

FUNCIONES CON PARÁMETROS OBLIGATORIOS Y OPCIONALES

- Función con un número **fijo** de parámetros **obligatorios**

```
(define función  $x_1 \dots x_n$ )  
  <Cuerpo de la función>  
)
```

Función equivalente

```
(define función  
  (lambda ( $x_1 \dots x_n$ )  
    <Cuerpo de la función>  
  )  
)
```

- Función con un número **indeterminado** de parámetros **opcionales** que se almacenan en una lista:
variable

```
(define función  
  (lambda variable  
    <Cuerpo de la función>  
  )  
)
```

- Función con un número **fijo** de parámetros **obligatorios** y un número **indeterminado** de parámetros opcionales
 - Los parámetros obligatorios se almacenan en variables individuales
 - Los parámetros opcionales se almacenan todos en un misma lista **z**

(define función

(lambda (x₁ ... x_n . z)

<Cuerpo de la función>

)

)

- El **punto** que situado entre **x_n** y **z** es **imprescindible**
- El punto ha de estar separado.

:: Función con un número **fijo** de parámetros

```
(define (datos-personales-1 nombre apellido1 apellido2)
  (display nombre)
  (display " ")
  (display apellido1)
  (display " ")
  (display apellido2)
  (newline)
)
```

:: Función equivalente

```
(define datos-personales-1-lambda
  (lambda (nombre apellido1 apellido2)
    (display nombre)
    (display " ")
    (display apellido1)
    (display " ")
    (display apellido2)
    (newline)
  )
)
```

:: Ejemplo

```
(datos-personales-1 "Ana" "Ramos" "Rueda")
```

:: Función con un número **variable** de parámetros

```
(define datos-personales-2
  (lambda lista-datos
    (ver-lista-cadenas lista-datos)
  )
)
```

:: Función que muestra una lista de cadenas

```
(define (ver-lista-cadenas lista)
  (cond
    ((null? lista) (newline))
    (else
      (display (car lista))
      (display " ")
      (ver-lista-cadenas (cdr lista))
      (newline)
    )
  )
)
```

:: Ejemplo

```
(datos-personales-2 "Ana" "Ramos" "Rueda" "Avda. Rosaleda 19" "Ponferrada")
```

:: Función con un número fijo de parámetros
:: y un **variable** de parámetros

```
(define datos-personales-3  
  (lambda (nombre apellido1 apellido2 . resto)  
    (datos-personales-1 nombre apellido1 apellido2)  
    (ver-lista-cadenas resto)  
  )  
)
```

:: Ejemplo

```
(datos-personales-3 "Ana" "Ramos" "Rueda" "Avda. Rosaleda 19" "Ponferrada" "León")
```