

1. La siguiente tabla propone una **lista de expresiones** cuyo resultado debe **convertirse a String**. La conversión debe de realizarse de forma **eficiente** indicando el código de conversión y el resultado final.

#	Inicial	Código	Final
1	123	String.valueOf(123)	"123"
2	43.21	String.valueOf(43.21)	"43.21"
3	'W'	String.valueOf('W')	"W"
4	true	String.valueOf(true)	"true"
5	2 + 3	String.valueOf(2 + 3)	"5"
6	20 / 3.0	String.valueOf(20 / 3.0)	"6.666666666666667"
7	(char)('\u0041' + 4)	String.valueOf((char)('\u0041' + 4))	"E"
8	!(2 <= 2)	String.valueOf(!(2 <= 2))	"false"

2. La siguiente tabla propone una **lista de expresiones** cuyo resultado debe **convertirse a String**. La conversión debe de realizarse de forma **rápida** indicando el código de conversión y el resultado final.

#	Inicial	Código	Final
1	2016	"" + 2016	"2016"
2	3.1415	"" + 3.1415	"3.1415"
3	'?'	"" + '?'	"?"
4	true	"" + true	"true"
5	6 % 3	"" + 6 % 3	"0"
6	5 - 3.0	"" + (5 - 3.0)	"2.0"
7	'\u20AC'	"" + '\u20AC'	"€"
8	'\u0041' > 'a'	"" + ('\u0041' > 'a')	"false"

3. La siguiente tabla muestra un lista de supuestos en cada uno de los cuales se realiza la **comparación global** de dos variables que contienen sendas cadenas de texto definidas tal cual se muestra en la tabla. Se pretende compararlas utilizando el código:

**str1.equals(str2)**

Escribir el resultado de la comparación.

#	Variables	Resultado
1	String str1 = "JOSE";	false
	String str2 = "PEPE";	
2	String str1 = "\120\105\120\105";	true
	String str2 = "PEPE";	
3	String str1 = "" + 2016;	false
	String str2 = String.valueOf(2017);	
4	String str1 = "P" + "E" + "P" + "E";	true
	String str2 = "\u0050\u0045\u0050\u0045";	
5	String str1 = "jose";	false
	String str2 = "pepe";	
6	String str1 = "\u0050\u0045\u0050\u0045";	true
	String str2 = "pepe".toUpperCase();	
7	String str1 = "josé";	false
	String str2 = "JOSÉ";	
8	String str1 = "pepe".replace('e', 'E').replace('p', 'P');	true
	String str2 = "\120\u0045PE";	
9	String str1 = " P E P E ".trim();	false
	String str2 = " PEPE ".trim();	
10	String str1 = "PE".concat("PE");	true
	String str2 = str1;	
11	String str1 = "PERÓXIDO DE PROLIPENO".substring(0, 4);	false
	String str2 = (2016 + "PEPE").substring(4);	
12	String str1 = "PePePóTaMo".toLowerCase().substring(0, 4);	true
	String str2 = "pEpEpÓTaMo".toLowerCase().substring(0, 4);	
13	String str1 = "PePePóTaMo";	false
	String str2 = "pEpEpÓTaMo";	

4. La siguiente tabla muestra un lista de supuestos en cada uno de los cuales se realiza la **comparación global ignorando es estado de mayúsculas/minúsculas** de dos variables que contienen sendas cadenas de texto definidas tal cual se muestra en la tabla. Se pretende compararlas utilizando el código:

**str1.equalsIgnoreCase(str2)**

Escribir el resultado de la comparación.

#	Variables	Resultado
1	String str1 = "JOSE";	false
	String str2 = "PEPE";	
2	String str1 = "\120\105\120\105";	true
	String str2 = "PEPE";	
3	String str1 = "" + 2016;	false
	String str2 = String.valueOf(2017);	
4	String str1 = "P" + "E" + "P" + "E";	true
	String str2 = "\u0050\u0045\u0050\u0045";	
5	String str1 = "jose";	false
	String str2 = "pepe";	
6	String str1 = "\u0050\u0045\u0050\u0045";	true
	String str2 = "pepe".toUpperCase();	
7	String str1 = "josé";	true
	String str2 = "JOSÉ";	
8	String str1 = "pepe".replace('e', 'E').replace('p', 'P');	true
	String str2 = "\120\u0045PE";	
9	String str1 = " P E P E ".trim();	false
	String str2 = " PEPE ".trim();	
10	String str1 = "PE".concat("PE");	true
	String str2 = str1;	
11	String str1 = "PERÓXIDO DE PROLIPENO".substring(0, 4);	false
	String str2 = (2016 + "PEPE").substring(4);	
12	String str1 = "PePePóTaMo".toLowerCase().substring(0, 4);	true
	String str2 = "pEpEpÓTaMo".toLowerCase().substring(0, 4);	
13	String str1 = "PePePóTaMo";	true
	String str2 = "pEpEpÓTaMo";	

5. La siguiente tabla muestra un lista de supuestos en cada uno de los cuales se realiza la **comparación lexicográfica** de dos variables que contienen sendas cadenas de texto definidas tal cual se muestra en la tabla. Se pretende compararlas utilizando el código:

**str1. compareTo(str2)**

Escribir el resultado de la comparación.

#	Variables	Resultado
1	String str1 = "JOSE";	-6
	String str2 = "PEPE";	
2	String str1 = "\120\105\120\105";	0
	String str2 = "PEPE";	
3	String str1 = "" + 2016;	-1
	String str2 = String.valueOf(2017);	
4	String str1 = "P" + "E" + "P" + "E";	0
	String str2 = "\u0050\u0045\u0050\u0045";	
5	String str1 = "jose";	-6
	String str2 = "pepe";	
6	String str1 = "\u0050\u0045\u0050\u0045";	0
	String str2 = "pepe".toUpperCase();	
7	String str1 = "josé";	32
	String str2 = "JOSÉ";	
8	String str1 = "pepe".replace('e', 'E').replace('p', 'P');	0
	String str2 = "\120\u0045PE";	
9	String str1 = " P E P E ".trim();	-37
	String str2 = " PEPE ".trim();	
10	String str1 = "PE".concat("PE");	0
	String str2 = str1;	
11	String str1 = "PERÓXIDO DE PROLIPENO".substring(0, 4);	2
	String str2 = (2016 + "PEPE").substring(4);	
12	String str1 = "PePePóTaMo".toLowerCase().substring(0, 4);	0
	String str2 = "pEpEpÓTaMo".toLowerCase().substring(0, 4);	
13	String str1 = "PePePóTaMo";	-32
	String str2 = "pEpEpÓTaMo";	

6. La siguiente tabla muestra un lista de supuestos en cada uno de los cuales se realiza la **comparación lexicográfica ignorando es estado de mayúsculas/minúsculas** de dos variables que contienen sendas cadenas de texto definidas tal cual se muestra en la tabla. Se pretende compararlas utilizando el código:

**str1. compareToIgnoreCase(str2)**

Escribir el resultado de la comparación.

#	Variables	Resultado
1	String str1 = "JOSE";	-6
	String str2 = "PEPE";	
2	String str1 = "\120\105\120\105";	0
	String str2 = "PEPE";	
3	String str1 = "" + 2016;	-1
	String str2 = String.valueOf(2017);	
4	String str1 = "P" + "E" + "P" + "E";	0
	String str2 = "\u0050\u0045\u0050\u0045";	
5	String str1 = "jose";	-6
	String str2 = "pepe";	
6	String str1 = "\u0050\u0045\u0050\u0045";	0
	String str2 = "pepe".toUpperCase();	
7	String str1 = "josé";	0
	String str2 = "JOSÉ";	
8	String str1 = "pepe".replace('e', 'E').replace('p', 'P');	0
	String str2 = "\120\u0045PE";	
9	String str1 = " P E P E ".trim();	-69
	String str2 = " PEPE ".trim();	
10	String str1 = "PE".concat("PE");	0
	String str2 = str1;	
11	String str1 = "PERÓXIDO DE PROLIPENO".substring(0, 4);	2
	String str2 = (2016 + "PEPE").substring(4);	
12	String str1 = "PePePóTaMo".toLowerCase().substring(0, 4);	0
	String str2 = "pEpEpÓTaMo".toLowerCase().substring(0, 4);	
13	String str1 = "PePePóTaMo";	0
	String str2 = "pEpEpÓTaMo";	

7. La siguiente tabla muestra un lista de supuestos en cada uno de los cuales se realiza la llamada a alguno de los métodos de la clase String que no se han visto en los ejercicios anteriores. Infórmese del funcionamiento de cada uno de ellos en la documentación de la clase String y escriba en cada caso el código necesario y el resultado de evaluar dicho código.

#	Código	Resultado
1	Comprobar si la cadena de caracteres "pEpEpÓTaMo" termina con la secuencia "TaMo"	
2	Comprobar si la cadena de caracteres "pEpEpÓTaMo" empieza con la secuencia "pepe"	
3	Averiguar cuantos caracteres tiene la cadena de caracteres "pEpEpÓTaMo"	
4	Comprobar si la cadena de caracteres "pEpEpÓTaMo" contiene la secuencia "Ep"	
5	Averiguar el carácter que ocupa la posición 5 en la cadena de caracteres "pEpEpÓTaMo"	
6	Averiguar la posición en la que se encuentra el primer carácter 'E' en la cadena de caracteres "pEpEpÓTaMo" empezando a buscar por la izquierda	
7	Averiguar la posición en la que se encuentra el primer carácter 'E' en la cadena de caracteres "pEpEpÓTaMo" empezando a buscar por la derecha	
8	Transformar la cadena de caracteres "pEpEpÓTaMo" buscando y sustituyendo cada aparición de la secuencia "pEpE" por la secuencia "jOsE"	
9	Comprobar si el fragmento indicado de la cadena de caracteres "pEpEpÓTaMo" coincide con el fragmento indicado de la cadena de caracteres "álamos"	
10	Comprobar si el fragmento indicado de la cadena de caracteres "pEpEpÓTaMo" coincide con el fragmento indicado de la cadena de caracteres "álamos" ignorando la disparidad mayúsculas/minúsculas	
11	A partir de la cadena de caracteres "12-15-21-28-35-41" compuesta por una secuencia de ítems separados entre sí por la secuencia "-", extraer dichos ítems y generar con ellos un array	
12	Capitalizar la palabra "pEpEpÓTaMo"	