Argentina programa 4.0

Expresiones Regulares

"Desarrollador Java Inicial"

Agenda

Argentina programa 4.0

- Motivación
- Validando un número
- Expresiones regulares
- Regexp 101
- Extraer información

Motivación

Argentina programa 4.0

- Cuando una aplicación toma información del “exterior”, ya sea de otro sistema, de un archivo o input del usuario, no siempre puede ser en cualquier formato
- Números, decimales, teléfonos, direcciones, dinero, fechas, etc...
- En general no es nuestro sistema el que debe adaptarse a cualquier situación, sino que nosotros tenemos el deber de establecer qué datos vamos a aceptar o si tomamos datos de un sistema establecido, relevar cuál es el formato de los mismos

Validando un número

Una primera aproximación a esto puede usar los métodos que ya conocemos y esperar que arrojen una excepción.

```
Integer.parseInt("167")

Float.parseFloat("30,5")

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss z");

ZonedDateTime zonedDateTime = ZonedDateTime.parse("2015-05-05 10:15:30 Europe/Paris", formatter);
```

Con esto pasan 2 cosas:

1) No es muy práctico para manejar a nivel código
2) Puede haber mucha más variedad de formatos de las que se nos ofrece en estas clases

Expresiones Regulares

Argentina programa 4.0

- Modelo para expresar un patrón que debe cumplir una cadena de texto
- Implementado en todos los lenguajes

CBU	41190418135201	[0-9]{22}
	META.MAMUT.NARANJA	[A-Z]+\.[A-Z]+\.[A-Z]+
CUIL	20-33605906-1	[0-9][0-9]\-[0-9]{6,8}\-[0-9]
Monto	137.2	[0-9]+(\.[0-9]{1,2})?

Probemos con:

- Nro de telefono
- email

Character	Brief Description	Simple Example
.	any character (wildcard)	w..kly
^	Begins with	^where
[ ]	character set	[0-9]
|	either or	hi|bye
+	Once or more	me+
*	zero occurances or more	me*
{}	exact number of times	l{4}
\	special character operations	\w
$	Ends with	$bar

Usos en Java: Validación / Búsqueda

Argentina programa 4.0

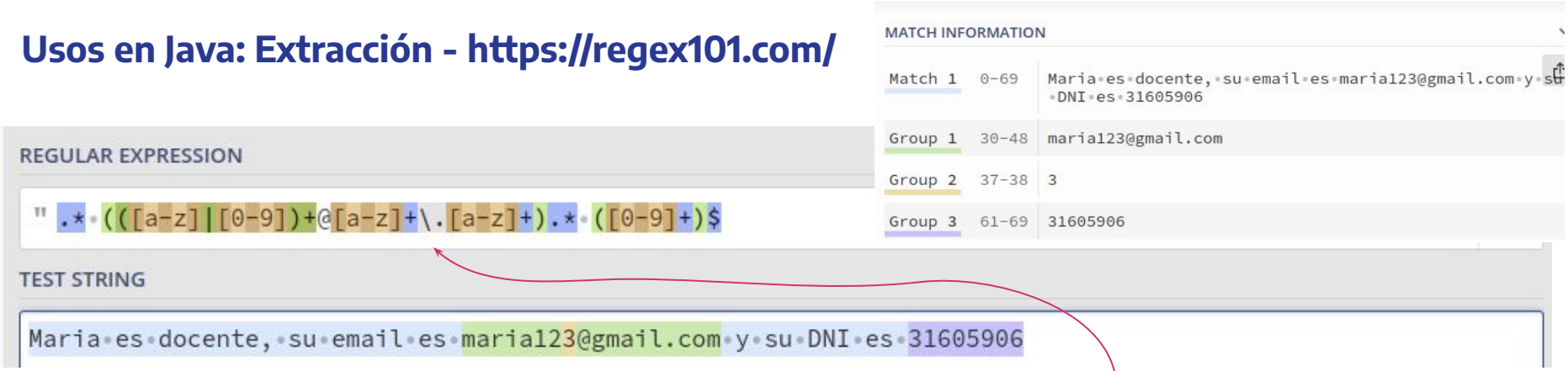
```java
import java.util.regex.Pattern;


String linea = "ramonperez@gmail.com";
linea.matches("([a-z]|[0-9])+@[a-z]+\\.[a-z]+"); // -> True


//-----------------------------------------------


String regex = "([a-z]|[0-9])+@[a-z]+\\.[a-z]+";
final Pattern pattern = Pattern.compile(regex);
pattern.matcher(linea).matches(); // -> True
```

Usos en Java: Extracción - https://regex101.com/




```
final String regex = ".* (([a-z]|[0-9])+@[a-z]+\\.[a-z]+).* ([0-9]+)$";
        final String string = "Maria es docente, su email es maria123@gmail.com y su DNI es
31605906";

        final Pattern pattern = Pattern.compile(regex, Pattern.MULTILINE);
        final Matcher matcher = pattern.matcher(string);

        while (matcher.find()) {
            System.out.println("Full match: " + matcher.group(0));

            for (int i = 1; i <= matcher.groupCount(); i++) {
                System.out.println("Group " + i + ": " + matcher.group(i));
            }
        }
```

Notar que la exp regular en Java y en el sitio cambian justo antes del punto ".", en el primero lleva doble escape \\, mientras que en el sitio solo uno \

Actividad

Argentina programa 4.0

- En ejercicio-archivo, agregar la posibilidad de ponerle tipos a los campos:
  - Cadena/String
  - Email
  - Fecha (dd/mm/aaaa)
  - numero
- Por ejemplo --tipos cadena,numero,email

Referencias

Argentina programa 4.0

- https://regex101.com/
- https://www.baeldung.com/java-string-to-date
- https://towardsdatascience.com/an-introduction-to-regular-expressions-5dd762afc5e4

Argentina
programa
4.0

Gracias!