En una función, y siempre con carácter opcional e informativo, existe la posibilidad de declararla como **PURE** o **IMPURE**. Un función *pura* es una función que devuelve siempre el mismo valor para unos parámetros de entrada dados. Una función es *impura* si para unos mismos parámetros puede devolver valores diferentes.

Normalmente las funciones impuras lo son porque dependen de alguna variable o señal global. Poner **PURE** o **IMPURE** en la declaración de la función no hace a esa función pura o impura, en realidad no hace nada, es sólo un comentario.

Un ejemplo típico de función impura es la función predefinida `now`. Esta función devuelve el tiempo que ha pasado desde el inicio de la simulación. Su definición es la siguiente:

```
IMPURE FUNCTION now RETURN delay_length;
```

El tipo `delay_length` es un subtipo del predefinido `time`. Esta función es evidentemente impura, ya que, dependiendo del tiempo que haya pasado durante la simulación, devuelve una cosa u otra.

Especificación de función y procedimiento

En muchas ocasiones resulta útil declarar la función antes de declarar su cuerpo por motivos de visibilidad, etc. En estos casos la declaración se hace igual que se ha visto pero se sustituye el **IS** por un punto y coma dando por finalizada la especificación:

```
-- Procedimiento:
PROCEDURE nombre [ (parámetros) ] ;

-- Función:
[PURE | IMPURE] FUNCTION nombre [ (parámetros) ] RETURN tipo ;
```

A continuación se presenta un ejemplo donde se define un procedimiento que calcula los valores máximo y mínimo de los números contenidos en un vector cuyo rango viene especificado en el propio tipo que se llama `vec`.

```
-- más corto: PROCEDURE lims(cjt: vec; min,max: INOUT integer) IS
PROCEDURE lims(CONSTANT cjt:IN vec; VARIABLE min,max:INOUT integer) IS
VARIABLE ind: integer;
BEGIN
```