



# THE DEVELOPER'S CONFERENCE

## **Expressão Regular - Cookbook**

**Mario Guedes**  
Artesão de Software

**Um bom código  
deve ser  
autoexplicativo  
e, óbvio, sem  
comentários**

"Good code should be self explanatory and obvious without comments."

Regular expressions:



# RegEx é Vida!



- Poderosa ferramenta para manipulação de texto
- *“Tão simples quanto um comando de busca e tão poderoso quanto uma linguagem de programação”*  
- Jeffrey E. F. Friedl
- Presente em todas as linguagens e ferramentas de desenvolvimento



# Temos 2 atores



- **Máquina RegEx:** É a implementação da RegEx – onde temos vários “sabores”. O Delphi adota a PCRE: *Expressão Regular Compatível com Perl*
- **Linguagem hospedeira:** É a linguagem que oferece e interage com os recursos de RegEx.  
No Delphi temos a unit: `System.RegularExpressions`



# Basicamente são 3 pilares



- O **padrão** a ser combinado – A Expressão Regular propriamente dita
- O **texto** a ser analisado – Um texto alvo
- Os **modificadores** – Sinalizadores que modificam o comportamento da máquina RegEx



# Tem que ter “caractere” 😊



THE  
DEVELOPER'S  
CONFERENCE

- Uma regex é definida à partir de pequenas unidades de blocos de montagem:
  - **Meta-caractere:** Caracteres **com** função especial dentro da regex
  - **Caractere literal:** Caracteres **sem** função especial dentro da regex



# Os meta-caracteres

. ? \* + ^ \$ | [ ] { } ( ) \ -

- **Representantes:** Representa um caractere, como o ponto ou uma lista
- **Quantificadores:** Determina a quantidade de ocorrências de um átomo
- **Âncoras:** Combina com uma determinada posição no texto alvo com o circunflexo
- **Diversos:** Todo o resto





THE  
DEVELOPER'S  
CONFERENCE

# EXEMPLOS PRÁTICOS



arrayof.io/DelphiRegEx.pdf



THE  
DEVELOPER'S  
CONFERENCE



5 estrelas por favor 😊



<https://thedeveloperconf.com/tdc/2019/avaliacao>





# THE DEVELOPER'S CONFERENCE