

Actividad

Actividad 1

1.1

```
public class RegEx {  
    private static void checkMatch(String str, String pat){  
        Pattern pattern = Pattern.compile(regex:pat);  
        Matcher matcher = pattern.matcher(input:str);  
        System.out.println((matcher.matches()? "Si": "No") + " hace match.");  
    }  
    public static void main(String[] args) {  
        checkMatch(str:"abc", pat:"abc");  
        checkMatch(str:"abc", pat:"cba");  
    }  
}
```

gex.RegEx > main >

- RegEx (run) x

run:
Si hace match.
No hace match.
BUILD SUCCESSFUL (total time: 0 seconds)

1.2

```
        checkMatch(str:"abc", pat:".*abc.*");  
        checkMatch(str:"cba", pat:".*abc.*");  
    }  
}
```

gex.RegEx > main >

- RegEx (run) x

run:
Si hace match.
No hace match.
BUILD SUCCESSFUL (total time: 0 seconds)
||

```

public class RegEx {
    private static void checkMatch(String str, String pat){
        Pattern pattern = Pattern.compile(regex:pat);
        Matcher matcher = pattern.matcher(input:str);
        System.out.println((matcher.find()? "Si": "No") + " hace match.");
    }
    public static void main(String[] args) {
        /*
        checkMatch("abc", "abc");
        checkMatch("abc", "cba");
        */

        checkMatch(str:"abc", pat:".*abc.*");
        checkMatch(str:"cba", pat:".*abc.*");
    }
}

```

- RegEx (run) ×

```

run:
Si hace match.
No hace match.
BUILD SUCCESSFUL (total time: 0 seconds)

```

1.3

```

public class RegEx {
    private static void checkMatch(String str, String pat){
        Pattern pattern = Pattern.compile(regex:pat);
        Matcher matcher = pattern.matcher(input:str);
        System.out.println(pat + " " + (matcher.matches()? "si": "no") + " hace match en " + str);
    }
    public static void main(String[] args) {
        /*
        checkMatch("abc", "abc");
        checkMatch("abc", "cba");
        */

        checkMatch(str:"abcxxx", pat:"^abc.*");
        checkMatch(str:"xxxabc", pat:"^abc.*");
    }
}

```

regex.RegEx > checkMatch >

out - RegEx (run) ×

```

run:
^abc.* si hace match en abcxxx
^abc.* no hace match en xxxabc
BUILD SUCCESSFUL (total time: 0 seconds)

```

1.4

```

    }
    }

    checkMatch( str: "Abc", pat: "^[aA]bc.*");
    checkMatch( str: "xabc", pat: "^[aA]bc.*");
}
}

regex.RegEx > main >
ut - RegEx (run) x
run:
^[aA]bc.* si hace match en Abc
^[aA]bc.* no hace match en xabc
BUILD SUCCESSFUL (total time: 0 seconds)

```

1.5

```

    }
    }

    checkMatch( str: "aBCDe", pat: "[a-zA-Z]{5,10}");
    checkMatch( str: "hola", pat: "[a-zA-Z]{5,10}");
}
}

regex.RegEx > main >
ut - RegEx (run) x
run:
[a-zA-Z]{5,10} si hace match en aBCDe
[a-zA-Z]{5,10} no hace match en hola
BUILD SUCCESSFUL (total time: 0 seconds)

```

1.6

```
    checkMatch( str: "123", pat: "^[^\\d].*");
    checkMatch( str: "x123", pat: "^[^\\d].*");
}

}
```

regex.RegEx > main >

out - RegEx (run) x

```
run:
^[^\\d].* no hace match en 123
^[^\\d].* si hace match en x123
BUILD SUCCESSFUL (total time: 0 seconds)
```

1.7

```
    checkMatch( str: "123", pat: ".*[^\\d]$");
    checkMatch( str: "123x", pat: ".*[^\\d]$");
}
```

regex.RegEx > main >

out - RegEx (run) x

```
run:
.*[^\\d]$ no hace match en 123
.*[^\\d]$ si hace match en 123x
BUILD SUCCESSFUL (total time: 0 seconds)
```

1.8S

```
    checkMatch( str: "aaa", pat: "(a|b)+");  
    checkMatch( str: "xxx", pat: "(a|b)+");  
}  
}
```

regex.RegEx > main >

out - RegEx (run) x

run:
(a|b)+ si hace match en aaa
(a|b)+ no hace match en xxx
BUILD SUCCESSFUL (total time: 0 seconds)

1.9

```
    checkMatch( str: "123", pat: ".*1(?!2).*");  
    checkMatch( str: "111", pat: ".*1(?!2).*");  
}  
}
```

regex.RegEx > main >

out - RegEx (run) x

run:
.*1(?!2).* no hace match en 123
.*1(?!2).* si hace match en 111
BUILD SUCCESSFUL (total time: 0 seconds)

Actividad 2

```
Scanner sc = new Scanner( source: System.in);
String email;
System.out.println( x: "Introduce email: ");
email = sc.nextLine();
checkMatch( str: email, pat: "^([\\w-]+(\\. [\\w-]+)*@[A-Za-z0-9]+(\\. [A-Za-z0-9]+)* (\\. [A-Za-z]{2,})$)"
);
}
}
```

regex.RegEx > main >

out - RegEx (run) x

```
run:
Introduce email:
0224969@up.edu.mx
^([\\w-]+(\\. [\\w-]+)*@[A-Za-z0-9]+(\\. [A-Za-z0-9]+)* (\\. [A-Za-z]{2,})$) si hace match en 0224969@up.edu.mx
BUILD SUCCESSFUL (total time: 5 seconds)
```

Actividad 3

```
if ("123".matches( regex: ".*1(?:2).*" )) System.out.println( x: "Si matchea");
else System.out.println( x: "No matchea");
```

gex.RegEx > main >

- RegEx (run) x

```
run:
No matchea
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
String str = "blanco-rojo:amarillo.verde_azul";
String [] cadenas = str.split( regex: "[-:._]" );
for( int i = 0; i < cadenas.length; i++)
    System.out.println( cadenas[i] );
```

regex.RegEx > main >

ut - RegEx (run) x

```
run:
blanco
rojo
amarillo
verde
azul
BUILD SUCCESSFUL (total time: 0 seconds)
```

