METRICA

CONSULTING

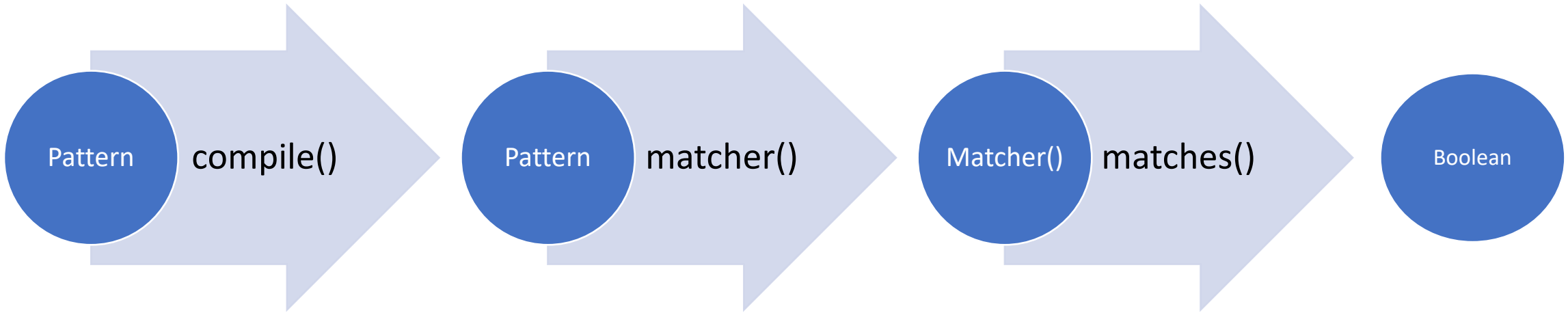
Expresiones regulares

EXPRESIONES REGULARES 01

Expression

```
/([A-Z])\w+/g
```

02 ESTRUCTURA COMPLETA




```
import java.util.regex.*
```

RICA
CONSULTING

02

EJEMPLO

```
Pattern pattern = Pattern.compile("^[0-9]", Pattern.CASE_INSENSITIVE);
Matcher matcher = pattern.matcher("+3491111256");
boolean matchFound = matcher.matches();
```

03

USO DESDE STRING

`boolean`	`matches(String regex)` Tells whether or not this string matches the given regular expression.
`String`	`replaceAll(String regex, String replacement)` Replaces each substring of this string that matches the given regular expression with the given replacement.
`String`	`replaceAll(String regex, String replacement)` Replaces each substring of this string that matches the given regular expression with the given replacement.
`String`	`replaceFirst(String regex, String replacement)` Replaces the first substring of this string that matches the given regular expression with the given replacement.
`String[]`	`split(String regex)` Splits this string around matches of the given regular expression.
`boolean`	`startsWith(String prefix)` Tests if this string starts with the specified prefix.
`String`	`replace(CharSequence target, CharSequence replacement)` Replaces each substring of this string that matches the literal target sequence with the specified literal replacement sequence.

03
SINTAXIS

[ABC]	1 de los contenidos
[^ABC]	cualquier no contenido
[0-9]	uno en el intervalo
[A-Z^[AEIOU]]	Exclusión en el intervalo

03 METACARÁCTERES

| uno u otro

. comodín

^ principio

$ final

\d dígito

\s espacio

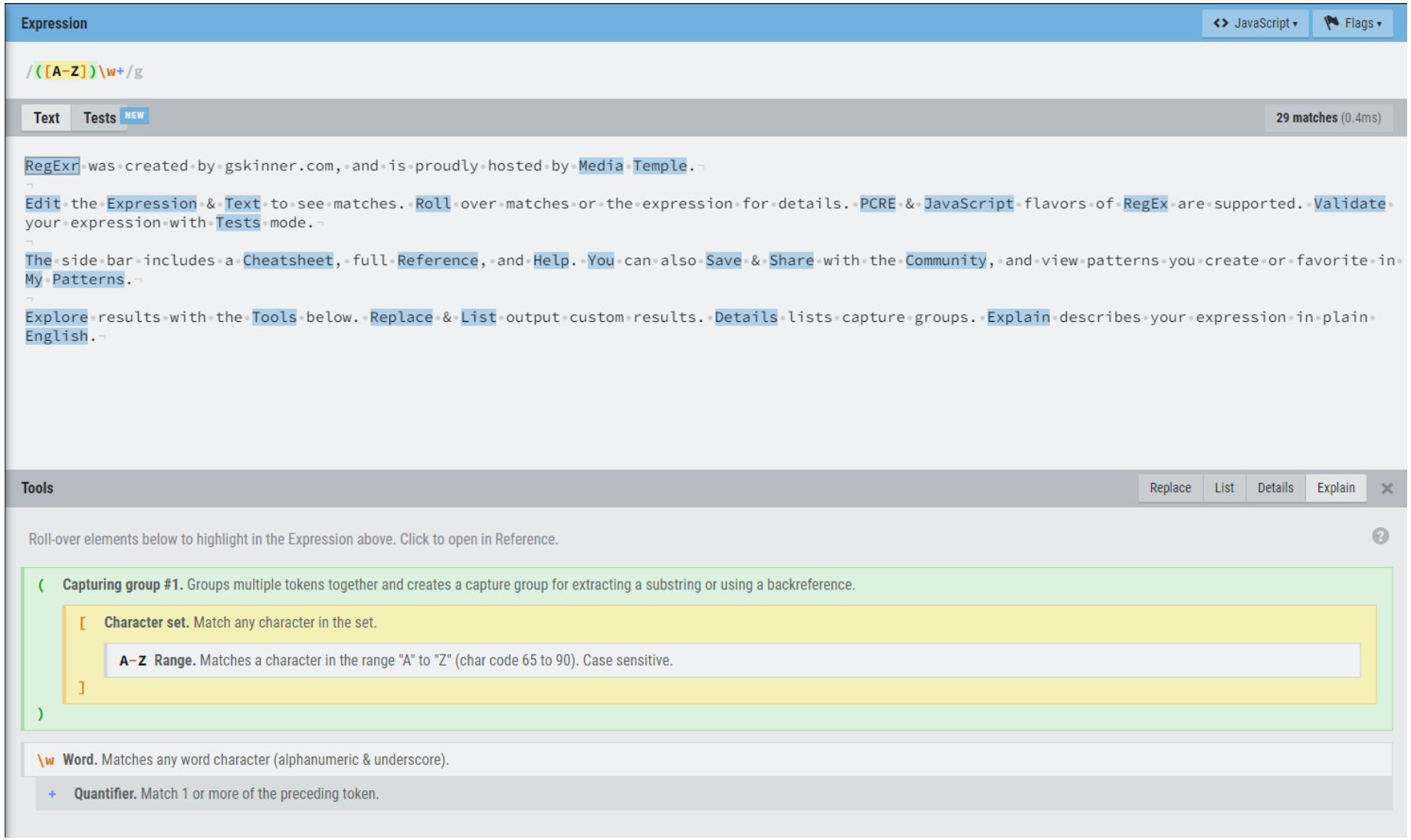
\b principio o final de palabra

\D no es dígito

03 CUANTIFICADORES

n+	al menos uno
n*	cualquier cantidad
n?	0 o 1
n{x}	se repite x veces
n{x,y}	entre x e y veces
n{x, }	al menos x veces

04 regexr.com



04

EJEMPLOS

Expression

```
/^[0-9]{8}[A-Z]$/
```

```
12345678A
A12345678B
00000000B
00000000BA
```

Expression

```
/\b[a-zA-Z](\d|\w|\.)*@\w*\.\w{2,}\b/
```

Tools

`\b` **Word boundary.** Matches a word boundary position between a word character and non-word character or position (start / end of string).

`[` **Character set.** Match any character in the set.

- `a-z` **Range.** Matches a character in the range "a" to "z" (char code 97 to 122). Case sensitive.
- `A-Z` **Range.** Matches a character in the range "A" to "Z" (char code 65 to 90). Case sensitive.

`]`

`(` **Capturing group #1.** Groups multiple tokens together and creates a capture group for extracting a substring or using a backreference.

- `\d` **Digit.** Matches any digit character (0-9).
- `|` **Alternation.** Acts like a boolean OR. Matches the expression before or after the `|`.

05
EJERCICIOS

Una expresión que contiene una letra entre la a y la c

Una expresión que contiene una palabra que empiece por la a o la c

Cualquier cantidad de letras terminadas en un número mayor que 4

No contiene tres números seguidos

RICA TEIII CONSULTING

05

Regexp Basics - is it a eight bit unsigned number?
https://www.codewars.com/kata/567e8f7b4096f2b4b1000005

Regex validate PIN code
https://www.codewars.com/kata/55f8a9c06c018a0d6e000132

Regexp basics - parsing prices
https://www.codewars.com/kata/56833b76371e86f8b6000015

Count the smiley faces!
https://www.codewars.com/kata/583203e6eb35d7980400002a

Credit card issuer checking
https://www.codewars.com/kata/5701e43f86306a615c001868

String incrementer
https://www.codewars.com/kata/54a91a4883a7de5d7800009c