

# Introdução as Expressões Regulares

Processamento de Linguagem Natural

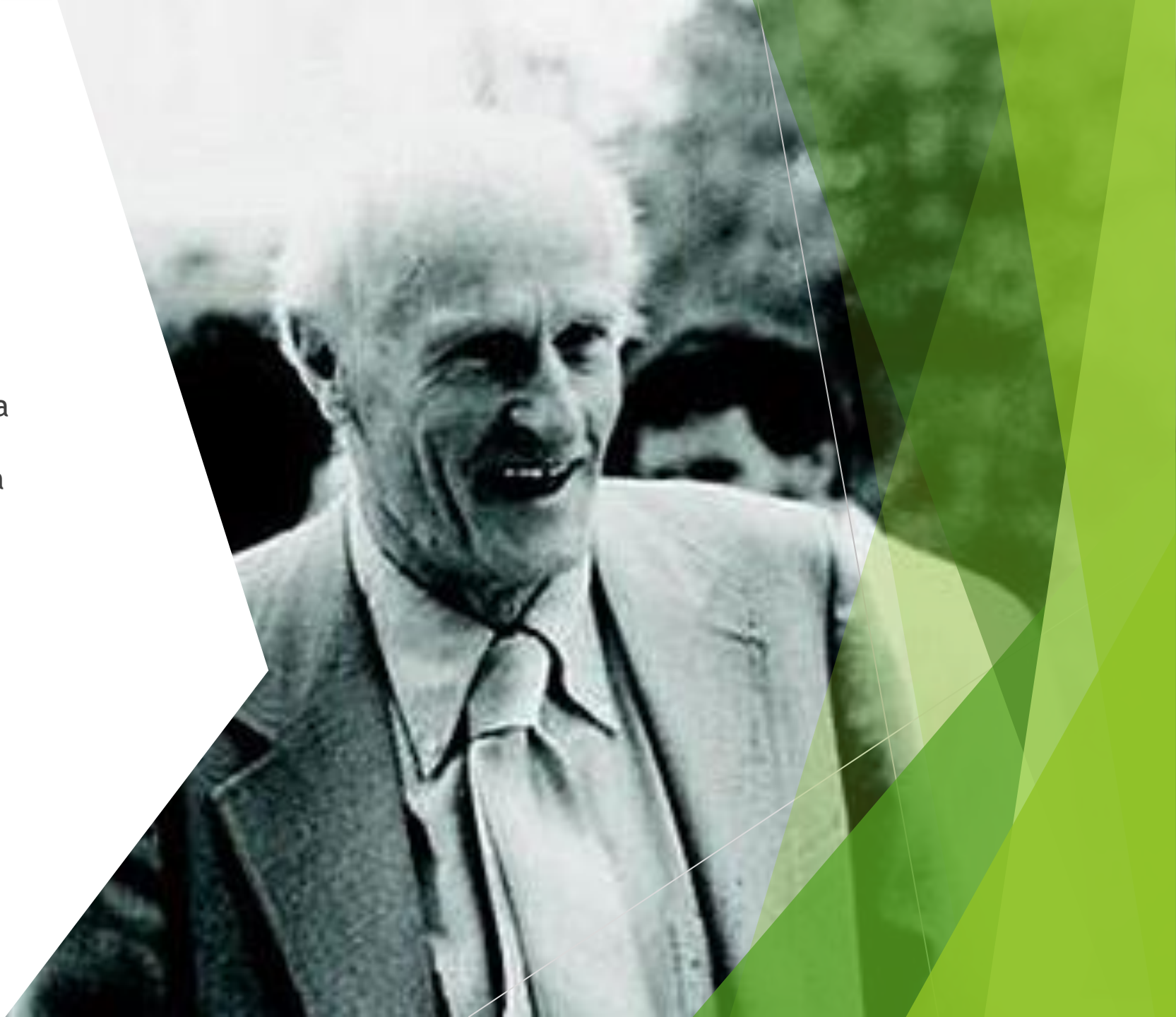
Prof. Leandro Alvim, D. Sc.

# Agenda

- ▶ O que são expressões regulares ?
- ▶ Aplicações
- ▶ Motivação
- ▶ Regras da Linguagem
- ▶ Exemplos
- ▶ Ferramentas
- ▶ Exercícios

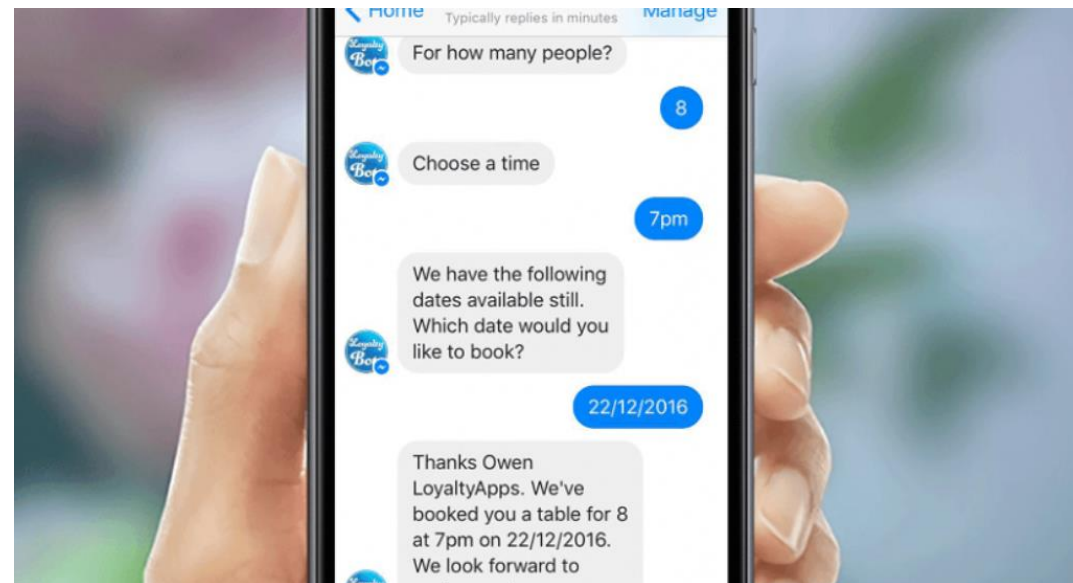
# O que são expressões regulares?

- ▶ Expressão regular (regex)
  - ▶ Notação formal expressa numa sequência de caracteres para descrever um padrão de busca
  - ▶ Linguagem Regular, Stephen Cole Kleene em 1950
  - ▶ A partir de 1980
    - ▶ Apareceram muitas variantes
    - ▶ Unix



# Aplicações

- ▶ Motores de busca
- ▶ Editores de texto
- ▶ Chat Bots
- ▶ E-mails + Agenda
- ▶ ...





# Motivação

- ▶ O que importa é a informação
- ▶ Como podemos buscar por *tempero*?
  - ▶ *Tempero*
  - ▶ *Tempeiro*
  - ▶ *Tenpero*
  - ▶ *Tempero*
  - ▶ *tempeiros*



# Motivação

- ▶ Como buscar por um padrão de *sentimento* a partir da escrita?
- ▶ Twitter, Whats app, ...
  - ▶ *Kkkk, kkkkkkk*
  - ▶ *Huaha, uhauhaha*



# Motivação



Buscar palavras que terminam com as letras *nte*

*Infelizmente, mente, contente, ...*



Buscar por números de telefone, cpf, ...

xxxxx-xxxx, xxx.xxx.xxx-xx, ..



Buscar por palavras duplicadas no texto

para para, a a, ...



Buscar por data

dd/mm/yyyy, d/m/yyyy, yyyy-mm-dd, ...



Extração de informação em Html

<abbrev>Dudu</abbrev>

# Regras da Linguagem

Conjunto	Significado
\d ou [0-9]	Dígito
\D ou [^0-9]	Não dígito
\s	Espaço
\S	Não espaço
\w ou [a-zA-Z0-9_]	Letra ou dígito
\W ou [^a-zA-Z0-9_]	Não Letra ou dígito
.	Qualquer caracter (exceto \n)
\b	Borda



# Regras da Linguagem

Símbolo/Operador	Significado
[]	Conjunto de caracteres
()	Expressão
^	Início da linha ou negação ([^ ])
\$	Fim da linha
	Ou lógico
-	Intervalo de caracteres
+	Pelo menos uma ocorrência do caracter anterior
*	Zero ou mais ocorrências do caracter anterior
?	Torna opcional o caracter anterior
{N}	N ocorrências

# Exemplos

Símbolo/Operador	Padrão	Busca por
+	K+	K KK KKK ...
*	KK*	K kk kkk ...
?	Tempei?ro	Tempero Tempeiro
.	Berin.ela	Berinjela Beringela

# Exemplos

Símbolo/Operador	Padrão	Busca por
[]	[Oo]bra	Obra obra
	Obra obra	Obra obra
[] e -	[a-zA-Z]a	aa ba ca da ... Aa Ba Ca ..
[] e ^	[^0-9]	<i>Qualquer caracter, menos dígitos</i>

# Exemplos

Símbolo/Operador	Padrão	Busca por
\d e {}	\d{1,3}	<i>Números da forma: x, xy ou xxz.</i>
\d e {}	\d{1,}	<i>Números com pelo menos um dígito</i>
\w e {}	\w{2,}	<i>duas ou mais letras ou dígitos</i>
\w e \d	[\w\d]{1,}	<i>letras soltas ou números ou palavras ou alfa numéricos</i>

# Exemplos

Símbolo/Operador	Padrão	Busca por
^	^Obra	<i>Uma linha que inicie com a palavra “Obra”</i>
\$	\.\$	<i>Pontos que aparecem no final de cada linha</i>
\w e \$ e {}	\w{1,}\.\$	Palavras ou dígitos no final de cada linha que terminam com ponto
\w e \$ e {} e ?	\w{1,}\.?\$	Palavras ou dígitos no final de cada linha que podem ou não terminar com ponto

# Ferramentas

## ► Online

► <https://regex101.com/>

The screenshot displays the regex101.com interface. The top navigation bar includes the site name, a user profile icon, a donate button, a contact link, a bug reports & feedback link, and a wiki link. The left sidebar contains a 'SAVE & SHARE' section with a 'Save Regex' button (ctrl+s), a 'FLAVOR' section with options for PCRE (PHP), ECMAScript (JavaScript), Python (selected with a green checkmark), and Golang, and a 'TOOLS' section with a 'Code Generator' button. The main area is divided into three panels: 'REGULAR EXPRESSION' with the input `[0o]bra` and a status bar indicating '2 matches, 119 steps (~106ms)'; 'TEST STRING' with the text 'Custo da obra de R\$ 50.690,10 com início em 10/2/2019 e término em 10/4/2019. Obra realizada pelo José da Silva, cpf:637.892.888-45, cel: (21)99876-6483.'; and 'EXPLANATION' which details the match process, including a list of characters for `[0o]` and a section on global pattern flags (`g` for global, `m` for multi-line). The 'MATCH INFORMATION' panel at the bottom right shows two matches: Match 1 (Full match, 9-13, obra) and Match 2 (Full match, 78-82, Obra).

# Ferramentas

- ▶ Python

- ▶ Módulo re (<https://docs.python.org/2/library/re.html>)

- ▶ Findall
    - ▶ Match
    - ▶ Search
    - ▶ Sub
    - ▶ Split



# Ferramentas

## ► Python

- Módulo re (<https://docs.python.org/2/library/re.html>)
  - Findall: retorna uma lista de strings com todos os casamentos sem sobreposição do padrão
  - Exemplo

```
>>> import re
>>> text = """Custo da obra de R$ 50.690,10 com início em 10/2/2019 e término em 1/04/2019.
... Obra realizada pelo José da Silva, cpf:637.892.888-45, cel: (21)99876-6483.
... """
>>> re.findall("[oO]bra",text)
['obra', 'Obra']
```

# Ferramentas

- ▶ Python

- ▶ Módulo re (<https://docs.python.org/2/library/re.html>)

- ▶ Match: retorna um objeto caso haja um casamento do padrão apenas no início da string

- ▶ Exemplo

```
>>> re.match("Custo",text)
<_sre.SRE_Match object; span=(0, 5), match='Custo'>
>>> re.match("Custo",text).group()
'Custo'
```

# Ferramentas

## ► Python

### ► Módulo re (<https://docs.python.org/2/library/re.html>)

- Search: retorna um objeto caso haja um casamento do padrão em qualquer ponto da string. Uma única vez.

### ► Exemplo

```
>>> re.search("[Oo]bra",text)
<_sre.SRE_Match object; span=(9, 13), match='obra'>
>>> re.search("[Oo]bra",text).group()
'obra'
```

# Ferramentas

## ▶ Python

- ▶ Módulo re (<https://docs.python.org/2/library/re.html>)
  - ▶ Sub: A partir de um padrão e uma string de origem, substitui este na string alvo
  - ▶ Exemplo

```
>>> re.sub("[Oo]bra", "construção", text)
'Custo da construção de R$ 50.690,10 com início em 10/2/2019 e término em 1/04/2019.\nconstrução realizada pelo José da Silva,
```

# Ferramentas

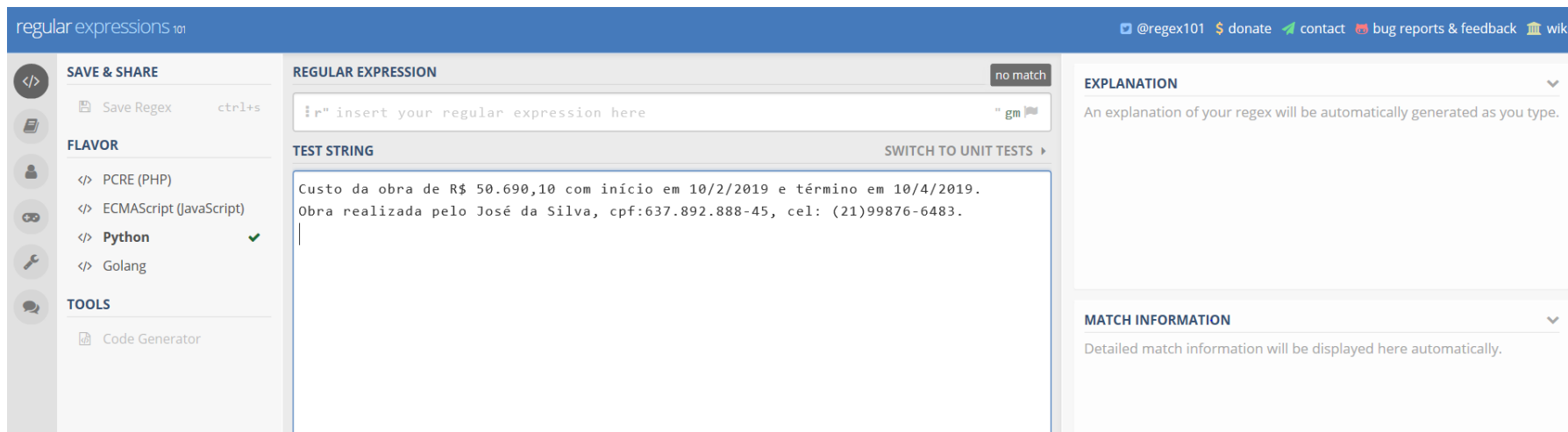
- ▶ Python
  - ▶ Módulo re (<https://docs.python.org/2/library/re.html>)
    - ▶ Split: Separa o texto de acordo com o padrão informado
    - ▶ Exemplo

```
>>> re.split("\W+",text)
['Custo', 'da', 'obra', 'de', 'R', '50', '690', '10', 'com', 'início', 'em', '10', '2', '2019', 'e', 'término', 'em', '1', '04']
```

# Exercícios

## ► Online

- Qual padrão para encontrarmos números de telefone ?
- Qual padrão para localizarmos cpf ?
- Qual padrão para localizarmos datas em vários formatos ?
- Qual padrão para localizarmos um nome ?
- Qual padrão para localizarmos dinheiro em qualquer valor ?



# Exercícios

- ▶ Python
  - ▶ Copie a *string* anterior para uma variável denominada *texto*
  - ▶ Utilize *findall* com os padrões desenvolvidos do exercícios anterior e mostre os resultados
  - ▶ Exemplo base

```
import re

text = """Custo da obra de R$ 50.690,10 com início em 10/2/2019 e término em 1/04/2019.
Obra realizada pelo José da Silva, cpf:637.892.888-45, cel: (21)99876-6483.
"""

resultados = re.findall("[oO]bra",text)

print("resultados da busca: ", resultados)
```