JAVA FUNDAMENTOS

EXPRESIONES REGULARES

Ing. Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com
gcoronelc@gmail.com

Índice

- Necesidad
- Expresiones Regulares

Necesidad

Al programar, muchas veces nos vemos en la necesidad de validar si ciertas cadenas de textos cumplen o no con un patrón específico, por ejemplo, saber si un texto esta compuesto en su totalidad por dígitos.

```
public static boolean isInt(String cadena) {
  try {
    Integer.parseInt(cadena);
    return true;
  } catch (Exception e) {
    return false;
  }
}
```

Necesidad

En otros casos necesitamos necesitamos verificar si un cadena existe dentro de otra:

```
public static int seek(String subCadena, String cadena) {
  int pos = cadena.indexOf(subCadena);
  return pos;
}
```

EXPRESIÓN REGULAR

Una expresión regular define un patrón de búsqueda para cadenas de caracteres.

Se puede utilizar para comprobar si una cadena contiene o coincide con el patrón.

El contenido de la cadena de caracteres puede coincidir con el patrón 0, 1 o más veces.

EXPRESIÓN REGULAR

Algunos ejemplos de uso de expresiones regulares pueden ser para:

- Comprobar que la fecha leída cumple el patrón dd/mm/aaaa
- Comprobar que un NIF está formado por 8 dígitos, un guión y una letra
- Comprobar que una dirección de correo electrónico es válida.
- Comprobar que una contraseña cumple unas determinadas condiciones.
- Comprobar que una URL es válida.
- Comprobar cuántas veces se repite dentro de la cadena una secuencia de caracteres determinada.
- Etc.

EXPRESIÓN REGULAR

El patrón se busca en la cadena de izquierda a derecha. Cuando se determina que un carácter cumple con el patrón este carácter ya no vuelve a intervenir en la comprobación.

Ejemplo:

La expresión regular "010" la encontraremos dentro de la cadena "010101010" solo dos veces: "010101010"

EXPRESIÓN REGULAR

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

public class Ejemplo{

  public static void main(String[] args) {
    boolean b = Pattern.matches("[0-9]+","1234");
    System.out.println(b);
  }
}
```