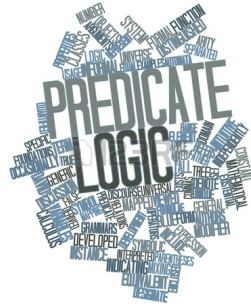


## Predica con el ejemplo

20 de septiembre de 2018



Marty (recuerda, Levi Strauss para tus padres), has de convencer a tus padres de que los predicados son la forma perfecta de expresar las propiedades que cumplen los programas que escriben. Tendrás que dar un buen ejemplo proporcionándoles una interesante colección de predicados que puedan utilizar posteriormente en su beneficio.

Comenzamos utilizando Dafny para definir predicados que posteriormente utilizaremos para especificar algoritmos. Dafny está instalado en Visual Studio Code en los laboratorios y os la podéis instalar en vuestro computador. Descarga el fichero **Predicando.dfy**. *Crea un fichero nuevo en VSCode con el nombre de tu avatar (sin espacios) y extensión dfy y ve copiando en él cada ejercicio a medida que los vas haciendo. Has de entregar ese fichero en la tarea correspondiente del CV.*

1. Primero aparece definido un predicado **esPar** que indica si un número es par. Fíjate en la sintaxis. Si cometes algún error sintáctico aparece subrayado en rojo y una explicación en la ventana de "Problemas". Has de guardar los cambios para que Dafny vuelva a comprobar la sintaxis. Debajo de la definición del predicado el método **TestPar** comprueba la validez del predicado para algunos ejemplos. *Descomenta de una en una cada una de esas líneas para observar qué sucede en cada caso y vuelve a comentarlo después anotando al lado lo que ha sucedido.*

2. A continuación se define un predicado `positivos` que indica que todos los elementos de una secuencia de enteros son positivos. De nuevo el método `TestPositivos` comprueba la validez del predicado en ciertos casos. *Descomenta de una en una cada una de esas líneas para observar qué sucede en cada caso y vuelve a comentarlo después anotando al lado lo que ha sucedido.*

Las secuencias son objetos inmutables que en muchas ocasiones serán útiles para representar la secuencia de elementos contenidos en un array. Por ejemplo, el predicado `positivosA` indica que todos los elementos de un array de enteros son positivos utilizando el predicado `positivos` sobre la secuencia de sus elementos, la cual viene representada por `a[..]` o también `a[0..a.Length]`.

3. El tercer ejemplo define un predicado **iguales** que indica si todos los elementos de una secuencia son iguales, y otro **igualesA** que indica lo mismo para un array. El método **TestIguales** comprueba la validez del predicado en ciertos casos. De nuevo el método **TestIguales** comprueba la validez del predicado en ciertos casos. *Descomenta de una en una cada una de esas líneas para observar qué sucede en cada caso y vuelve a comentarlo después anotando al lado lo que ha sucedido.*

4. El predicado `distintosDe`, dada una secuencia de enteros y un entero, indica si todos los elementos de la secuencia son distintos de dicho entero. De nuevo el método `TestDistintosDe` comprueba la validez del predicado en ciertos casos.
5. Escribe un predicado `distintos` que indique que los elementos de una secuencia son todos distintos entre sí, y otro `distintosA` para un array. El método `testDistintos` comprobará su validez para algunos casos, comprueba que obtienes los resultados esperados.
6. Escribe un predicado `contiguosDistintos` que indique que los elementos contiguos de una secuencia son distintos. El método `testContiguosDistintos` comprobará su validez para algunos casos, comprueba que obtienes los resultados esperados.
7. Escribe un predicado `ordenado` que indique que una secuencia está ordenada de manera no decreciente. El método `testOrdenado` comprobará su validez para algunos casos, comprueba que obtienes los resultados esperados.
8. Escribe un predicado `ordenadoEstricto` que indique que una secuencia está ordenada de manera estrictamente creciente. El método `testOrdenadoEstricto` comprobará su validez para algunos casos, comprueba que obtienes los resultados esperados.
9. El siguiente ejemplo define de dos formas distintas un predicado `igual` que indica si hay un elemento en la secuencia igual a uno dado. El método `testIgual` comprobará su validez para algunos casos. Los predicados existenciales son difíciles de demostrar. Observa que usando el cuantificador existencial Dafny no es capaz de demostrar el primer caso sin ayuda, mientras que con el operador `in` de las secuencias sí puede. Es una de las facilidades del tipo secuencia a tener en cuenta.
10. Escribe un predicado `primerCero` que indique si una posición de la secuencia es la primera posición (desde la izquierda) en que aparece un cero, o es igual a la longitud en caso de no haber ceros en la secuencia. El método `testPrimerCero` comprobará su validez para algunos casos, comprueba que obtienes los resultados esperados. *Puedes añadir tus propios casos de comprobación.*
11. El predicado `esPermutacion` determina si una secuencia es permutacion de otra utilizando la función `multiset` que permite obtener el multiconjunto de elementos de una secuencia dada. Es una forma cómoda de expresar este predicado. *Descomenta de una en una cada una de esas líneas para observar qué sucede en cada caso y vuelve a comentarlo después anotando al lado lo que ha sucedido.*