

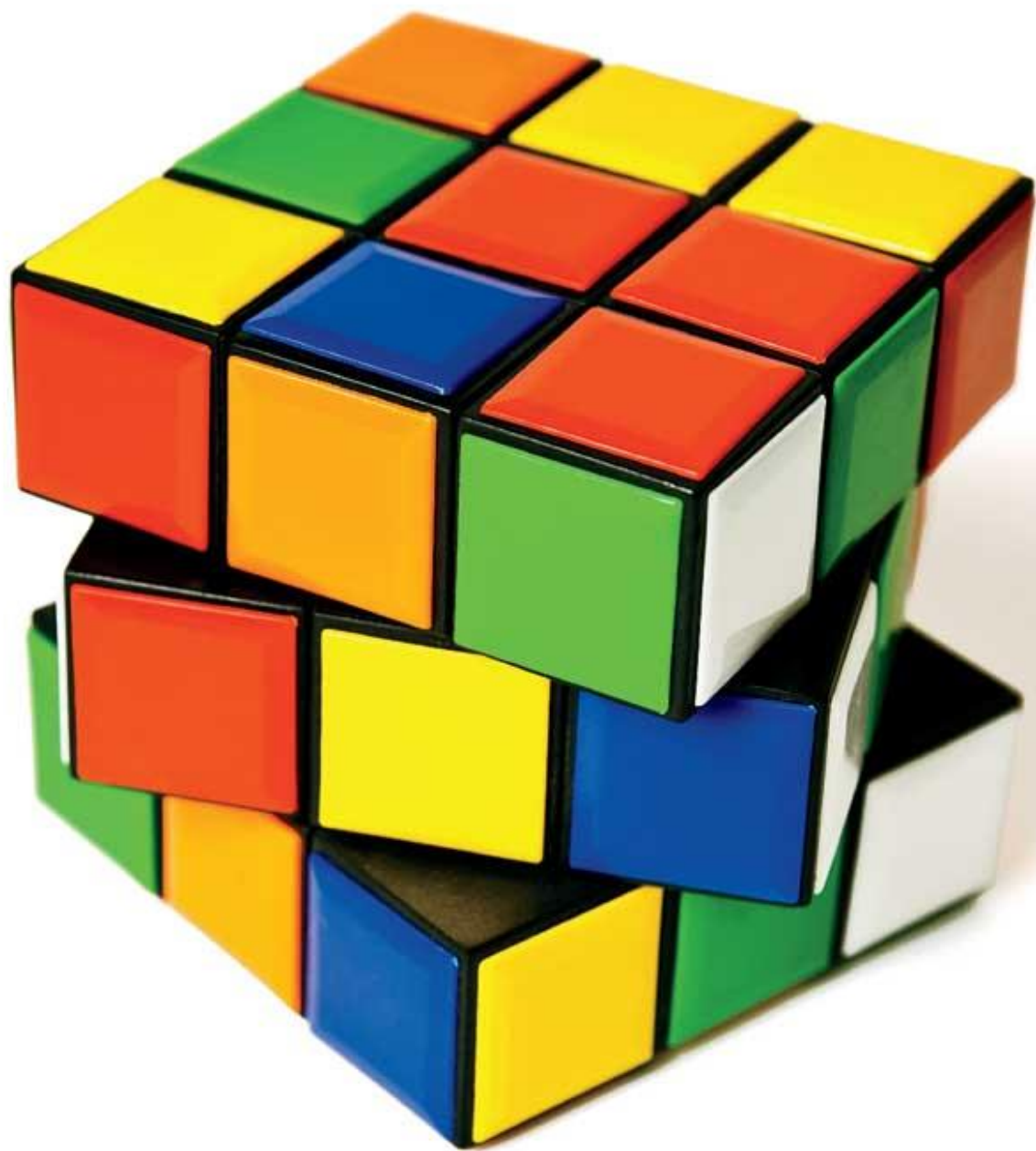


O Poder dos Padrões

Partilha de uma Visão Estratégica

Agradecimento





How to Solve a Rubik's Cube (Easy Move Notation)

Edited by NatK, Krystle, Mimi, BohemianWikipedian and 62 others

Four Parts: [■ First Layer](#) [■ Middle Layer](#) [■ Last layer](#) [■ Notations](#)

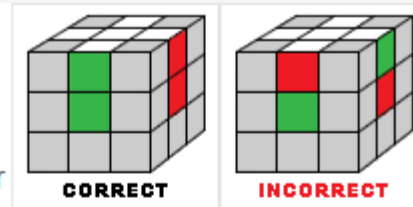
The Rubik's Cube can be very frustrating and may seem next to impossible to restore to its original configuration. However, once you know a few algorithms, it is very easy to solve. The method described in this article is the layer method: we first solve one face of the cube (first layer), then the middle layer, and finally the last layer.

■ Part 1 of 4: First Layer

- 1** Familiarize yourself with the **Notations** at the bottom of the page.

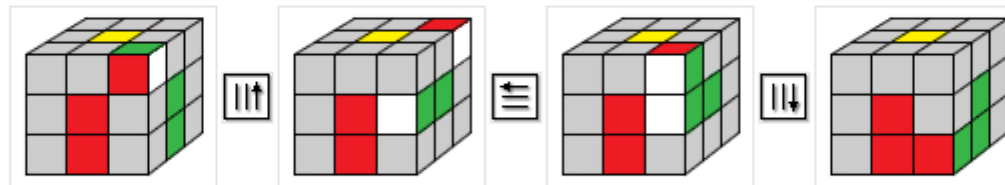
Part 1 of 4: First Layer

3 Solve the cross. Set into position the four edge pieces that contain white. (You should be able to do this by yourself without needing algorithms.) All four edge pieces can be placed in a maximum of eight moves (five or six in general).



- Place the cross at the bottom. Turn the cube over 180° so that the cross is now on the bottom.

4 Solve the four corners of the first layer, one by one. You should also be able to place the corners without needing algorithms. To get you started, here is an example of one corner being solved:



- At the end of this step, the first layer should be complete, with a solid color (in this case, white) at the bottom.

Part 2 of 4: Middle Layer

1 Place the four edges of the middle layer. Those edge pieces are the ones that do not contain yellow in our example. You need to know only one algorithm to solve the middle layer. The second algorithm is symmetrical to the first.

- If the edge piece is located in the last layer :



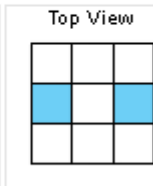
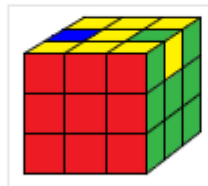
- If the edge piece is in the middle layer but in the wrong place or with the wrong orientation, simply use the same algorithm to place any other edge piece in its position. Your edge piece will then be in the last layer, and you just have to use the algorithm again to position it properly in the middle layer.

2 Verify correct positioning. Your cube should now have the first two layers complete and look like this (from the bottom side) :

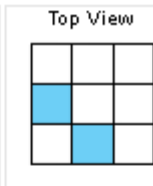
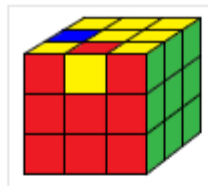
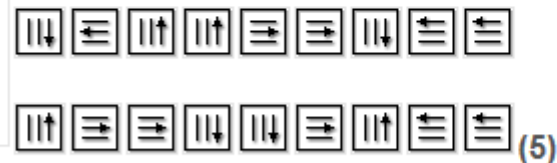


Part 3 of 4: Last layer

4 Orient the edges. You will need to know two algorithms for that last step :



Dedmore "H" Pattern



Dedmore "Fish" Pattern



- Note the DOWN, LEFT, UP, RIGHT, sequence to most of the Dedmore "H" and "Fish" algorithms. You really have only one algorithm to remember since :

$$(6) = \begin{matrix} \curvearrowright \\ \uparrow \uparrow \end{matrix} + (5) + \begin{matrix} \downarrow \downarrow \\ \curvearrowright \end{matrix}$$

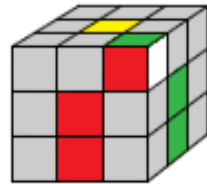
- If all four edges are flipped, perform the "H" pattern algorithm from any side, and you will have to perform that algorithm one more time to solve the cube.

Tal como na Geração de Software

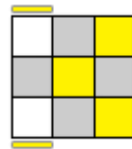
- Sequência definida de passos (procedimento)

Four Parts: ■ First Layer ■ Middle Layer ■ Last layer ■ Notations

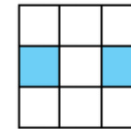
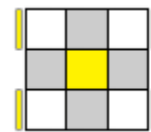
- Padrões



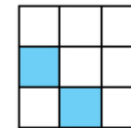
Two correctly oriented corners



No correctly oriented corners

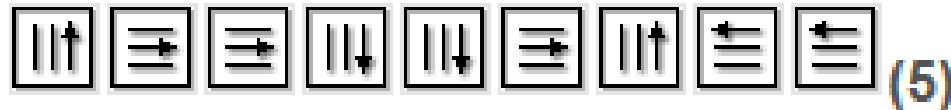


Dedmore
"H" Pattern

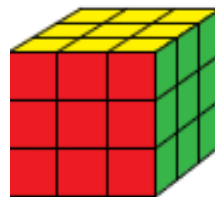


Dedmore
"Fish" Pattern

- Linguagem



- Resultado garantido





Quidgest

..... ge area ¶

A Engenharia do Software avança por Padrões ¶

www.quidgest.com | quidgest@quidgest.com ¶

Padrões no Genio

1. Padrões de persistência (estrutura de dados)
2. Padrões relacionados com os processos e o fluxo de trabalho
3. Padrões relacionados com lógica de negócio
4. Padrões relacionados com pesquisas, listas, formulários e interfaces
5. Padrões de segurança e auditoria
6. Padrões de identificação e depuração de erros (*debugging*)
7. Padrões de administração dos sistemas
8. Padrões que suportam a utilização internacional dos sistemas
9. Padrões que garantem a integração e a interoperabilidade
10. Padrões que suportam a engenharia do software
11. Padrões específicos de uma função ou área de negócio
12. Padrões e meios que facilitam a transição para Genio

Os Padrões reduzem a dificuldade!

$$8! \times 3^7 \times (12!/2) \times 2^{11}$$
$$= 43\ 252\ 003\ 274\ 489\ 856\ 000$$

Os Padrões são reprodutíveis!

- **A Engenharia do Software não é uma arte, é uma ciência!**
- **Os Padrões podem ser copiados. Os Padrões podem ser executados as vezes que quisermos. O resultado é garantido.**

Os Padrões são facilmente ensinados!

Em vez de 10 anos de formação,
10 semanas de formação



Os Padrões são automatizáveis!

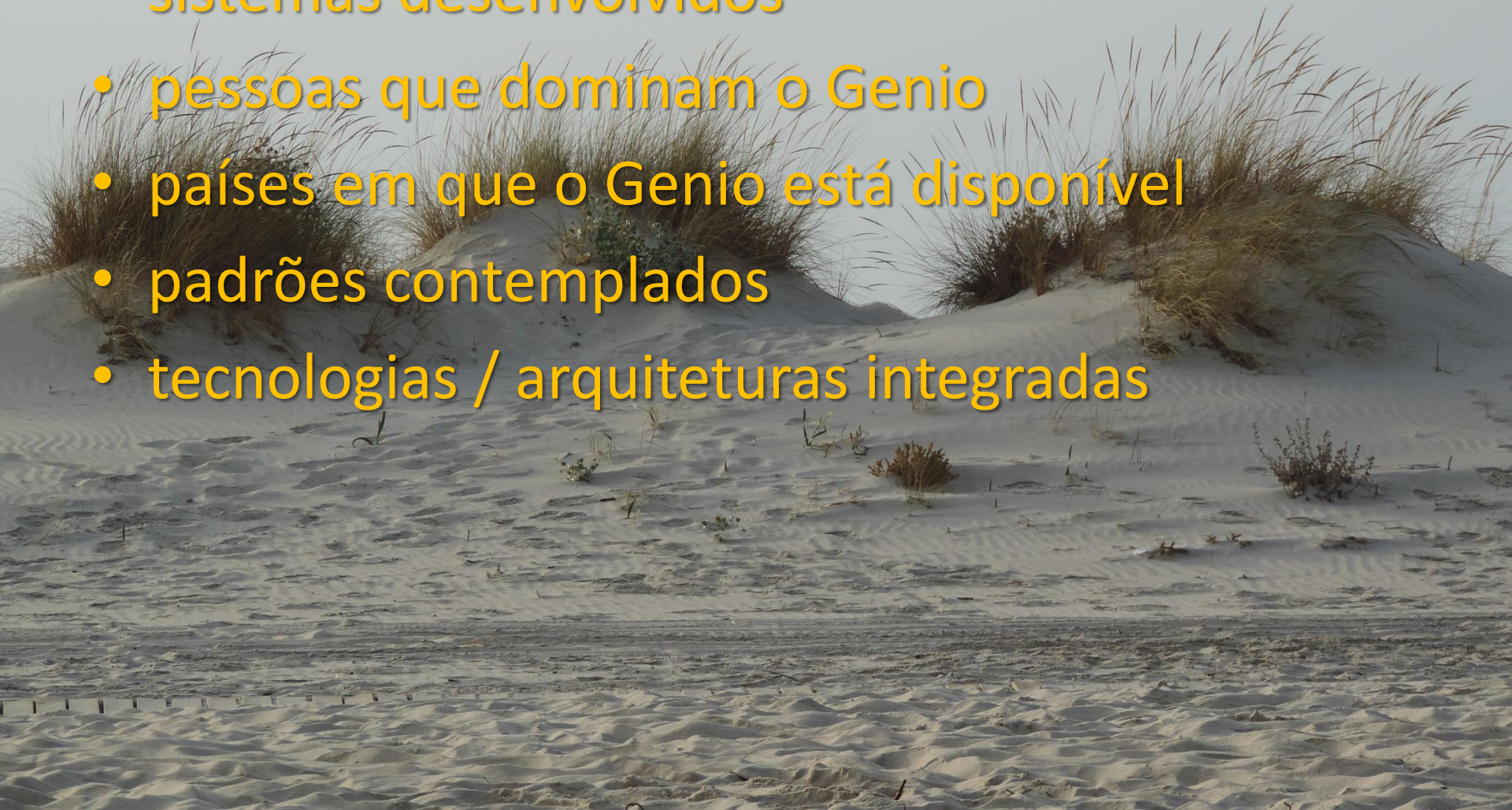
- A maior parte dos engenheiros informáticos formados no mundo é desperdiçada a escrever código que poderia ser automatizado, com base em padrões

Os Padrões são melhoráveis!

- **First Sight Model**, financiado pelo QREN, vai tornar ainda mais fácil o desenvolvimento com padrões (200 horas de investigação, com o apoio do Inov e da GTE)
- **Certificação CMMI** em curso

Os Padrões são escaláveis!

- sistemas desenvolvidos
- pessoas que dominam o Genio
- países em que o Genio está disponível
- padrões contemplados
- tecnologias / arquiteturas integradas



QuidNet

- Para tudo isto, precisamos e queremos ter parceiros

Quem quer ir longe, vai acompanhado!



Sistemas desenvolvidos

670 soluções diferentes em produção
10 áreas de desenvolvimento
Um conjunto muito bom de referências

Pessoas que dominam o Genio

Percurso de especialização em Genio

Documentação e Suporte

Parceiro especializado Actual Training

Engenho4All

Países em que o Genio está presente

Estamos em todos os continentes
Integramos as comitativas oficiais
Empresas Quidgest em 6 países

Padrões contemplados

First Sight Model e QSearch
Extensão do Modelo Genio
Modelação de mais alto nível
Visualização do Modelo

Tecnologias / arquiteturas integradas

Vigilância Tecnológica

Aplicações móveis (especialmente Android)

Automotive, Raspberry / Arduino