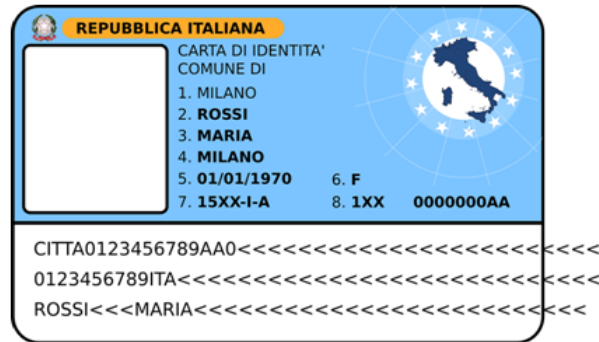


6.C. Expresiones regulares.

1. Expresiones regulares.

1.1. Expresiones regulares (I).

¿Tienen algo en común todos los números de DNI y de NIE? ¿Podrías hacer un programa que verificara si un DNI o un NIE es correcto? Seguro que sí. Si te fijas, los números de DNI y los de NIE tienen una estructura fija: X1234567Z (en el caso del NIE) y 1234567Z (en el caso del DNI). Ambos siguen un patrón que podría describirse como: una letra inicial opcional (solo presente en los NIE), seguida de una secuencia numérica y finalizando con otra letra. ¿Fácil no?



Pues esta es la función de las expresiones regulares: permitir comprobar si una cadena sigue o no un patrón preestablecido. Las expresiones regulares son un mecanismo para describir esos patrones, y se construyen de una forma relativamente sencilla. Existen muchas librerías diferentes para trabajar con expresiones regulares, y casi todas siguen, más o menos, una sintaxis similar, con ligeras variaciones. Dicha sintaxis nos permite indicar el patrón de forma cómoda, como si de una cadena de texto se tratase, en la que determinados símbolos tienen un significado especial. Por ejemplo "[01]+" es una expresión regular que permite comprobar si una cadena conforma un número binario. Veamos cuáles son las reglas generales para construir una expresión regular:

- Podemos indicar que una cadena **contiene un conjunto de símbolos fijo**, simplemente poniendo dichos símbolos en el patrón, excepto para algunos símbolos especiales que necesitarán un carácter de escape como veremos más adelante. Por ejemplo, el patrón **"aaa"** admitirá cadenas que contengan tres aes.
- **"[xyz]"**. Entre corchetes podemos indicar **opcionalidad**. Solo uno de los símbolos que hay entre los corchetes podrá aparecer en el lugar donde están los corchetes. Por ejemplo, la expresión regular **"aa[xy]"** admitirá como válidas las cadenas **"aaax"** y la cadena **"aaay"**. **Los corchetes representan una posición de la cadena que puede tomar uno de varios valores posibles.**
- **"[a-z]" " [A-Z]" " [a-zA-Z]"**. Usando el guión y los corchetes podemos indicar que el patrón admite cualquier carácter entre la letra inicial y la final. Es importante que sepas que se diferencia entre letras mayúsculas y minúsculas, no son iguales de cara a las expresiones regulares.
- **"[0-9]"**. Y nuevamente, usando un guión, podemos indicar que se permite la presencia de un dígito numérico entre 0 y 9, **cualquiera de ellos, pero solo uno.**

Con las reglas anteriores podemos indicar el conjunto de símbolos que admite el patrón y el orden que deben tener. Si una cadena no contiene los símbolos especificados en el patrón, en el mismo orden, entonces la cadena no encajará con el patrón. **Veamos ahora como indicar repeticiones:**

- "a?". Usaremos el interrogante para indicar que un símbolo puede **aparecer una vez o ninguna**. De esta forma la letra "a" podrá aparecer una vez o simplemente no aparecer.
- "a*?". Usaremos el asterisco para indicar que un símbolo puede **aparecer una vez o muchas veces**, pero **también ninguna**. Cadenas válidas para esta expresión regular serían "aa", "aaa" o "aaaaaaaaa".

- **"a⁺"**. Usaremos el símbolo de suma para indicar que **otro símbolo debe aparecer al menos una vez**, pudiendo repetirse cuantas veces quiera.
- **"a{1,4}"**. Usando las llaves, **podemos indicar el número mínimo y máximo de veces que un símbolo podrá repetirse**. El primer número del ejemplo es el número 1, y quiere decir que la letra "a" debe aparecer al menos una vez. El segundo número, 4, indica que como máximo puede repetirse cuatro veces.
- **"a{2,}"**. También es posible indicar solo el número mínimo de veces que un carácter debe aparecer (sin determinar el máximo), haciendo uso de las llaves, indicando el primer número y poniendo la coma (no la olvides).
- **"a{5}"**. A diferencia de la forma anterior, si solo escribimos un número entre llaves, sin poner la coma detrás, **significará que el símbolo debe aparecer un número exacto de veces**. En este caso, la "a" debe aparecer exactamente 5 veces.
- **"[a-z]{1,4}[0-9]+"**. Los indicadores de repetición se pueden usar también con corchetes, dado que los corchetes representan, básicamente, un símbolo. En el ejemplo anterior se permitiría de una a cuatro letras minúsculas, seguidas de al menos un dígito numérico.

Ejercicio resuelto

Con lo visto hasta ahora ya es posible construir una expresión regular capaz de verificar si una cadena contiene un DNI o un NIE, ¿serías capaz de averiguar cuál es dicha expresión regular?

Solución

La expresión regular que permite verificar si una cadena contiene un DNI o un NIE es la siguiente: **"[XYxy]?[0-9]{1,9}[A-Za-z]"**; aunque no es la única solución.

Nota: En realidad esta solución no es del todo precisa, ya que hay ciertas letras que no se crean con el algoritmo de generación de letra del DNI (concretamente las letras I, O, U y Ñ).