**Introdução**

MIP é a sigla em inglês para *Programação Inteira Mista*, um importante ramo da otimização matemática e uma tecnologia fantástica para modelagem e solução de problemas que envolvem tomadas de decisão em uma extensa gama de domínios como: Cadeia de produção, logística, tabelamento de jogos, geração de energia elétrica, e muito mais.

Aqui, você encontrará muitos exemplos práticos e acessíveis para a aprendizagem e aplicação de MIP para solucionar problemas reais. Muitos anos de experiência acadêmica e na indústrias estão sendo combinados para desenvolver esse material, que tem como objetivo ajudá-lo a adotar as melhores práticas desde o começo.

Faz parte dos nosso comprometimento manter esse material disponível e de graça para todos.

Um dos principais objetivos do Mip Master é tornar a tecnologia de MIP acessível para além da comunidade de pesquisa operacional, especialmente para a comunidade de *data science* (ciência de dados). Com isso em mente, os exemplos práticos compartilhados por nós são desenvolvidos para serem os mais didáticos possível (sem comprometer o desempenho) e serão todos implementados na linguagem *Python*.

Após o estudo desse material, você deverá ser capaz de:

1. Identificar quando um problema pode ser modelado com MIP;
2. Modelar uma variedade de problemas utilizando MIP;
3. Implementar e solucionar modelos utilizando um *solver* MIP em Python.

Expomos um pouco de teoria básica mas a ênfase é em aplicações. Primeiramente, ensinamos como utilizar MIP com vários exemplos práticos e acessíveis. Você encontrará modelos e amostras de códigos que poderá personalizar, expandir e utilizar em seus próprios projetos. À medida que você se sentir motivado, você terá oportunidade de se aprofundar na teoria e avançar para o próximo nível.

Ainda que programação é um componente importante deste curso, somente habilidades básicas de *Python* serão pré-requisitadas. Se você é iniciante em *Python*, você também aprenderá muita programação ao longo do caminho.

Esperamos que você aproveite esse conteúdo e que lhe seja muito útil. Esperamos também que você sempre volte para aprender mais e utilizar MIP com confiança, começando pelo seu próximo projeto!

**Estudos de Casos**

Se desejar avançar direto para os estudos de casos, a seguir está uma lista dos casos disponíveis. Tenha em mente que estamos frequentemente adicionando novos casos à lista. Recomendamos explorar os casos na ordem que estão listados, especialmente se você ainda não está familiarizado com MIP.

1. [TicTech](https://github.com/mip-master/learning_mip/tree/master/tictech)- Ilustra como implementar um modelo muito simples de otimização.   
   *Autor: Aster Santana, Jul 2020*

Conceitos abordados:

* + Os três passos para solucionar um problema prático;
  + Os três componentes de uma formulação matemática;
  + Modelagem com variáveis de decisão binárias;
  + Chamando um *solver* MIP.

1. [Ukulele-la♬-la♫](https://github.com/mip-master/learning_mip/tree/master/ukulelelala) - Uma versão simplificada de um problema prático de abastecimento de demanda.   
   *Autor: Aster Santana, Jul 2020*

Conceitos abordados:

* + Variáveis de decisão inteiras;
  + Notação de somatório;
  + Restrições “se-então”;
  + Restrições usando *Big-M*;
  + Complemento de uma variável binária;
  + Arquivos LP.

1. [Paper Tree](https://github.com/mip-master/learning_mip/tree/master/paper_tree) - Uma versão simplificada de um problema prático de planejamento de produção.  
   *Autor: Aster Santana, Jul 2020*

Conceitos abordados:

* + Problema do fluxo de rede;
  + Restrição da conservação de fluxo;
  + Modelo de formatação de dados para otimização.

Gostaria de contribuir com um estudo de caso também?! Nós providenciamos [modelos](https://github.com/mip-master/learning_mip/tree/master/templates) e ficaríamos felizes em ajudá-lo a construir o seu!

**A Tecnologia MIP**

Vamos aprofundar um pouco mais na tecnologia MIP.

**O que é MIP?**

MIP é a sigla em inglês para Programação Inteira Mista.

* “Programação” é sinônimo de “Otimização”.
* “Inteira” é para variáveis de decisão inteira, que inclue variáveis binários como um caso especial.
* “Mista” é para indicar que pode haver uma mistura de variáveis de decisão inteiras e contínuas.

Juntando esses conceitos temos que: “MIP é um problema de otimização que envolve variáveis de decisão inteiras/binárias e contínuas”

Caso você não esteja familiarizado com otimização, não se preocupe! Providenciamos a base necessária a medida que apresentamos of estudos de casos. Por hora, tudo que você precisa saber é que um problema de otimização consiste em minimizar ou maximizar uma função objetivo sujeita a um conjunto de restrições, onde tanto a função objetivo como as restrições são escritas em função das variáveis de decisão.

Uma variável de decisão pode ser o número de produtos a ser manufaturado em determinado maquinário, ou se o abastecimento de uma loja vem do Depósito A ou do Depósito B. Exemplos de funções objetivo incluem minimizar o desperdício e maximizar os lucros. Restrições podem representar disponibilidade de recursos, como o número máximo de horas de trabalho disponível, e políticas de operacionalização.

No fim das contas, nosso objetivo é definir um conjunto de soluções viáveis (factíveis) do problema e identificar aquelas que têm o melhor valor objetivo.

Para isso, existem três passos fundamentais:

1. **Definição**: Entender o problema em detalhes, que inclui a identificação dos objetivos principais, dos dados e dos requisitos;
2. **Modelagem**: Construir uma representação precisa do problema – No nosso caso, primeiramente uma formulação matemática e depois um modelo de otimização que deve ser interpretado pelo computador;
3. **Solução**: A busca pela melhor solução – No nosso caso, utilizando um MIP *solver*.

Você irá se surpreender ao ver a variedade de problemas práticos que pode ser solucionado com o que você vai aprender aqui!

Mais alguns fatos:

* MIP é uma extensão da Programação Linear (em inglês LP), que representa a classe mais básica de problemas de otimização;
* Às vezes se utiliza o acrônimo MILP ao invés de MIP para enfatizar a natureza linear do problema;
* Alguns preferem a palavra “otimização” ao invés de “programação”. Neste caso, o acrônimo passa a ser MIO ou MILO;
* Nem todo MIP é um problema de otimização, no sentido que nem sempre estamos interessados em maximizar ou minimizar uma função objetivo. Em alguns casos, queremos encontrar apenas a solução factível, i.e., a solução que atenda à todas as restrições simultaneamente. Quebra-cabeças, como Sudoku, são bons exemplos disso.
* Existem muitas classes de problemas de otimização. Essas classes se definem, por exemplo, baseadas em pressupostos de modelagem, como determinísticos ou estocásticos; tipos de variáveis de decisão presentes no modelo, sendo elas contínuas, inteiras ou ambas; ou pelo tipo de expressões que definem o objetivo e suas restrições, podendo ser linear ou não linear, convexa ou não convexa. Nosso foco são *programas determinísticos lineares inteiros mistos*.

**Por que MIP?**

MIP é uma tecnologia de tomada de decisão poderosa. Aqui vão quatro características do MIP que dão suporte à essa afirmação: representatividade, robustez, tratabilidade e garantia do ótimo.

* **Representatividade**: O primeiro passo para solucionar um problema real usando otimização é escrever uma formulação matemática precisa do problema. Esse processo é conhecido como modelagem – utilizaremos essa terminologia com bastante frequência. Representatividade significa que uma grande classe de problemas reais podem ser modelados usando MIP.
* **Robustez**: Modelagem, seja usando MIP ou outra tecnologia, não é uma tarefa trivial. E no ambiente dinâmico em que vivemos, os problemas reais tendem a mudar com muita frequência. Como você verá nos estudos de casos, MIP é flexível e adaptável, ou seja, robusto. Na maioria das vezes, é fácil extender ou modificar um MIP para incorporar novos requisitos do problema.
* **Tratabilidade**:A modelagem de um problema de negócios é o primeiro passo. Eventualmente, teremos que solucionar esse modelo. E como se pode imaginar, quanto maior o número de variáveis e restrições, mais difícil será para resolver o modelo. A boa notícia é que modelos práticos com *milhões* de variáveis e restrições podem ser solucionados eficientemente. Outra boa notícia é que o *solver* MIP faz o trabalho duro por você!
* **Garantia do ótimo**: O que significa solucionar um problema de otimização? Normalmente significa encontrar a melhor solução factível em relação a um objetivo. É mais fácil de entender com um exemplo. Suponha que você queira ajustar um modelo de regressão não-linear à um conjunto de dados usando um algoritmo de gradiente descendente. Como saber se a solução encontrada é a que melhor se ajusta? A solução encontrada depende do ponto de partida escolhido. Então, a não ser que seu modelo não-linear tenha uma estrutura especial, você nunca saberá com certeza se a solução encontrada é a melhor possível. As soluções provenientes de um MIP *solver*, por outro lado, vem com um *gap de otimalidade* – chamado também de *gap de dualidade* ou simplesmente *MIP gap*. Se esse gap (intervalo) é zero, nenhuma solução com melhor objetivo existe. Se o intervalo é maior que zero, a solução pode não ser ótima mas ainda assim temos um limite superior de “quão longe” a solução encontrada está da melhor solução que se pode obter.

Recapitulando, MIP é uma excelente tecnologia porque:

* Tem aplicabilidade para uma extensa gama de classes de problemas;
* É bastante flexível e adaptável;
* Pode solucionar problemas grandes, podendo ter milhões de variáveis e restrições;
* A solução vem com um gap de otimalidade que informa o quão perto ou longe a solução se encontra da melhor solução possível.

**Por que a MIP agora?**

Para responder essa questão, vamos dar uma olhada na história da teoria por trás da MIP e seus *solvers*.

Primeiramente, a MIP não é uma tecnologia nova. Aqui temos uma tradução de uma citação do [livro](https://www.springer.com/gp/book/9783540682745) *50 Years of Integer Programming 1958-2008, From the Early Years to the State-of-the-Art*.

*“Em 1978, Ralph E. Gomory transformou o campo da programação inteira quando publicou um breve artigo que descreve seu inovador algoritmo para programação puramente inteira e anunciou que seu método poderia ser refinado para entregar um algoritmo finito para programação inteira.”*

A tratabilidade de um MIP era um problema inícialmente. Porém, houve um tremendo progresso acerca da solução de MIP. Aqui está uma tradução de um trecho de um [artigo](http://www.mit.edu/~dbertsim/papers/Machine%20Learning%20under%20a%20Modern%20Optimization%20Lens/Logistic%20Regression-From%20Art%20to%20Science.pdf) escrito por Dimitris Bertsimas e Angela King:

*“No período entre 1991-2015, os avanços algorítmicos na Otimização Linear Inteira Mista (MILO) juntamente com melhorias em hardware resultaram no incrível aumento do fator de aceleração da solução de problemas MILO em 450 bilhões de vezes.”*

O número 450 bilhões é impressionante, não é? E esse número continua crescendo a cada ano, como pode ser visto em relatórios de desempenho de MIP *solvers* comerciais como [CPLEX](https://www.ibm.com/analytics/cplex-optimizer) e [Gurobi](https://www.gurobi.com/).

Por várias décadas, a MIP tem transformado operações em muitas indústrias, incluindo [tabelamento de funcionários de empresas aéreas](https://www.isye.gatech.edu/news/airline-optimization-isye), [tabelamento de eventos esportivos](http://www.sports-scheduling.com/in-the-news.html), e toda a área de logistica e de produção. Todavia, inicialmente, somente aqueles que estavão desenvolvendo a teoria da MIP eram aptos a utilizar essa tecnologia para solucionar problemas reais.

Essa realidade começou a mudar nos anos 90, quando o número de artigos científicos mostrando como modelar e solucionar modelos MIP começou a crescer. Por volta desse período, os *solvers* comerciais [Xpress](https://en.wikipedia.org/wiki/FICO_Xpress) e [CPLEX](https://www.ibm.com/analytics/cplex-optimizer) começaram se popularizar. Alguns anos depois, no início dos anos 2000, dois dos primeiros *solvers* MIP *open-source*, [GLPK](https://en.wikipedia.org/wiki/GNU_Linear_Programming_Kit) e [COIN-OR CBC](https://en.wikipedia.org/wiki/COIN-OR#CBC), foram publicados. Outro grande competidor do lado comercial, [Gurobi](https://www.gurobi.com/), foi fundado em 2008.

Com isso, uma comunidade maior começou a adotar MIP para solucionar problemas desafiadores em várias indústrias. No entanto, algumas linguagens de modelagem específicas, como [GAMS](https://en.wikipedia.org/wiki/General_Algebraic_Modeling_System), [AMPL](https://en.wikipedia.org/wiki/AMPL), ou [AIMMS](https://en.wikipedia.org/wiki/AIMMS), ainda eram necessárias para modelar e solucionar problemas usando MIP. Isso limitava de certa forma a utilização de MIP à comunidade de pesquisa operacional.

Outra onda de democratização da tecnologia MIP veio com a popularização do *Python* como a principal linguagem de programação para cientistas e analistas de dados. Várias *interfaces* em *Python* para pacotes de otimização emergiram, como [Pyomo](https://en.wikipedia.org/wiki/Pyomo), [Mip](https://pypi.org/project/mip/), and [PuLP](https://coin-or.github.io/pulp/). Gurobi, em particular, fez um grande esforço para tornar o seu pacote de otimização facilmente acessível via [gurobipy](https://www.gurobi.com/documentation/9.0/quickstart_mac/py_python_interface.html), uma *interface* em *Python* que dispõe de uma documentação bem estruturada. CPLEX também optou por tomar uma rota semelhante.

Agora, qualquer cientista de dados (e outros profissionais da área) podem facilmente alavancar a tecnologia de MIP para solucionar diversos problemas de natureza combinatória. No entanto, o mito de que a otimização é exclusiva de pesquisadores operacionais ainda perdura.

**MIP para Cientistas de Dados**

Além do fato de que MIP é uma tecnologia poderosa e acessível mesmo para aqueles fora da comunidade de pesquisa operacional, podemos afirmar que **ciência de dados e MIP é uma combinação muito poderosa**.

Em várias aplicações, parte dos dados de entrada para modelos MIP é gerado por um grupo de cientista de dados. Um exemplo clássicos são dados de demanda, em que os cientistas de dados predizem a demanda e a saída do algoritmo de previsão se torna a entrada do modelo MIP. Nesse exemplo, predição e otimização são feitos de maneiras desconexas. Essa desconexão tem criado um distanciamento entre o que estamos fazendo atualmente e o que poderíamos alcançar com a análise de dados.

Pra citar um exemplo prático, considere o caso em que cientistas de dados e pesquisadores operacionais solucionaram, de forma colaborativa, um problema real em que MIP foi utilizado para discretizar e maximizar faturamento sobre dez mil curvas probabilísticas, simutaneamente, geradas por um algoritmo de *machine learning*.

Aqui está a tradução de uma citação de Ed Rothberg, CEO do Gurobi, extraída de um [vídeo](https://www.youtube.com/watch?v=RDOsP-gUWgQ) de dois minutos:

*“A MIP complementa muito bem outras técnicas analíticas como machine learning, veremos mais e mais empresas construindo aplicações que combinam machine learning e otimização.”*

Concluindo, existe muito potencial de benefícios em aproximar MIP e ciência de dados (*machine learning*).

**É difícil aprender MIP?**

Essa é uma pergunta delicada. A resposta depende muito do nível de aperfeiçoamento que se almeja alcançar. Comparado à *machine learning*, seria justo dizer que:

* É mais difícil aprender regressão linear básica e técnicas de classificação do que MIP.
* *Deep learning* e *neural networks*, por outro lado, são certamente mais difíceis de apender do que MIP.

Perceba que todas essas técnicas de *machine learning* são por si só aplicações de otimização, e a MIP é um tipo de otimização.

Por exemplo, todas as vezes que tentamos ajustar um modelo à um conjunto de dados, estamos procurando por parâmetros do modelo que minimizem o erro ou maximizem uma função de *likelihood*. Portanto, esse é um problema de otimização, apesar de que tipicamente, os cientistas de dados não a enxergam dessa maneira. E porque não? A resposta é simples: ao minimizar a soma do quadrado dos erros, existe uma *formúla fechada* para a os parâmetros que melhor ajustam a curva aos dados. A fórmula esconde o problema de otimização. Agora, substitua o modelo linear por um modelo exponencial, como faríamos para modelar uma doença contagiosa. Nesse caso, não existe uma fórmula *fechada*, então teremos que abordar o problema com otimização de forma explícita, por exemplo, com um algoritmo de gradiente descendente.

Voltando à pergunta central, existem alguns níveis de instrução que se pode alcançar em otimização. Incluindo ser capaz de:

1. **Admitir** que um problema é um problema de otimização quando é o caso.
2. **Reconhecer** quando um problema pode ser formulado como um problema de otimização.
3. **Modelar/Formular** problemas práticos como problemas de otimização.
4. **Solucionar** problemas de otimização com *solvers* e algoritmos já existentes.
5. **Criar** algoritmos customizados para solucionar problemas desafiadores de otimização.
6. **Desenvolver** teorias para solucionar classes inteiras de problemas de otimização.

O nível 3 é o que requer mais prática, pois não existe receita para fazer modelagem usando MIP, é meio que uma arte. O nível 4 é o mais divertido para a maioria das pessoas. É quando você aperta o botão para executar o programa e acompanha o progresso de otimização através do *log* do solver, como no exemplo abaixo (observe como o MIP gap, última coluna, converge para zero).

Nodes Cuts/

Node Left Objective IInf Best Integer Best Bou nd ItCnt Gap

\* 0+ 0 6.11449e+07 5.06574e+ 07 17.15%

0 0 5.45123e+07 284 6.11449e+07 5.45123e+ 07 2230 10.85%

\* 0+ 0 5.75671e+07 5.45123e+ 07 5.31%

\* 0+ 0 5.69224e+07 5.45123e+ 07 4.23%

0 0 5.45151e+07 284 5.69224e+07 Cuts: 2 48 2773 4.23%

\* 0+ 0 5.67861e+07 5.45151e+ 07 4.00%

0 0 5.45175e+07 284 5.67861e+07 Cuts: 2 43 3350 3.99%

0 0 5.45176e+07 284 5.67861e+07 Cuts: 1 42 3823 3.99%

\* 0+ 0 5.59982e+07 5.45176e+ 07 2.64%

0 0 5.45178e+07 284 5.59982e+07 Cuts: 1 40 4397 2.64%

\* 0+ 0 5.53996e+07 5.45178e+ 07 1.59%

\* 0+ 0 5.51997e+07 5.45182e+ 07 1.23%

0 0 -1.00000e+75 0 5.51997e+07 5.45182e+ 07 4397 1.23%

\* 0+ 0 5.47518e+07 5.45182e+ 07 0.43%

No início, você passará bastante tempo nos níveis 3 e 4, que são o foco principal do projeto Learning MIP. O nível 6 é o qual encontraremos profissionais de Pesquisa Operacional com grau de Ph.D. O nível 5 também é muito divertido, mas requer mais experiência. Para alcançar o nível 6, você deverá entender de maneira aprofundada a teoria que está por trás dos níveis 3-5. Podemos indicar a direção para aqueles interessados, mas primeiro focamos somente na teoria que o fará um *praticante consciente e bem informado*.

**Sumário**

Muito bem! Até agora você aprendeu que:

* MIP é uma tecnologia muito poderosa para tomada de decisão e baseada em otimização.
* MIP pode ser aplicado em uma variedade de domínios por conta de sua representatividade espetacular.
* MIP é adaptável e flexível, o que nos permite encorporar novos requisitos do problema real de forma rápida e eficiente.
* Podemos solucionar MIP com milhões de variáveis, graças aos recentes avanços teóricos e computacionais.
* A tecnologia MIP se tornou acessível muito além da comunidade de pesquisa operacional (graças aos MIP *solvers,* comerciais e *open-source*), e é por isso que você está aqui.
* A maioria das técnicas de *machine learning* são baseadas em otimização.
* Esse conteúdo vai levá-lo ao nível de conhecimento necessário para solucionar problemas reais sem perder tempo desnecessáriamente com teorias de otimização.

**Próximos passos**

Esperamos que você esteja pronto e animado para começar sua jornada no mundo do MIP! Ou avançar para o próximo nível, se MIP não é mais novidade para você.

Agora é a hora de ver exemplos práticos e aprender como modelar e solucionar problemas de otimização reais como o Mr. Mip! Recomendamos que comece pelo estudo de caso [TicTech](https://github.com/mip-master/learning_mip/tree/master/tictech).

**Conheça o Mr. Mip**

Antes de começar, nos permita introduzi-lo ao Mr. Mip, ele é nosso exemplo para aprendizagem e aplicação da tecnologia de MIP. Mr. Mip é um consultor que é mestre em utilizar otimização matemática e programação para solucionar problemas de tomadas de decisão. Você irá aprender ao observar Mr. Mip em ação através dos estudos de casos. Para aproveitar ao máximo essa experiência, você deve focar em entender como Mr. Mip pensa. Ele é bem sistemático, então fica fácil reconhecer seu padrão de pensamento.

**Configurando um Solver MIP**

Você vai precisar de um *solver* para solucionar modelos de MIP. Um *solver* MIP é basicamente uma implementação profissional de um algorítimo de [branch and bound](https://en.wikipedia.org/wiki/Branch_and_bound) combinado com o método de [cutting-plane](https://en.wikipedia.org/wiki/Cutting-plane_method).

Ainda que existem muitos *solvers* disponíveis para solucionar MIP, nós listamos somente quatro dos *solvers* mais tradicionais (confira esse [link](https://en.wikipedia.org/wiki/List_of_optimization_software) para uma lista mais completa de *softwares* de otimização). Todos são *solvers* patenteados, exceto pelo CBC, que é *open-source* e faz parte do projeto [COIN-OR](https://www.coin-or.org/). Existem licenças acadêmicas para os três *solvers* patenteados listados abaixo.

Se você é iniciante e qualificado para uma licença acadêmica, Gurobi tem uma documentação bem estruturada e amigável. Se você não se qualifica para uma licença acadêmica, você pode começar com o CBC-PuLP. A boa notícias é que a sintaxe de modelagem desses *solvers* são bem similares, ou seja, se você aprender como usar um *solver* é fácil trocar para outro.

* [CBC](https://github.com/coin-or/Cbc)
  + Tipo: Open-source
  + *Interface Python*: [PuLP](https://pypi.org/project/PuLP/), [Guia de Instalação](https://coin-or.github.io/pulp/main/installing_pulp_at_home.html#installation)
* [CPLEX](https://www.ibm.com/analytics/cplex-optimizer)
  + Tipo: Patenteado
  + *Interface Python*: [DOcplex](https://cdn.rawgit.com/IBMDecisionOptimization/docplex-doc/master/docs/index.html), [Guia de Instalação](https://cdn.rawgit.com/IBMDecisionOptimization/docplex-doc/master/docs/getting_started_python.html)
* [Gurobi](https://www.gurobi.com/)
  + Tipo: Patenteado
  + *Interface Python*: [gurobipy](https://www.gurobi.com/documentation/9.0/quickstart_mac/py_python_interface.html), [Guia de Instalação](https://www.gurobi.com/documentation/9.0/quickstart_mac/ins_the_anaconda_python_di.html#section:Anaconda)
* [Xpress](https://www.fico.com/en/products/fico-xpress-optimization)
  + Tipo: Patenteado
  + *Interface Python*: [xpress](https://pypi.org/project/xpress/), [Guia de](https://www.msi-jp.com/xpress/learning/square/01-python-interface.pdf) Instalação

**Filosofia**

A meta de Mip Master com o projeto Learning MIP é democratizar a tecnologia de MIP para que seja utilizada além da comunidade de Pesquisa Operacional.

Tradicionalmente, o ensino de MIP foi planejado para começar com uma revisão detalhada da teoria por trás da tecnologia de MIP. Especificamente, estudantes têm gastado muito tempo aprendendo álgebra linear, desigualdades lineares, algoritmos simplex, etc. Só então os estudantes tem a chance de aprender formulação. A implementação, parte mais divertida, tem sido a última componente em cursos tradicionais que ensinam MIP.

Mip Master tem uma abordagem diferente. Focamos primeiro na aplicação e, a medida que for necessário e somente depois que existe motivação suficiente, introduziremos a teoria. Na realidade, começamos com formulação e implementação já no primeiro estudo de caso, pois estamos seguros que a experiência será muito mais significativa e gratificante dessa forma.

Acreditamos que essa é a abordagem correta, mesmo para a comunidade de pesquisa operacional, isso porque é assim que o cérebro humano funciona. Você vê um problema e tenta resolvê-lo com as ferramentas que tem a disposição, e se não ficar satisfeito com o resultado, você procura ferramentas mais sofisticadas. E no momento que você reconhecer que ferramentas mais complexas são necessárias, você já adquiriu a motivação e clareza necessária para entender a teoria.