

# Introducción a la Programación

## Primer Entregable de laboratorio

### Ejercicio 5

Implementar la función `combinacionesMenoresOIguales :: Integer -> Integer`, cuya especificación es la siguiente:

```
problema combinacionesMenoresOIguales (n:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  {  
  requiere:  $\{n \geq 1\}$   
  asegura:  $\{res = \sum_{i=1}^n \sum_{j=1}^n \text{if } i * j \leq n \text{ then } 1 \text{ else } 0 \text{ fi}\}$   
}
```

# Introducción a la Programación

## Primer Entregable de laboratorio

### Ejercicio 4

Usando recursión, implementar la función `sumaPrimerosNImparesEspecial :: Integer -> Integer`, cuya especificación es la siguiente:

```
problema sumaPrimerosNImparesEspecial (n:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  {  
    requiere:  $\{n \geq 1\}$   
    asegura:  $\{res = \sum_{i=1}^{2n-1} \text{if } (i \bmod 2 = 0) \text{ then } 0 \text{ else } 2 * i + 2 \text{ fi}\}$   
}
```

# Introducción a la Programación

## Primer Entregable de laboratorio

### Ejercicio 3

Implementar la función `prod :: Integer ->Integer`, cuya especificación es la siguiente:

```
problema prod (n:ℤ) : ℤ {  
    requiere: { $n \geq 1$ }  
    asegura: { $res = \prod_{i=1}^{2n} (i^2 + 2i)$ }  
}
```

# Introducción a la Programación

## Primer Entregable de laboratorio

### Ejercicio 2

Implementar la función `sumaDigitos :: Integer -> Integer`, cuya especificación es la siguiente:

```
problema sumaDigitos (n:ℤ) : ℤ {  
  requiere: { $n \geq 0$ }  
  asegura: {res es la suma de todos los dígitos de  $n$ }  
}
```

# Introducción a la Programación

## Primer Entregable de laboratorio

### Ejercicio 1

Implementar la función `sumaMenosQueMax :: (Int, Int, Int) -> Bool`, cuya especificación es la siguiente:

```
problema sumaMenosQueMax (t:  $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ ) : Bool {  
    requiere: {true}  
    asegura: {res = true  $\Leftrightarrow \max(t_0, t_1, t_2) > \min(t_0, t_1, t_2) + \text{medio}(t_0, t_1, t_2)$ }  
}  
  
problema max (a, b, c:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  {  
    requiere: {true}  
    asegura: {res es el máximo valor entre a, b y c}  
}  
  
problema min (a, b, c:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  {  
    requiere: {true}  
    asegura: {res es el mínimo valor entre a, b y c}  
}  
  
problema medio (a, b, c:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  {  
    requiere: {true}  
    asegura: {( $\exists l : \text{seq}(\mathbb{Z})$ )( $|l| = 3 \wedge \text{esPermutacion}(l, <a, b, c>) \wedge l[0] \leq l[1] \leq l[2] \wedge \text{res} = l[1]$ )}  
}  
  
pred esPermutacion (l1, l2:  $\text{seq}(\mathbb{Z})$ ) {  
    ( $\forall x : \mathbb{Z}$ )  $\text{cantidadDeApariciones}(l1, x) = \text{cantidadDeApariciones}(l2, x)$   
}
```