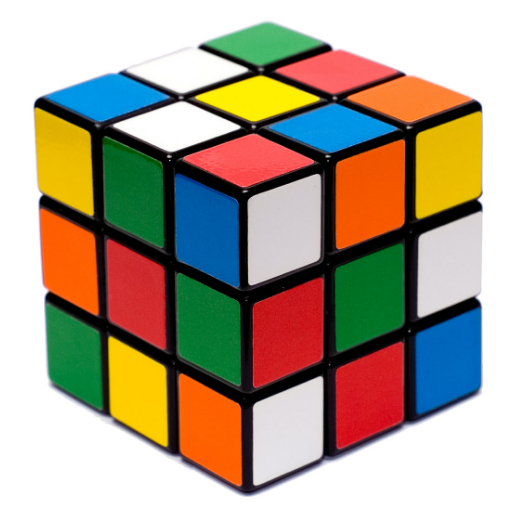
**Abner da Conceição Ferreira**

**Giovanna dos Santos Ferreira**

**Lógica de Programação**

**A arte de pensar ordenadamente**

****

**1°EDIÇÃO**

**AMPARO-SP**

**2023**

|  |
| --- |
| Esta obra está licenciada em Creative Commons.  <a rel="license" href="http://creativecommons.org/licenses/by-nc/4.0/"><img alt="Licença Creative Commons" style="border-width:0" src="https://i.creativecommons.org/l/by-nc/4.0/88x31.png" /></a><br /><span xmlns:dct="http://purl.org/dc/terms/" property="dct:title">Lógica de Programação-A arte de pensar ordenadamente</span> de <a xmlns:cc="http://creativecommons.org/ns#" href="https://github.com/abnerdev23/L-gica-de-Programa-o--A-arte-de-pensar-ordenadamente/tree/main" property="cc:attributionName" rel="cc:attributionURL">Abner da Conceição Ferreira e Giovanna dos Santos Ferreira</a> está licenciado com uma Licença <a rel="license" href="http://creativecommons.org/licenses/by-nc/4.0/">Creative Commons - Atribuição-NãoComercial 4.0 Internacional</a>.<br />Baseado no trabalho disponível em <a xmlns:dct="http://purl.org/dc/terms/" href="https://github.com/abnerdev23" rel="dct:source">https://github.com/abnerdev23</a>.<br />Podem estar disponíveis autorizações adicionais às concedidas no âmbito desta licença em <a xmlns:cc="http://creativecommons.org/ns#" href="2023" rel="cc:morePermissions">2023</a>. |

|  |
| --- |
| This work is licensed under the Creative Commons Atribuição-NãoComercial 4.0 Internacional License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/4.0/. |

|  |
| --- |
|  |

**Dedicatória**

.

À minha esposa Marta, que sempre me apoiou em todos os meus Projetos,Faculdades e também nas adversidades da vida, pois “ ao lado de um grande homem sempre tem uma grande mulher”.e você é esta pessoa que Deus colocou na minha vida, te amo muito .

À minha filha Giovanna que abraçõu a carreira de tecnologia e dedicou muitos anos de estudos, finais de semana, abdicando até da vida social para estudar e se especializar e está começando a colher os frutos.Que Deus te abençõe grandemente nesta jornada, pois tecnologa é análoga a carreira de Medicina e de Professor, se estuda para sempre, não pode parar, aliás isto é muito bom, pois mitiga os riscos de Alhzeimer quando chegarmos na melhor idade.

À meu filho Artur que está descobrindo novos caminhos, e encontrou um motivo para estudar; solucionando o problema das pessoas com a tecnologia.

Aos meus alunos que sempre me incentivaram a produzir aulas melhores e a continuar estudando e pesquisando.

À todos os professores pelos quais eu passei, e tiveram paciência comigo e acreditaram que aquele menino poderia se tornar “alguém na vida” , me impondo limites, cobrando tarefas e seminários ( que aliás melhoraram minha timidez).

Aos meus pais por me incentivarem sempre a estudar, pois tive poucos brinquedos, mas livros nunca faltaram na casa deles.

Finalmente, seria eu ingrato se não agradecesse a Deus por ter me emprestado inteligência, força de vontade, determinação, resiliência e acima de tudo o desejo de compartilhar com outras pessoas 100% do 1% que sei, mas com certeza de uma forma clara, transparente e traduzida, pois é isto o que falta nos dias atuais para uma educação de qualidade.

*“Os verdadeiros artistas criam coisas reais e que serão usadas”.*

*(Steve Jobs)*

**Sobre os Autores**

**Abner da C.Ferreira**

Abner Ferreira nascido em Guarulhos-São Paulo em 1973, sempre gostou de estudar mas a virada de chave mesmo foi SENAI, onde se formou em 1989 no curso de Eletricista de Manutenção.Trabalhou na Indústria,Comércio, Empreendedor e autodidata, percebeu qua a chave para abrir as portas que precisava estava nos estudos.Graduado em Matemática e Física pela Universidade Guarulhos\_UNG, Graduado em Pedagogia pela Faculdade da Aldeia de Carapicuíba-FALC e Graduando em Engenharia da Computação pela UNIVESP, lecionou em diversos colégios particulares, sindicato dos hotéis de Guarulhos e atualmente é professor efetivo da Secretria da Educação do Estado de São Paulo.Atuou como supervisor do PIBID do Instituto Federal de São Paulo-pólo Guarulhos onde orientava dez residentes da Graduação em Matemática.

**Giovanna dos Santos Ferreira**

Giovanna Ferreira, nascida em Guarulhos-São Paulo em 2003, é graduada em Análise e Desenvolvimento de Sistemas pela UNIP-Universidade Paulista, desenvolveu vários TCCs , resolveu empreender em 2023 recém formada prestando serviços para a YPÊ-Química Amparo , na área da TI, em parceria uma grande empresa de Tecnologia de Amparo, a Shark It.

**Prefácio**

O objetivo desta obra é facilitar o aprendizado de qualquer linguagem de programação de uma forma divertida e bem visual, pois em qualquer curso o mais importante é a didática e, se o professor ensinar sobre manga nas aulas, posteriormente deverá ter exercícios de manga, trabalhos sobre manga e consequentemente prova sobre manga, o que infelizmente, na maioria das vezes não acontece.Nos apoiamos em grandes autores para escrevermos esta obra, então, não temos aqui a pretensão de forma alguma de excluir ou até mesmo denegrir nenhum curso técnico ou de graduação, mas sim temos o intuito de colaborar com o aprendizado e desenvolvimento não somente de jovens, mas de qualquer pessoa que queiraa estudar programação.

Bons estudos!

**LISTA DE FIGURAS**

Figura 1- 10

Figura 2- 10

Figura 3- 10

**SUMÁRIO**

1. [**Introdução 8**](#_bookmark1)
2. [**Módulo 1-Uma breve história da computação 8**](#_bookmark1)
3. [**Módulo 2-Introdução a Lógica de Programação 8**](#_bookmark1)
4. [**Módulo 3-Mas afinal, o que são Algoritmos? 8**](#_bookmark1)
5. [**Módulo 4-Linguagens de programação 8**](#_bookmark1)
6. [**Módulo 5-Ambientes de Desenvolvimento-IDEs 8**](#_bookmark1)
7. [**Módulo 6-Microcontroladores e Microprocessadores 8**](#_bookmark1)
8. [**Módulo 7-O Microcontrolador Arduino 8**](#_bookmark1)
9. [**Módulo 8-Variáveis e Constantes na programação 8**](#_bookmark1)
10. [**Módulo 9-O que são Arrays? 8**](#_bookmark1)
11. [**Módulo 10-Operadores e a tabela-verdade 8**](#_bookmark1)
12. [**Módulo 11-Estruturas de Decisão/Seleção 8**](#_bookmark1)
13. [**Módulo 12-Estruturas de Repetição 8**](#_bookmark1)
14. [**Módulo 13-Funções na programação 8**](#_bookmark1)

[**CONCLUSÃO 23**](#_bookmark21)

[**REFERÊNCIAS 24**](#_bookmark22)

# INTRODUÇÃO

É muito comum aos iniciantes na área de Tecnologia da Informação-TI, ficarem completamente perdidos em relação a que linguagem escolher, qual é a melhor, qual é a mais fácil ou qual é a mais usada no mercado de trabalho atualmente.Digo isto, pois aconteceu comigo e com vários amigos, que aliás desistiram antes mesmo de conhecerem o mais legal da TI, a área de desenvolvimento.

O objetivo deste curso é de facilitar aos egressos na profissão, para que não desistam, pois é o que mais acontece nas Universidades e cursos de tecnologia, sejam eles cursos técnicos ou cursos profissionalizantes, devido a inversão no ensino da Lógica, linguagem, IDE, e um treinamento específico através de projetos e não de um único TCC, visto que o tempo é muito precioso e as tecnologias se tornam defasadas em apenas alguns meses.

Quando fui estudar Engenharia da Computação, fiquei apavorado quando fui apresentado a linguagem JAVA, não por dizerem que ela era difícil, o que não é verdade, mas por não terem ensinado a Lógica de programação primeiro, que é fundamental para qualquer linguagem.Pois bem, estudei a mesma durante um semestre, fui atrás de livros em bibliotecas virtuais e percebí que tinha algo de muito errado naquela metodologia.Como ensinar uma linguagem para um aluno que nunca vira uma IDE na vida? Como iniciar por uma linguagem que era orientada a objetos sem antes ensinar uma linguagem estruturada e a diferença entre as duas?Chegou então o grande e temido dia da prova impressa de JAVA, estudei muito ( e de forma errada ) para tirar a média apenas, que era nota 5.Será se o problema estava em mim?

Os motivos acima me levaram a me aprofundar no assunto, pois apesar de já programar em C, devido ao arduíno que me abriu a mente para a tecnologia, e a desenvolver uma metodologia, sobre o quê estudar primeiro antes de se aventurar num PYTHON por exemplo sem saber de conceitos fundamentais e nos primeiros exercícios desanimar e abandonar o curso.Isto que aconteceu comigo, acontece todos os dias nas Universidades, Faculdades, cursos livres ou em escolas técnicas.Longe de mim de denegrir a imagem de tais escolas/universidades, mas não sou o primeiro e nem serei o último e o objetivo aqui é desmistificar para que haja avanços significativos na sua nova vida profissional.

Espero do profundamente que as teorias aqui apresentadas, que estudei em várias obras referenciadas no final desta obra, sejam de grande proveito para você, e que possamos abrir portas no “labirinto dos ratos”, que nos impuseram quando crianças, dizendo que programação é só para nerds, só para homens,etc...

Na verdade, programação é para quem tem o desejo, para quem tem vontade de aprender, de ajudar as pessoas, solucionar problemas da sociedade.

É bem verdade que, existem altos salários na TI, mas, você não deve escolher esta profissão por este motivo, pois existem grandes desafios também, um deles é o de se manter sempre atualizado em relação as novas tecnologias, pois senão, ficará desatualizado e com certeza logo ficará fora do mercado de trabalho.

Uma dica bem interessante que funciona com com estuda tecnologia é, se você estuda por livros, estude o tema, abra uma IDE e vá praticar, você vai perceber que se esquecer de digitar um ponto e vírgula no seu código, ele vai “bugar”.Outra coisa é a velocidade, quanto mais treinar, mais rápido ficará naquela tecnologia, com certeza.Agora, você que gosta de assistir vídeos do youtube e odeia ler,tem solução; assista aos vídeos, umas cinco aulas por exemplo, um módulo se for curto e, abra o VSCode e vá praticar, pois ninguém se torna programador apenas assistindo vídeos na internet, mas sim, praticando, fazendo projetos, participando de comunidades, estudando documentações em sites oficiais,etc...

Outra dica muito importante, faça todos os exercícios deste livro, pois estão numa ordem lógica de aprendizagem e nunca se contente apenas com isto, invista em você, compre livros, eles são caros dependendo o ponto de vista, veja; se você pagar R$200,00 por um livro de HTML e CSS você acha um absurdo não é mesmo? Então, de repente isto é um lanche com seus amigos e você não achou caro.Se seu salário hoje for de R$1800,00 e pretende aumentá-lo para R$3000,00 por exemplo, saiba que o investimento é muito baixo em relação ao custo-benefício (16% de investimento e retorno de 66,6% aproximadamente).É claro que, isto não acontece de forma instantânea, mas é rápido em se tratando de tecnologia, depende de você exclusivamente.

A última dica é, utilize o método HBC (Horas-Bunda-Cadeira), mas, em estudo e prática efetivamente, sem celular, redes sociais, distrações,etc... Crie um ambiente de estudo favorável, não na cama de forma confortável, pois dará sono.Separe um lugar da casa com seu computador, fone de ouvido se necessário para entender a semântica e a sintaxe da linguagem.E, lembre-se: “ Nao existe cozinheiro sem chegar perto do fogão”, ou seja ; saia dos videos e vá programar!

# Módulo 1- Uma breve história da computação

A história da computação inicia com a necessidade do homem de calcular, pois computar significa calcular, contar.Voltando então na história, a computação nasceu então entre 5000 a 7000 anos atrás aproximadamente, com a invenção do **àbaco** pelos mesopotâmios,conforme mostra a figura 1, posteriormente aprimorado pelos chineses e romanos.



Fig.1

Em 1863, o padre William Oughtred inventou a **régua de cálculos**, que calculava logaritmos e fazia multiplicações e divisões,figura 2.



Fig.2

Em 1642, Pascal criou a primeira calculadora mecânica da história, a **Pascaline**,fig.3.Ela fazia somas e subtrações,figura 3.



Fig.3

Em 1831Joseph M.Jacquard desenvolveu o **tear programável**, para o corte de tecidos de forma automática, figura 4.



Fig.4

Em 1801, Charles Babbage criou a **máquina das diferenças**, que resolvia funções de trigonometria , logaritmos e polinômios de forma simples sem energia elétrica, apresentada na figura 5.Aparece então, a primeira mulher na computação, Augusta Ada Byron Lovelace, que encantada pelo Tear programável, é convidada por Babbage para ajudá-lo neste novo projeto.Ada Lovelace se preocupava com a parte da programação através de símbolos enquanto Babbage com os números e a parte mecânica.Uma coisa interessante ,por questões políticas e financeiras, Babbage morreu e seu projeto foi construído posteriormente.Outra curiosidade é que a máquina tinha as características dos computadores atuais; entrada, saída, memória expansível e uma central de processamento, uma tecnologia muito avançada para a época.

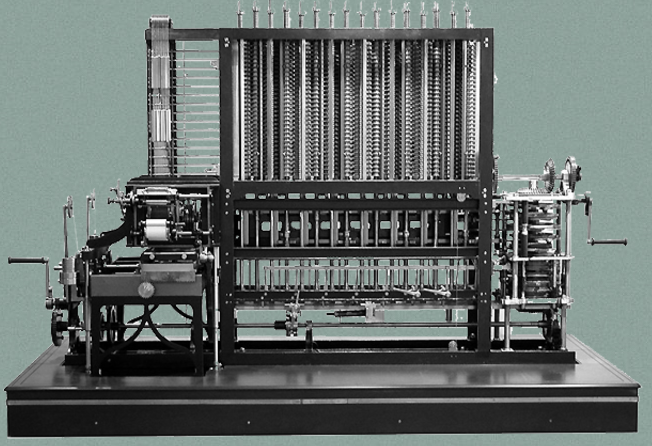


Fig.5

Naquela época, a única utilidade que todos viam para o computador, era o de realizar cálculos, mas, Ada Lovelace afirmou que não, o computador daria para trabalhar com símbolos e imagens também, com um propósito geral.

Em 1847,George Boole, desenvolveu a Lógica Moderna ou **Lógica Booleana**, o sistema binário e a tabela-verdade que consiste em apenas dois valores 0 ou 1, que é a forma que o computador entende( figura 6).

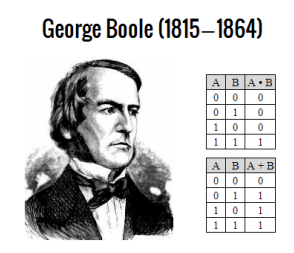


Fig.6

Em 1890, seria realizado um censo nos Estados Unidos e Hermann Holerith resolveu desenvolver uma máquina baseada no Tear de Jacquard e na lógica de Boole, deu tão certo que ele criou a Empresa Tabulating Machine Company, que mais tarde se fundiria a mais três empresas e mudaria de nome, passando a se chamar Computing Tabulating Recording Corporation.Hoje conhecida como IBM.

Por volta de 1944 a Universidade de Harvard (EUA) desenvolveram o **Mark I.**fig.7



Fig.7

Não poderíamos deixar de citar Alan Turing, que adorava correr fig 8, mas que era um grande matemático que estudava Inteligência Artificial e criptografia, inspirado é claro em John Von Neumann, outro grande matemático.Nesta época, estava acontecendo a 2ª grande guerra mundial (1945) e Turing fora chamado para decifrar a máquina de Hitler, a **Enigma** fig.9.Alan Turing então, que era muito bom em codificar coisas, teria agora que fazer o contrário “ descodificar” uma máquina que matava milhões de soldados americanos.Foi então que ele criou uma máquina mais poderosa que a Enigma, denominada de “**A Bomba”** inicialmente e mais tarde rebatizada de **A máquina de Turing** fig.10**.**

Fig.8 fig.9

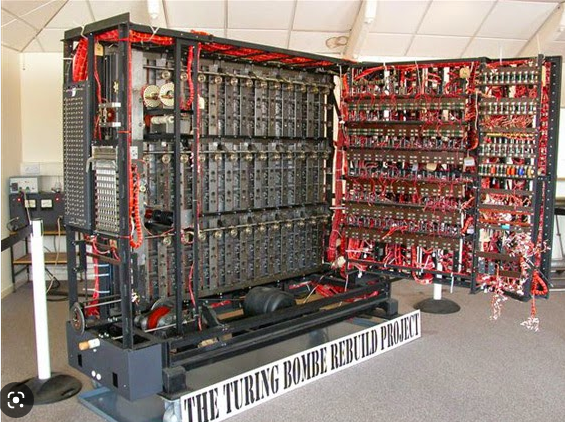


Fig.10

## Gerações de computadores

Os computadores estão dividivos em quatro gerações, sendo elas:

**1ª Geração** (1946-1959)

Eram gigantescos pesavam 30 toneladas , mediam aproximadamente 25 metros e utilizavam válvulas.

**2ª Geração** (1959-1964)

Com o avanço da tecnologia, surgiu o circuito impresso e o transístor, que reduziu consideravelmente o tamanho dos computadores.

**3ª Geração** (1964-1970)

Surge nesta época o circuito integrado que continha vários transístores embutidos nele.Além do computador diminuir ainda mais de tamanho, ele começa a ficar mais barato.

**4ª Geração** (1970- até hoje)

Na quarta geração, surgem os microprocessadores que possuem vários circuitos integrados embutidos.Foi desenvolvido o computador Altair que não possuia monitor, nem mouse, pois isso não existia até o momento.Como o lançamento foi um sucesso pois vendeu 10 vezes mais que o esperado, um jovem chamado Bill Gates resolveu criar uma linguagem para este computador a Altair Basic.Outro jovem chamado Steve Jobs, percebeu que aquele computador não era fácil de ser usado por pessoas comuns e resolveu acrescentar um monitor, para que o usuário enxergasse o que estava acontecendo dentro do computador, deu o nome a ele de APPLE I, por volta de 1979.

Outras empresa fizeram algo parecido mas, sempre incrementando, tais como os computadores LISA (1983) e a MACHINTOSH (1984) que inseriram a interface gráfica atual e o mouse.

O jovem Bill Gates então fundou a Microsoft para papalelo com a APPLE.Como seus programas estavam inferiores aos de Jobs, resolveu fazer uma parceria APPLE-MICROSOFT que seria desfeita em breve e cada um iria para seu lado, mas segundo ele nunca se tornaram inimigos.





Steve Jobs Bill Gates

# Módulo 2-Introdução a Lógica de Programação

Definimos lógica de programação a forma de se ordenar uma sequência de instruções, de forma que faça sentido e que ao final, solucione o problema do cliente. .Vejamos na prática como isto funciona.

Carlos está com fome e quer comer miojo, ele vai até o armário e descobre que não tem miojo em casa.Carlos precisa decidir o que fazer, pois a cada minuto, seu estômago “ronca” cada vez mais.Decidido, ele pensa:

|  |
| --- |
| Início   1. Ir ao mercado; 2. Comprar miojo; 3. Voltar para casa; 4. Colocar 300 ml de água em uma panela para ferver no fogão por 3 minutos; 5. Ao iniciar a fervura, abrir o pacote de miojo e colocar na panela; 6. Observar o tempo; 7. Após 2 minutos, colocar o tempero pronto; 8. Aos 3 minutos, desligar e despejar num prato; 9. Comer o miojo.   Fim |

A questão é a seguinte, esta sequência de instruções, não precisa necessariamente estar nesta ordem.Vejamos esta outra sequência:

|  |
| --- |
| Início   1. Comprar miojo; 2. Ir ao mercado; 3. Voltar para casa; 4. Colocar 300 ml de água em uma panela para ferver no fogão por 3 minutos; 5. Ao iniciar a fervura, abrir o pacote de miojo e colocar na panela; 6. Observar o tempo; 7. Após 2 minutos, colocar o tempero pronto; 8. Aos 3 minutos, desligar e despejar num prato; 9. Comer o miojo.   Fim |

Percebemos claramente que algo deu errado neste tipo de pensamento.Existe uma sequência de instruções, claras, porém, estão numa ordem bem confusa.

Primeiro Carlos compra miojo, depois vai ao mercado.Mas, onde ele comprou o miojo se ele foi ao mercado em seguida?

Muito bem, a Lógica de Programação serve exatamente para isto, ordenar de forma clara, e numa ordem que faça total sentido para qualquer um, isso mesmo, eu disse qualquer um, possa realizar a tarefa.Agora, imagine se você não consegue organizar de forma clara, um conjunto de instruções para alguém que nunca fez miojo na vida, se vai conseguir “ensinar” um computador a realizar uma ou mais tarefas.Entende agora o grau de importância de se aprender Lógica de Programação e algoritmos antes de entrar de cabeça em qualquer linguagem de programação?

Existem cursos que ensinam diretamente lógica de programação juntamente com a linguagem de programação.Não que seja proibido, mas veja, o aluno acha que está avançando, mas quando se encontra sozinho para desenvolver uma solução, sem aqueles exercícios prontos, adivinha? Não vai conseguir, e vai ter que voltar e estudar lógica desde o início.

# Módulo 3- Mas afinal, o que são Algoritmos?

No módulo anterior estudamos a Lógica de Programação que é organizar um conjunto de instruções de forma que faça sentido. A este conjunto de instruções, damos o nome de algoritmos.

Os algoritmos são utilizados para representar a solução de um problema e são instruções “ensinadas” aos computadores e executadas por eles.

Quando aprendemos a base dos algoritmos, podemos aplicar este conhecimento em qualquer linguagem de programação (C, C#, Java, Javascript, Python, etc...).

Quando temos um problema cotidiano, e conseguimos resolver este problema, através de uma determinada solução, não quer dizer que esta seja a única solução possível. Outra pessoa pode resolver o mesmo problema utilizando outra solução. Essa é a parte mais interessante dos algoritmos pois cada programador pensa de forma diferente e resolverá um mesmo problema com soluções e linguagens de programação diferentes.

1. **Algoritmo Computacional**

Como vimos na história da computação, o computador realiza tarefas, entretanto essas tarefas necessitam de instruções precisas, claras, detalhadas e que ele entenda. Chamamos essas instruções específicas de **Programa**.

1. **Formas de Representação**

Existem três formas de representarmos os algoritmos:

1. Descrição narrativa;
2. Fluxograma;
3. Pseudocódigo
4. **Descrição Narrativa**

Na Descrição Narrativa, a sequência de passos é descrita exclusivamente na língua portuguesa. Exemplo:

Média Bimestral de um aluno

* Digitar as 4 notas de provas;
* Calcular a média aritmética;
* Se a média for maior ou igual a 5, mostrar na tela a mensagem “Aprovado!”;
* Senão, mostrar na tela a mensagem “Reprovado”.

1. **Fluxograma**

O Fluxograma é a representação gráfica dos algoritmos utilizando formas geométricas. Os símbolos que vamos dar como exemplo abaixo, variam de autor para autor. Outro aspecto importante sobre fluxogramas, é que a visualização e entendimento é bem melhor do que a Descrição Narrativa, entretanto, o uso deles é bem interessante e eficaz quando trabalhamos com pequenos programas, dificultando quando o programa possui muitas linhas de código.

Exemplo:

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Formas Geométricas Específicas** | |
|  | Início e Fim do Fluxograma |
|  | Entrada de Dados |
|  | Saída de Dados |
|  | Atribuição |
|  | Decisão |

1. **Pseudocódigo.**

O pseudocódigo é uma forma natural de escrevermos nosso código, se você quiser se aprofundar neste assunto, visite o canal <https://www.youtube.com/watch?v=6OIADpFImtc> que ensina sobre o compilador Portugol.

# Informação x Dados x Conhecimento

Os **dados** elementos a serem armazenados, tratados e processados.

Exemplos:

O rg de uma pessoa :20.554.512-4;

O nome de um funcionário(a): ‘Marcela’;

O endereço do funcionário(a): ‘rua Sabiá’;

A profissão de um funcionário na empresa: ‘Arquiteto’.

Temperaturas mínima e máxima em Amparo nesta semana: 9° e 23°.

Possibilidade de chuvas em Amparo daqui a três dias: 89%.

Já a **informação** é o conjunto estruturado e organizado dos dados, ou seja, um depende do outro.

Exemplo:

Funcionário:Marcela

Rg: 20.554.512-4

Endereço: Rua Sabiá

Profissão: arquiteta

**Conhecimento** por sua vez, é a aplicação prática da informação por uma pessoa, após o processamento da mesma.

Marcela pretende lavar roupas nesta semana, ainda não tem dia certo, mas tem muita roupa e sua lavanderia é descoberta.

Estava “mexendo” no celular, viu exetamente na hora em que o ligou que estava 18° de temperatura, mas não ligou muito.Ao continuar pesquisando no Instagram, Facebook, chegou num site de notícias locais que tinha a seguinte informação:

Temperaturas mínima e máxima em Amparo nesta semana: 9° e 23°.

Possibilidade de chuvas em Amparo daqui a três dias: 89%.

Marcela então pensou que se iria chover daqui a três dias, ela teria que tomar uma decisão rápida, ou esperaria passar as chuvas para lavar suas roupas, ou lavá-las imediatamente para que desse tempo de secá-las.Foi o que ela fez, lavou tudo e resolveu seu problema.

Marcela viu os **dados,** no celular, mas como estes estavam soltos, descnectados, não faziam muito sentido.Em seguida ela viu a **Informação**, mas de nada adianta ter acesso a mesma se não houver uma ação, a aplicação prática daquilo em sua vida; lavar a roupa suja.Este último é o **Conhecimento**.

# Banco de Dados

Para guardamos valores, dinheiro, documentos, escrituras, diamantes,ouro, devemos ter um local seguro, idôneo e não corrompido.Este lugar, que é chamado de cofre, deve ter uma senha que somente o dono daquela conta possa acessar, e mesmo que este cofre ou conta corrente esteja em um Banco particular ou público, os gerenciadores do mesmo não podem ter acesso a esta senha, somente aos valores que ali se encontram.



<https://pt.dreamstime.com/fotografia-de-stock-ilustra%C3%A7%C3%A3o-da-porta-do-cofre-de-banco-vista-dianteira-image31022572>

<https://www.keyato.com.br/cofres/cofre-digital-pequeno-value-mini>

Muito bem, no mundo digital os **dados** são estes valores, e seguem as mesmas regras, devem ser armazenados em um local seguro.Este local se chama Banco de Dados.

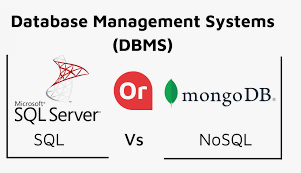
Existem diversos Sistemas que gerenciam os Bancos de Dados, estes sistemas tem o nome de Sistema de Gerenciamento de Banco de Dados-**SGBD.**

Alguns exemplos de SGBD são mostrados logo abaixo:





Os SGBDs são divididos em duas classes, os **Relacionais** (Sql)e os **Não-Relacionais**(NoSql).O Banco de Dados que iremos utilizar aqui no nosso curso é o Mongo DB (NoSql) e lá aprofundaremos mais neste assunto.



# Tipos de Dados

Dados são as informações tratadas e armazenadas no computador.Existem 3 tipos de dados:

1. Literais;
2. Numéricos;
3. Lógicos.

Os **dados Literais**  são as letras, dígitos e caracteres especiais.

Exemplos: **string** ou **char** dependendo da Linguagem de programação

Os **dados Numéricos**  são os números que podem ser o conjunto dos números inteiros ( negativos,zero,positivos) ou o conjunto dos números Reais ( que engloba números naturais, inteiros, decimais,frações,etc...).Exemplos: **int,float,const.**

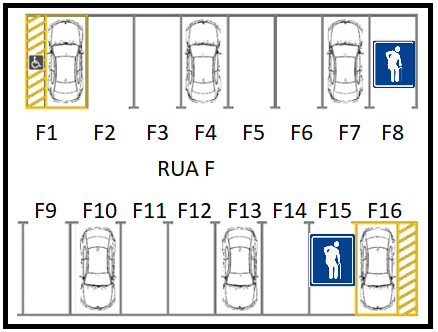
Os **dados Lógicos**  são usados para **Verdadeiro ou Falso.**

# Módulo 5-Variáveis e Constantes na programação

**Variáveis** são espaços reservados na memória do computador.Este espaço reservado tem um endereço e pode ser alterado durante a execução do programa, ou seja, pode variar.

Já a **Constante**, é um espaço reservado na memória, possui um endereço mas ao contrário da variável, não pode ser alterada durante a execução do programa, será como o próprio nome diz, constante até o final.

Imaginemos um estacionamento com cada vaga demarcada e com o nome de cada rua conforme a figura abaixo.

****

Todas as vagas são para carros, porém existem duas vagas destinadas a idosos e duas vagas destinadas a deficientes físicos, que são os tipos de vagas que são reservadas por lei a estas pessoas.Outra coisa a ser levada em consideração é que estas vagas são para veículos pequenos, não cabendo então uma caminhonete,uma moto, um ônibus ou até mesmo um caminhão.Estes, exigem **outro tipo de vaga**.

Analogamente é a memória do computador, quando criamos uma variável, estamos reservando um espaço para guardar ali dentro algum dado.

As variáveis devem ter:

1. Um nome;
2. O tipo de dado;
3. A Informação.
4. **Critérios para criar o nome de uma variável**

Existem alguns critérios universais para a criação do nome uma variável, vejamos quais são eles.

1. Os nomes devem começar sempre por uma letra.

Exemplos:

Media, Telefone. telefone, salario.

1. **Não** devem conter caracteres especiais.

Exemplos:

%porcentagem, &media, s@lario.

1. **Não** devem conter espaços em branco.

Exemplos:

Data de nascimento, media de salario, endereco do cliente.

Correto: data\_de\_nascimento, media\_salario, end\_cliente.

1. **Não** devem conter hífens entre as palavras.No lugar do hífen devemos utilizar underline.

Exemplos:

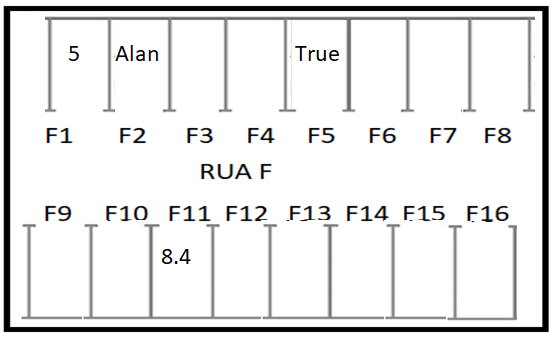
Data-de-nascimento, media-salario.

Correto: data\_nascimento, media\_salario.

Obs:As Linguagens de programação são sensitivas (case-sensitive), ou seja, se você colocar o nome de uma variável de media, com “m” minúsculo , quando você chamar esta variável por Media com “M” maiúsculo, o computador não entenderá, pois se trata de outra variável.

1. O **sinal de igual (= )** em Linguagem de programação é utilizado para **atribuir um valor a uma variável**(colocar um dado dentro dela ).Quando precisamos utilizar o sinal de igual em operações, usamos o igual dobrado (==).Não confunda.

Observe na figura abaixo do estacionamento e imagine agora que no lugar dos carros, colocaremos alguns dados.



Preciso colocar o número 5 dentro da variável F1.Ela deve ter um nome, o tipo e a informação correto?

1 int F1 = 5 ;

O número 5 é do tipo inteiro, portanto na **sintaxe** ( falaremos sobre este assunto mais a frente) desta Linguagem de Programação inteiro é representado por **int.**O nome desta variável é **F1** e a informação a ser armazenada é o próprio número **5.**

Neste caso lemos: A variável F1 recebe (=) o número 5.

Vejamos como serão guardados os outros valores:

Alan, True (verdadeiro), 8.4

Perceba que Alan é um nome formado por caracteres, então será do tipo **string.**

True que significa verdadeiro, é do tipo **lógico (booleano)**

A variável F11 receberá um número que não é inteiro, este número é racional, um número com vírgula.Neste caso e nesta linguagem utilizaremos o **float** para identificar o seu tipo**.**

2 string F2 = “Abner”; // ( aqui pode ser aspas simples ou duplas dependendo da linguagem)

3 bool F5 = TRUE;

4 float F11 = 8.4;

Viu como não é tão difícil, é só “avisar” o computador o tipo de dado, utilizar o código correto (sintaxe) e utilizar a lógica que dará certo.

Você deve ter percebido que foram utilizados o ponto e virgula (**;**) ao final de cada linha, um número no início de cada linha que representa a própria linha para que possamos nos localizar se houver algum erro, e que no comentário da linha 2, eu digitei barra barra (//), que serve exatamente para este fim se quisermos comentar algo naquela linha sem alterar o programa.

Se quisermos comentar algo mais extenso, devemos utilizar barra-asterico, asterisco- barra ( /\* \*/ ).Exemplo de código com comentários:

|  |
| --- |
| /\* Ligar Led com botão  Autor:Abner da C.Ferreira  Data:12/03/2015  \*/  int led\_A = 11; // led ligado na porta digital 11  int botao\_A = 9; // botão conectado ao pino digital 9  int val = 0; // variável para guardar o valor lido  void setup() {  pinMode(led\_A, OUTPUT); // determina a porta digital 11 como saída  pinMode(botao\_A, INPUT); // determina a porta digital 9 como entrada  }  void loop() {  val = digitalRead(botao\_A); // lê o botão A  digitalWrite(led\_A, val); // liga o LED depois de ter lido o botão A  } |

Todos os comentários que estão na cor azul, não irão interferir de nenhuma forma no código, pois na hora de rodar o programa irá “pular” tais comentários.

Segue abaixo o esquema de ligação no **arduino** do nosso código :

|  |
| --- |
|  |

Fonte: O autor

Agora que já entendemos o que são dados,variáveis, constante e seus tipos, vamos conversar um pouco sobre microcontroladores.

# Microcontroladores

Mas você deve estar se perguntando porque estamos estudando microcontroladores num curso de Linguagem de Programação,Lógica,Node,etc?

Então, quando fui estudar Linguagem de Programação tive muita dificuldade pois, tudo era muito abstrato, ou pelo menos quase tudo, tinha uns comandos meio malucos e esse tal de “imprima na tela” ,”escreva na tela”, eu ficava meio perdido.Não consegui avançar muito pois a **didática** ( forma de ensinar) não era boa.Fiz um curso do SENAI de eletricista de manutenção em 1998 que me ajudou e me ajuda até hoje na compreensão de lógica de programação e de IoT ( Internet Of Things-Internet das coisas ), o que é um circuito fechado, interruptores, motores,etc...Mas o “pulo do gato” veio em 2005 quando criaram um negócio na Itália chamado **Arduino**, uma plataforma robusta, reprogramável e que se ligava a outros módulos, de código aberto (open source ) que permite não somente que você copie os códigos pois ele é livre para ser estudado, mas também os criadores permitem que você copie e fabrique seu próprio arduino ( só não pode usar a marca registrada do infinito), de resto, você pode brincar a vontade usando a imaginação.Eu particularmente conheci o arduino em meados de 2013, quando estudava robótica educacional através do Lego Mindstorms, que é excelente, mas muito cara.Então como ensinar robótica nos colégios particulares se a plataforma Lego custava R$3000,00 aprocximadamente e teriam que ser vários kits.E na escola pública, quem pagaria por isto?Os alunos? O professor? O Estado? O dono do colégio?

Pensando nestas perguntas e como acabara de conhecer o arduino, juntando meus conhecimentos de Física, Eletricista,Eletrônico e Matemático, resolvi me aprofundar neste universo de faça você mesmo ( que hoje é conhecido como Movimento Maker) e tomei ali uma decisão, ensinar robótica, programação e princípios de eletrônica na rede pública sozinho assim que tivesse oportunidade.Esta oportunidade veio quando me mudei de Guarulhos para Amparo interior de São Paulo, onde meu projeto foi aceito e leciono robótica e matemática em um colegio de tempo integral.

O site oficial do arduino é **arduino.cc,** lá você pode estudar e se aprofundar ainda mais pois existe uma documentação imensa.Claro que vou deixar também os melhores canais do youtube como indicação, mas saibam que estudaram por lá.

Percebí que quanto mais eu estudava programação pelo arduino que é bastante concreto, pois é visual, eu vejo claramente o que está acontecendo quando eu programo,que eu entendia melhor a Lógica de Programação.Então começei a associar por analogia a Linguagem de Programação do arduino, que é baseada em C e C++ com uma outra Linguagem que aliás é muito temida por muitos programadores, mas, que é mal compreendida pois faltam conceitos básicos de Lógica de Programação.Sim, estou falando do JAVA.

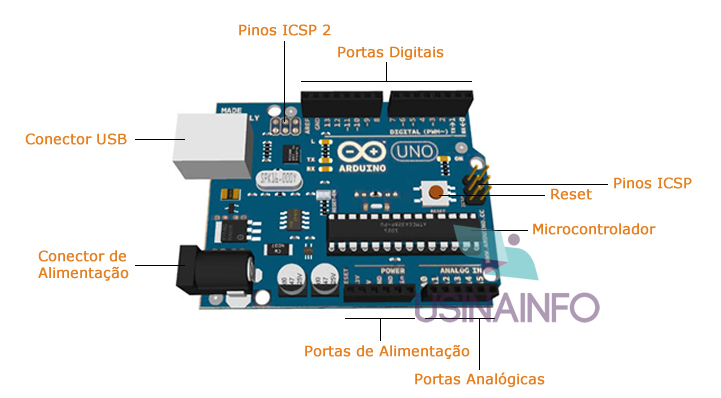
E é por este motivo que vamos iniciar nossos estudos em Lógica de Programação, fazendo, programando, acertando, errando, corrigindo e copiando, sim,pois no início é um tal de copia e cola da internet que vou te falar, mas, quem nunca, não é mesmo?

O problema é que você deve aprender a programar utilizando a sintaxe correta da linguagem ( o código certo), pois cada linguagem tem a sua própria sintaxe.

Os microcontroladores são dispositivos pequenos, plataformas que podem ser programadas e reprogramadas, contém um chip e memória.Possuem também as portas de entrada e de saída que são as mesmas.Quando queremos que uma porta ou pino seja de entrada, programamos esta como **INPUT** e quando queremos que ela seja saída, como **OUTPUT.** Ele tem uma filosofia muito importante que é chamada de **OPEN SOURCE,** ou seja , de Código Aberto.Isto quer dizer que você pode copiar ou melhorar qualquer código que esteja tanto na plataforma oficial quanto na internet pois é autorizado, inclusive copiar e reproduzir o hardware,isto mesmo copiar o arduino.Mas, tem uma ressalva, a marca registrada abaixo não pode ser copiada e reproduzida de forma alguma, ok?

Este dispositivo funciona com uma tensão de entrada entre 5V a 12V, mas i ideal mesmo é 9V, para trabalhar bacaninha.

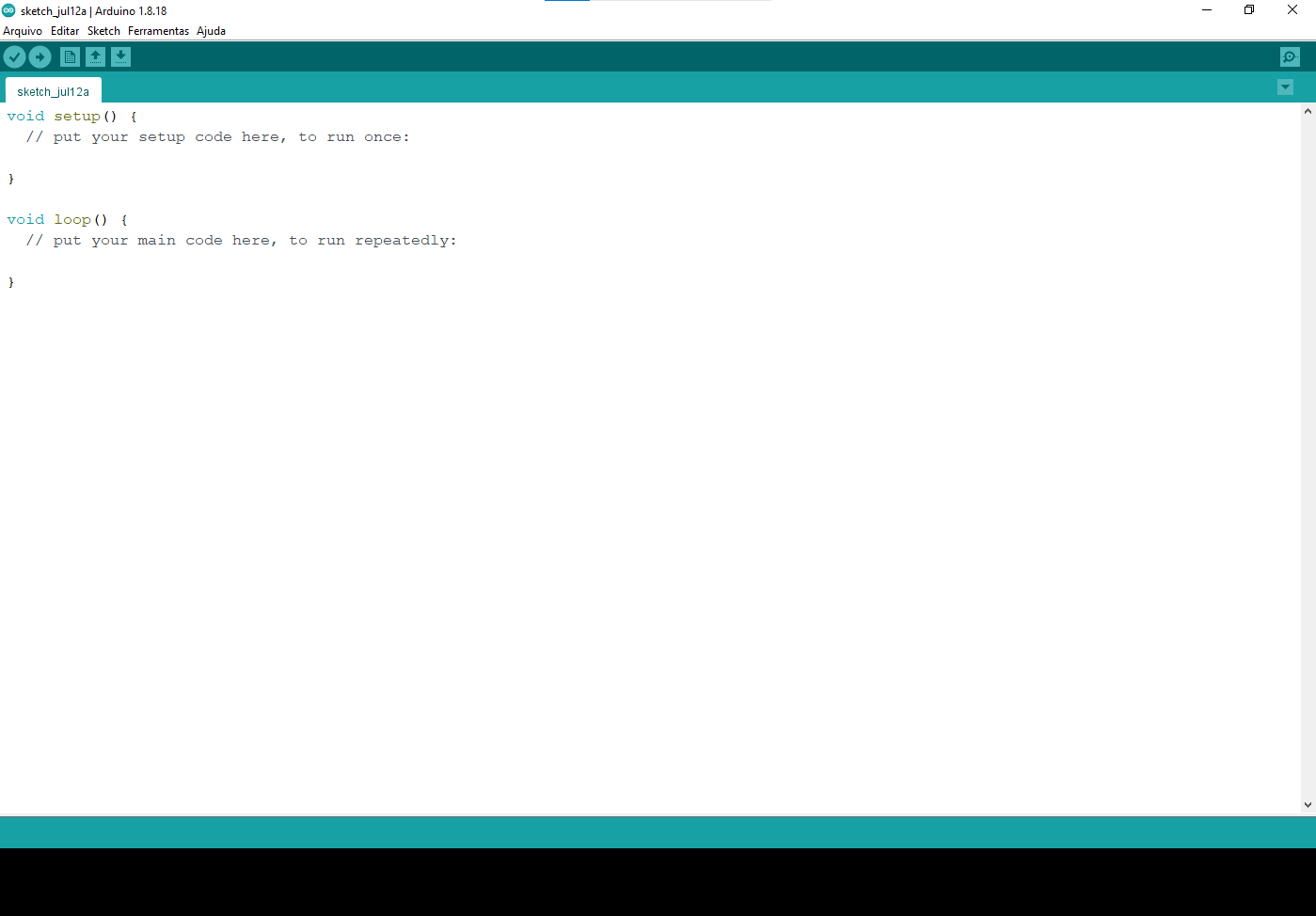




Fonte:UsinaInfo

**Ambiente de Desenvolvimento Integrado**

Toda Linguagem de programação deve ser escrita em algum lugar concorda? Pois então, este lugar se cham Ambiente de Desenvolvimento Integrado a **IDE (***Integrate Development Integrate***).**Na IDE você consegue desenvolver tudo pois existem vários recursos para auxiliá-lo, inclusive você baixa gratuitamente no site arduino.cc.Um dos recursos é escolher qual arduino será utilizado, fazer download, fazer upload, escrever código, compilar código e etc...

****

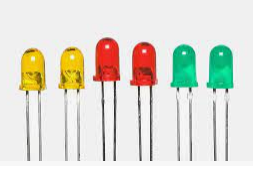
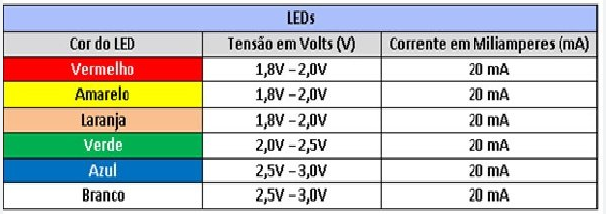
Version 1.8

Fonte: o autor

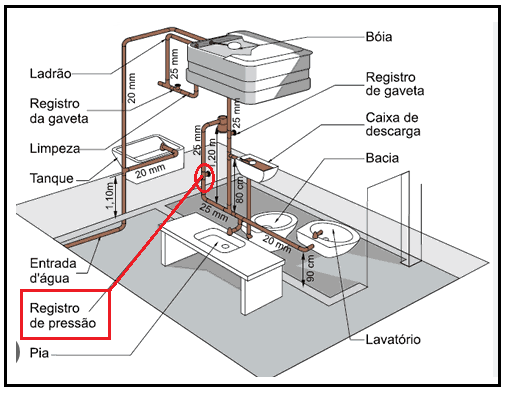
**5.-Alguns componentes eletrônicos**

Nesta apostila faremos algumas montagens utilizando não somente o arduino, mas também leds, resistores e botões (push-botton).Cada um deles tem sua função específica.

O Led é uma pequena lâmpada que de tensão que varia de 1,5V a 3V, dependendo de sua cor é claro.

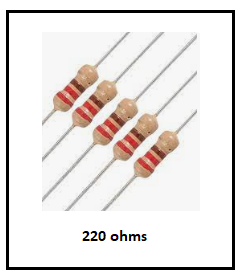
 

Já os resistores funcionam como “ redutores” de tensão/corrente para não queimarmos nossos leds, pois em cada porta do arduino sai 5 volts.O resistor então que utilizaremos muito aqui é chamado de resistor do Led, varia entre **220 ohms e 330 ohms.**



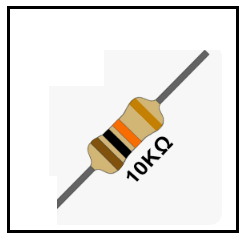
O registro de água serve para abrir ou fechar a água, mas também conseguimos diminuir a pressão da água se estiver muito forte.

O resistor faz exatamente isto, porém ele possui um valor fixo.



**Botão de pressão ou PUSH BOTTON**

Temos outro componente que será usado , o nome dele é push bottton ou simplesmente botão de pressão.Ele também necessita de um resistor para ser ligado ao arduino.Este resistor tem um valor de 10Kohms, isso mesmo 10 000 ohms.Portanto, muita atenção, pois as vezes seu código não funcionará pois utilizou o resistor errado, 100Kohms por exemplo no lugar de 10Kohms.



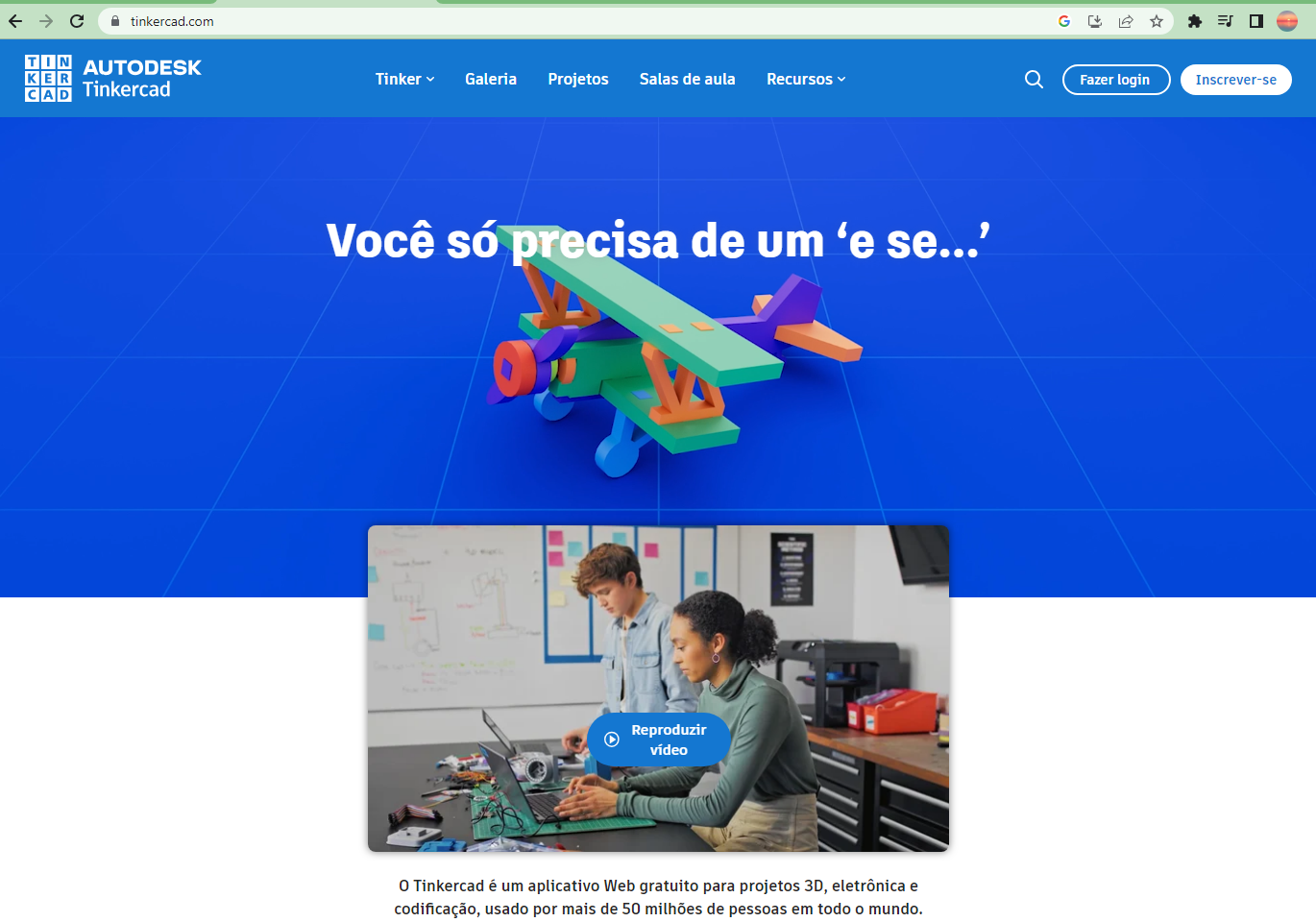
**Módulo 5.1-Simulador Thinkercad**

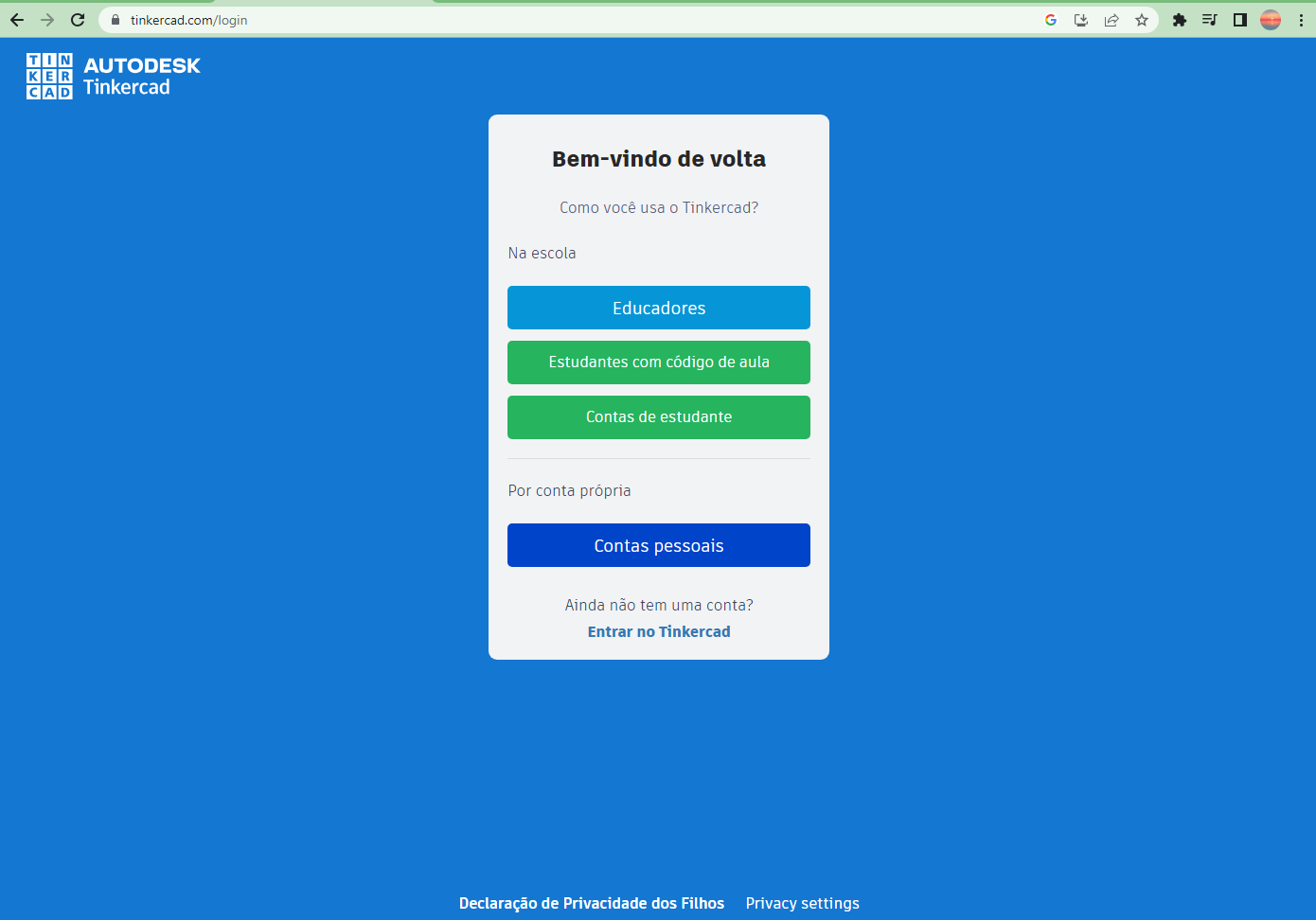
O Thinkercad é uma ferramenta muito útil a qualquer pessoa que deseja desenvolver agum projeto na área de programação, circuitos eletrônicos ou até mesmo criar uma modelagem para uma impressora 3D.

O interessante desta ferramenta, é que como a maioria dos componentes eletrônicos são muito caros e nem sempre temos eles a mão, podemos criar a vontade, testar sem nos preocuparmos de não queimar nada.

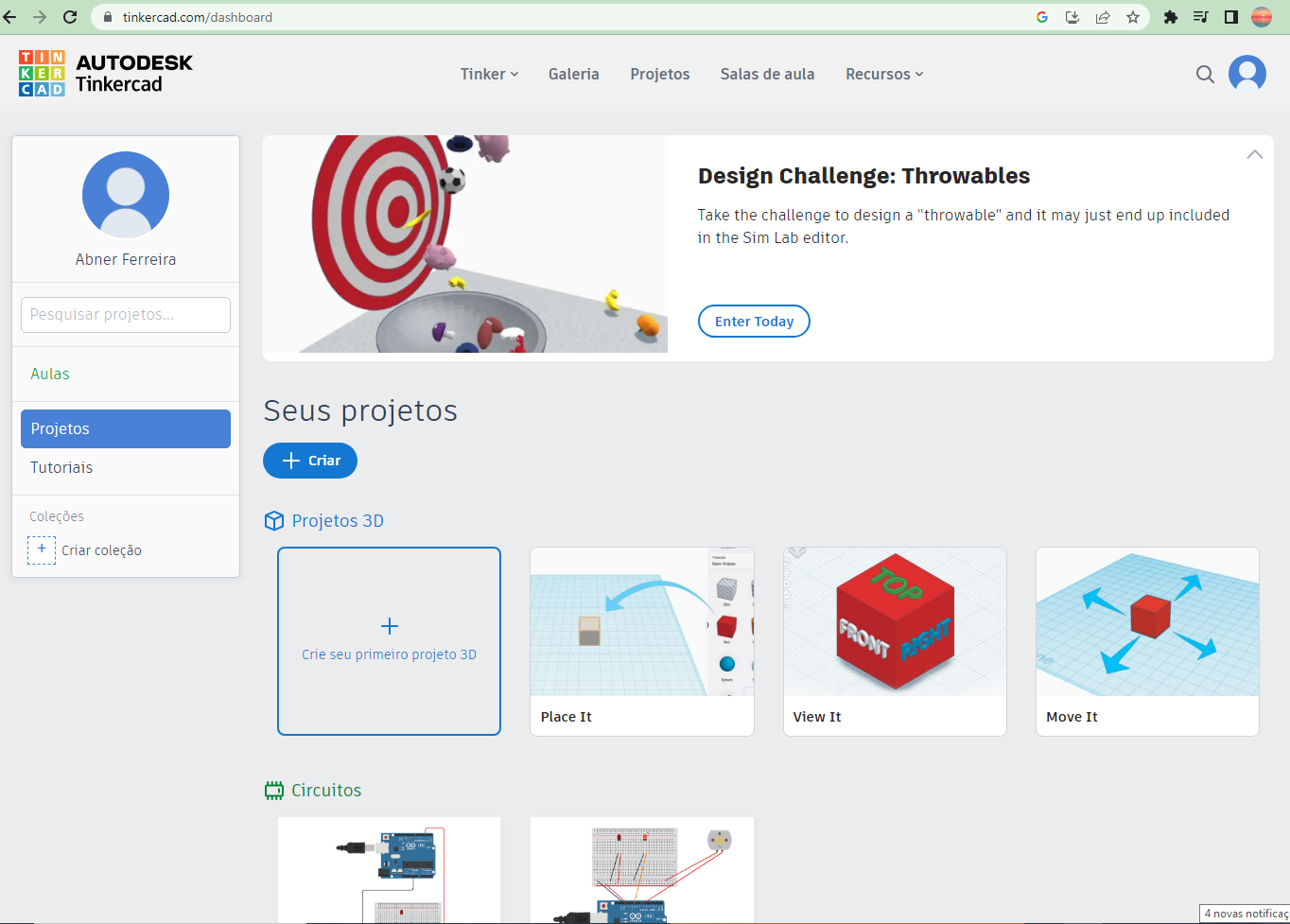
O site desta ferramenta é <https://www.tinkercad.com>, lá você se cadastra com um email, desenvolve seus projetos e olha que legal, eles ficam salvos lá para uso futuro.

Vejamos sua interface:

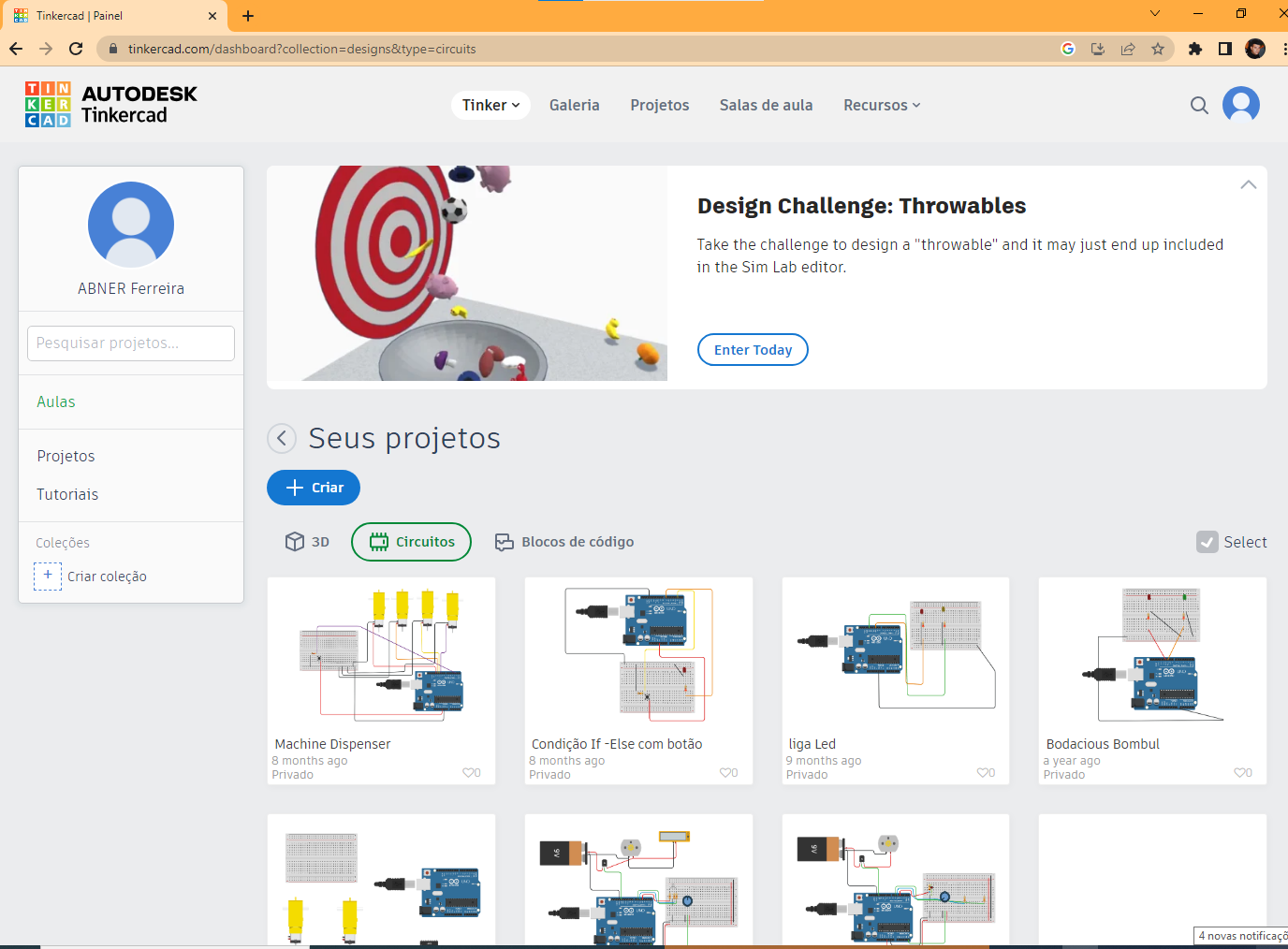
****

****

Tela de login



Tela geral

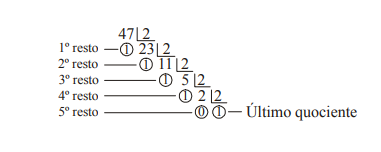
****

Tela de circuitos

Agora, que você já conhece o simulador, vamos realizar o seu cadastro.

**6-Como o computador entende os Dados?**

O computador só entende números binários, que é uma linguagem de baixo nível.Isto quer dizer que se digitarmos o número 47 (sistema decimal), o computador vai entender **101111,** pois serão realizadas divisões sucessivas pelo número dois, quando o resto der zero você pára e pega todos os restos da direita para a esquerda “subindo”, como no exemplo abaixo.



Se quisermos transformar um número binário em sistema decimal, o que fazer?

Por exemplo o número 10100?

Faça uma tabela e escreva/digite o binário em cada quadradinho na segunda linha.Em seguida, da direita para a esquerda, escreva na primeira linha.Eleve o primeiro quadradinho ao número 1, o segundo ao número 2 e assim sucessivamente até o último quadradinho.Resolva as potências somente dos binários que o número é 1.Some tudo e encontrará o resultado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 24 | 23 | 22 | 21 | 20 |
| 1 | 0 | 1 | 0 | 0 |
| **16** | não | **4** | não | não |

16+ 4= 20

O número procurado é 20.

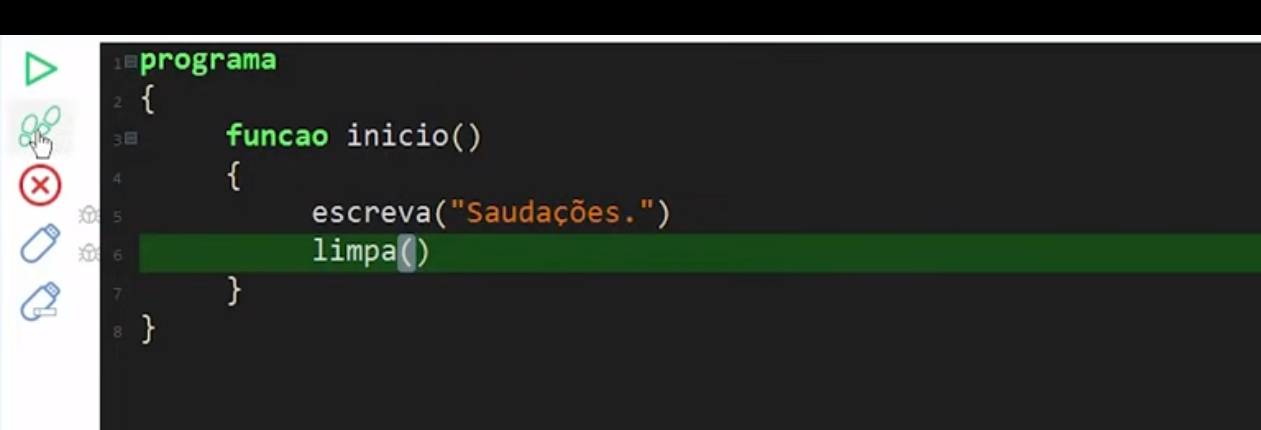
**5.1-Entrada e Saída de dados**

Como você pode ver, o computador só entende números binários, ou seja 0 ou 1, Alto ou Baixo, HIGH ou LOW, ligado ou desligado e, funciona basicamente da seguinte forma; Entrada de dados, processamento destes dados e saída dos mesmos.

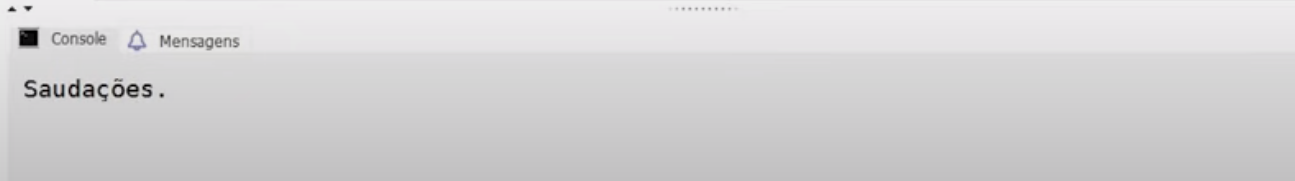
No pseudocódigo existem duas formas disto acontecer; o Leia e o Escreva.

O Escreva é quando algo é mostrado(impresso) na tela(monitor) ao usuário.

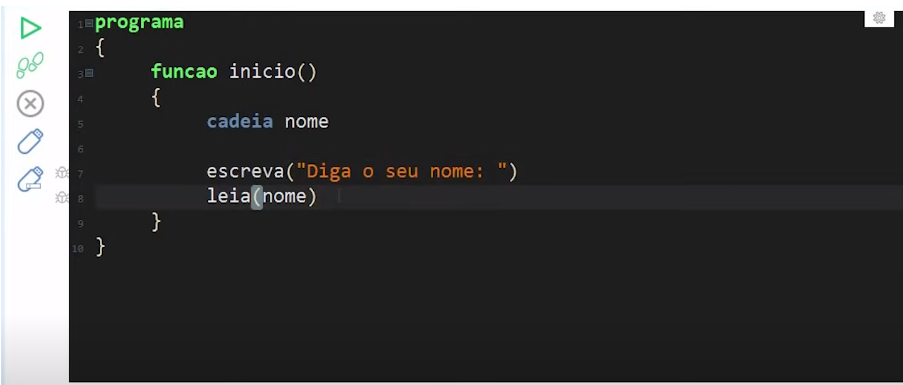
Exemplo de função escreva:



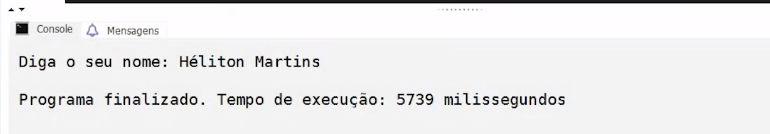
O que será mostrado na tela para o usuário



Exemplo de função Leia:



Veja o que aparece na tela para o usuário:



Outro exemplo:

|  |
| --- |
| Algoritmo Se\_esta\_autorizado\_dirigir  **Var** idade:**inteiro**;  Início  **Leia** idade;  If idade >=18  **Escreva** “ Pode seguir viagem”;  Else  **Escreva** “ Terei que apreender seu veículo e chamar seus pais”;  Fim |

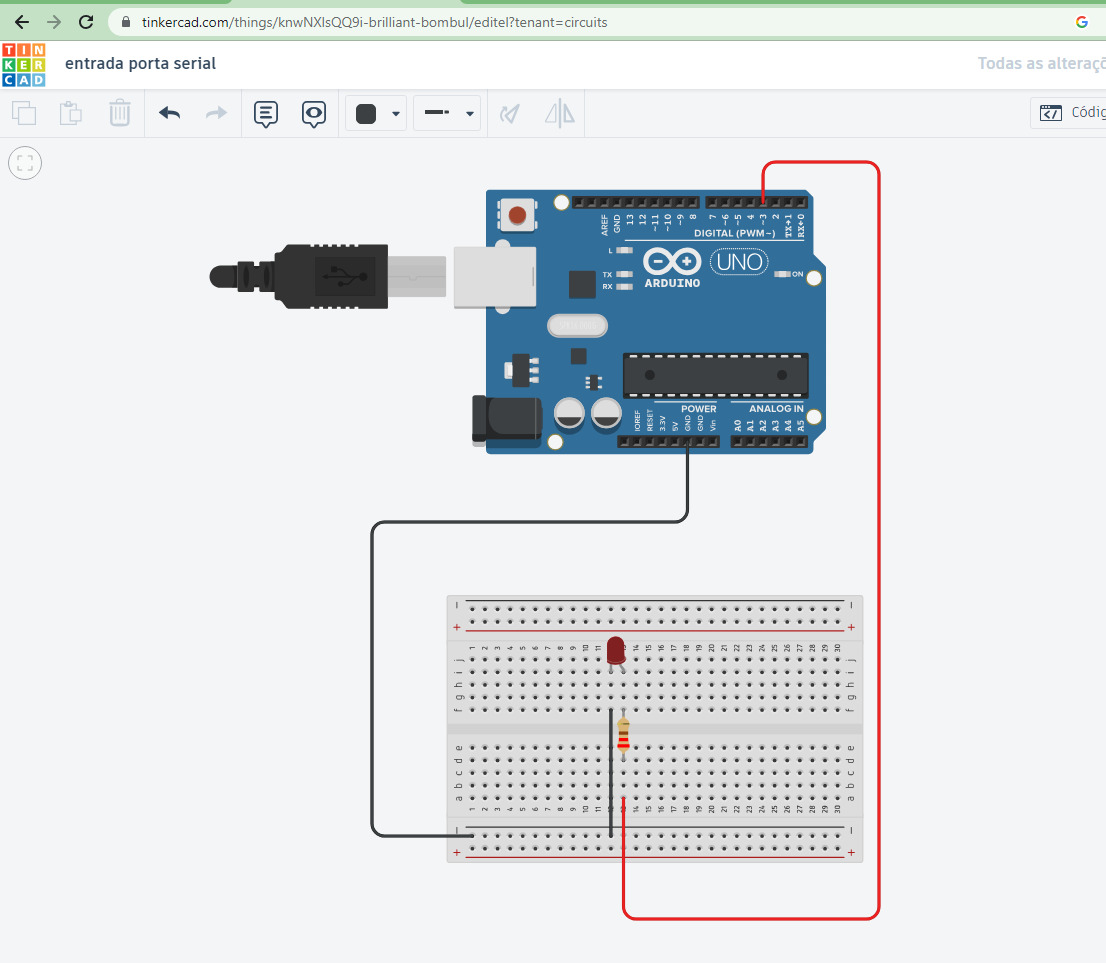
No arduino, podemos utilizar uma tela para “conversarmos” com o programa, ou seja entrar com os dados mais exatamente.O nome desta tela é **Monitor Serial.**

Vamos entrar agora no simulador Thinkercad.

No projeto abaixo, vamos ligar um led ao arduino e todas as vezes que digitarmos o número 1, o programa entenderá que é para ligar o mesmo, caso contrário, se eu digitar o número 0, o led será desligado.

È um programa muito simples, porém estes conceitos serão muito importantes futuramente e não podemos pular de forma alguma.

|  |
| --- |
| /\*  Entrada de dados no arduino pela porta serial  Autor: Abner da Conceição Ferreira  data:13/08/2014  \*/  //declarando a variável led na porta 3  #define led 3  int valor\_lido;// valor lido será um número inteiro teclado  void setup()//neste bloco definimos quem vai fazer o quê  {  Serial.begin(9600);  pinMode(led, OUTPUT);  }  void loop() /\*neste bloco acontece  a repetição, os comandos automatizados\*/    {  if(Serial.available()>0) /\*aqui é analisado se  existe um valor digitado maior que 0 \*/  {  valor\_lido = Serial.read (); //aqui é feita a entrada dos dados  }  if (valor\_lido =='1') // se o valor for 1  {  digitalWrite(led, HIGH); // o led acende  } else{ // senão    digitalWrite(led, LOW); // o led apaga    }  } |



OBS: Perceba o uso de comentários ( // ou /\* \*/) como é de extrema importância, se daqui a 3 anos você precisar abrir este código e realizar uma implementação, se estiver comentado (documentado) , você não terá problemas e se lembrará de absolutamente tudo.Entretanto , se tiver preguiça e não comentar nada, confiando na sua memória, terá sérios problemas e seu trabalho será enorme.

**Analogia entre linguagens**

|  |  |  |
| --- | --- | --- |
| **Linguagem** | **Ação a ser executada** | |
| (\*)Pseudocódigo | LEIA | ESCREVA |
| C | scanf | printf |
| Arduino | Serial.read  digitalRead(na porta digital)  analogRead(na porta analógica) | Serial.println (no monitor serial)  digitalWrite (na porta digital)  analogWrite (na porta analogica) |
| Javascript | prompt  input | alert  console.log |

**(\*)**Lembrando que pseudocódigo não é uma linguagem de programação, e sim, um método de se escrever um código numa linguagem natural de fácil entendimento.

**Módulo 7-OPERADORES**

Operadores são diversos símbolos utilizados nas Linguagens de Programação que representam cálculos, ordem dos dados e atribuições.

Os tipos de dados usados nos operadores são :

|  |  |
| --- | --- |
| **ARITMÉTICOS** | **ATRIBUIÇÃO** |
| DIVISÃO / | SIMPLES = |
| ADIÇÃO + | COM ADIÇÃO += |
| SUBTRAÇÃO - | COM SUBTRAÇÃO -= |
| MULTIPLICAÇÃO \* | COM MULTIPLICAÇÃO \*= |
| RESTO % | COM DIVISÃO /= |
| INCREMENTO ++ | COM MÓDULO %= |
| DECREMENTO -- |  |

|  |  |
| --- | --- |
| **LÓGICOS** | **RELACIONAIS** |
| Conjunção and/&&/e | IGUAL == |
| DISJUNÇÃO or/ou/II | MAIOR > |
| NEGAÇÃO not/não | MENOR < |
|  | DIFERENTE != ou <> |
|  | MAIOR OU IGUAL >= |
|  | MENOR OU IGUAL <= |
|  |  |

**7.1-OPERADORES ARITMÉTICOS**

Vamos abrir agora o nosso simulador TinkerCad e montar o circuito abaixo:

|  |
| --- |
|  |

Fonte:<https://www.youtube.com/watch?v=oOioNOQdO74&list=PLBN3lYxRDbp2teC_Bl79ajtkqRolPgDqt&index=43>

Nesse caso vamos realizar as 4 operações básicas da matemática que são: adição, subtração, multiplicação e divisão.

Exemplo: Precisamos somar 15 + 2, vejamos como fica:

Se quisermos realizar a operação 15 – 2, é só alterar na variável “res”.

Exemplo:

res= a – b;

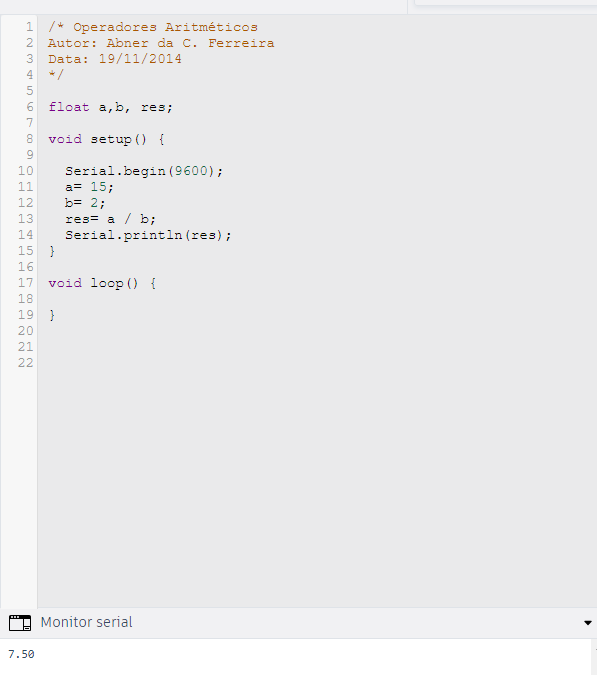
E terá como resultado o número 13.

Na multiplicação 15 X 2, ficará a \* b que terá como resultado 30.

E por fim na divisão que seria 15 : 2 , ficará a / b que terá como resultado 7.

Por que será que isso aconteceu, sendo que o resultado deveria ser 7.5?

Porque as variáveis que estão sendo utilizadas no início do programa (escopo) são do tipo inteiro (int), mas na verdade deveriam ser do tipo decimal (float). Vejamos no exemplo abaixo:

****

Nos operadores aritméticos se quisermos descobrir se um número é par ou ímpar, devemos dividir o primeiro número por 2. Se o resto for 0 o primeiro número é par, senão será ímpar. A sintaxe que devemos usar nesse caso é o %. Vejamos no exemplo abaixo:



No exemplo acima o resultado deu 1, ou seja, o primeiro número (15) é ímpar.

Mas, e se quisermos que exista uma condição de que se o resto for par imprima na tela “este número é par”, senão, imprima na tela “ este numero é impar”.

Vejamos:



Os comando if e else veremos mais adiante com profundidade.

Obs: incremento e decremento falaremos mais adiante quando estudarmos as Estruturas de Repetição.

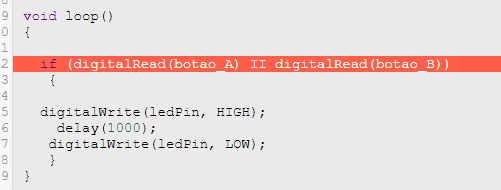
**7.1-OPERADORES LÓGICOS**

**( OR-OU-||)**

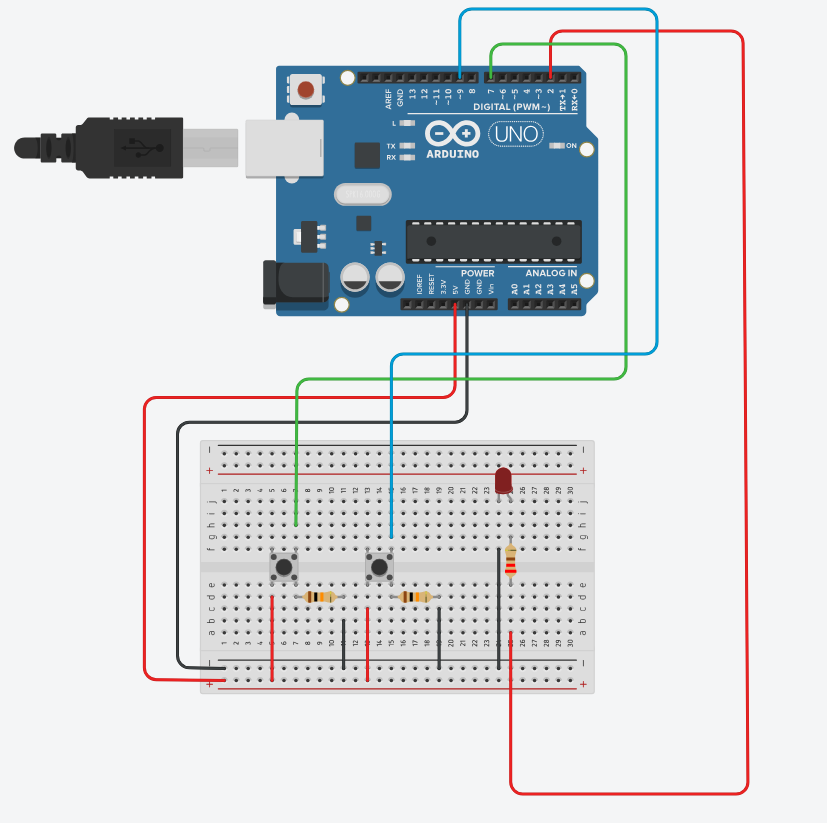
Quando temos que trabalhar com operadores lógicos,, devemos ter em mente que duas situações serão analisadas e será tomada uma decisão, realizada uma instrução.No caso do operador **OU,** que em inglês é **OR,** sua sintaxe é **||.**Mas, veja que interessante, vários cursos e professores pelos quais passei, nunca haviam me ensinado onde se localizava esta barra no teclado.Vejamos na prática então, pois de nada adianta saber codificar e não se localizar no teclado.Este símbolo **|** fica do lado esquerdo da letra Z, então é só segurar o SHIFT e digitar duas vezes que está pronto.

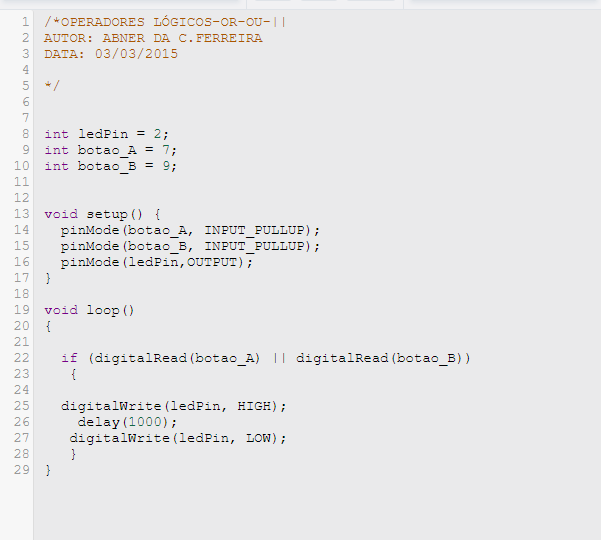


E nem pense em substituir o || pela letra I maiúscula que não dará certo.Observe o exemplo abaixo:

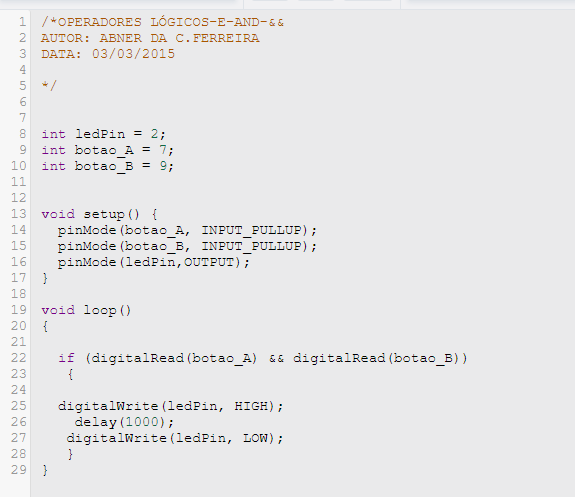


Problema:Foi solicitado para que montássemos um circuito com um led e que este tenha dois botões.Quando for pressionado botão A **OU** o botão B, o led acenderá.Vamos ao código:





**( AND-E-&&)**

Problema:Foi solicitado para que montássemos um circuito com um led (usaremos o mesmo circuito), e que este tenha dois botões.Quando for pressionado botão A **E** o botão B, o led acenderá.Vamos ao código:

Veja que o circuito é o mesmo, só teremos uma dificuldade aqui no simulador, não teremos como testar apertando os dois botões ao mesmo tempo, mas, se você realizar esta montagem, dará certo.

**-Estruturas de Decisão/Seleção**

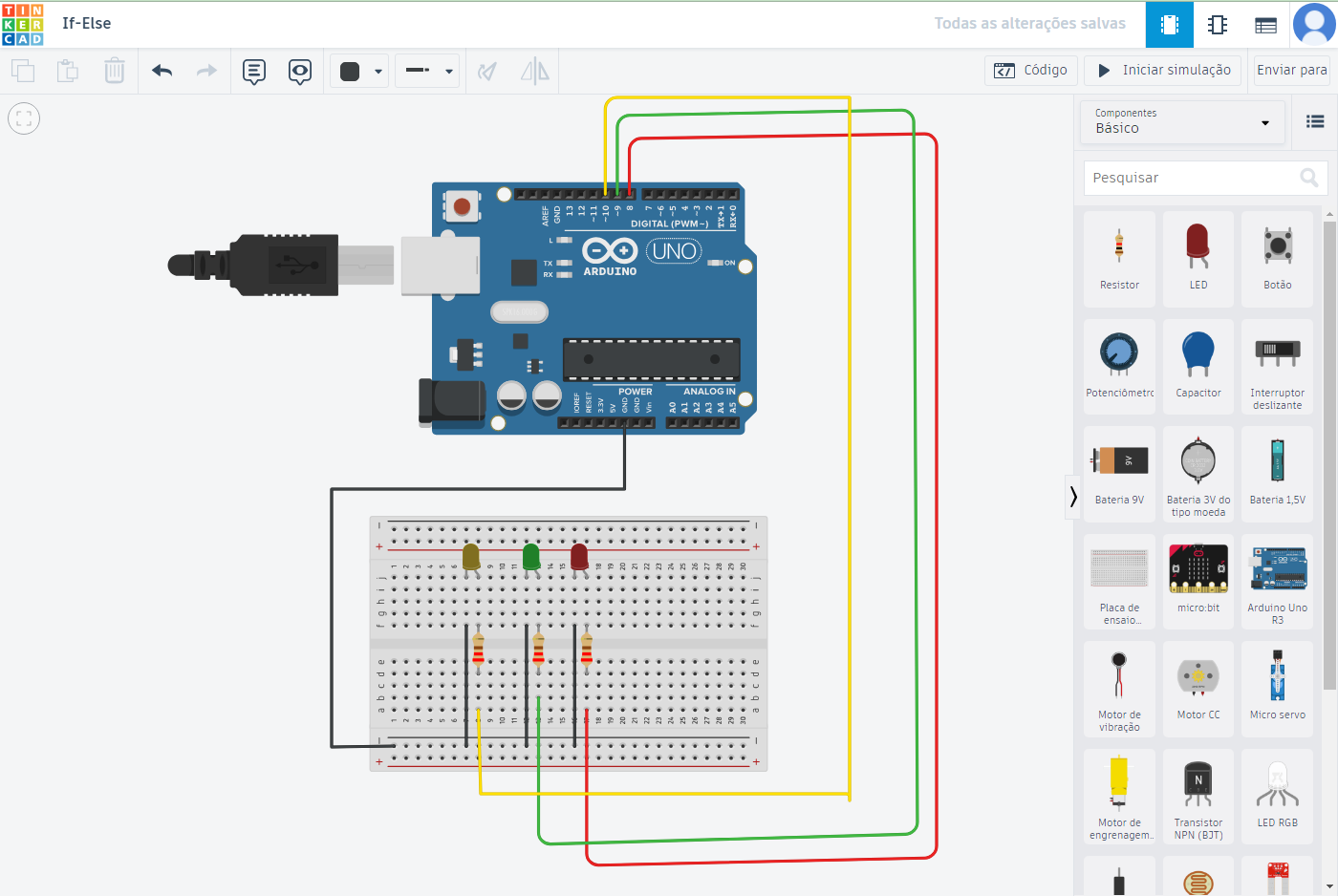
As estruturas de decisão são comandos que avaliam condições lógicas, comparando e validando resultados.Temos as seguintes estruturas if-else e switch-case.

Se e senão **(IF/ELSE)**

Problema: Preciso acender três Leds; um amarelo, um verde e um vermelho, digitando no teclado 1 para acender o led amarelo, 2 para acender o led verd e 3 para acender o led vermelho.

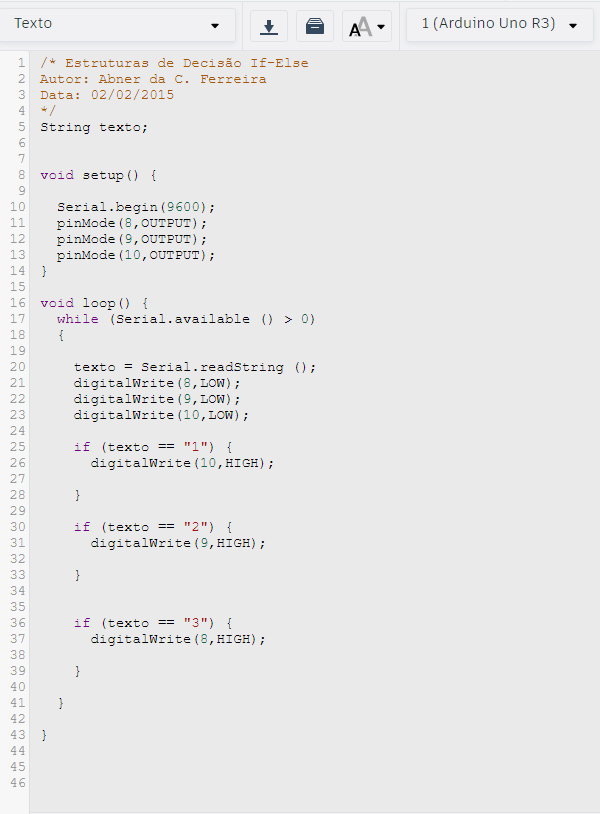
Se, nenhum desses três números forem digitados, os leds deverão continuar apagados.

Vamos montar o circuito abaixo utilizando o TinkerCad.



Vamos utilizar resistores de 220 ohms, para não queimar os leds.

Vejamos o código:



Já no código abaixo utilizamos o comando Els, que significa Senão. Se não acontecer o que está descrito no bloco do “if”, deverá acontecer o que está dentro do “else”.

Exemplo: Se o número não for par, ele só poderá ser ímpar.

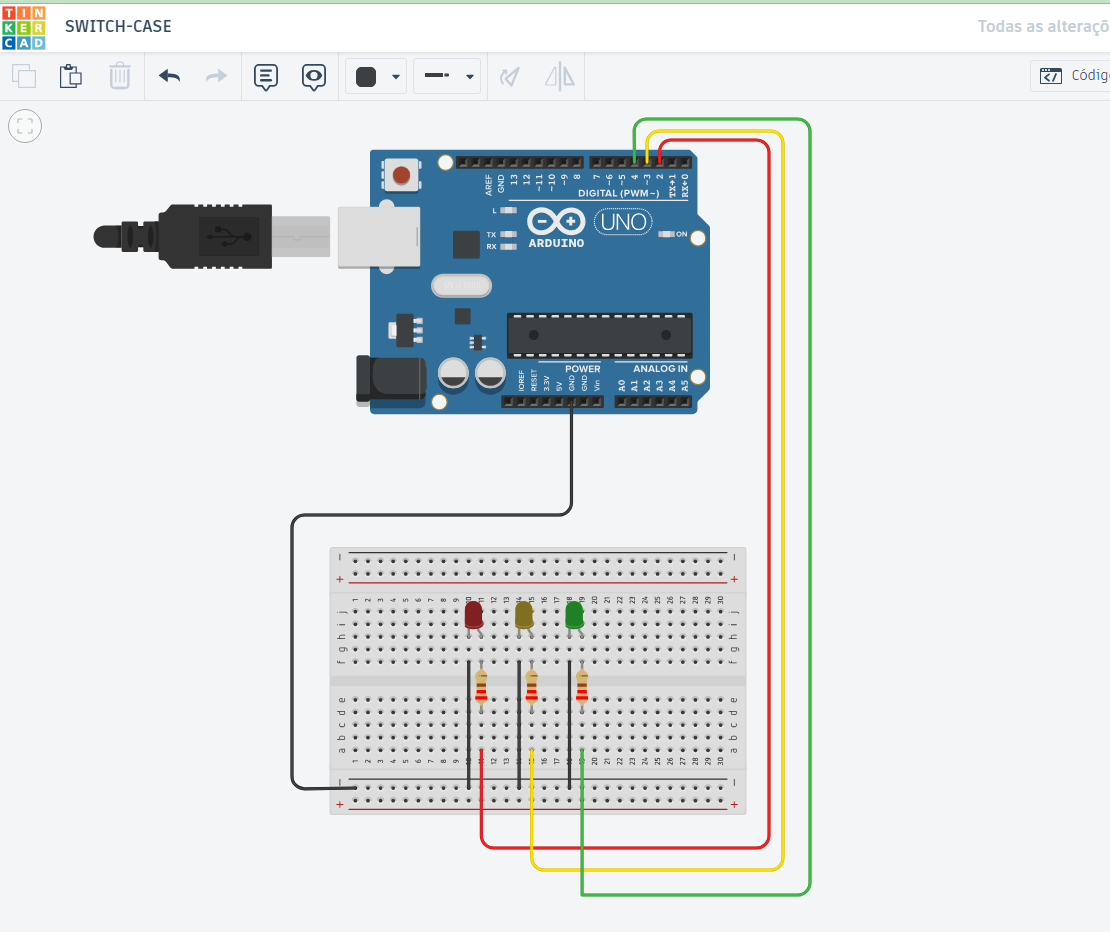


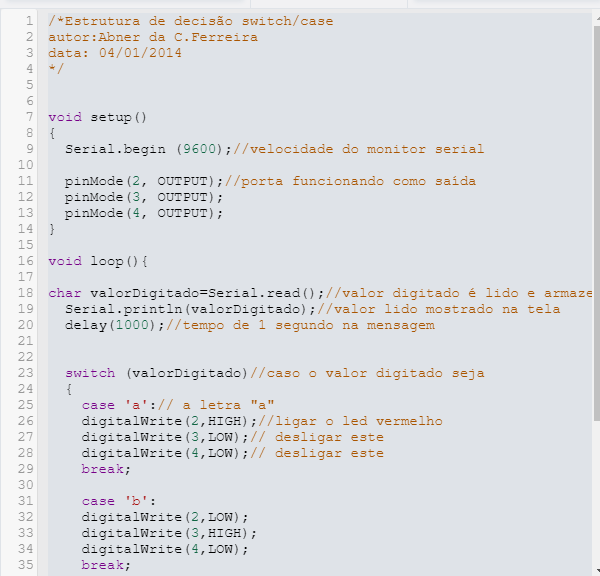
Escolha-Caso **(SWITCH/CASE)**

Quando começamos a utilizar muito repetidamente a estrutura d “if”, devemos repensar e utilizar a estrutura SWITCH-CASE ,pois além de ser mais prática, torna o código mais limpo e menor.

Problema: Quero acender 3 leds, digitando no teclado do meu computador.O primeiro Led será vermelho, o segundo amarelo e o terceiro será verde.Os códigos que que vou digitar são as letras “a”,”b”, e “c” respectivamente, ou seja, do tipo **char**.

Monte o circuito abaixo e vamos entender a estrutura Switch-case.





Esta estrutura é muito utilizada na programação como MENU de Opções.

Quando não é digitado nenhuma das opções, podemos se quisermos digitar **default.**No caso acima, quando nada é digitado aparece na tela a mensagem” digite o codigo correto para acender os leds”.

1. **Módulo -Estruturas de Repetição**

As estruturas de repetição são comandos que um bloco de instruções seja executado várias vezes até que a condição pré-estabelecida seja satisfeita.No jargão dos programadores, estas estrtuturas são chamados de loops ou laços de repetição.Temos as seguintes estruturas for, while, do-while.

**For (Para/Faça)**

Este comando deve ser usado quando conhecemos o número exato de repetições. A variável utilizada neste caso deve ser do tipo **inteiro** ou **literal.** A **condição** pré-estabelecida é validada no ínicio de cada loop, enquanto a condição for verdadeira as instruções dentro do laço serão executadas, no momento em que esta condição for falsa, o laço será encerrado.

|  |
| --- |
| No comando **for** usaremos o incremento (++) ou o decremento (--), que é somente somar 1 ou subtrair 1 da variável.  Exemplos de:  incremento i++ ou i=i+1 ou n= n+1  decremento i - - ou i= i-1 ou n=n-1 |

Antes de irmos para o exemplo, não tem como falar do comando **FOR** sem falar de **ARRAYS.**

**O que são Arrays?**

Arrays são vários valores dentro de uma única variável. Na verdade é uma **Matriz**,também conhecido como **Vetor**, com valores(dados) que podem ser de um único tipo ou de vários tipos string, boolean, integer, float, etc .Estes dados guardados no array podem ser acessados através de um índice.

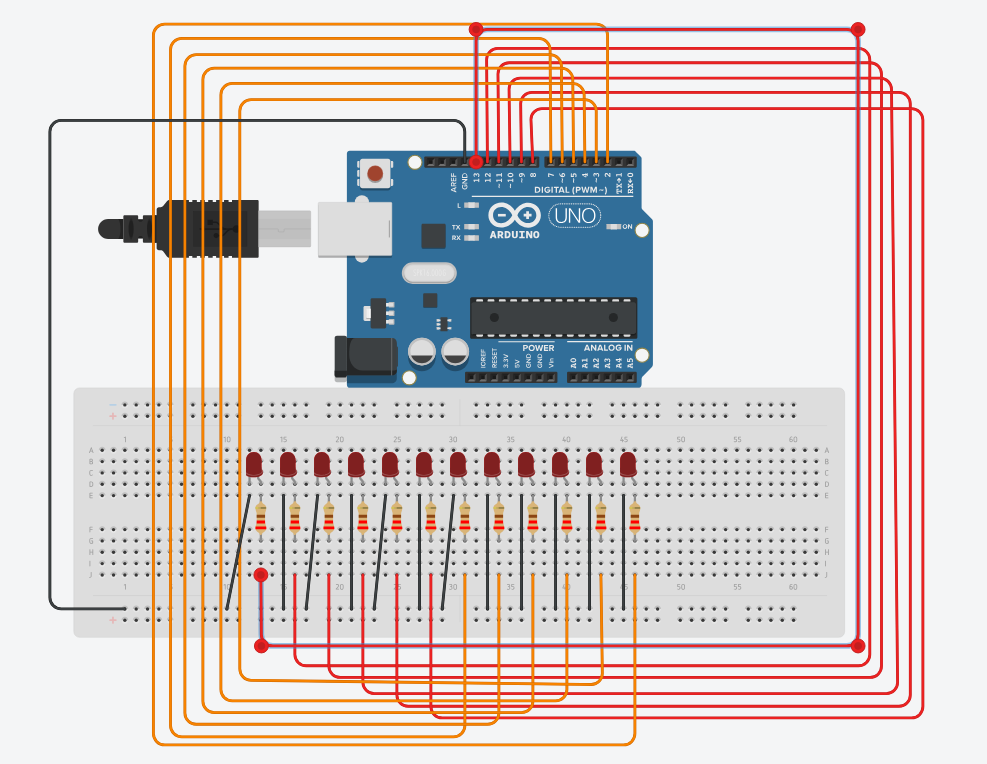
No exemplo abaixo preciso guardar 5 valores do tipo inteiro no **array media.**No array é obrigatório o uso de colchetes [ ].Já para declarar os valores, estes devem estar dentro de chaves **{ }.**

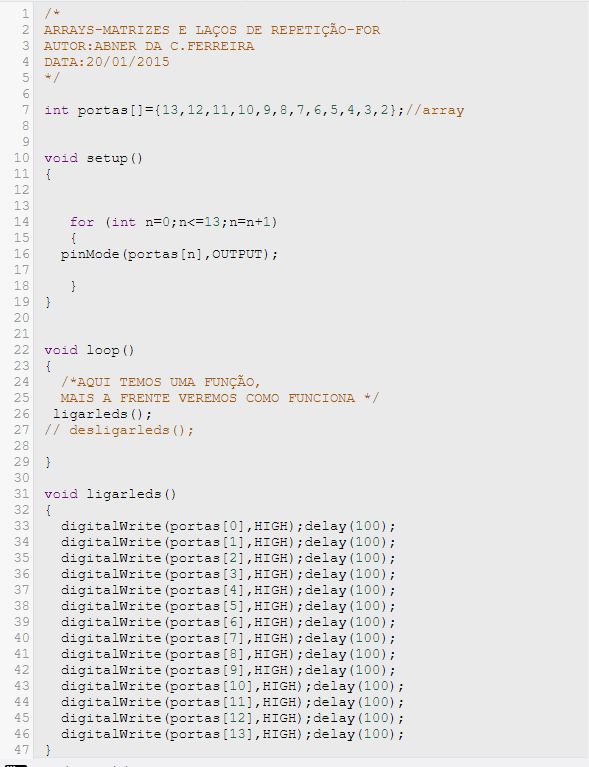
int media [ 5 ] = { 40,10,8,9,87 };

Agora, você não é obrigado a declarar oa quantidade de valores dentro dos colchetes, pode deixar em branco que também funciona.Observe.

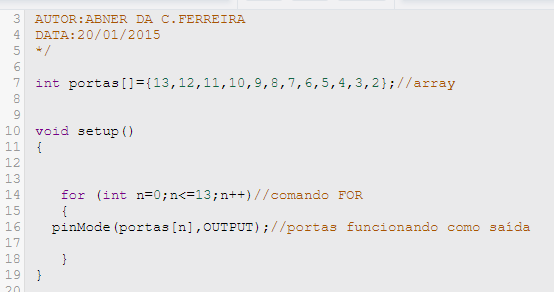
int media [ ] = { 40,10,8,9,87 };

Vejamos o exemplo abaixo

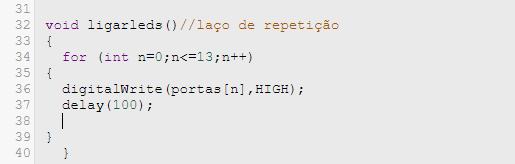




Observe que na linha 14 do código, o n=n+1 foi substituído por n++ que é a mesma coisa, aliás é bem mais prático.Funciona da mesma forma.



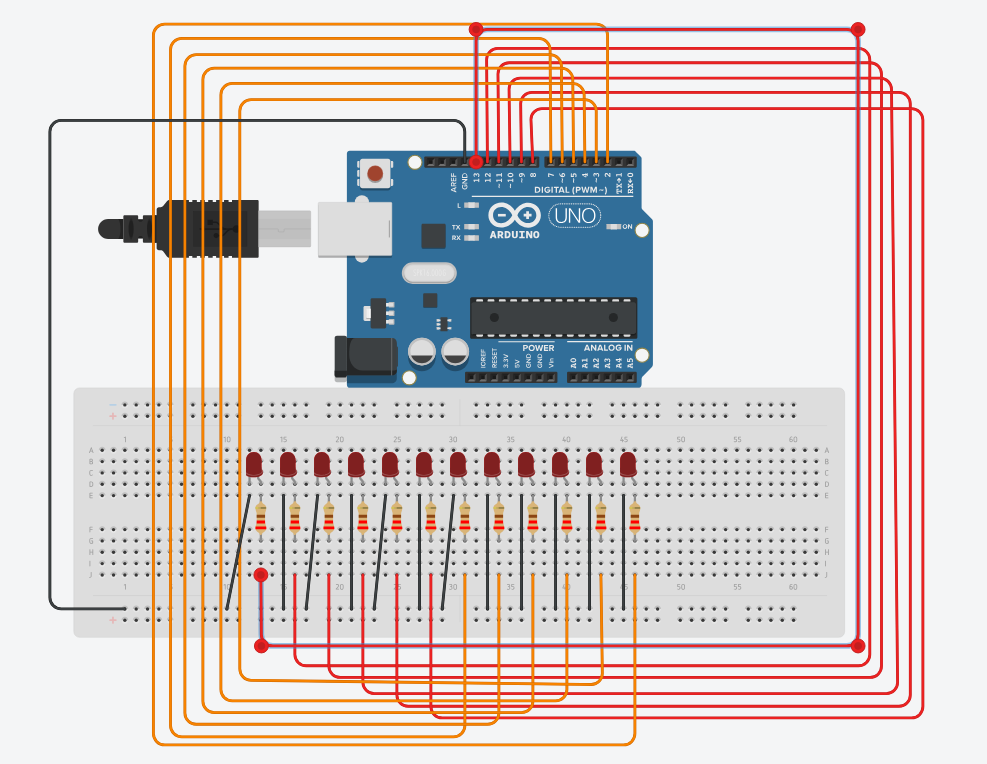
Veja que se eu souber utilizar o comando **FOR,** posso reduzir e deixar meu código mais limpo.Deletei da linha 33 até a linha 43 e substitui pelo comando abaixo.O resultado será o mesmo.



Ainda no comando **FOR**, e ainda falando de **ARRAYS**,será se conseguimos reproduzir a sinalização de uma pista de aeroporto utilizando o exemplo de Montagem acima?Como ficaria o código?



Problema: Utilizando 12 leds apenas, simule uma pista de aterrisagem de aeroporto, que acende e apaga sem parar.





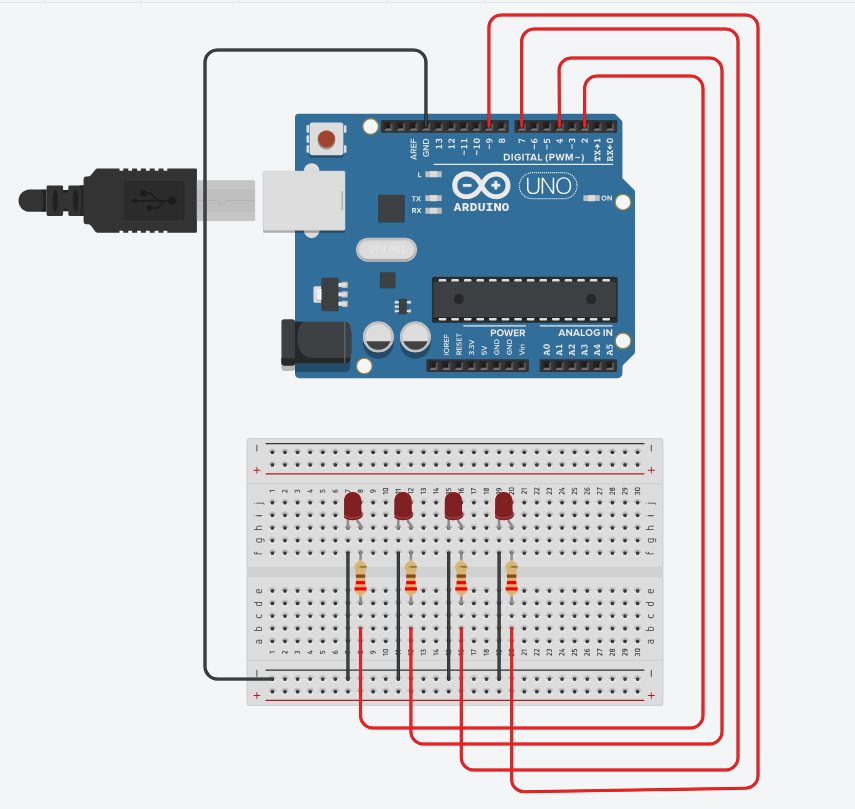
**O que é Identação?**

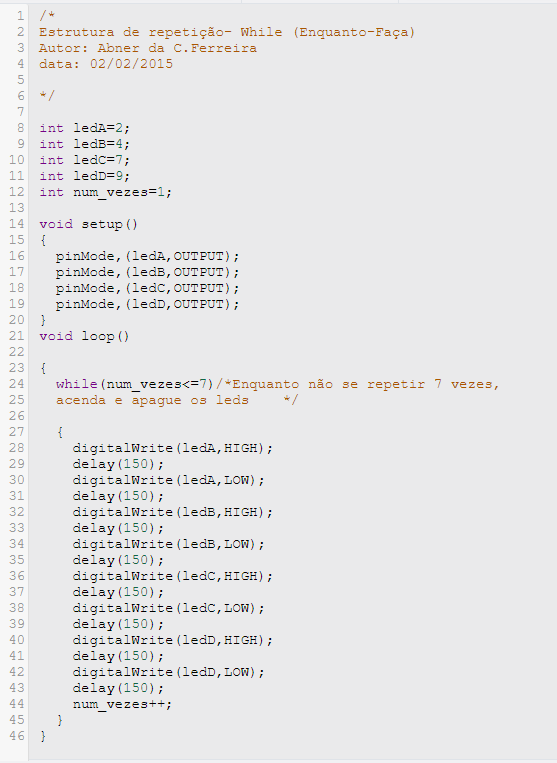


**While (Enquanto/Faça)**

Este comando deve ser usado quando **não conhecemos** o número exato de repetições. **Primeiro ele verifica a** condição, **depois ele faz** o que foi programado.

Vejamos o exemplo abaixo





No código acima, **Enquanto** a condição (Serial.available()>0) for maior que zero, ou seja, existir no teclado números maiores que zero, mas que sejam 1,2 ou 3 , ele executa o bloco de instruções abaixo.

**While-1º avalia a condição, depois executa as instruções.**

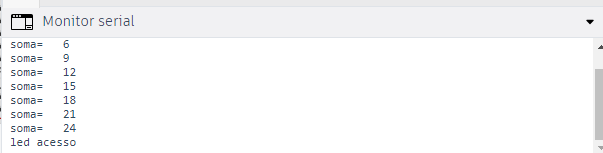
Este comando é interesante para a validação de senha num sistema, pois **enquanto** o usuário não digitar a senha correta “ 1234” vai ficar se repetindo na tela “ senha incorreta, digite novamente”.

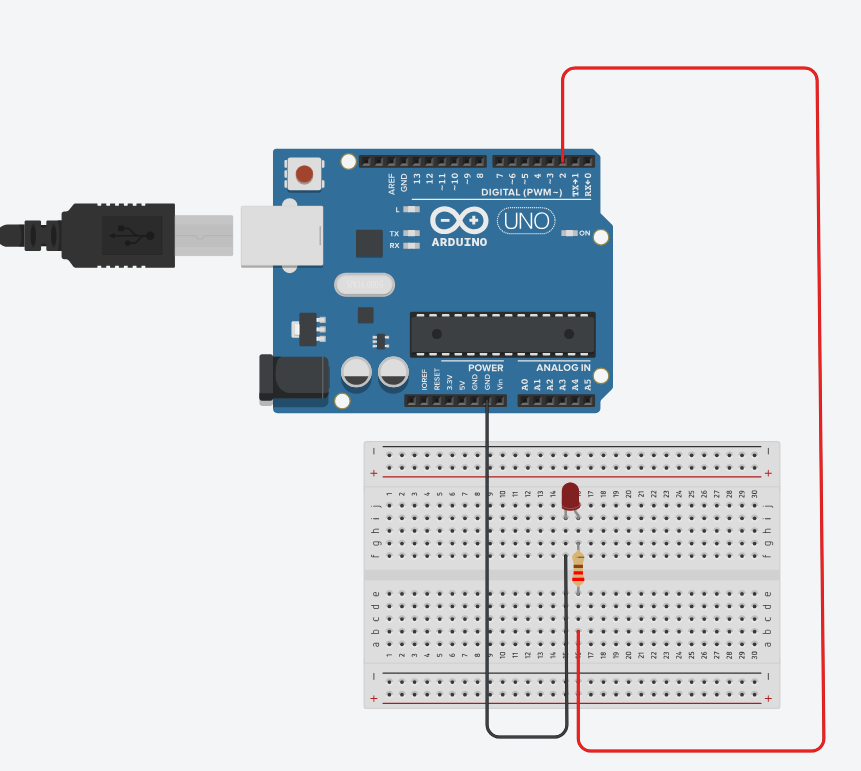
**Do-While (Faça/Enquanto)**

Este comando deve ser usado para testar a **condição de validação** do laço apenas **no final** do comando, realizando as instruções dentro do laço pelo menos uma vez. **Primeiro ele faz** o que foi programado, **depois ele verifica** a condição.

Problema:Pretende-se realizar uma contagem de 3 em 3, iniciando do 0.Se esta contagem der 24, um led deve acender.







**9-Funções na programação**

**CONCLUSÃO**

Neste Projeto Integrado Multidisciplinar VIII, foram realizadas diversas atividades relacionadas com as disciplinas de Desenvolvimento de Software para Internet, Programação Orientada a Objetos II e Tópicos Especiais de Programação Orientada a Objetos.

Com o auxílio da disciplina de Programação Orientada a Objetos II foi possível realizar a conexão entre os dados digitados pelo usuário na interface com o banco de dados, ou seja, conectar toda a parte de front-end (interface) com o back-end (banco de dados).

Com o auxílio da disciplina Desenvolvimento de Software para Internet foi possível desenvolver o método CRUD para o usuário, que nada mais é do que Cadastrar, Visualizar, Editar e Excluir dados que se encontram armazenados no banco de dados.

Já a disciplina Tópicos Especiais de Programação Orientada a Objetos, auxiliou na elaboração de um protótipo não funcional de um aplicativo Android, onde o usuário pode se cadastrar, visualizar, editar e excluir seus dados, que no caso também ficarão armazenados no banco de dados SQLite e pode ser acessado do celular.

Portanto, o desenvolvimento da aplicação de teleatendimento médico para consulta de pacientes Connect Virtual Health é totalmente possível e viável, pois a demanda pós-pandemia ainda é grande no setor da saúde, mitigando assim os custos para os convênios médicos. Em relação aos usuários é muito interessante, pois economiza tempo de locomoção e dinheiro aos mesmos, possibilitando assim uma consulta médica no conforto de sua residência. Já em relação aos médicos, facilitará bastante o atendimento aos pacientes, visto que por ser virtual, reduz a exposição de ambos, paciente-médico, a quaisquer tipos de vírus/bactérias, não se esquecendo também da economia de tempo/dinheiro do profissional que pode atender o paciente de forma remota em qualquer lugar.

**REFERÊNCIAS**

AMARAL, Wagner Machado do. **Android Studio | SQLITE | Criar BD e listar dados em Listview.** Youtube. Ano 2020.

**Disponível em:** https:/[/www](http://www.youtube.com/watch?v=hRptcOy1g2M).[youtube.com/watch?v=hRptcOy1g2M](http://www.youtube.com/watch?v=hRptcOy1g2M)

**Acesso em:** 13/11/2022.

BOCARD, Taysa. **O que são aplicativos? Definição da desenvolvedora Usemobile.**

Ano 2021.

**Disponível em:** https://usemobile.com.br/aplicativo-movel/

**Acesso em:** 21/11/2022.

BR, Programador. **Firebase - O que é e para que serve? Ep. 1.** Youtube. Ano 2019.

**Disponível em:** https:/[/www](http://www.youtube.com/watch?v=hx2k0XZ_2Cw).[youtube.com/watch?v=hx2k0XZ\_2Cw](http://www.youtube.com/watch?v=hx2k0XZ_2Cw)

**Acesso em:** 22/11/2022.

CONTENT, Redator Rock. **Conheça Firebase: a ferramenta de desenvolvimento e análise de aplicativos mobile.** Ano 2019.

**Disponível em:** https://rockcontent.com/br/blog/firebase/

**Acesso em:** 22/11/2022.

DEITEL. **C#: Como Programar**. 1°Edição. Editora: Pearson, 2003. E-book. ISBN 9788534614597.

**Disponível em:** https://plataforma.bvirtual.com.br/Leitor/Publicacao/282/epub/0

**Acesso em:** 10/11/2022.

FERREIRA, Valdir. **CRUD MVC USANDO BANCO DE DADOS \*\*\*\*SQL SERVER\*\*\***

**asp.net core / .NET 6 C#.** Youtube. Ano 2021.

**Disponível em:** https:/[/www](http://www.youtube.com/watch?v=zr3QiQDZ0-k).[youtube.com/watch?v=zr3QiQDZ0-k](http://www.youtube.com/watch?v=zr3QiQDZ0-k)

**Acesso em:** 03/11/2022.

G, Ariane. **O que é CSS? Guia Básico para Iniciantes.** Ano 2022.

**Disponível em:** https:/[/www.h](http://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css)o[stinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css](http://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css)

**Acesso em:** 19/11/2022.

GUNJI, José Cassiano Grassi. **Livro Texto Unidade I, II, III e IV- Tópicos Especiais de Programação Orientada a Objetos** - São Paulo: Editora Sol, 2020.

**Disponível em:** https://ava.ead.unip.br/

**Acesso em:** 06/11/2022.

HOSTGATOR. **C-Sharp: Entenda a linguagem de programação multiparadigma.** Ano 2021.

**Disponível em:** https:/[/www.h](http://www.hostgator.com.br/blog/csharp-linguagem-de-programacao-)o[stgator.com.br/blog/csharp-linguagem-de-programacao-](http://www.hostgator.com.br/blog/csharp-linguagem-de-programacao-) multiparadigma/

**Acesso em:** 14/11/2022.

IMPACTA, Redação. **Você sabe o que é Visual Studio?.** Ano 2017.

**Disponível em:** https:/[/www.im](http://www.impacta.com.br/blog/voce-sabe-o-que-e-visual-studio/)p[acta.com.br/blog/voce-sabe-o-que-e-visual-studio/](http://www.impacta.com.br/blog/voce-sabe-o-que-e-visual-studio/)

**Acesso em:** 19/11/2022.

MACHADO, Emerson. **CURSO COMPLETO DE ANDROID STUDIO! INSTALAÇÃO E CONFIGURAÇÃO (#01).** Youtube. Ano 2022.

**Disponível em:** https:/[/www](http://www.youtube.com/watch?v=PZMgUGcWXPc&t=468s).[youtube.com/watch?v=PZMgUGcWXPc&t=468s](http://www.youtube.com/watch?v=PZMgUGcWXPc&t=468s)

**Acesso em:** 05/11/2022.

MANZANO, José Augusto N G. **Programação de Computadores com C#**. 1°Edição. Editora: Saraiva, 2014. E-book. ISBN 9788536519470.

**Disponível em:** https://integrada.minhabiblioteca.com.br/#/books/9788536519470/

**Acesso em:** 10/11/2022.

MARINHO, Salatiel Luz**. Livro Texto Unidade I, II, III e IV- Desenvolvimento de Software para Internet**- São Paulo: Editora Sol, 2022.

**Disponível em:** https://ava.ead.unip.br/

**Acesso em:** 04/11/2022.

MARINHO, Salatiel Luz**. Livro Texto Unidade I, II, III e IV- Programação Orientada a Objetos II** - São Paulo: Editora Sol, 2020.

**Disponível em:** https://ava.ead.unip.br/

**Acesso em:** 05/11/2022.

MARQUES, Rafael. **O que é HTML? Entenda de forma descomplicada**. Ano 2022.

**Disponível em:** https:/[/www.h](http://www.homehost.com.br/blog/tutoriais/o-que-e-html/)o[mehost.com.br/blog/tutoriais/o-que-e-html/](http://www.homehost.com.br/blog/tutoriais/o-que-e-html/)

**Acesso em:** 19/11/2022.

MARTINS, Vinicius. **ASP.NET: o que é e como usar para criar páginas dinâmicas?.**

Ano 2021.

**Disponível em:** https://blog.betrybe.com/framework-de-programacao/asp-net-o-que-e/

**Acesso em:** 15/11/2022.

NOLETO, Cairo. **CRUD: as 4 operações básicas do banco de dados!.** Ano 2021. **Disponível em:** https://blog.betrybe.com/tecnologia/crud-operacoes-basicas/ **Acesso em:** 19/11/2022.

NOLETO, Cairo. **Framework: o que é, como ele funciona e para que serve?.** Ano 2020.

**Disponível em:** https://blog.betrybe.com/

**Acesso em:** 14/11/2022.

Oracle. **O que é um Banco de Dados?.** Ano 2022.

**Disponível em:** https:/[/www.oracle.co](http://www.oracle.com/br/database/what-is-database/)m[/br/database/what-is-database/](http://www.oracle.com/br/database/what-is-database/)

**Acesso em:** 18/11/2022.

ROCHA, Leonardo. **Cadastro simples com persistência de dados - Android Studio - [parte 1].** Youtube. Ano 2021.

**Disponível em:** https:/[/www](http://www.youtube.com/watch?v=3sAjgk6aAyI).[youtube.com/watch?v=3sAjgk6aAyI](http://www.youtube.com/watch?v=3sAjgk6aAyI)

**Acesso em:** 13/11/2022.

ROVEDA, Ugo. **JAVASCRIPT: O QUE É, PARA QUE SERVE E COMO FUNCIONA O JS?.** Ano 2021.

**Disponível em:** https://kenzie.com.br/blog/javascript/

**Acesso em:** 19/11/2022.

SILVA, Eduardo. **Firebase: o que é e quando usar no desenvolvimento mobile?.** Ano 2020.

**Disponível em:** https://blog.geekhunter.com.br/firebase-o-que-e-e-quando-usar-no- desenvolvimento-mobile/

**Acesso em:** 22/11/2022.

TIAGO. **Android Studio: O Que É E Como Desenvolver Apps Nele.** Ano 2020.

**Disponível em:** https://mundodevops.com/blog/android-studio/

**Acesso em:** 21/11/2022.

VIEIRA, Danielle. **SQLite: o que é, como funciona e qual é a diferença entre o MySQL.** Ano 2021.

**Disponível em:** https:/[/www.h](http://www.hostgator.com.br/blog/sqlite-o-que-e-como-funciona-e-qual-)o[stgator.com.br/blog/sqlite-o-que-e-como-funciona-e-qual-](http://www.hostgator.com.br/blog/sqlite-o-que-e-como-funciona-e-qual-) e-a-diferenca-entre-o-mysql/

**Acesso em:** 23/11/2022.

ZUCHER, Vitor. **O que é padrão MVC? Entenda arquitetura de softwares!.** Ano 2020.

**Disponível em:** https:/[/www.le](http://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc)w[agon.com/pt-BR/blog/o-que-e-padrao-mvc](http://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc)

**Acesso em:** 16/11/2022.

|  |
| --- |
| história da computação:  <https://edisciplinas.usp.br/pluginfile.php/5521150/mod_resource/content/2/%20Histo%CC%81ria%20da%20Computac%CC%A7a%CC%83o%20-%202020.pdf>  <https://igeo.ufrgs.br/museudetopografia/images/acervo/artigos/Histria_do_baco.pdf>  ada lovelace  <http://www.ime.unicamp.br/~apmat/ada-lovelace/>  <https://universosquanticos.wordpress.com/2017/07/08/logica-paraconsistente-maquina-de-turing/>  <https://docs.arduino.cc/built-in-examples/digital/Button#code>  <https://cdn.arduino.cc/reference/pt/language/variables/data-types/char/>  <https://docs.arduino.cc/learn/starting-guide/getting-started-arduino>  <https://materialpublic.imd.ufrn.br/curso/disciplina/2/61/1/4>  <https://www.usinainfo.com.br/blog/o-que-e-arduino/>  <https://www.makerhero.com/blog/monitore-sua-planta-usando-arduino/>  <https://www.youtube.com/watch?v=qTta8m3CTEo>  <https://wiki.sj.ifsc.edu.br/images/f/f9/Sistemas_de_numeracao_uberlandia.pdf> |