

## 1. Objetivo

Implementar un multiplicador concurrente mediante el mecanismo de mensajes usando la librería nSystem.

## 2. Ejercicio de Laboratorio

A continuación se presenta una solución para un multiplicador concurrente seguro que a medida que recibe un número, lo multiplica por el valor que tiene almacenado y posteriormente actualiza ese valor con el resultado. Se le pide que implemente la función *nMain()* en la cual solicite que el usuario ingrese por teclado los valores, y muestre el resultado de cada multiplicación realizada. Cuando ingrese -1, se detiene la ejecución.

Tenga en cuenta que debe crear el multiplicador usando *makeMult()* y destruirlo con *destroyMult()*. La salida debe ser similar a la mostrada en la Figura 1.

```
#include <nSystem.h>
```

```
#define MULT 1
#define DEST 2
```

```
typedef struct {
    nTask serv;
} *Mult;
```

```
typedef struct {
    int code;
    double f, p;
} Req;
```

```
void multProc() {
    double p= 1.0;
    for (;;) {
        Req *prq;
        nTask t;
        prq = (Req *) nReceive(&t, -1);
        if (prq->code==MULT) {
            p = p * prq->f;
            prq->p = p;
            nReply(t, 0);
        }
        else { /*DEST*/
            nReply(t, 0);
            nExitTask(0);
        }
    }
}
```

```
Mult makeMult() {
    Mult m = (Mult) nMalloc(sizeof *m);
    m->serv = nEmitTask(multProc);
    return m;
}
```

```
double mult(Mult m, double f) {
    Req rq;
    rq.code = MULT;
    rq.f = f;
    nSend(m->serv, &rq);
    return (rq.p);
}
```

```
void destroyMult(Mult m) {
    Req rq;
    rq.code = DEST;
    nSend(m->serv, &rq);
    nWaitTask(m->serv);
    nFree(m);
}
```

```
[so2018@so2018-pc archivos-so]$ ./mensaje
Ingrese un número:
2
1.000000 * 2.000000 = 2.000000
Ingrese un número:
3
2.000000 * 3.000000 = 6.000000
Ingrese un número:
5
6.000000 * 5.000000 = 30.000000
Ingrese un número:
0.5
30.000000 * 0.500000 = 15.000000
Ingrese un número:
-1
Estadísticas finales:

Nro. de cambios de contextos implícitos: 0
[so2018@so2018-pc archivos-so]$
```

Figura 1 Ejemplo salida por pantalla del programa.