

TI 3001 C

Analítica de datos y herramientas de inteligencia artificial

Expresiones regulares

Tecnológico de Monterrey



¿Qué son las expresiones regulares?

En este tema vamos a ver como extraer información de texto usando las expresiones regulares.

Las **expresiones regulares** son:

- Son secuencias de caracteres que especifican un **patrón de búsqueda**.
- Son una **fórmula** para buscar dentro de texto ciertas coincidencias.
- Son **patrones** utilizados para encontrar una determinada combinación de caracteres dentro de una cadena de texto, gracias a ellos se puede extraer información importante.



¿Qué son las expresiones regulares?

Las **expresiones regulares** se usan para buscar, contar, reemplazar y / o validar ciertos patrones de texto.

Ejemplos:

- Validar un correo electrónico, eliminar espacios dobles.
- Extraer información como correos, teléfonos, fechas, etc.



¿Cómo usar las expresiones regulares?

Para usar las expresiones regulares se necesita:

1. **La expresión regular**
2. **El texto a manipular.**



¿Qué pasa si quiero extraer de un directorio de 100 personas sus correos?

Los correos son distintos, cómo podríamos crear una fórmula para extraer todos los correos o modificar la información.

Laura Sanchez
442 153-32-42
lasanch@hotmail.com
<http://www.laurasan.com>

Pedro Lopez
448 342 33 12
pedrolo54@gmail.com
<https://www.pedro.lopez.com.mx>

Violeta Perez
443 214 43 72
violeta45@outlook.com
violeta.net

$(\cdot *)$ Exp[resio]nes
regul[are]s

```
/[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z]+/g
```

¿Qué pasa si quiero extraer de un directorio de 100 personas sus correos?

Las expresiones regulares nos van a permitir crear fórmulas para **extraer información**. No solamente podemos extraer información de los correos, sino los teléfonos o las páginas web.

```
/[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z]+/g
```

```
Carlos•Arturo→|↵  
449•123•45•67↵  
carlos_@hotmail.com↵  
www.carlos.com↵  
↵  
Manuel•Alejandro↵  
448-234-56-78↵  
alejandro@outlook.com↵  
https://www.manuel.alejandro.com.mx  
http://alejandro.com.mx↵
```

(*) Exp[re]siones
regul[ar]es

Expresiones regulares

Las expresiones regulares no solamente nos sirven para **extraer información**, sino también son importantes para **validar información**. Por ejemplo: Un correo electrónico.

Página para **validar expresiones regulares**, me permite escribirlas y validar las coincidencias que encuentra:

regexr.com

(*****) Exp[re]siones
regul[ares]

Módulo regex

El lenguaje de programación Python, en su librería estándar, nos proporciona el modulo **regex** el cual es utilizado para trabajar con expresiones regulares.

findall(): Encuentra todos los subtextos donde coincide la expresión regular y devuelve estas coincidencias como una lista. **(patrón, texto)**

Ejemplo:

```
1 import re
2 texto = input("Introduce un mensaje: ")
3 print(re.findall("is", texto)) # (Expresión regular, texto)
```

Shell ×

```
>>> %Run ER_Match.py
```

```
Introduce un mensaje: She is my sister Lis
['is', 'is', 'is']
```


¿Cómo funcionan las expresiones regulares?

Las expresiones regulares son una fórmula va a buscar dentro de nuestro texto coincidencias. Que es una coincidencia, que el texto de arriba, sea el mismo de abajo.

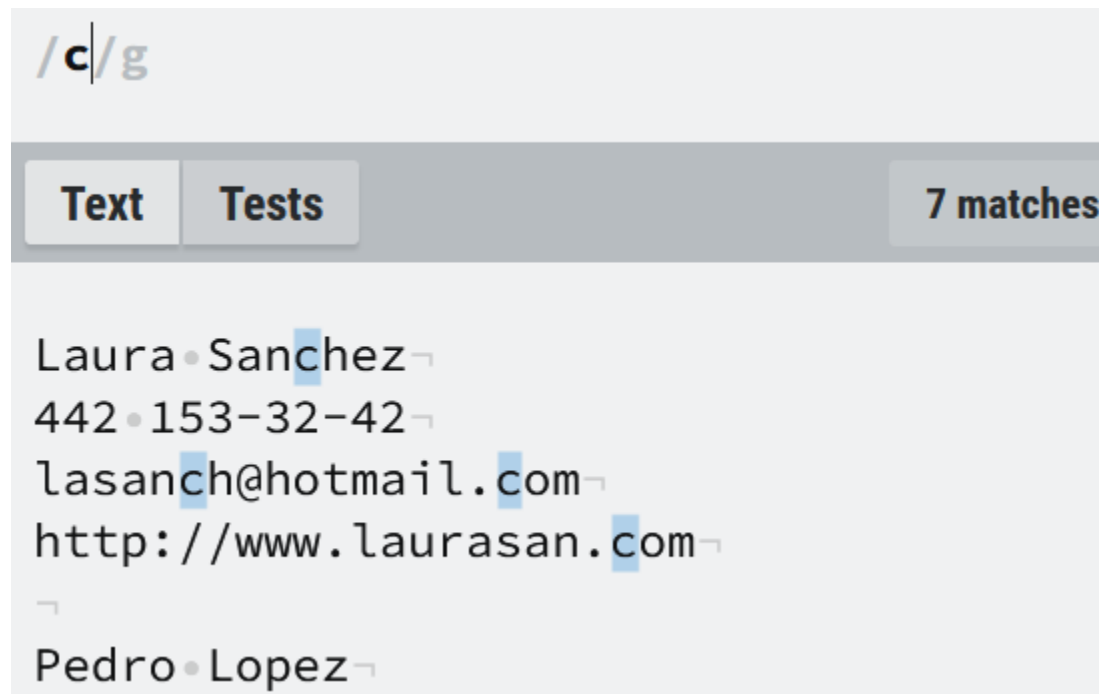
```
/[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z]+/g
```

```
Carlos•Arturo→  
449•123•45•67  
carlos_@hotmail.com  
www.carlos.com  
  
Manuel•Alejandro  
448-234-56-78  
alejandros@outlook.com  
https://www.manuel.alejandros.com.mx  
http://alejandros.com.mx
```

(. *) Exp[re]siones
regul[ares]

Expresiones regulares

Por ejemplo, coloco la letra **c**. Lo que hace la expresión regular es buscar todas las coincidencias de la letra **c**. Encuentra 7 coincidencias.



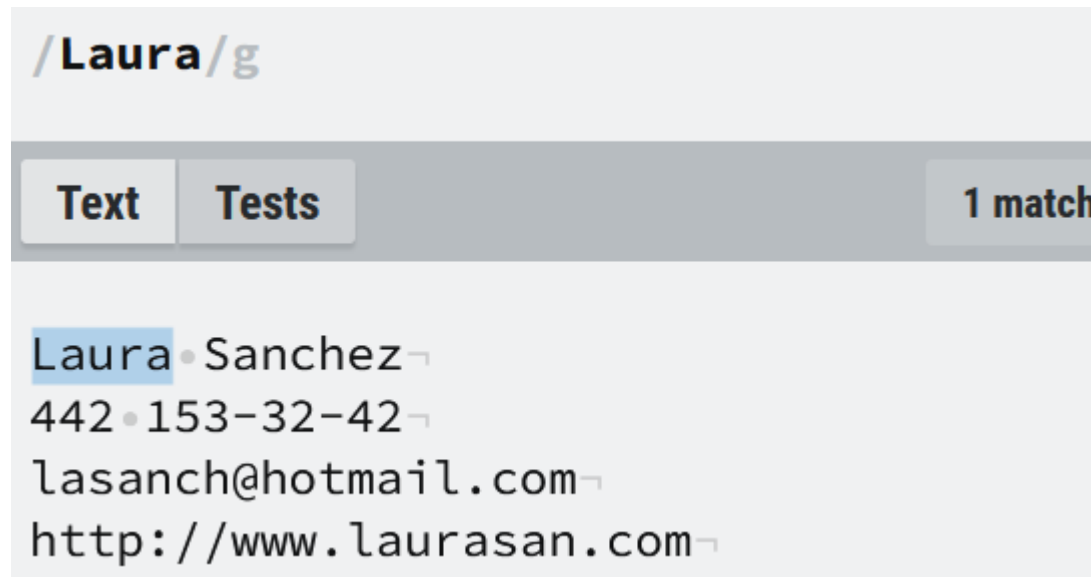
The image shows a screenshot of a web-based regular expression testing tool. At the top, the search pattern `/c/g` is entered. Below the input field, there are two tabs: **Text** and **Tests**. To the right of these tabs, it says **7 matches**. The main area displays the following text with blue highlights indicating the matches for the letter 'c':

```
Laura•Sanchez↵  
442•153-32-42↵  
lasanch@hotmail.com↵  
http://www.laurasan.com↵  
↵  
Pedro•Lopez↵
```

Expresiones regulares

Por ejemplo, coloco **Laura**. Lo que hace la expresión regular es buscar la letra L seguida de la letra a, luego la letra u , r y a. Encuentra 1 coincidencia. Busca carácter por carácter.

* **g** es una bandera, estamos haciendo una búsqueda global. Si le quitamos el global, solamente busca la primera coincidencia.



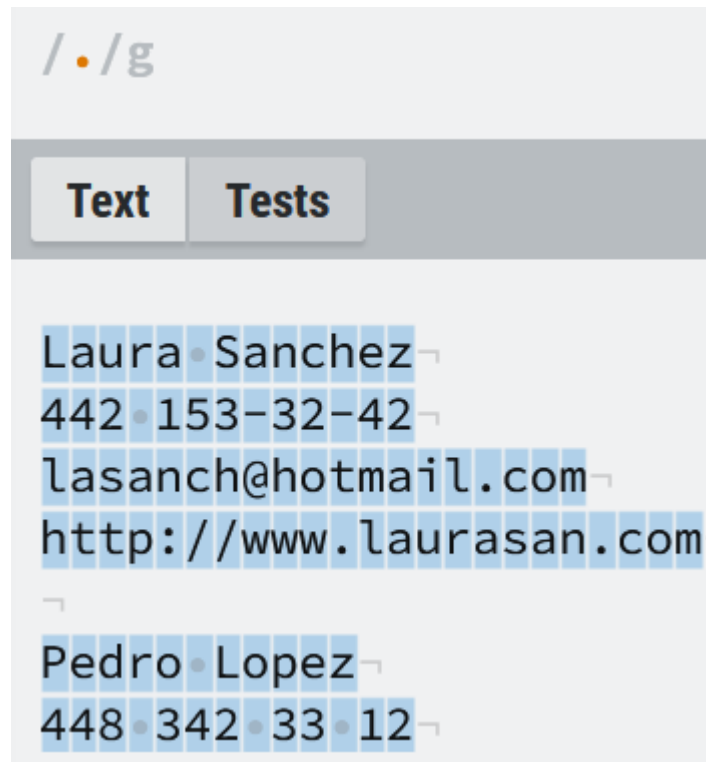
Metacaracteres

- . Cualquier caracter excepto nueva línea.
- \d** Dígitos (0-9)
- \D** No dígitos (0-9)
- \w** Caracter de palabra (a-z, A-Z, 0-9, _)
- \W** No caracter de palabra
- \s** Espacio en blanco (espacio, tab, nueva línea)
- \S** No espacio en blanco (espacio, tab, nueva línea)
- ** Cancela caracteres especiales
- ^** Inicio de una cadena de caracteres (string)
- \$** Fin de una cadena de caracteres

Metacaracteres

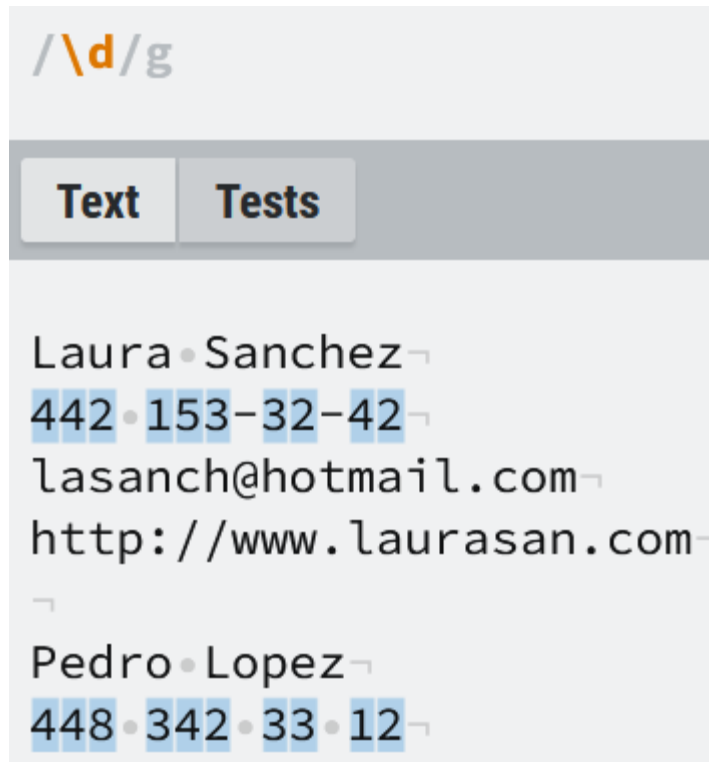
Punto(.) Cualquier carácter excepto el salto de línea

- Encuentra todos los caracteres.



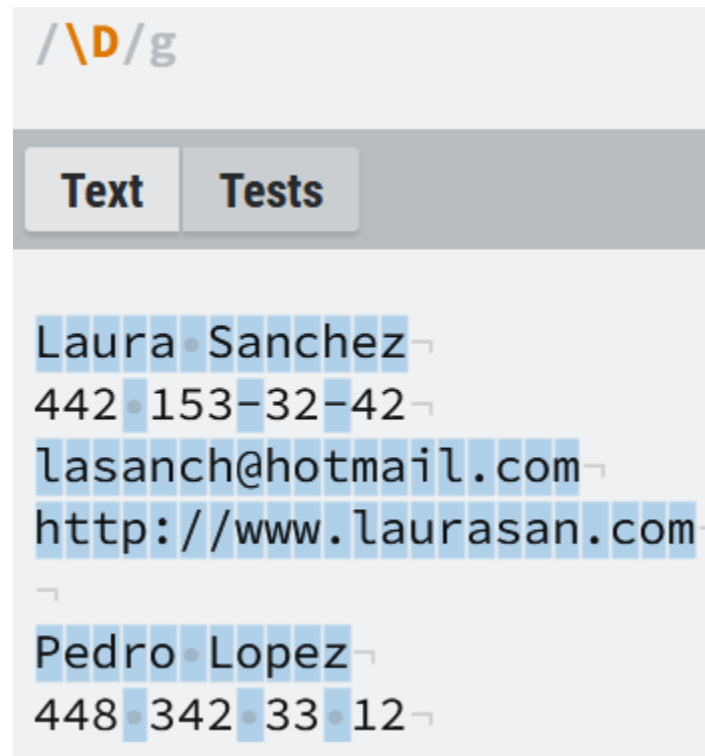
Metacaracteres

(\d) Dígitos (0 – 9)



Metacaracteres

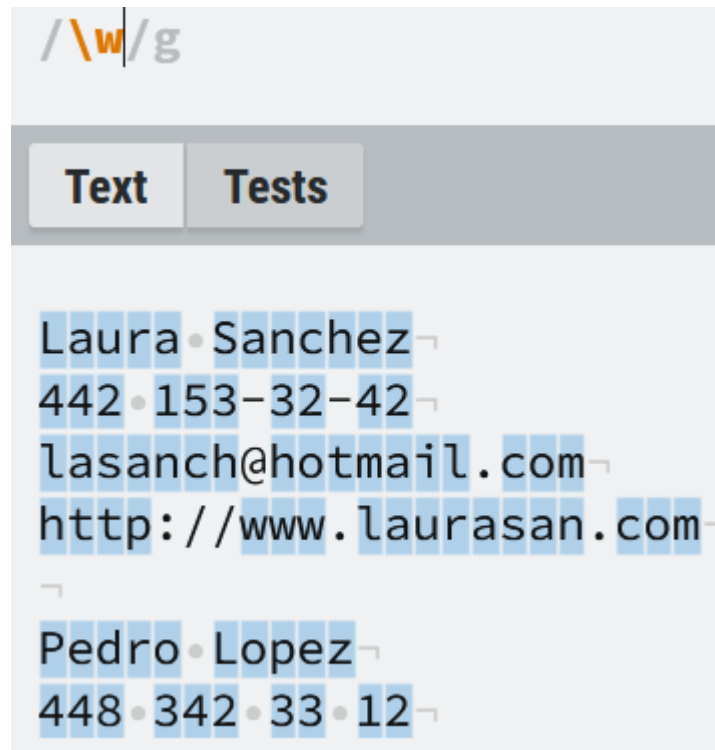
(\D) Todo lo que no sea un número



Metacaracteres

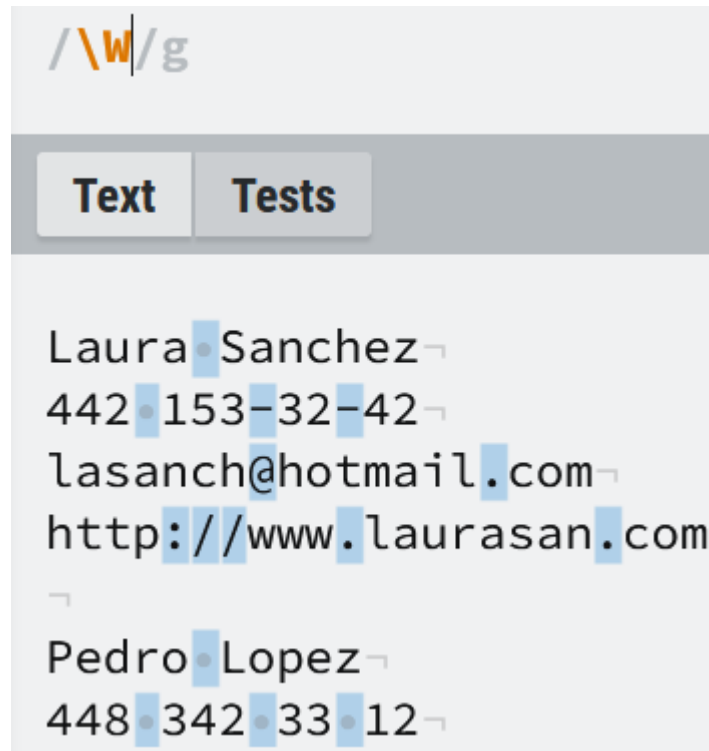
(\w) Caracter de palabra (a-z, A-Z, 0-9, _)

Busca de la a-z, A-Z, 0-9 y guion bajo.



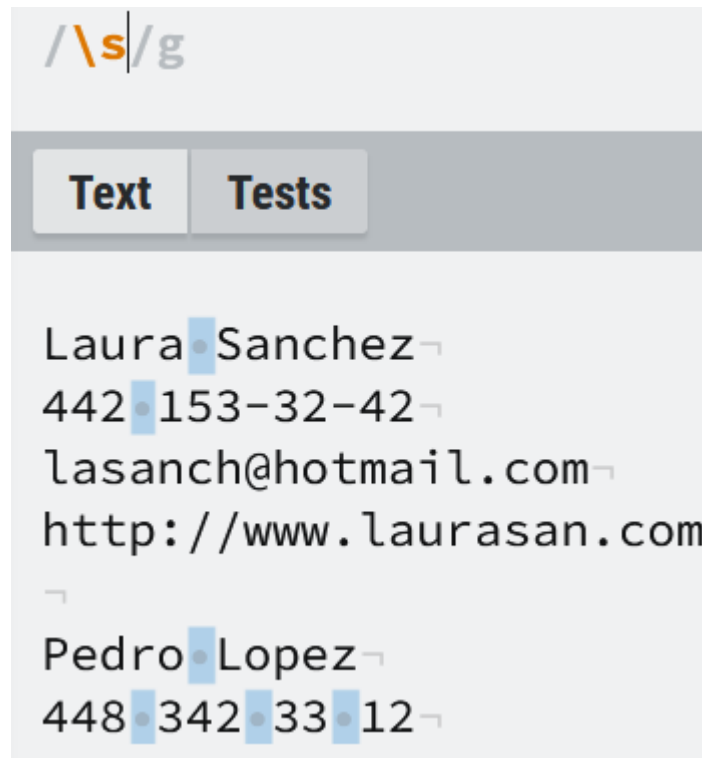
Metacaracteres

(\W) No es un caracter de palabra



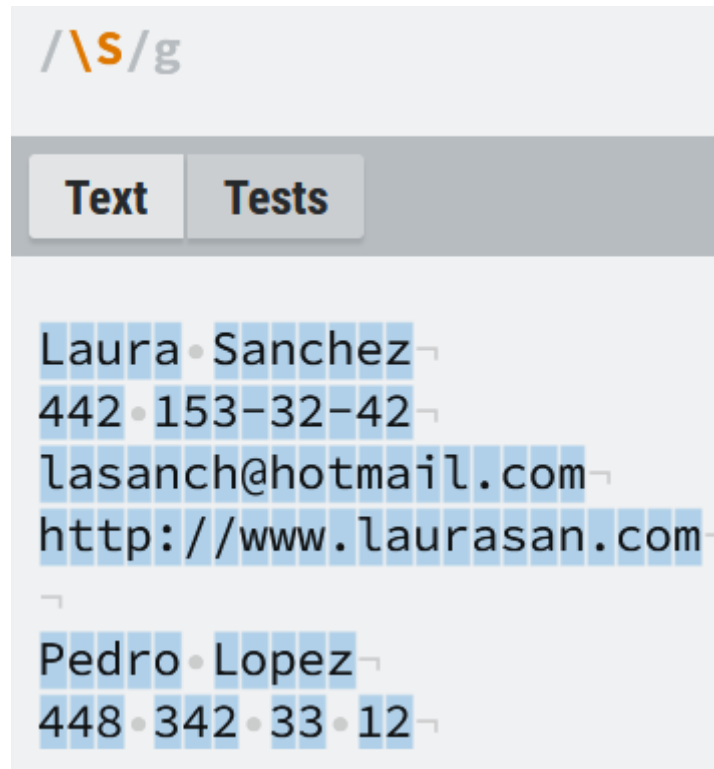
Metacaracteres

(\s) Espacios, tabulaciones y nuevas líneas



Metacaracteres

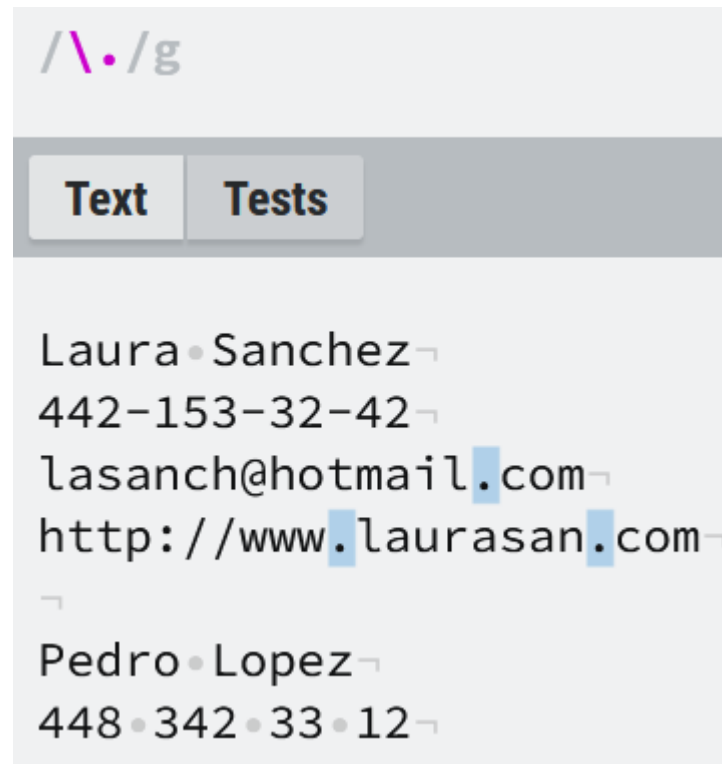
(\S) No espacio en blanco, tab y nueva línea



Metacaracteres

(\) Cancela caracteres especiales

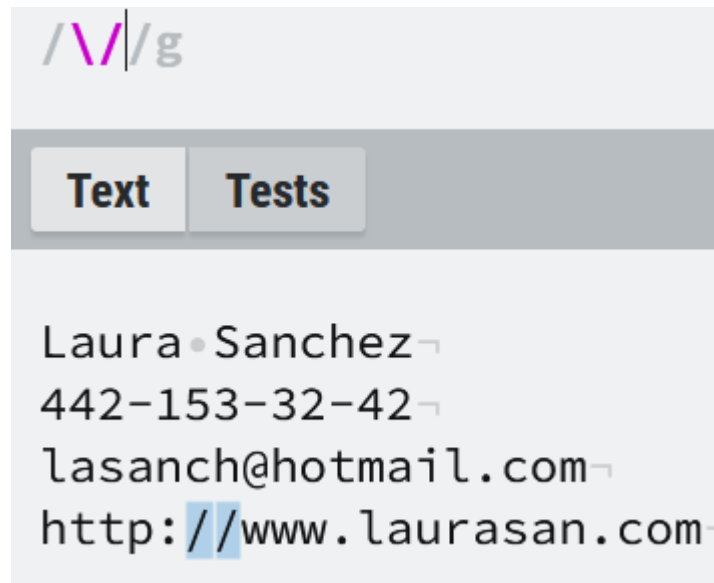
Si quiero encontrar un punto en el texto. Cancela carácter especial punto.



Metacaracteres

(\) Cancela caracteres especiales

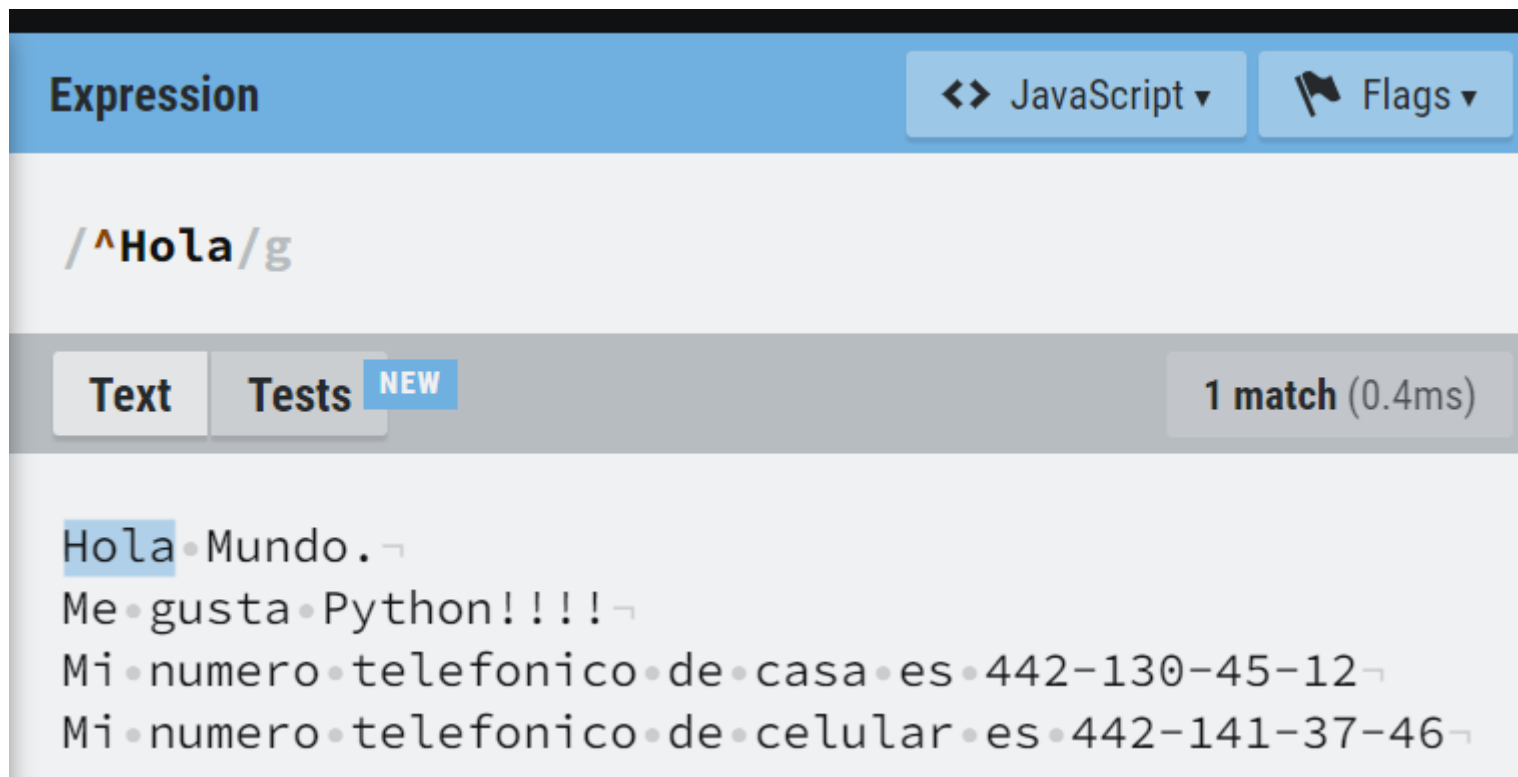
- Si quiero encontrar una diagonal.
- Casi todos los símbolos le tenemos que poner **backslash **, ya que son metacaracteres o símbolos especiales. Ya que los símbolos significan algo.



Metacaracteres

(^) Inicio de una cadena de caracteres

- Si quiero que una línea empiece con la palabra “Hola”



Metacaracteres

(\$) Fin de una cadena de caracteres

- Si quiero que una línea termine con “Mundo.”

```
/Mundo.$/g
```

Text

Tests

Hola•Mundo.↵

Me•gusta•Python!!!!↵

Mi•numero•telefonico•de•casa•es•442-130-45-12↵

Mi•numero•telefonico•de•celular•es•442-241-37-46↵

Mi•numero•telefonico•de•oficina•es•442-380-14-22

Metacaracteres

(\$) Fin de cadena de caracteres

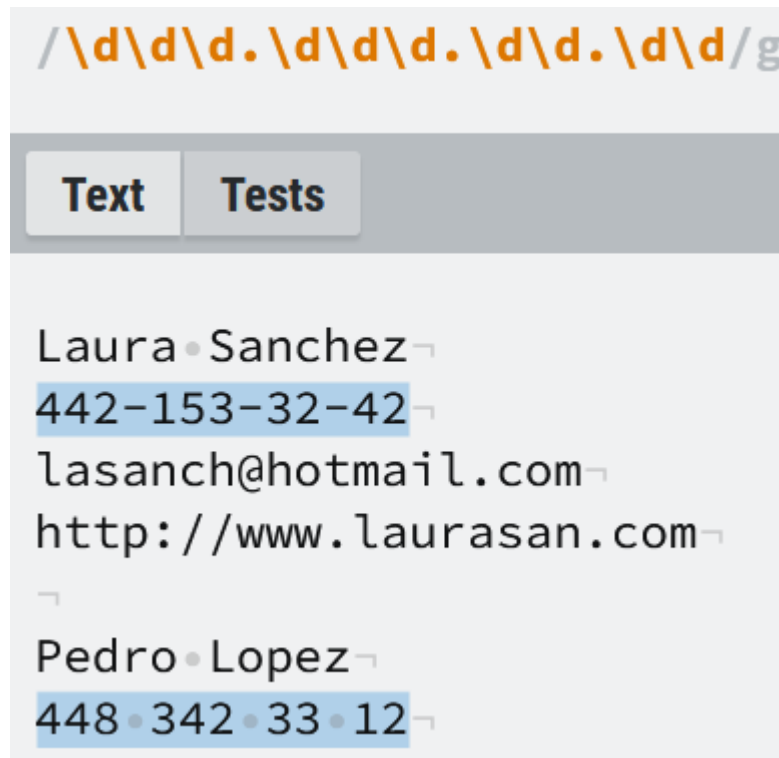
- Si quiero que una línea termine con “Mundo.”
- Agregar flag **m** para tomar texto multilínea. Que lea cada línea por separado.

The screenshot shows a regex testing tool. The expression `/Mundo.$/gm` is entered in the top field. Below it, there are tabs for 'Text' and 'Tests' (with a 'NEW' button). The 'Text' tab is active, showing a multi-line input: 'Hola.Mundo.', 'Me.gusta.Python!!!!', 'Mi.numero.telefonico.de.casa.es.4', and 'Mi.numero.telefonico.de.celular.e'. To the right, a panel titled 'Expression Flags' with a help icon (?) lists the flags: 'g'lobal (checked with a blue checkmark), 'c'ase 'i'nsensitive (checked with a grey checkmark), 'm'ultiline (checked with a blue checkmark), 's'ingle line (dotall) (checked with a grey checkmark), and 'u'nicode (checked with a grey checkmark).

Expresión regular

Ejemplo: Extraer todos los números telefónicos.

Primero tres dígitos juntos, luego espacio o guion, otros tres dígitos, espacio o guion, luego dos y dos. El punto involucra cualquier carácter excepto salto de línea. `\d\d\d.\d\d\d.\d\d.\d\d`



Cuantificadores

Estos símbolos representan cuantas veces se repiten los caracteres

***** **0 o más**

+ **1 o más**

? **0 o 1**

{3} **Numero exacto**

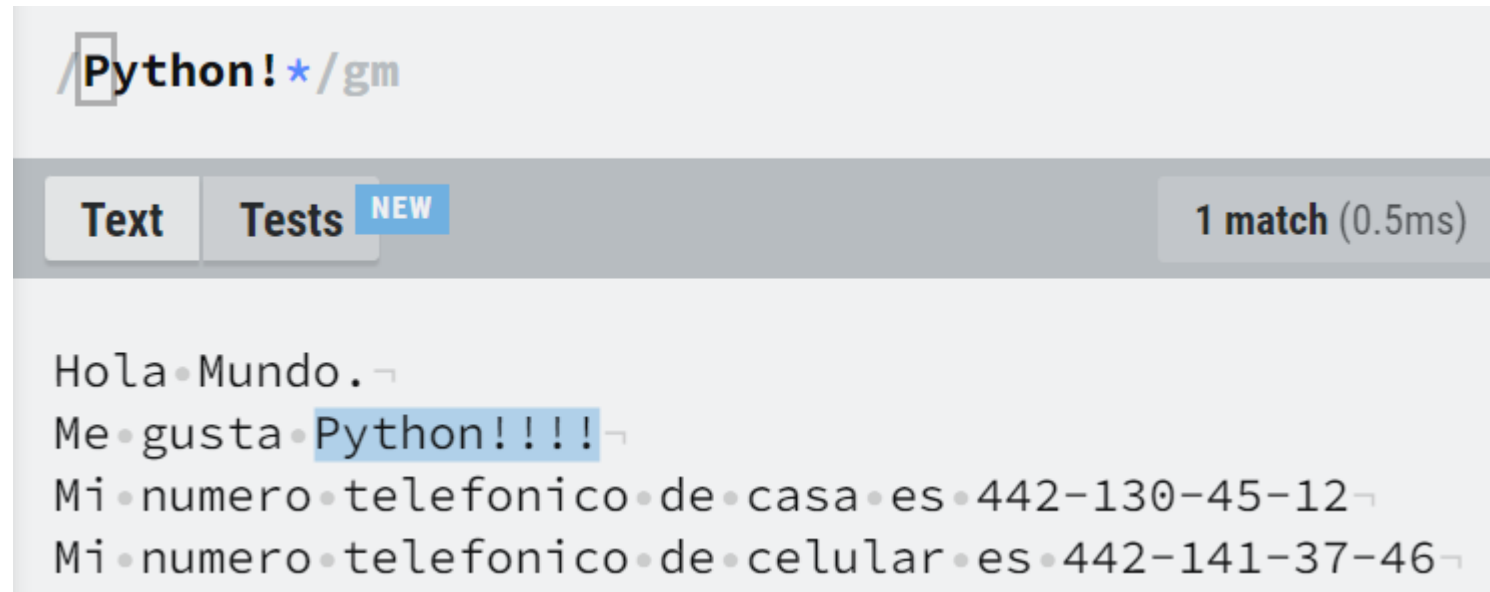
{n,} **Numero n+**

{3,4} **Rango de números (Mínimo, Máximo)**

Cuantificadores

(*) Cero o más veces

La admiración se repita 0 o más veces



The screenshot shows a web-based regex testing tool. At the top, the search bar contains the pattern `/Python!*/gm`. Below the search bar, there are two tabs: **Text** and **Tests**, with the **Tests** tab being active and marked with a **NEW** badge. To the right of the tabs, it displays **1 match (0.5ms)**. The main area shows four lines of text, each followed by a right arrow icon. The second line, `Me gusta Python!!!!`, has the text `Python!!!!` highlighted in blue, indicating a successful match.

```
/Python!*/gm
```

Text **Tests** **NEW** **1 match (0.5ms)**

Hola Mundo. →
Me gusta Python!!!! →
Mi numero telefonico de casa es 442-130-45-12 →
Mi numero telefonico de celular es 442-141-37-46 →

Cuantificadores

(+) Una o más veces

La admiración se repita 1 o más veces

The screenshot shows a web-based regex testing tool. At the top, the regex pattern `/Python!+/gm` is entered. Below the input, there are two tabs: **Text** and **Tests**, with the **Tests** tab being active and marked with a **NEW** badge. To the right of the tabs, it displays **1 match (0.6ms)**. The main area shows four lines of text with the matches highlighted in blue:

- Hola•Mundo.↵
- Me•gusta•Python!!!!↵
- Mi•numero•telefonico•de•casa•es•442-130-45-12↵
- Mi•numero•telefonico•de•celular•es•442-141-37-46↵

Cuantificadores

(?) 0 o 1 vez

La admiración se repita 0 o 1 vez

```
/Python! ? /gm
```

Text Tests **NEW** 1 match (0.3ms)

```
Hola•Mundo.↵  
Me•gusta•Python!!!!↵  
Mi•numero•telefonico•de•casa•es•442-130-45-12↵  
Mi•numero•telefonico•de•celular•es•442-141-37-46↵
```

Cuantificadores

{n} Número n exacto

Busca la cantidad exacta de elementos. Va a identificar a la palabra **Python** mas dos signos de admiración.

```
/Python!{2}/gm
```

Text

Tests

NEW

1 match (0.5ms)

Hola•Mundo.↵

Me•gusta•Python!!!!↵

Mi•numero•telefonico•de•casa•es•442-130-45-12↵

Mi•numero•telefonico•de•celular•es•442-141-37-46↵

Cuantificadores

{n, } Número n o más elementos

Busca n o más elementos del caracter.

```
/Python!{2,}/gm
```

Text

Tests

NEW

1 match (0.5ms)

Hola•Mundo.↵

Me•gusta•Python!!!!↵

Mi•numero•telefonico•de•casa•es•442-130-45-12↵

Mi•numero•telefonico•de•celular•es•442-141-37-46↵

Cuantificadores

{min, max } Rango de números mínimo y máximo

Siempre va a ir por la mayor cantidad

```
/Python!{2,3}/gm
```

Text **Tests** **NEW** **1 match** (0.4ms)

Hola•Mundo.↵
Me•gusta•Python!!!!↵
Mi•numero•telefonico•de•casa•es•442-130-45-12↵
Mi•numero•telefonico•de•celular•es•442-141-37-46↵

Expresión regular

Ejemplo: Extraer los nombres del directorio con dos textos

\w Caracter, dígito o guion bajo.

+ Uno o más caracteres

\s Espacio

^ Al inicio de la línea

\$ Al final de la línea

Al inicio de la cadena de texto, encuentres caracteres de palabra, uno o más, un espacio, caracteres de palabra, uno o más al final de la cadena de texto.

```
/^\\w+\\s\\w+$/gim
```

Text

Tests

```
Laura • Sanchez ↵  
442-153-32-42 ↵  
lasanch@hotmail.com ↵  
http://www.laurasan.com ↵  
Pedro • Lopez ↵  
448 • 342 • 33 • 12 ↵
```

Expresión regular

Ejemplo 2: Extraer los nombres del directorio con dos textos y espacio.

\w Caracter o guion bajo.

+ Uno o más caracteres

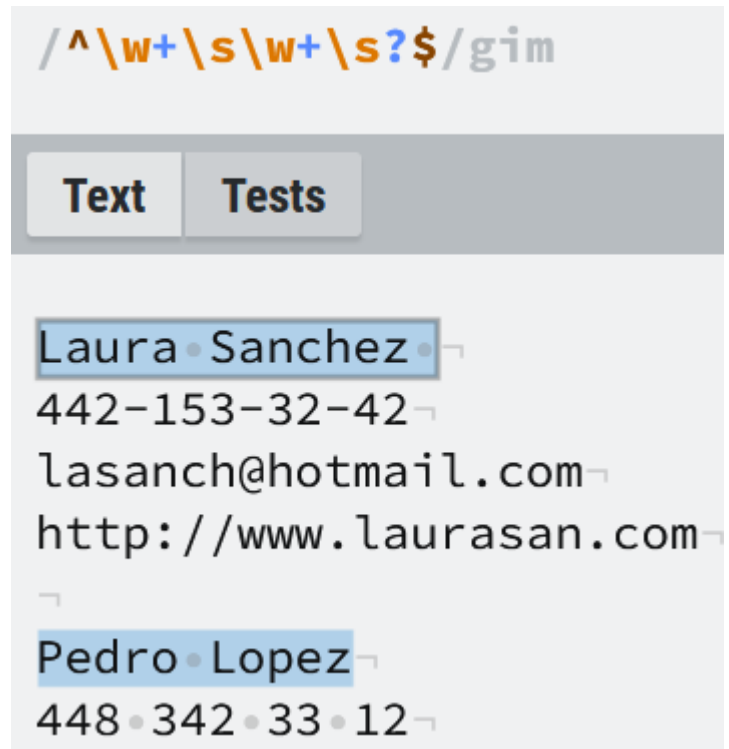
\s Espacio, tabulador o salto de línea

^ Al inicio de la línea

\$ Al final de la línea

? O o más

Al inicio de la cadena de texto, encuentres caracteres de palabra, uno o más, un espacio, caracteres de palabra, uno o más, un espacio cero o más veces al final de la cadena de texto.



Expresión regular

Ejemplo 3: Extraer los nombres del directorio con dos textos o uno.

\w Caracter o guion bajo.

+ Uno o más caracteres

\s Espacio

^ Al inicio de la línea

\$ Al final de la línea

? O o más

Al inicio de la cadena de texto, encuentres caracteres de palabra, uno o más, un espacio, caracteres de palabra, uno o más, un espacio cero o más al final de la cadena de texto.

\s?\w+? Opcional espacio y carácter de palabra.

```
/^\\w+\\s?\\w+?\\s?$|/gim
```

Text

Tests

Laura Sanchez

442-153-32-42

lasanch@hotmail.com

http://www.laurasan.com

Pedro

448-342-33-12

pedrolo54@gmail.com

https://www.pedro.lopez.com.mx

Violeta Perez

443-214-43-72

violeta45@outlook.com

violeta.net

Expresión regular

```
/^\\w+\\s?\\w+?\\s?$ /gim
```

Text **End.** Matches the end of the string, or the end of a line if the multiline flag (m) is enabled.

Laura Sanchez

442-153-32-42

lasanch@hotmail.com

http://www.laurasan.com

Pedro

448-342-33-12

pedrolo54@gmail.com

https://www.pedro.lopez.com.mx

Violeta Perez

443-214-43-72

violeta45@outlook.com

violeta.net

Grupos

- [] Encuentra caracteres en corchetes
- [^] Encuentra caracteres que no están dentro de corchetes
- | Condicional O
- () Grupos

Grupos

() Grupos

Ejemplo: Identificar números telefónicos y agrupar la lada.

The screenshot shows a regex testing interface. At the top, the regex pattern `/(\d{3})-\d{3}-\d{2}-\d{2}/gm` is entered. Below the pattern, there are tabs for 'Text' and 'Tests', with 'Tests' being the active tab. The test text contains several lines, including phone numbers. Two phone numbers are highlighted with blue boxes: '442-130-45-12' and '442-141-37-46'. A tooltip points to the second number, displaying the match and range information. The 'Tools' section at the bottom left is also visible.

```
/(\d{3})-\d{3}-\d{2}-\d{2}/gm
```

Text Tests **NEW**

Hola • Mundo. ↵
Me • gusta • Python!!!! ↵
Mi • numero • telefonico • de • casa • es • 442-130-45-12 ↵
Mi • numero • telefonico • de • celular • es • 442-141-37-46 ↵

match: 442-141-37-46
range: 113-125

Tools

group #1: 442

Grupos

() Grupos

Ejemplo: Identificar los números telefónicos que comiencen con 1 o 2 después de la lada 442.

```
/442-(1|2)\d{2}-\d{2}-\d{2}/gm
```

Text Tests **NEW** 2 matches (0.5ms)

Hola•Mundo.↵
Me•gusta•Python!!!!↵
Mi•numero•telefonico•de•casa•es•442-130-45-12↵
Mi•numero•telefonico•de•celular•es•442-241-37-46↵
Mi•numero•telefonico•de•oficina•es•442-380-14-22↵

Grupos

[] Encuentra caracteres en corchetes

Ejemplo: Identificar los números telefónicos que comiencen con 1 o 3 después de la lada 442.

```
/442-[13]\d{2}-\d{2}-\d{2}/gm
```

Text Tests **NEW** 2 matches (0.3m)

Hola•Mundo.↵
Me•gusta•Python!!!!↵
Mi•numero•telefonico•de•casa•es•442-130-45-12↵
Mi•numero•telefonico•de•celular•es•442-241-37-46↵
Mi•numero•telefonico•de•oficina•es•442-380-14-22↵

Grupos

[] Encuentra caracteres en corchetes

Ejemplo: Identificar las letras y dígitos.

The screenshot shows a regex testing tool interface. At the top, the regex pattern `/[a-zA-Z0-9]/gm` is entered. Below the pattern, there are tabs for 'Text' and 'Tests', with 'Tests' being the active tab and marked as 'NEW'. To the right of the tabs, it says '136 matches (0.5ms)'. The main area displays five lines of text with individual characters highlighted in blue boxes, indicating they are matches for the pattern. The text is: 'Hola Mundo.', 'Me gusta Python!!!!', 'Mi numero telefonico de casa es 442-130-45-12', 'Mi numero telefonico de celular es 442-241-37-46', and 'Mi numero telefonico de oficina es 442-380-14-22'.

```
/[a-zA-Z0-9]/gm
```

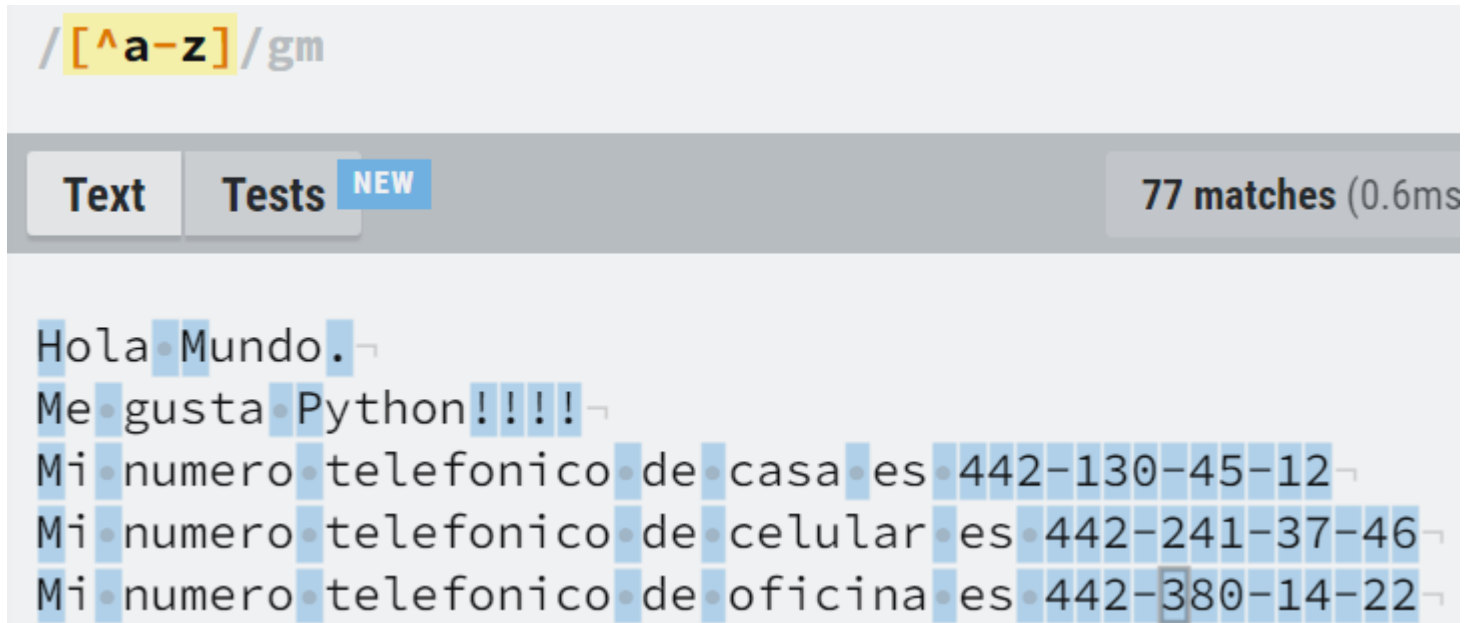
Text Tests **NEW** 136 matches (0.5ms)

Hola•Mundo.↵
Me•gusta•Python!!!!↵
Mi•numero•telefonico•de•casa•es•442-130-45-12↵
Mi•numero•telefonico•de•celular•es•442-241-37-46↵
Mi•numero•telefonico•de•oficina•es•442-380-14-22↵

Grupos

[^] Encuentra caracteres no están dentro de corchetes

Ejemplo: Identificar todos los caracteres que no son letras minúsculas.



The screenshot shows a regex testing interface. At the top, the pattern `/[^a-z]/gm` is entered. Below the pattern, there are tabs for 'Text' and 'Tests', with 'Tests' being the active tab and marked as 'NEW'. To the right of the tabs, it says '77 matches (0.6ms)'. The main area displays the text being tested, with individual characters highlighted in blue boxes to indicate matches. The text is: 'Hola Mundo. Me gusta Python!!!! Mi numero telefonico de casa es 442-130-45-12 Mi numero telefonico de celular es 442-241-37-46 Mi numero telefonico de oficina es 442-380-14-22'. The matches are all non-lowercase characters: spaces, periods, exclamation marks, hyphens, and digits.

```
/[^a-z]/gm
```

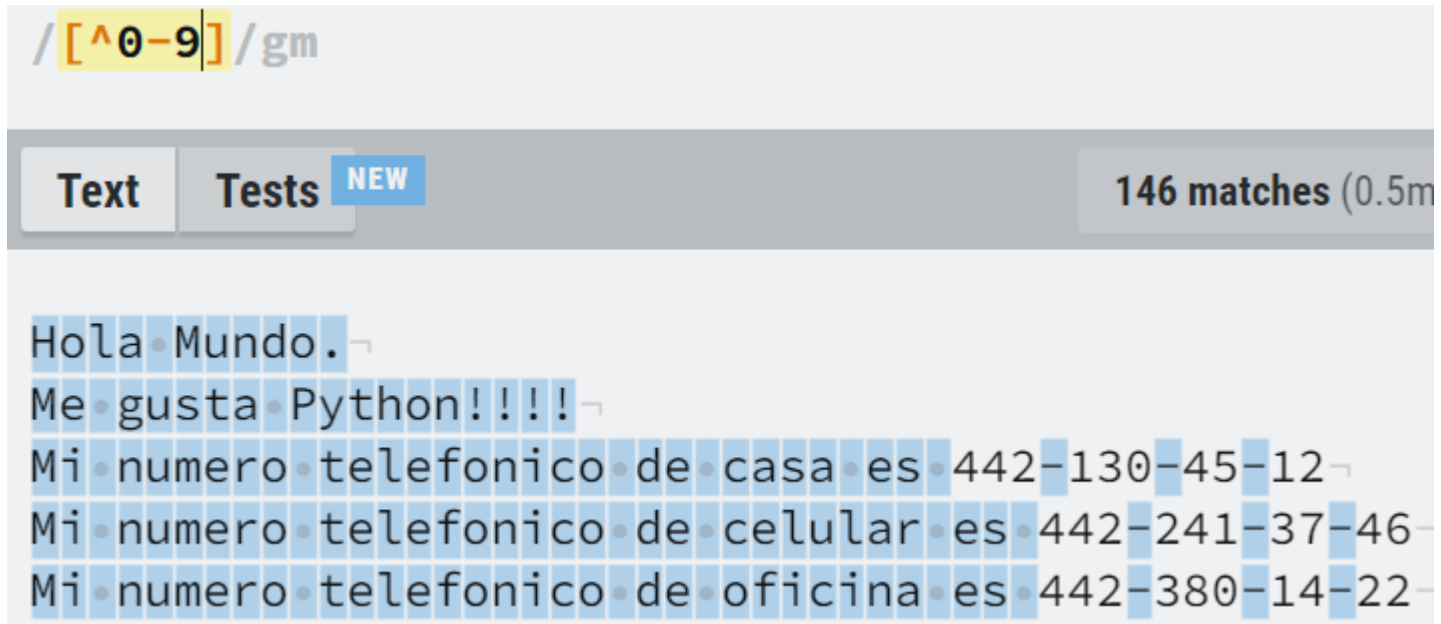
Text Tests **NEW** 77 matches (0.6ms)

Hola Mundo. Me gusta Python!!!! Mi numero telefonico de casa es 442-130-45-12 Mi numero telefonico de celular es 442-241-37-46 Mi numero telefonico de oficina es 442-380-14-22

Grupos

[^] Encuentra caracteres que no están dentro de corchetes

Ejemplo: Identificar todos los caracteres que no son dígitos.



The screenshot shows a web-based regex testing interface. At the top, the regex pattern `/[^0-9]/gm` is entered. Below the input field, there are tabs for 'Text' and 'Tests', with a 'NEW' button next to 'Tests'. To the right, it says '146 matches (0.5m)'. The main area displays the text 'Hola Mundo. Me gusta Python!!!! Mi numero telefonico de casa es 442-130-45-12 Mi numero telefonico de celular es 442-241-37-46 Mi numero telefonico de oficina es 442-380-14-22' with each character highlighted in a blue box. The non-digit characters (spaces, punctuation, and letters) are highlighted, demonstrating the matches found by the regex.

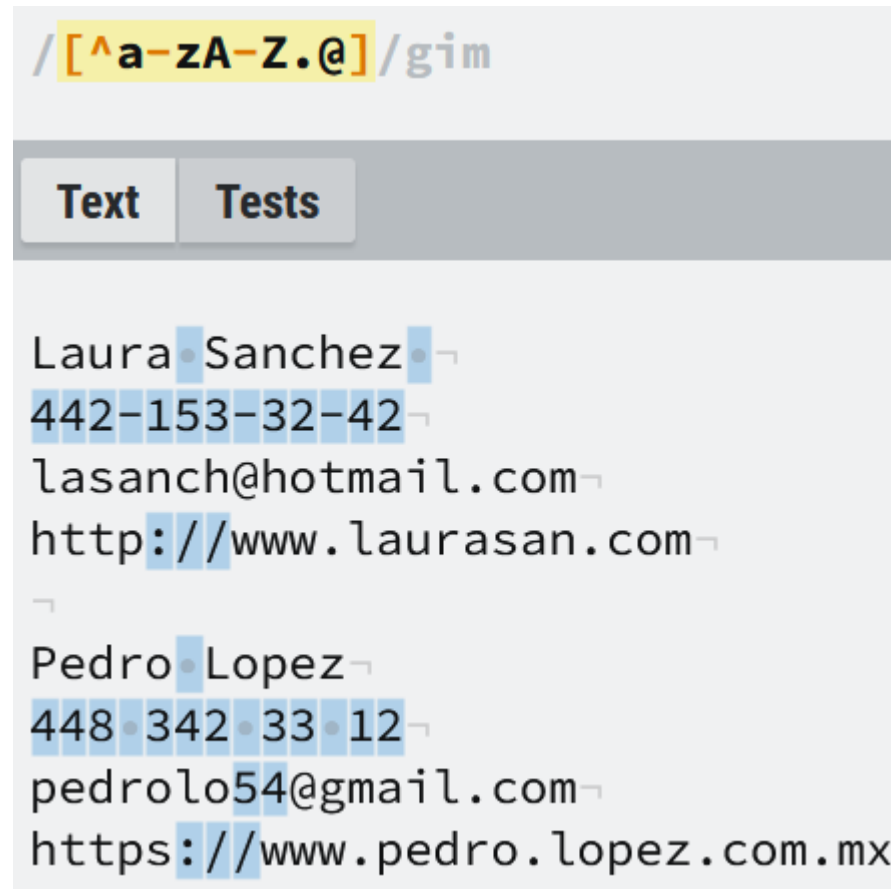
Grupos

- **[0-5]+** En los grupos **no** es necesario el uso del **slash ** para los metacaracteres como el punto.
- **[a-zA-Z.@]** Encuentra caracteres del abecedario en mayúsculas o minúsculas, puntos, arrobas.



Grupos

- `[^a-zA-Z.@]` No caracteres del abecedario en mayúsculas o minúsculas, puntos, arrobas.



Grupos

() Nos permiten comparar entre algunos valores

Ejemplo 1: Extraer todos los números telefónicos con espacio o guion.

Primero tres dígitos juntos, luego espacio o guion, otros tres dígitos, espacio o guion, luego dos y dos.

```
/\d\d\d(\s|-)\d\d\d(\s|-)\d\d(\s|-)\d\d/g
```

Text

Tests

```
Laura • Sanchez ↵  
442-153-32-42 ↵  
lasanch@hotmail.com ↵  
http://www.laurasan.com ↵  
↵  
Pedro • Lopez ↵  
448 • 342 • 33 • 12 ↵
```

Grupos

Ejemplo 2: Extraer todos los números telefónicos haciendo uso de grupos y cuantificadores.

Tres dígitos, luego un espacio o guion, otros tres dígitos, espacio o guion, luego dos dígitos, espacio o guion y dos dígitos.

```
/\d{3}(\s|-)\d{3}(\s|-)\d{2}(\s|-)\d{2}/gim
```

Text	Tests
Laura • Sanchez	442-153-32-42
lasanch@hotmail.com	http://www.laurasan.com
Pedro • Lopez	448-342-33-12
pedrolo54@gmail.com	https://www.pedro.lopez.com.mx
Violeta • Perez	443-214-43-72
violeta45@outlook.com	violeta.net

Grupos

() Nos permiten comparar entre algunos valores

Ejemplo 3: Encuentre los número de teléfono con ladas 442, 443 y 448 solamente.

Conjunto de caracteres **2, 3 u 8**.

```
/44[238][\s-.]\\d{3}[\s-.]\\d{2}[\s-.]\\d{2}
```

Text	Tests
Laura • Sanchez	
442-153-32-42	
lasanch@hotmail.com	
http://www.laurasan.com	
Pedro • Lopez	
448-342-33-12	
pedrolo54@gmail.com	

```
/44[238][.-.]\\d{3}[-.-.]\\d{2}[-.-.]\\d{2}
```

Text	Tests	NEW
Laura • Sanchez		
442-153-32-42		
laura.sanchez@hotmail-mx.com		
http://www.laurasan.com		
Pedro • Lopez		
448-342-33-12		
pedro-lopez54@gmail.com		
https://www.pedro.lopez.com.mx		

Expresión regular

Ejercicio: Seleccionar correo electrónico.

[a-zA-Z] una o más letras.

`/\w+@\w+\.\com/gm`

Text Tests

Laura • Sanchez •
442-153-32-42
lasanch@hotmail.com
http://www.laurasan.com

Pedro • Lopez
448 • 342 • 33 • 12
pedrolo54@gmail.com
https://www.pedro.lopez.com.mx

`/\w+@\w+\.[a-zA-Z]+/gm`

Text Tests

Laura • Sanchez •
442-153-32-42
lasanch@hotmail.com
http://www.laurasan.com

Pedro • Lopez
448 • 342 • 33 • 12
pedrolo54@gmail.com
https://www.pedro.lopez.com.mx

Expresión regular

Ejercicio: Seleccionar correo electrónico.

```
/[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z]+/gm
```

Text

Tests NEW

Laura•Sanchez↵
442•153•32•42↵
laura.sanchez@hotmail-mx.com↵
http://www.laurasan.com↵
↵
Pedro•Lopez↵
448•342•33•12↵
pedro-lopez54@gmail.com↵
https://www.pedro.lopez.com.mx↵
↵
Violeta↵
443•214•43•72↵
violeta_45@outlook.com↵
violeta.net↵

Expresión regular

Ejercicio: Seleccionar correo electrónico.

```
/[a-zA-Z0-9._\-\+@][a-zA-Z0-9._\-\+@]\.[a-zA-Z]+/gm
```

Text Tests **NEW**

Laura • Sanchez
442 • 153-32-42
laura.sanchez@hotmail-mx.com
http://www.laurasan.com

Pedro • Lopez
448 • 342 • 33 • 12
pedro-lopez54@gmail.com
https://www.pedro.lopez.com.mx

Violeta
443 • 214 • 43 • 72
violeta_45@outlook.com
violeta.net



Gracias

