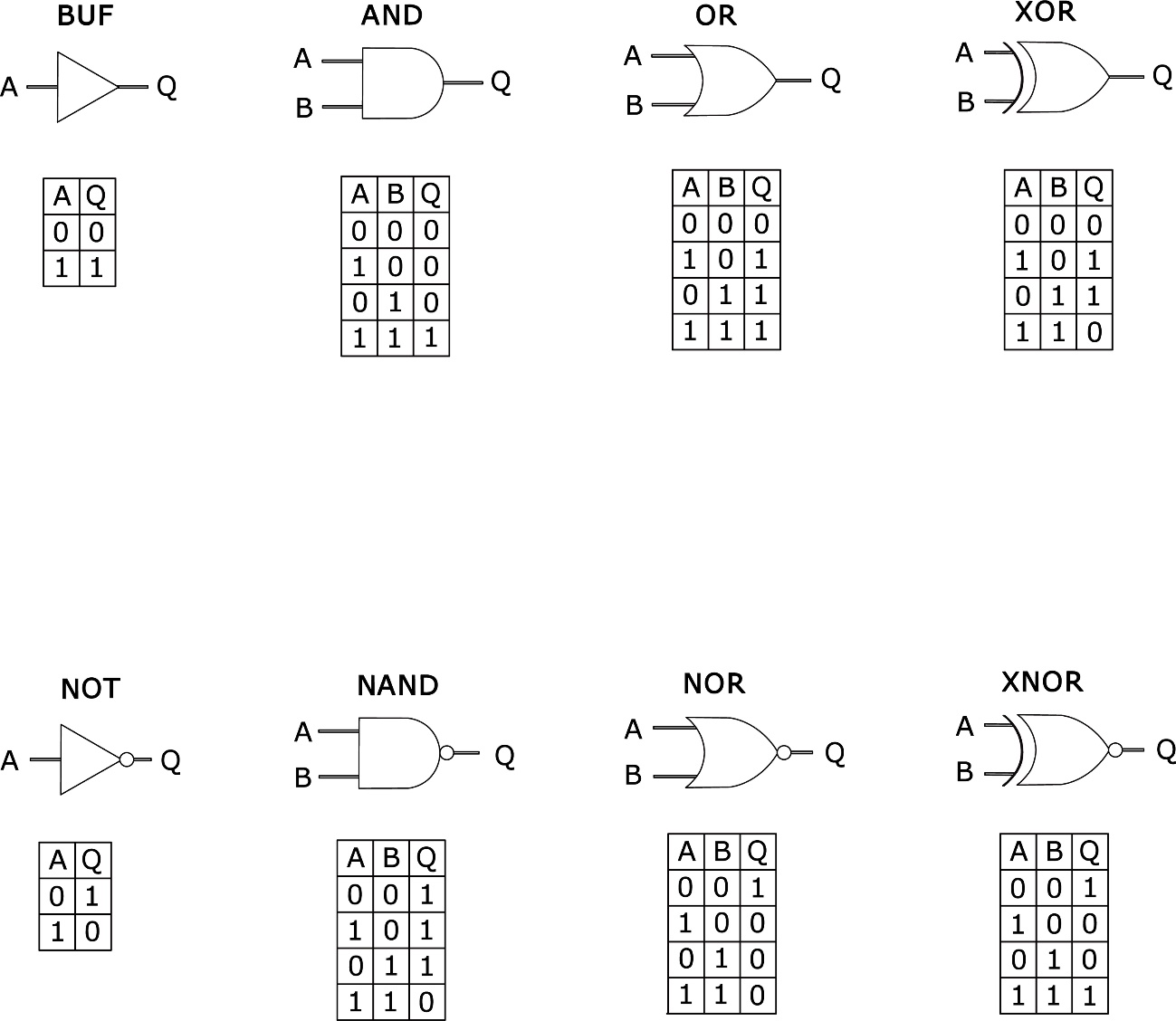
**Introdução da Unidade**



**Objetivos da Unidade**

Ao longo desta Unidade, você irá:

* Esclarecer a simbologia de operações lógicas;
* Discutir as regras da álgebra Booleana;
* Analisar os conceitos de porta lógica.

Você sabe onde podemos encontrar conceitos de álgebra booleana, conhecida como álgebra digital? Se olharmos ao nosso redor, com certeza você encontrará dispositivos que possuam uma placa de circuito integrado – é um circuito eletrônico que contém uma miniaturização de diversos componentes eletrônicos, do tipo transistores, diodos, resistores e capacitores, armazenados em um chip de silício – em que este conceito é fortemente aplicado. Veja como exemplo: seu relógio, smartphones, notebooks ou computadores, sistemas de segurança, chuveiros e muitos outros.

As placas de circuitos impressos são feitas em uma placa de cobre que serve de base para a transferência do desenho de um circuito. Esse desenho contém os terminais e as pistas de interligação dos componentes. Essa placa e esses componentes quase nunca são visualizados, mas têm seu papel para o bom funcionamento do circuito em questão. Um exemplo disso seria um simples rádio.

Essas placas de circuitos são de vital importância para o nosso cotidiano. Para entendermos as definições e os conceitos para esses e outros dispositivos dentro de lógica digital, precisaremos desenvolver a fundamental competência para seu aprendizado, que é: conhecer e compreender os princípios de arquitetura e organização de computadores. Assim, vamos destacar os objetivos de aprendizagem que visamos alcançar com os estudos da presente unidade. São eles:

* Conhecer e apresentar aos estudantes os conceitos e aplicações de Álgebra Booleana.
* Conhecer e correlacionar a determinação e simplificação de expressões lógicas.
* Conhecer e apresentar portas lógicas: conceitos, símbolos e tipos.
* Conhecer e despertar a reflexão da aplicação de circuitos: digitais, combinacionais e sequenciais.

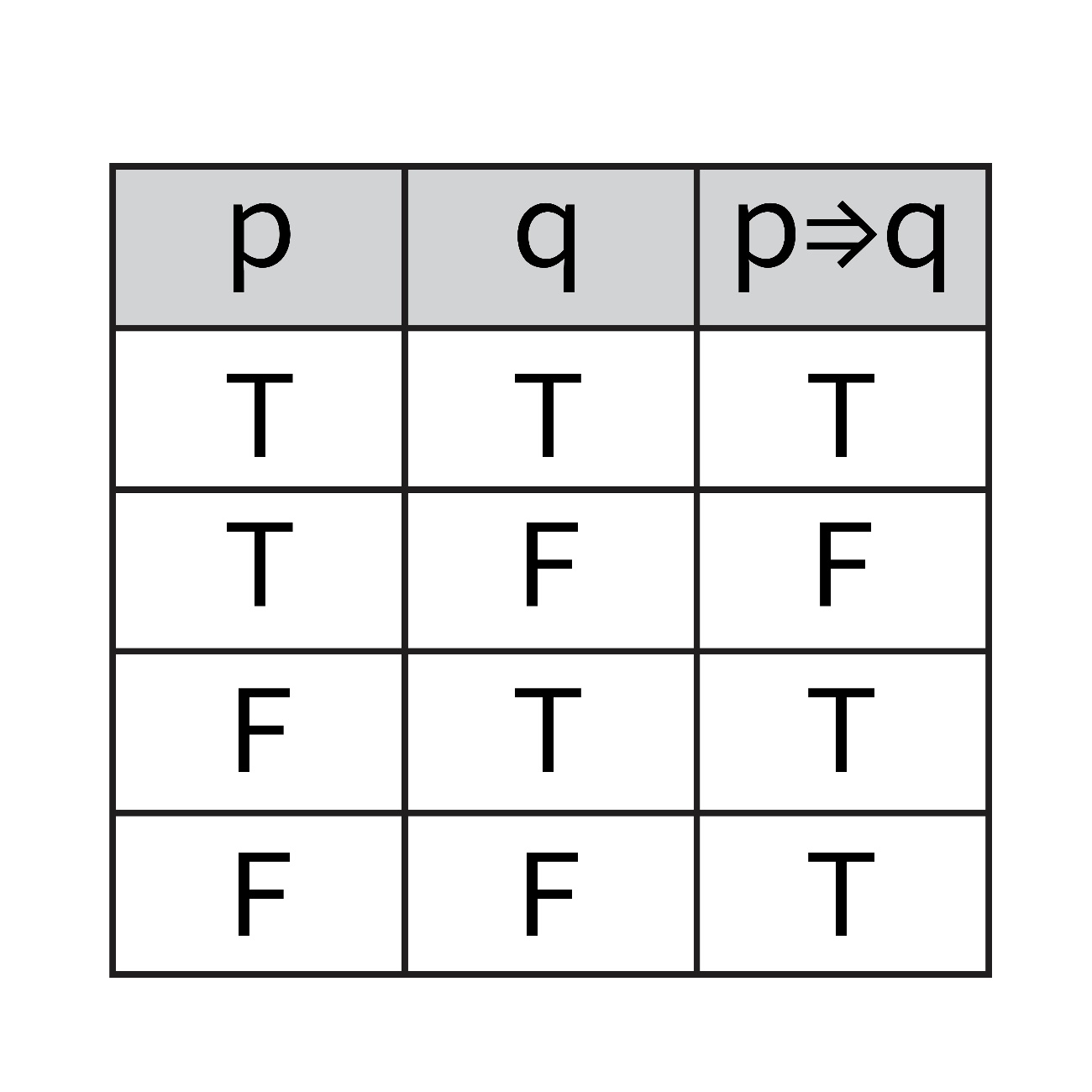
Você encontrará tudo isso em seu material didático, demonstrado de uma forma objetiva, simples e exemplificada, com exercícios e sugestões.

Para isso, aproximamos a teoria da prática apresentando uma possível empresa em que você trabalha, cuja atividade-fim é a fabricação de placas de circuitos impressos (PCI). Você atua como engenheiro em um processo de criação de PCIs. As placas são fabricadas de acordo com normas e procedimentos passo a passo, nos quais cada placa de circuito impresso é feita de acordo com seu diagrama, formando, assim, trilhas ou pistas. Essas trilhas são responsáveis pela condução de corrente elétrica pelos componentes eletrônicos que podem compor a placa. Esses componentes, com o uso da corrente elétrica, fazem com que a PCI funcione em sua totalidade.

Ao considerar este contexto, vamos associar a teoria e a sua aplicação prática. Está preparado para iniciarmos nosso aprendizado?

Sigamos em frente!

**Introdução da Aula**



**Qual é o foco da aula?**

Na primeira aula desta unidade, aprenderemos operações lógicas básicas conhecidas como NOT, AND e OR.

**Objetivos gerais de aprendizagem**

Ao longo desta aula, você irá:

* Identificar a simbologia de operações lógicas;
* Compreender as tabelas-verdade e porta lógica;
* Interpretar os diagramas interruptores.

Situação-problema

Na unidade anterior, aprendemos como realizar conversões entre bases do sistema de numeração binário, decimal, octal e hexadecimal. Agora que entendemos bem essas conversões e conceitos, principalmente os do sistema de numeração binário, passaremos a conhecer e aprender sobre álgebra booleana, a qual é muito utilizada no meio computacional e eletroeletrônico. Aqui, aprenderemos operações lógicas básicas conhecidas como NOT, AND e OR. A partir delas iremos criar as tabelas verdadeiras para cada caso. Essas tabelas demonstram logicamente o que está entrando de informação e qual será o resultado de saída, sempre verdadeiro ou falso.

Um exemplo disso seria uma esteira rolante de caixa de supermercado. Ela rola quando nenhum objeto chega no sensor de chegada. Quando chega no sensor de chegada é enviado um sinal lógico (verdadeiro ou falso) e a esteira para de rodar.

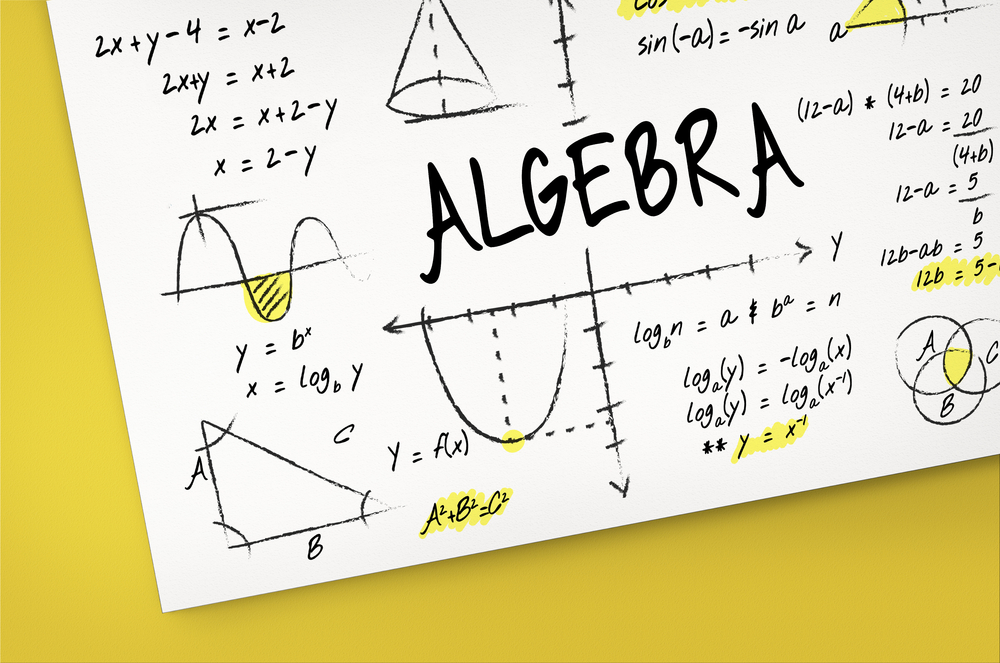
Para seguir com nossos estudos e associar a teoria com a prática, vamos desenvolver um projeto que utiliza um diagrama de lógica digital e conceitos de álgebra booleana. Para tal, será necessário que você considere as descrições a seguir:

O primeiro passo é a criação de um diagrama do circuito de um interruptor. Um diagrama de circuito é o primeiro passo para se chegar ao processo de criação de uma placa de circuito impresso. Um interruptor é um dispositivo que é ligado a um ponto de um circuito, podendo ter um de dois estados, sendo: ligado ou desligado (ou, ainda, estar aberto ou fechado). No estado "fechado" o interruptor permitirá que a corrente elétrica passe através do ponto, permitindo, assim, que uma lâmpada seja acesa, por exemplo. Já no estado "aberto" não se permite a passagem da corrente elétrica pelo ponto, fazendo com que a lâmpada fique desligada. Você deverá usar para o estado "fechado" o valor 1 e para o estado "aberto" o valor 0, devendo o circuito ter dois interruptores (em paralelo). Depois de feito o diagrama em paralelo, faça o diagrama em série e sua respectiva tabela-verdade.

Com essa criação você estará aproximando os conceitos aprendidos com a prática profissional. Como iniciaremos nossos estudos com introdução à álgebra booleana, vamos entender as ideias e conhecer os conceitos aplicados a esse assunto e compreender o uso de operadores lógicos. Para que tenha um ótimo aproveitamento, sugiro que utilize seu material didático como guia e execute todas as pesquisas sugeridas para facilitar seu entendimento a esse aprendizado. Não se esqueça, também, dos outros materiais que estão à sua disposição, complementando seu aprendizado.

Desejo um bom estudo para você!

**Álgebra de Boole**



Caro estudante, você já sabe que estudaremos nesta aula uma introdução à álgebra de Boole.

Antes de falarmos de George Boole, considerado o pai e criador da álgebra booleana, vamos ver um pouco do histórico relacionado ao surgimento dos conceitos da álgebra de Boole.

Leibniz, antecessor aos estudos de Boole, foi responsável pelos conceitos básicos ao desenvolvimento da conhecida álgebra de Boole.

Veja um pouco da sua biografia.

**Gottfried Wilhelm Von Leibniz** - Leibniz nasceu na cidade de Leipzig (Alemanha), em 1º de julho de 1646 e faleceu em 14 de novembro de 1716. Ele entrou para a faculdade com apenas 15 anos de idade e fez os cursos de Direito, Matemática, Teologia e Filosofia. Nos seus 17 anos de idade, ele já possuía seu diploma de bacharel e com 20 anos de idade já se encontrava pronto para possuir o título de doutorado em direito, que lhe foi negado devido à sua pouca idade. Mas ele não parou por aí e foi para a Universidade de Altdorf receber seu diploma de doutor.

Leibniz é, hoje, considerado um dos últimos eruditos que possuía um conhecimento universal. Dentro da nossa área de estudo, ele propôs e contribuiu de muitas formas:

* Foi o primeiro estudioso a propor o uso de sistema binário.
* Paralelo a Newton, ele foi o criador do cálculo.
* Ele criou e demonstrou alguns conceitos de princípios de lógica (GONÇALVES, 2009, p. 4, grifo nosso):

1. *Todas as nossas ideias são formadas a partir de um pequeno número de ideias simples, que formam o alfabeto do pensamento humano.*
2. *Ideias complexas procedem dessas ideias simples por uma combinação uniforme e simétrica, análoga à multiplicação.*

*Esses conceitos de princípios de ideias com o tempo foram sendo aperfeiçoados*.

* Durante toda a sua vida, Leibniz possui apenas trabalhos não concluídos ou rascunhos sobre esses assuntos, não havendo publicado nada.
* Vários séculos depois de Leibniz, no início do século XX, a lógica formal requer uma negação unária e variáveis quantificadas sobre algum universo do qual estamos utilizando.

\_\_\_\_\_\_\_

**➕ Pesquise mais**

Veja a [Biografia](http://ecalculo.if.usp.br/historia/leibniz.htm)Completa de Leibniz, antecessor de George Boole.

\_\_\_\_\_\_\_

**George Boole** - Foi proveniente de uma família simples e modesta. Nasceu em 2 de novembro de 1815 e faleceu em 8 de dezembro de 1864. Com 12 anos de idade já havia aprendido o latim, sem ter qualquer formação acadêmica. Ele é considerado o gênio cujo trabalho deu origem à invenção dos computadores.

Boole ficava pensando em como se davam os processos de raciocínio dos humanos, presentes no dia a dia. Desse modo, notou que podia representá-los na forma de lógica formal e tentou colocar na forma de matemática o raciocínio lógico. Diante disso, utilizou como base os conceitos de Aristóteles e de Leibniz para formar a lógica clássica (GONÇALVES, 2009).

\_\_\_\_\_\_\_

**➕ Pesquise mais**

Veja a [Biografia](http://ecalculo.if.usp.br/historia/boole.htm)Completa de Boole.

\_\_\_\_\_\_\_

Principais ideias de Boole (GONÇALVES, 2009, p. 7):

Para corresponder a palavras, preposições ou frase, utilizava de símbolos algébricos p, q, r, x, y, e z.

Boole queria criar um sistema de operações algébricas do tipo soma, multiplicação e criação de um método para que pudesse resolver equações.

Seria necessário formular uma linguagem simbólica de pensamento para a Álgebra de Boole.

Para chegar em um resultado por essa linguagem, nunca se chegaria a resultados numéricos, mas sim a resultados de conclusão lógica.

Essa sua conclusão lógica recebeu o nome de ‘Álgebra do pensamento’.

\_\_\_\_\_\_\_

**💭 Reflita**

*“Boole é considerado um dos fundadores da ciência da computação, mesmo não existindo computadores em seus dias”* (GONÇALVES, 2009, p. 6, grifo nosso).

\_\_\_\_\_\_\_

**Álgebra de Boole** - Para sistemas digitais e circuitos digitais de computadores existem suas respectivas análises e projetos, dentro dos termos da matemática, porém utilizados dentro do conceito de Álgebra de Boole, e não de matemática sim*ples. Os princípios que são considerados básicos formam seu tratado denominado An Investigation of the Laws of Thought on Wich to Found Mathematical Theories of Logical and Probabilities*, do ano de 1854 (GONÇALVES, 2009). No ano de 1938, o assistente de pesquisas, *Claude Shannon*, do departamento de Engenharia Elétrica do MIT – *Massachusets Institute of Technology*–, sugeriu que a já conhecida álgebra booleana poderia ser usada para resolver problemas relativos a projeto de circuitos de comutação de relés (SHANNON, 1938).

\_\_\_\_\_\_\_

**📖 Vocabulário**

**Relés** – Um dispositivo eletromecânico, formado por um magneto móvel, que se deslocava unindo dois contatos metálicos, como uma chave. Verifique o [artigo](https://www.hardware.com.br/termos/rele-relay)Relés.

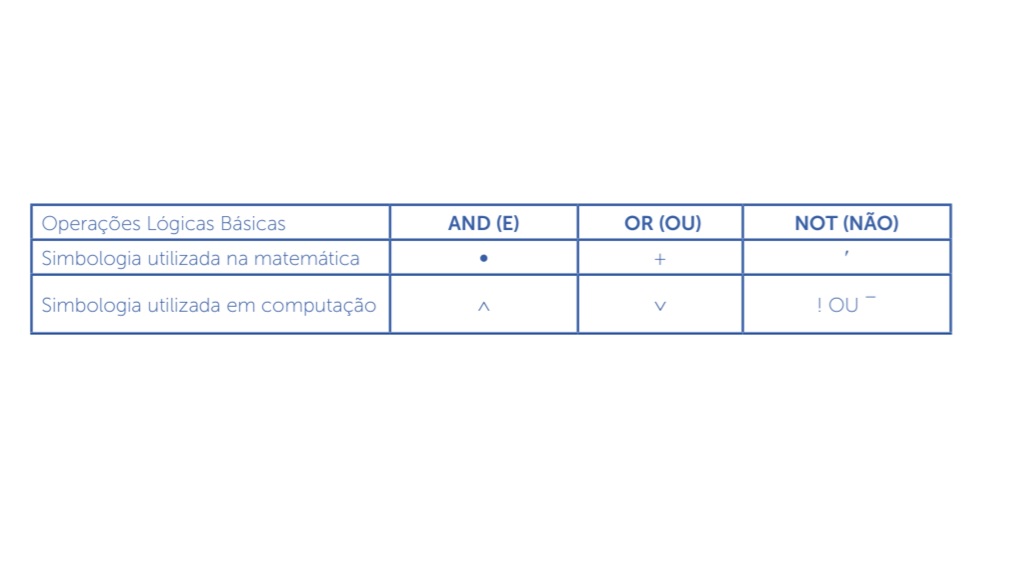
**Circuitos de Comutação** – É um processo onde se pode interligar dois ou mais processos entre si, neste caso, utilizando-se dos relés, que nada mais é que uma chave.

\_\_\_\_\_\_\_

A sua técnica foi largamente utilizada na análise e no projeto de circuitos eletrônicos digitais. Seu nome é considerado um marco para a área de sistemas digitais.

Como em qualquer uso de álgebra convencional, a álgebra booleana utiliza-se de variáveis e operações lógicas. Essa variável pode ter o valor lógico 1 (verdadeiro) ou 0 (falso). Cada variável pode assumir um único valor, sendo ele: 1 ou 0, verdadeiro ou falso, *true*ou *false*, sim ou não, aberto ou fechado, aceso ou apagado, entre outros.

As operações lógicas básicas são: AND (E), OR (OU) e NOT (NÃO), conforme o apresentado na Tabela a seguir.

Simbologia de Operações Lógicas. Fonte: O autor.

Podendo ser usado da seguinte forma:

A AND B = A • B = A ∧ B

A OR B = A + B = A ∨ B

NOT A = !A = A

Veja um exemplo prático:

A = Ana viaja

B = Ana brinca

A ∧ B → Ana viaja **e** Ana brinca

A ∨ B → Ana Viaja **ou**Ana Brinca

!A → Ana **não**viaja

A partir das 3 operações lógicas temos:

AND → Produto Lógico

OR → Soma Lógica

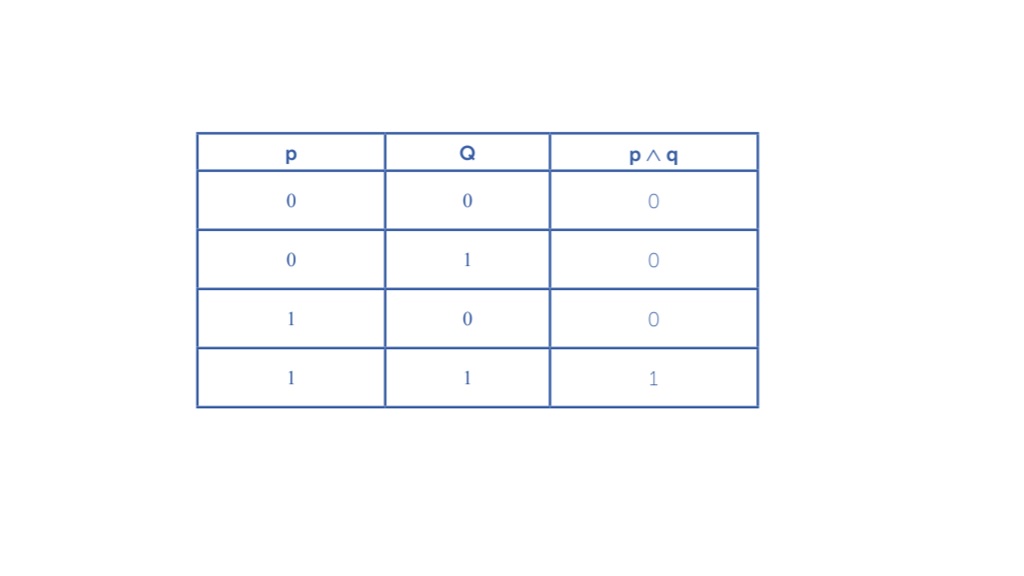
Not → Negação

Sabendoessa simbologia de operações lógicas e tendo em vista essa interpretação lógica, podemos criar as tabelas-verdade. Usamos nas tabela-verdade as variáveis p e q.

**Tabela-verdade AND, OR, NOT e porta lógica NAND, NOR, XOR e XNOR**



Nessa tabela usamos a proposições p ∧ q. Para que p ∧ q sejam verdadeiras, as duas proposições têm de ser verdadeiras (STALLINGS, 2003). Isso nos leva à seguinte tabela-verdade:

Tabela-verdade AND. Fonte: O autor.

Visualize na tabela AND que o valor será verdadeiro (1) sempre que p e q são verdadeiros.

\_\_\_\_\_\_\_

**📝 Exemplificando**

x = 3

y = 5

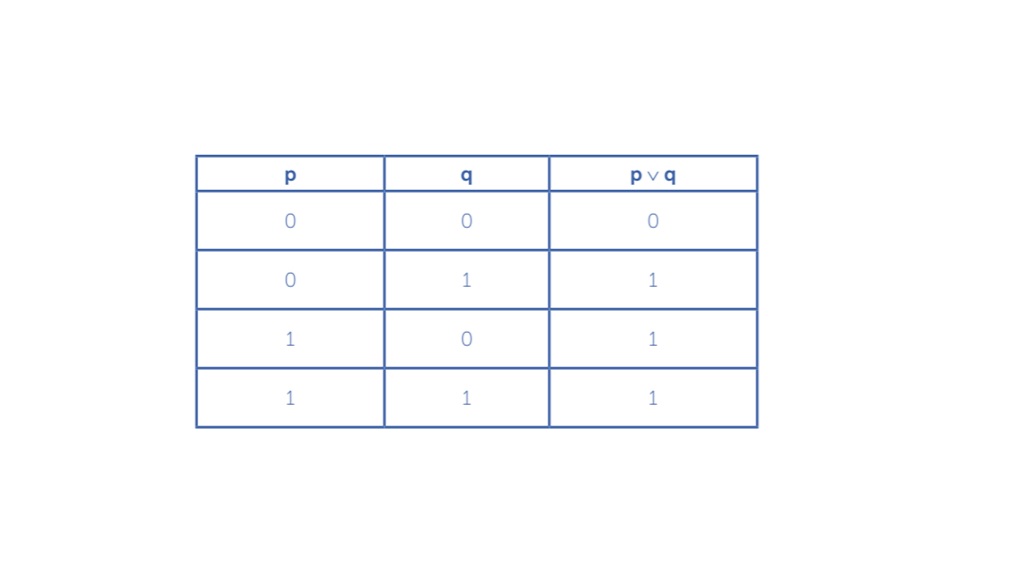
A expressão (x = 4) ∧ (y=5) é verdadeira (1) ou Falsa (0)?

(x = 4) ∧ (y=5) → (x = 4) representa p e (y=5) representa q, temos:

(falso) ∧ (verdade) = (veja a segunda linha da tabela-verdade. Temos o valor 0 (falso) e o valor verdade (1), logo temos o valor para p ∧ q = 0 (falso).

\_\_\_\_\_\_\_

Na tabela OR também usamos a mesma proposição, porém, basta que uma delas seja verdadeira para que p ∨ q seja verdadeiro. Veja sua tabela-verdade:

Tabela-verdade OR. Fonte: O autor.

Veja na tabela-verdade OR que o resultado sempre será negativo (0) quando todos os valores de p e q são negativos. Veja no exemplo.

\_\_\_\_\_\_\_

**📝 Exemplificando**

k = azul

w = verde

A expressão (k = vermelho) ∨ (w = verde) é verdadeira ou falsa?

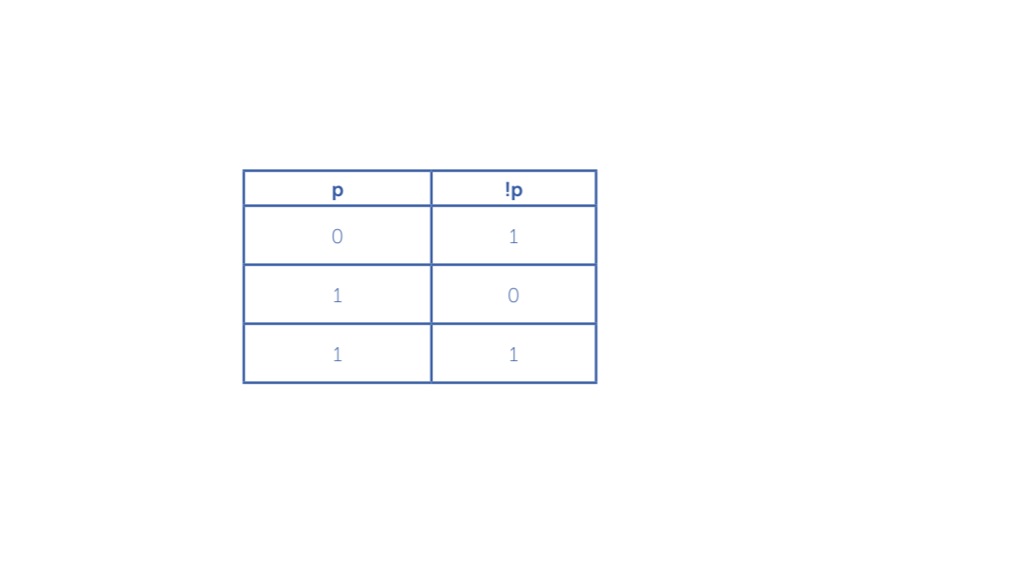
(k = vermelho) ∨ (w = verde) →

Falso ∨ verdadeiro = (veja segunda linha da tabela-verdade)

O resultado para esse p ∨ q é 1 (verdade)

\_\_\_\_\_\_\_

Essa tabela-verdade dá-se pela negação da proposição, invertendo seu valor. A tabela-verdade ficará assim:

Tabela-verdade NOT. Fonte: O autor.

A negação sempre será o inverso do valor em questão.

\_\_\_\_\_\_\_

**🔁 Assimile**

p = 8

!p = representa todos os números menos o 8.

\_\_\_\_\_\_\_

As tabelas-verdade são usadas para definir valores para as portas lógicas, portanto, podemos chamar de porta lógica AND, porta lógica OR e porta lógica NOT.

\_\_\_\_\_\_\_

**💪 Faça você mesmo**

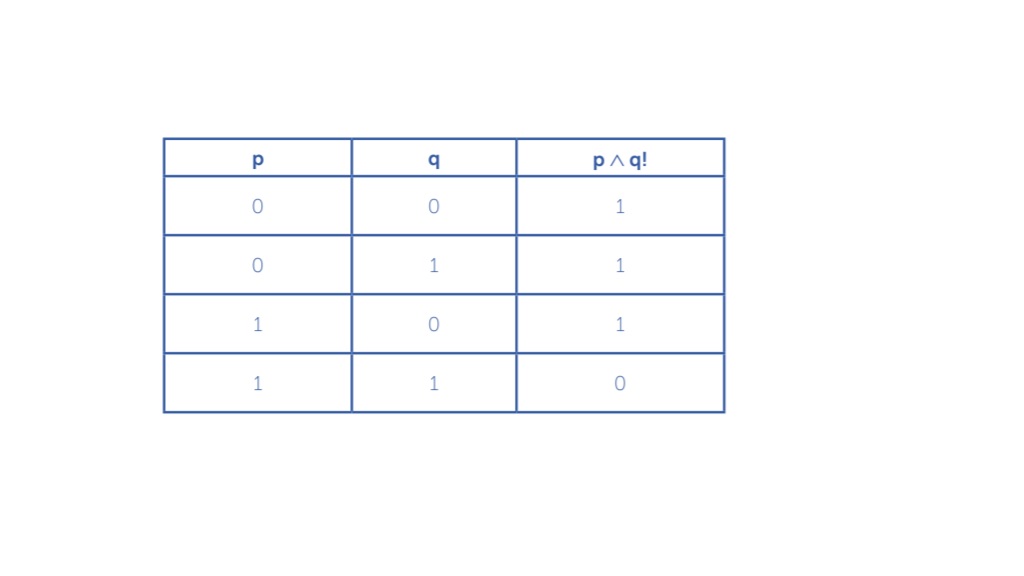
Se c = céu azul e d = dia chuvoso, cria a tabela-verdade para **p ∧ q**

* c = falso e d = falso
* c = falso e d = verdade
* c = verdade e d = falso
* c = verdade e d = verdade E qual seria o resultado para !d?

\_\_\_\_\_\_\_

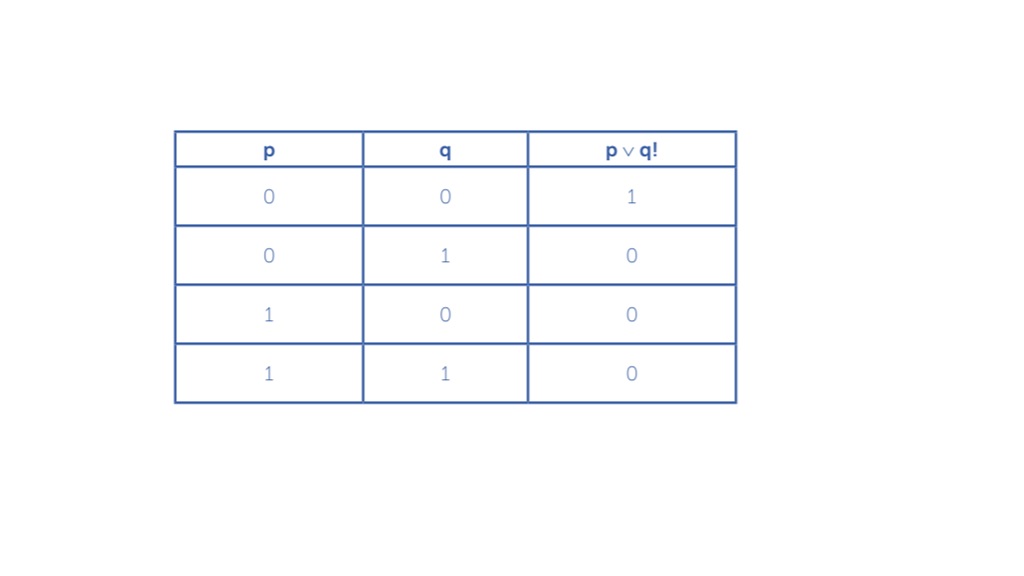
Observe outras portas lógicas utilizadas a partir das portas lógicas básicas AND, OR e NOT:

Equivale a uma porta lógica AND seguida de uma NOT → (p ∧ q)!

Tabela-verdade NAND. Fonte: O autor.

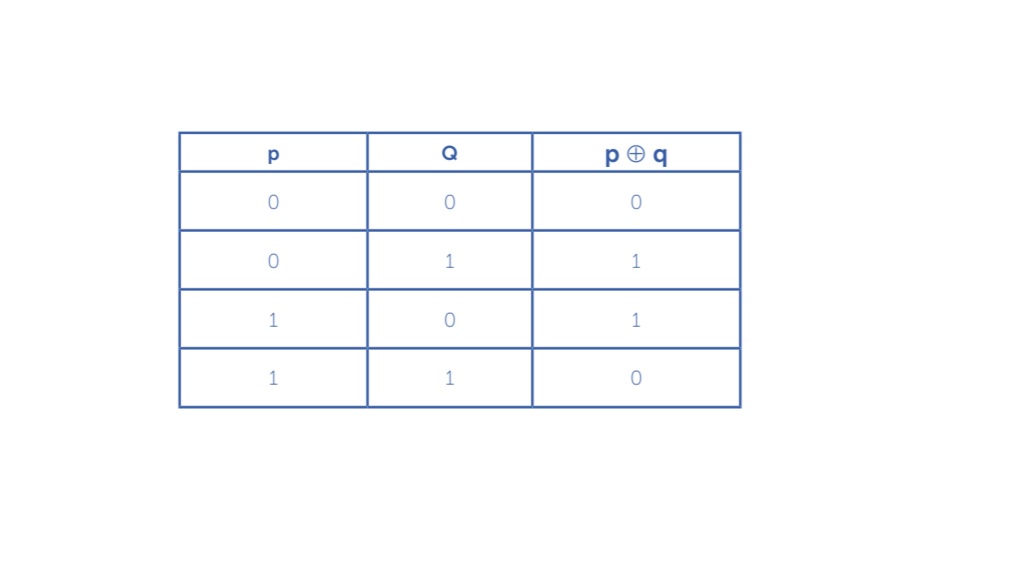
Perceba que os resultados, pela negação, são o inverso do p ∧ q (AND).

**Porta Lógica NOR** - Equivale a uma porta lógica OR seguida de NOT → (p ∨ q)!

Tabela-verdade NOR. Fonte: O autor.

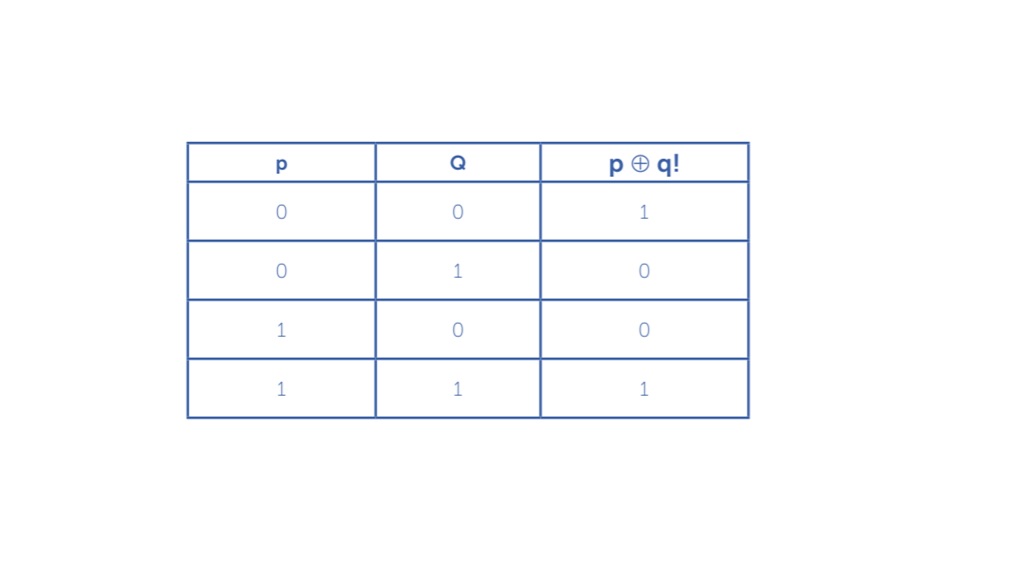
Os resultados, pela negação, são o inverso do p ∨ q (OR).

**Porta Lógica XOR** - Conhecida como OU EXCLUSIVO (⊕). Ela compara dois valores e, se o resultado for diferente, mostra como saída o valor 1 → p ⊕ q.

Tabela-verdade XOR. Fonte: O autor

Somente terá resultado como verdadeiro quando os valores de p e q tiverem resultado diferente entre eles.

**Porta Lógica XNOR** - Essa porta é o complemento da XOR. Se analisar, verá que nada mais é que a negação da porta XOR→ (p ⊕ q)!

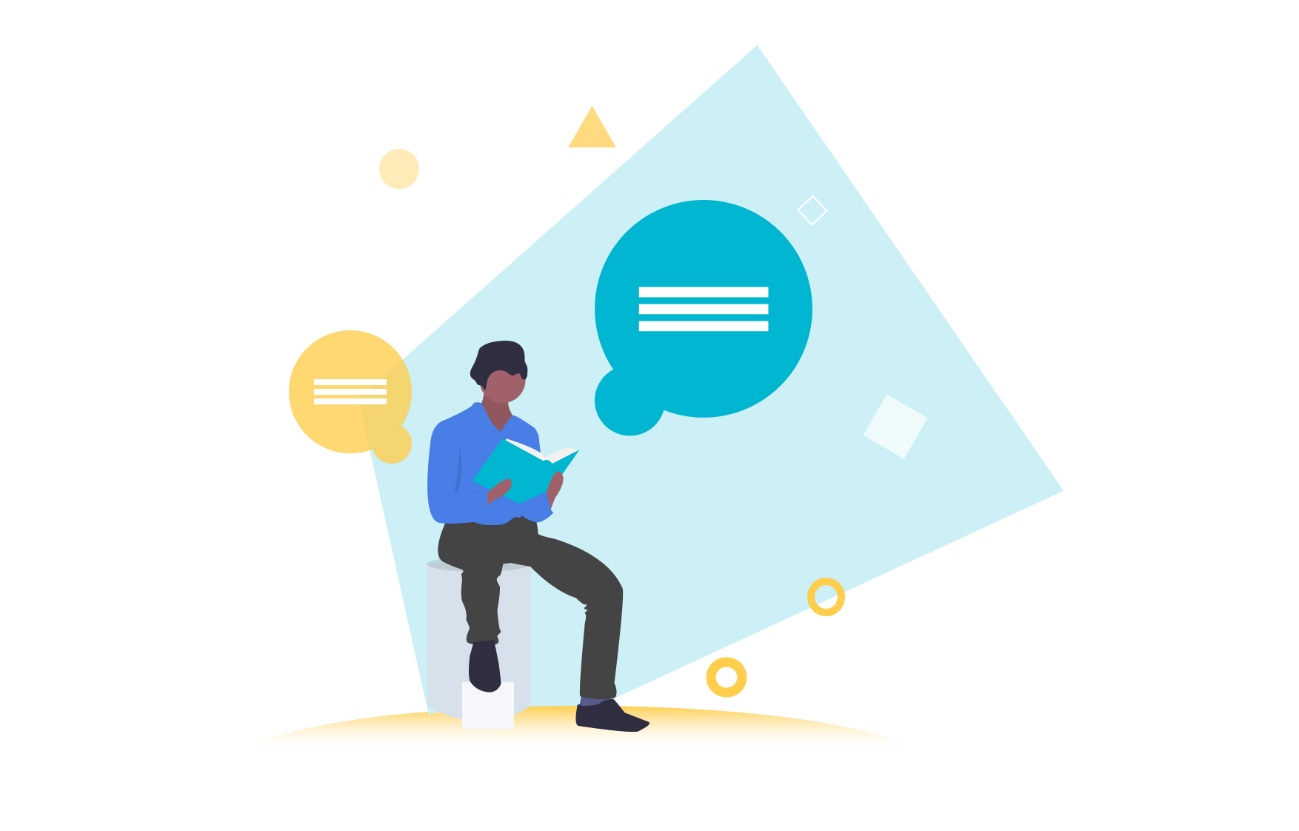
Tabela-verdade XNOR. Fonte: O autor.

**➕ Pesquise mais**

Conheça um pouco mais [sobre](ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia861_1s10/notas_de_%20aula/topico3_IA861_1s10.pdf)Álgebra de Boole, Conjuntos Clássicos e Lógica.

Assista ao [vídeo](https://www.youtube.com/watch?v=STc90QFwNjw&feature=youtu.be)Introdução à Lógica para rever os conceitos aprendidos.

**Conclusão**

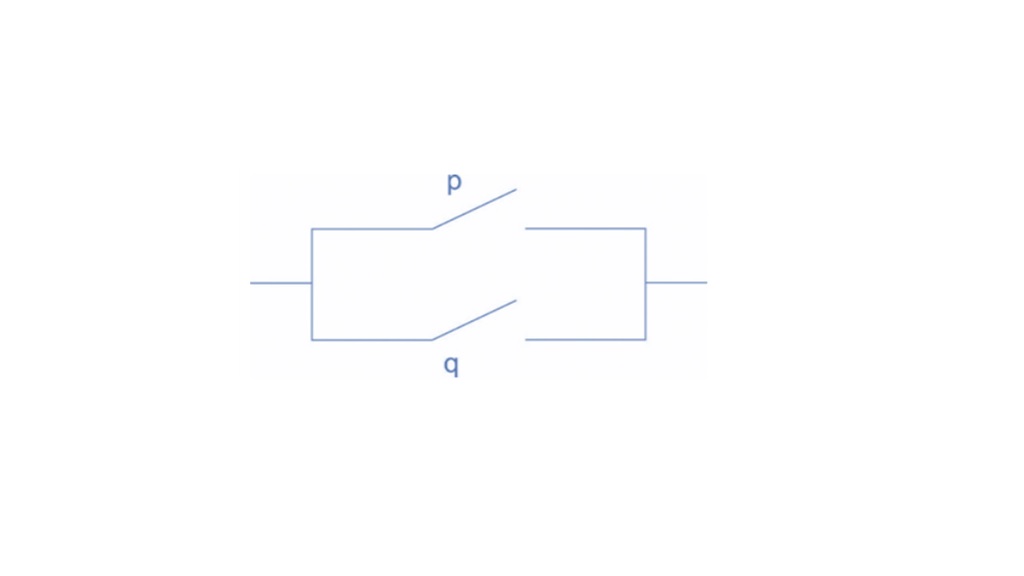


Agora, podemos iniciar o desenvolvimento do diagrama do interruptor. O primeiro passo é a criação desse diagrama de circuito de um interruptor. Um diagrama de circuito é o primeiro passo para se chegar ao processo de criação de uma placa. Um interruptor é um dispositivo que é ligado a um ponto de um circuito, podendo ele ter um de dois estados, sendo: ligado ou desligado (ou ainda estar aberto ou fechado). No estado "fechado", o interruptor permitirá que a corrente elétrica passe através do ponto, permitindo assim que uma lâmpada seja acesa, por exemplo. Já no estado "aberto", não se permite a passagem da corrente elétrica pelo ponto, fazendo com que a lâmpada fique desligada. Você deverá usar para o estado "fechado" o valor 1 e para o estado "aberto" o valor 0, devendo o circuito ter dois interruptores (em paralelo). Depois de feito o diagrama em paralelo, faça o diagrama em série e sua respectiva tabela verdade.

Antes de iniciarmos, é necessário que você conheça [sobre](https://www.tecmundo.com.br/como-e-feito/18501-como-as-placas-de-circuito-impresso-sao-produzidas.htm)“Como as placas de circuitos impressos são produzidas”.

Agora que você já conhece os conceitos e passos para a elaboração de uma PCI simples, para criar a PCI é necessário “confeccionar” seu diagrama do circuito de um interruptor.

Nosso diagrama, por possuir dois interruptores, ficará em paralelo, indicado por p ∨ q.

Diagrama interruptor OR. Fonte: O autor.

Neste diagrama não passa corrente se, e somente se, p = 0 e q = 0. Isso significa que ambos estão “abertos” e correspondem à tabela-verdade OR.

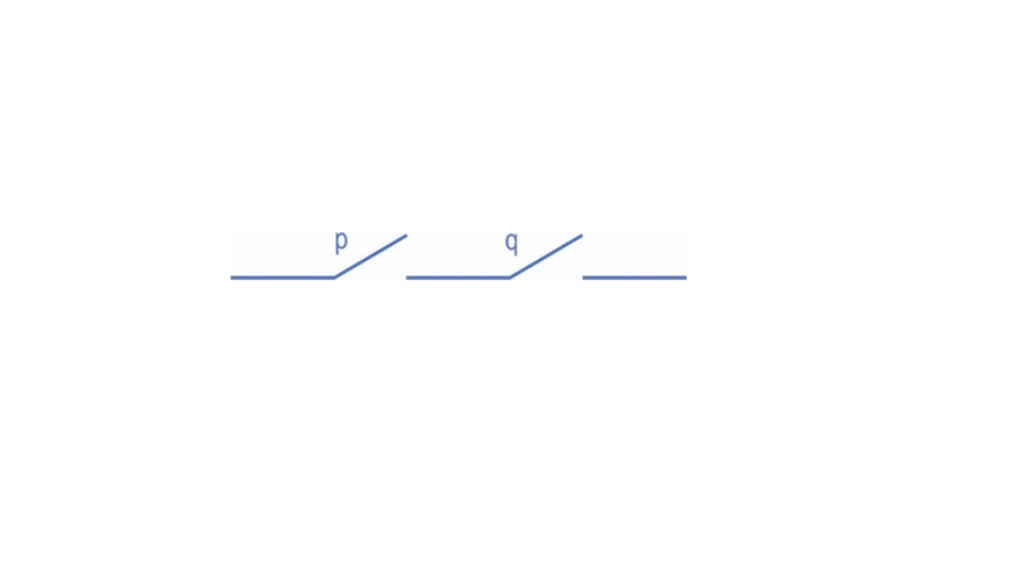
Veja na tabela-verdade os valores de p e q como zero.

Se montarmos uma tabela-verdade OR para o interruptor, ficaria assim:

Tabela-verdade OR para estado do interruptor. Fonte: O autor.

Dessa maneira, notamos que o interruptor pode ser “ligado” se p estiver fechado ou se q estiver fechado. Somente ficará desligado se p e q estiverem abertos, ou seja, desligado.

Agora, o diagrama do circuito em série é indicado por **p ∧ q**.

Diagrama interruptor AND. Fonte: O autor.

Nesse diagrama passa corrente se, e somente se, p = 1 e q = 1. Isso significa que ambos estão “fechados” e correspondem à tabela-verdade AND.

Veja na tabela-verdade os valores de p e q como 1.

E a tabela-verdade AND para o interruptor ficaria assim:

Tabela-verdade AND para estado do interruptor. Fonte: O autor.

Assim, notamos que o interruptor pode ser “ligado” se p estiver fechado e se q estiver fechado. Somente ficará desligado se p ou q estiverem abertos.

Desse modo, resolvemos nossa tabela-verdade com todos os valores para o interruptor e desenhamos o diagrama do interruptor.

\_\_\_\_\_\_\_

**⚠️ Atenção!**

Lembre-se sempre de que o valores relacionados ao 0 são desligados, falsos, sempre na negação.

Os valores relacionados a 1 são sempre relacionados à ligado, verdadeiros, sempre na veracidade.