

PROJETO PARA INTRODUÇÃO A CIÊNCIA DA COMPUTAÇÃO:

ICC Helper

Alexandre Eduardo Coelho Oliveira

1. Introdução

1.1. Objetivo

Este projeto tem como objetivo possuir uma interface gráfica em que ao entrar com os valores o sistema irá realizar **conversões de números entre bases** e o **cálculo com aritmética binária**, apresentando os resultados com os mesmo métodos ensinados em sala caso seja possível com os valores entrados.

1.2. Público-alvo

O público-alvo são os estudantes da disciplina **Introdução a Ciência da Computação** (ICC) que desejam auxílio no assunto passado em sala.

2. Desenvolvimento

A seguir estarão as principais informações no desenvolvimento do projeto. Todas elas estão sujeitas a mudanças em possíveis futuras atualizações no projeto.

2.1. Tecnologias e Ferramentas Utilizadas

Para a realização deste projeto que consiste em uma página web, foram utilizados:

- HyperText Markup Language (HTML). Consiste em uma linguagem de que marca o tipo de conteúdo apresentado.
- Cascading Style Sheets (CSS). Consiste em uma linguagem que estiliza o conteúdo marcado pela HTML.
- JavaScript (JS). Consiste em uma linguagem de programação interpretada, com o seu principal uso sendo, para páginas Web.
- React. Consiste em um Biblioteca JavaScript com o objetivo de facilitar o desenvolvimento de páginas dinâmicas.

- CSS Modules. Consiste em isolar cada arquivo de CSS em escopos fechados, evitando conflitos de estilos.
- TypeScript (TS). Consiste em uma extensão do JavaScript, sendo sua principal adição os tipos.
- Vite. Consiste em uma ferramenta de construção de projetos de frontend de forma leve e rápida.
- Git. Consiste em uma ferramenta de versionamento de código.
- GitHub. Consiste em uma ferramenta para armazenamento repositórios.
- Vercel. Consiste em uma plataforma com suporte para por sites no ar.

2.2. Cores

Como no primeiro momento deste projeto cor não foi considerado um detalhe de grandes preocupações, foi escolhido como as **duas principais cores o branco e o preto**.

Essas cores criam facilmente um contraste uma com a outra, o que facilita a leitura, como também por serem neutras vão influenciar menos na recepção do conteúdo.

Uma observação que as cores branco e preto escolhidas **não foram em sua versão pura**, já que elas possuem alto contraste e podem ser agressivas aos olhos.

Quatro cores adicionais são um **vermelho**, um laranja, um **verde** e um **azul**. O vermelho e verde é para **representar campos com respostas válidas ou inválidas**. A laranja representa um aviso de algo que não pode fazer. Já o azul o link em meu nome.

A lista de cores em hexadecimal são:

- Cor de Background: **#F9F9F9** - Branco
- Cor Primária: **#121212** - Preto
- Cor de Erro: **#FF0000EE** – Vermelho
- Cor de Aviso: **#EE9E09** – Laranja
- Cor de Sucesso: **#00FF40C0** - Verde
- Cor do meu Link: **#1A7EFF** – Azul

2.3. Layout

O layout da página foi desenvolvido para ser simples e facilmente responsivo (capacidade de se adaptar ao tamanho das telas), por dois motivos: celeridade e ausência de informações desnecessárias.

Desta forma, o layout contém de fato somente três blocos:

- **Botões de Função:** O primeiro bloco são os dois botões que definem qual das funções, Sistemas de Numeração ou Aritmética Binária, será usada. O botão selecionado ficará com seu fundo preenchido e desativado.
- **Formulário da Função:** O segundo bloco é o formulário onde o usuário entrará com os dados do qual ele deseja resposta. O formulário que será exibido depende do botão que foi selecionado.
- **Seção de Respostas:** O terceiro bloco por padrão não é visível, ele somente aparecerá caso ao clicar no botão de submissão do formulário, “Converter” em Sistemas de Numeração e “Calcular” em Aritmética Binária, e os dados entrados sejam válidos.

2.4. Versões

Houve ao todo 13 versões, começando 0.1.0 ao 0.11.0 e as duas finais sendo a 1.0.0-alpha-1 e 1.0.0-alpha-2. Essas versões foram feitas em 15 dias.

Cada versão tinha como objetivo adicionar funcionalidades necessárias para o projeto. No momento, está na versão 1.0.0-beta.

3. Estrutura e Funcionalidades

3.1. Aprofundando na Estrutura

3.1.1. Sistemas de Numeração

É a primeira estrutura apresentada. São suas entradas de dados:

- **Base:** é uma entrada numérica que representa a base (quantidade de símbolos) do número. A base deve ser de 2 até 36. Menores que 2 não é possível, por ser o mínimo, já o limite ser 36 é por conveniência, já que esta base engloba os números de 0 a 9 e todas as letras do

alfabeto em maiúsculo. Seria uma adição problemática e com utilidades baixas mais do que isso. **O padrão é 2.**

- **Número:** é uma entrada de texto. Ela aceita qualquer coisa, mas no momento da submissão do formulário será verificado se a entrada é válida levando em consideração a base. **O padrão é vazio.** Algumas observações são:
 - Todo o dígito é convertido automaticamente para maiúsculo.
 - Ao digitar ponto (.) será convertido automaticamente para vírgula.
 - Ao ter uma vírgula como primeiro dígito, será adicionado automaticamente um 0 à esquerda.
 - Uma vírgula sem nada após ela, no momento da submissão será adicionado um 0 à direita.
- **É complemento:** é uma caixa de verificação, ela representa se o número é negativo ou não. Com ela marcada não há necessidade de digitar “-” (e nem pode) na entrada de número, porém caso seja binário é necessário digitar ele no complemento. Caso não esteja marcada o número é tratado como positivo. **Padrão é não está marcado.**
- **Máximo de Casas Decimais:** é uma entrada numérica que só aparece quando é digitado uma vírgula em qualquer base que não seja 2, 4, 8 e 16. Seu objetivo é para impedir que no inverso da TFN as casas decimais cresçam infinitamente. O mínimo de casas decimais é 1. **O padrão é 4.**
- **Métodos para Exibir:** são caixas de verificação que definem quais são os métodos que serão exibidos na seção do resultado, haverá caixas de verificação que estarão desativadas caso ele não seja um dos métodos para conversão de bases. É possível converter com nenhum método de conversão usado marcado. **Padrão é todos marcados.**

3.1.2. Aritmética Binária

É uma estrutura que aparece após clicar o botão “Aritmética Binária”, ela atualiza o conteúdo do formulário dinamicamente apresentando as entradas de dados para esta função:

- **Tamanho da Arquitetura:** é uma entrada numérica que vai de 1 ao infinito. Ela representa a quantidade de bits totais em número binário.

Padrão 1.

- **Tem bit de sinal:** é um caixa de verificação que tem a função de verificar se o bit de maior magnitude do número deve representar do sinal. Caso esteja desmarcada, será suposto que todos os números são positivos independentemente do bit de maior magnitude. Caso seja marcada o valor mínimo para o tamanho da arquitetura se torna 2.

Padrão desmarcado.

- **Seletor da Divisão de Bits:** uma caixa de seleção que permite selecionar a divisão dos bits da arquitetura para a parte inteira e para a parte fracionaria. Ao todo há 7 opções, indo de 0% a 80% para a parte fracionaria, o resto é para a parte inteira, e a última é manual, podendo decidir por si só a divisão. **Por padrão é inteiro.** Essa divisão segue as seguintes regras:

- Se não houver bit de sinal, deve haver ao menos 1 bit para a parte inteira. Caso haja, então ao menos 2 bits para a parte inteira.
- O bit inteiro em geral possui prioridade de aumento, então ao aumentar o tamanho da arquitetura, provavelmente, a parte inteira aumente primeiro que a fracionária.
- A soma das partes deve ser igual ao tamanho da arquitetura.

- **Seletor da Operação:** uma caixa de seleção que permite selecionar qual operação aritmética será feita. São 4 opções: Adição, Subtração, Multiplicação e Divisão. Com exceção da adição, todas as outras por padrão marcam que tem bit de sinal e desativam a opção de mudar isso. **Por padrão Adição.**

- **Tipo da Entrada do Número:** uma entrada com duas opções e só há como escolher uma: Binário ou Decimal. A escolha influencia em que tipo de dado será aceito pela entrada de número. **Por padrão binário.**

- **Entrada de Número:** entrada de texto. Há duas entradas dessas, a primeira representa o operando do lado esquerdo e a segunda o operando do lado direito. Pode ser digitado qualquer tipo de texto,

porém após a submissão será verificado se ela obedece a base. A base da entrada será 2, caso seja marcado o tipo Binário, ou será 10, caso seja marcado o tipo Decimal. Caso seja binário, os bits digitados respeitam a arquitetura, dividindo de forma correta para parte inteira e fracionária. **Padrão é vazio.** Algumas observações são:

- Todo o dígito é convertido automaticamente para maiúsculo.
 - Ao digitar ponto (.) será convertido automaticamente para vírgula.
 - Ao ter uma vírgula como primeiro dígito, será adicionado automaticamente um 0 à esquerda.
 - Uma vírgula sem nada após ela, no momento da submissão será adicionado um 0 à direita.
- **É complemento:** para cada entrada de número há uma caixa de verificação para saber se é negativo ou não. Caso qualquer uma seja marcada, automaticamente será dito que há bit de sinal e a opção de mudar isso será desativada. Padrão é desmarcado. Além disso:
 - Se houver bit de sinal e ela não estiver marcada, a entrada em binário do número não pode ter o bit de maior magnitude 1.
 - Se estiver marcada, a entrada em binário do número não pode ter o bit de maior magnitude 0.
 - Para decimal, impede a necessidade de digitar "-".
- **Modificador (long / double):** uma caixa de verificação que caso seja selecionada ela tratará que o número pode entrar com o dobro da arquitetura, tanto para parte inteira quanto para a fracionária, como também deve resultar com esse tamanho dobrado. Assim, a entrada de número quando para binários vai aceitar o dobro. Somente é possível usar este modificador para operações de Soma ou Subtração. Padrão desmarcado.
- **Representar em decimal:** uma caixa de verificação que ao ser marcado o resultado final será pego para a TFN, demonstrando o resultado para decimal. Padrão desmarcado.

3.2. Funcionalidades

3.2.1. Sistemas de Numeração

Após a submissão do formulário de Sistemas de Numeração será renderizado a seção com o resultado, caso as entradas do formulário sejam válidas.

A primeira parte a ser exibida será as dos detalhes. Nele há:

- **Base**, com o número da base passada na entrada.
- **Número de Entrada**, com, bem, o número passado como entrada.
- **É negativo**, com Sim ou Não, dependendo se a caixa de verificação “É complemento” foi marcada ou não.
- **Métodos Necessários**, com todos os métodos necessários, marcados como Métodos para Exibir, para a conversão para todas as bases em ordem. Se não houver nenhum necessário, a resposta será “Nenhum Método Necessário Selecionado”.

A segunda parte exibida são os métodos necessários e marcados para exibir em ordem. Cada método inicia com um título indicando qual é o método e para qual base está convertendo. Após isso faz o processo necessário e finaliza exibindo o número em sua base original e como ele é na nova base.

As ordens dos métodos são definidas pela base alvo:

1. **A primeira base buscada é a 2**, que em caso de já não ser será com a **Desagregação** para bases 4, 8 e 16 ou **Inverso da TFN** para a base 10. A única exceção é quando a base é diferente das ultimas citadas, sendo então o primeiro método a TFN para encontrar a base 10 e então após isso o Inverso da TFN para encontrar a base 2.
2. **Já a segunda é a 10**, que em caso de já não ser será com **exclusivamente com o TFN do número binário**. Relembrando da exceção em que ela é o primeiro caso seja uma base diferente de 2, 4, 8, 10 e 16, que neste caso será TFN com a base qualquer.
3. **A terceira base buscada é a 8**, que em caso de já não ser o único método é o da **Agregação de 3 em 3** com o binário.
4. **A quarta e última base buscada é a 16**, que em caso de já não ser, como a na base a 8, o único método é o da **Agregação de 4 em 4** com o binário.

A **única exceção à regra é o complemento**, que sempre que for marcado será o primeiro método, mostrando a operação de complemento do número binário.

Com a citação do complemento, caso o número seja um, todos os métodos usam a versão positiva e somente no resultado final ele são apresentados na sua versão de complemento.

Uma **curiosidade de código do complemento é este método não está inerente aos Sistemas de Numeração e sim de Aritmética Binária**, tudo que ela faz é pegar emprestado este método dela.

3.2.2. Aritmética Binária

Após a submissão do formulário de Aritmética Binária será renderizado a seção com o resultado, caso as entradas do formulário sejam válidas.

A primeira parte são as dos detalhes, ela é dividida em duas subpartes. A primeira subparte informa a arquitetura, com o total de bits, os bits reservados para a parte inteira e os bits reservados para a parte fracionária. A segunda subparte possui qual função é passada para a ULA, sendo uma das 4 operações aritméticas e o diagnostico que será retornado por ela, que varia entre Overflow, Underflow e NaN.

A segunda parte é a do cálculo. Em geral ela possui um bloco para os registradores, os blocos para cada operação, o bloco de resultado e caso a entrada seja decimal os blocos para pegar o numero na base 2.

Vamos detalhar isso:

- **Blocos do Inverso da TFN.** Estes blocos somente aparecem quando a o tipo de entrada numérica é marcada como decimal. Serão os dois primeiros blocos, exibindo a conversão do número na base 10 para a base 2. É importante ressaltar que o número respeita o tamanho da arquitetura (com correção de magnitude).
- **Bloco de Registradores.** Este bloco exibe os registradores em ordem a partir de 1. Os valores associados são os números binários respeitando arquitetura (com correção de magnitude), com exceção dos registradores quem recebem o resultado da operação. Em geral só há 3 registradores, dois para cada entrada e um para o resultado da operação, porém caso tenha o modificador será 6 registradores:

- Os dois primeiros representam a primeira entrada, o primeiro representa a primeira metade do número e o segundo a segunda metade.
 - Os próximos dois representam a segunda entrada, o terceiro a primeira parte e o quarto a segunda parte.
 - O quinto é a operação do Registrador 2 com o 4 (segunda parte com segunda parte).
 - O sexto é a operação do Registrador 1 e com o 3 (primeira parte com primeira parte). Lembrando que o ultimo carry da operação anterior interfere com esta operação.
 - Não sendo um registrador, porém uma representação visual que o resultado final é a concatenação (“+”) do registrador R6 com R5.
- **Bloco de Operações.** Estes blocos contém a operação com um layout similar a como seria fazer a operação da aritmética binária eletrônica no papel. Ela mostra os registradores que estão realizando a operação e cada linha da operação representa ou está associado. As mais diferentes são a divisão e complemento. A divisão não indica o cada parte representa e o complemento não sinalizar registrador, apenas referencia qual número está sofrendo complemento.
 - **Bloco do Resultado.** Este bloco sempre vai mostrar o resultado final na versão binária. Caso seja marcado para representar em decimal haverá uma TFN que recebe o binário positivo e mostra visualmente a TFN e o resultado final em decimal, que pode ser negativo caso seja necessário realizar um complemento logo antes da TFN.

Curiosidades técnicas sobre este sistema é que foi necessário pegar emprestado muitos métodos do sistema de numeração, como os métodos de conversão e o tratamento com os números (com principal exceção de tratamento a correção de magnitude).

4. Desafios Técnicos:

Ao decorrer do desenvolvimento, houve diversos desafios para tornar o que esta página é agora. Dentre estes desafios, os principais e mais marcantes foram:

1. **Receber os números.** Mesmo os chamando de números, para o computador eles são um conjunto de caracteres. Essa escolha foi considerada a mais ideal, já que a entrada numérica considera decimal, sendo uma problemática principalmente para bases maiores que 10, como 16, com as letras ao invés de números em si. Por fim, textos tem uma vantagem de ignorar 0s a esquerda dos inteiros e os 0s a direita dos fracionários, isso permite a realização de operações para todos os estes números, como um computador faria.
2. **Layout dos Resultados.** O principal objetivo é mostrar o caminho para chegar ao resultado e não somente apresentar ele. Para isso era necessário um cuidado de não só realizar o algoritmo da forma correta, como armazenar os dados exigidos para uma exibição correta e repassá-los. E com os dados em mãos, ainda era preciso decidir como seria o estilo para ter uma representação visual mais próxima e de forma intuitiva.
3. **Arquitetura.** A divisão da arquitetura da parte inteira e fracionária foi uma das causas de uma complexidade considerável de todo o algoritmo. Tudo começa com a busca de uma forma de o formulário não quebrar com as mudanças automáticas, manuais, de 1 em 1 ou de muitos bits do tamanho da arquitetura. Após isso, em vários momentos que foram necessários capturar as partes inteiras e fracionárias corretamente para aplicar situações específicas, como correção de magnitude.
4. **Multiplicação e Divisão.** Foram às operações mais difíceis de serem feitas. Isso ocorre porque elas nascem em cima do mesmo sistema das últimas operações, porém isso causa graves problemas para a realização dessas operações. Por causa disso, essas operações são uma das principais de causas da adição de várias exceções no código.
5. **Modificador.** O último desafio foi quando adicionado para permitir valores dobrados, isso também causou algumas exceções chatas no código para que fosse possível a representação visual de mais registradores.

5. Considerações Finais

Não há dúvidas sobre ser um projeto desafiador, a aplicação dos conceitos praticados no papel em código que tenta representar visualmente o mais próximo possível o que seria feito em papel.

Muitas ideias subtendidas, facilmente deduzidas ou sequer questionadas precisam ser escritas claramente para que o interpretador seja capaz de realizar da forma correta. Por exemplo, ninguém tem dificuldade de imaginar que a divisão de bits na arquitetura deve tomar cuidado com o tamanho total ou onde e que bit colocar na correção de magnitude, porém para o interpretador todos esses conceitos devem ser explícitos.

Em relação técnicas está longe de ser um projeto perfeito, já em relação à exibição do resultado final é bastante satisfatório. Entretanto, ainda é possível buscar boas melhoras, como na aparência, no que significa os elementos textuais na tela na parte do formulário tanto na parte de preenchimento quanto na parte de cometer um erro, na acessibilidade.

Possíveis futuras atualizações deste projeto devem se focar nas melhoras anteriormente citadas e também adição de testes de unidades, otimização, refatoração de códigos sujos e focar em tornar o site mais intuitivo e explicativo.