM, tornando o resultado mais claro para o utilizador.

## **6. Testes durante o desenvolvimento**

Durante todo o processo de implementação, foram utilizados diversos `console.log()` para monitorizar os resultados intermédios e validar o comportamento das operações. Isso permitiu testar cálculos, verificar entradas e garantir que as funções estavam a devolver os valores esperados.

## **7. Integração com o DOM**

Por fim, foram implementadas as interações com o DOM para tornar a aplicação totalmente funcional para o utilizador. Duas funções principais foram criadas:

### **Função do botão “Calcular”**

Extraí os valores inseridos nos campos de entrada e chama a função principal. Em seguida, exibe o resultado numa caixa visual que surge dinamicamente na interface.

### **Função do botão “Reset”**

Responsável por limpar os valores dos campos, restaurar parâmetros iniciais e ocultar a caixa de resultados, permitindo iniciar um novo cálculo.

# Cenários binários

Calculadora de bases Início Calcular Contato

Insira um numero

Binário

00100101

Converter a

Decimal

Reset Calcular

Resultado na base Decimal:

37

## 1. Complemento a Dois

Calculadora de bases Início Calcular Contato

Insira um numero

Binário

-00100101

Converter a

Binário

Reset Calcular

Resultado na base Binária:

1101101  
1

## 2. Conversão a hexadecimal

Calculadora de bases Início Calcular Contato

Insira um numero

Binário

00100101

Converter a

Hexadecimal

Reset Calcular

Resultado na base Hexadecimal:

25



# GIT, GITHUB

The screenshot shows a GitHub repository named 'ProjectoS12' by user 'Marcelrobl'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows three commits, all by 'Marcelrobl', with the latest commit 'a23532f' made 2 minutes ago. The repository contains three files: 'index.html', 'main.js', and 'styles.css', all committed 2 minutes ago. The 'README' file is currently empty, with a prompt to 'Add a README file'. A terminal window is overlaid on the right side of the page, showing the following commands and output:

```
MINGW64/C:/Users/Utilizador/OneDrive/Universidade/SI/P3_2_R0BI
Utilizador@DESKTOP-L1K8K53 MINGW64 ~/OneDrive/Universidade/SI/P3_2_R0BI
$ git init
Initialized empty Git repository in C:/Users/Utilizador/OneDrive/Universidade/SI/P3_2_R0BI/.git/
Utilizador@DESKTOP-L1K8K53 MINGW64 ~/OneDrive/Universidade/SI/P3_2_R0BI (main)
$ git add .
Utilizador@DESKTOP-L1K8K53 MINGW64 ~/OneDrive/Universidade/SI/P3_2_R0BI (main)
$ git commit -m "Primeito commit do projecto"
[main (root-commit) a23532f] Primeito commit do projecto
3 files changed, 480 insertions(+),
create mode 100644 index.html
create mode 100644 main.js
create mode 100644 styles.css
Utilizador@DESKTOP-L1K8K53 MINGW64 ~/OneDrive/Universidade/SI/P3_2_R0BI (main)
$ git remote add origin https://github.com/Marcelrobl/ProjectoS12.git
Utilizador@DESKTOP-L1K8K53 MINGW64 ~/OneDrive/Universidade/SI/P3_2_R0BI (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 4.26 KiB | 1.42 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```

## **Dificuldades e propostas de melhoras**

Uma das funções mais desafiantes de implementar foi a responsável pelo cálculo do complemento a dois e pelo tratamento de números negativos. Esta parte exigiu uma análise lógica mais profunda e cuidadosa, tornando-se uma das etapas mais complexas do desenvolvimento.

Durante todo o processo, dediquei muito mais tempo à construção da lógica da aplicação do que ao seu design visual. Como proposta de melhoria para versões futuras, gostaria de aprimorar a interface gráfica e tornar a aplicação mais interativa e intuitiva para o utilizador.

Uma das ideias iniciais que pretendo retomar é a inclusão de animações para a apresentação da janela de resultados, tornando a experiência visual mais dinâmica. Além disso, também considero importante otimizar o desempenho das funções internas, tornando-as mais eficientes e reduzindo a complexidade computacional sempre que possível.

## Conclusão

O desenvolvimento desta calculadora de conversão entre bases numéricas permitiu consolidar conhecimentos fundamentais de lógica de programação, estruturas condicionais e manipulação de dados em JavaScript. A implementação das funções de conversão — especialmente a que trata números negativos e o cálculo do complemento a dois — representou um dos maiores desafios do projeto, exigindo análise detalhada e construção de algoritmos próprios para garantir resultados precisos.

Ao longo do processo, ficou evidente a importância de organizar o código de forma modular e de testar cada etapa com cuidado, garantindo a correta interação entre as funções e a fiabilidade do programa final. Embora o foco principal tenha sido a lógica e o funcionamento interno da aplicação, o projeto também evidenciou oportunidades de melhoria na interface e na experiência do utilizador, nomeadamente através de aprimoramentos visuais, melhores interações com o DOM e possíveis otimizações de desempenho.

No conjunto, o projeto alcançou o objetivo proposto: criar uma ferramenta funcional, capaz de converter números entre várias bases de forma clara, rápida e precisa. Além disso, estabeleceu uma base sólida para futuras extensões, melhorias e aperfeiçoamentos, tanto na lógica quanto no design da aplicação.

## Referências

1. freeCodeCamp. (2023). *The JavaScript DOM Manipulation Handbook*. freeCodeCamp. <https://www.freecodecamp.org/news/the-javascript-dom-manipulation-handbook/>
2. DevMedia. (s.f.). *Trabalhando com DOM em JavaScript*. DevMedia. <https://www.devmedia.com.br/trabalhando-com-dom-em-javascript/29039>
3. freeCodeCamp. (2023). *JavaScript DOM Manipulation – Full Course for Beginners* [Video]. YouTube. <https://www.youtube.com/watch?v=5fb2aPlgoys>