



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Reentrega Trabajo Práctico 3

Programación orientada a Objetos.

Paradigmas y lenguajes de programación

Grupo: Zamba cálculo

Integrante	LU	Correo electrónico
Ignacio Niesz	722/10	ignacio.niesz@gmail.com
Leandro Vega	698/11	leandrogvega@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

## Índice

<b>1. Modelo de mediciones</b>	<b>3</b>
1.1. Unidad . . . . .	3
1.1.1. Producto de Unidades . . . . .	3
1.1.2. Unidad Básica . . . . .	4
1.1.3. Unidad Escalar . . . . .	4
1.2. Medida . . . . .	5
1.3. Number . . . . .	6

# 1. Modelo de mediciones

## 1.1. Unidad

```

1
2 Object subclass: #Unidad
3   instanceVariableNames: ''
4   classVariableNames: ''
5   poolDictionaries: ''
6   category: 'PLP-TP-2015C1'
7
8 "-----"
9
10 Unidad class
11   instanceVariableNames: ''
12
13   escalar
14     ^UnidadEscalar new.
15
16   metro
17     ^UnidadBasica deNombre:'metro'.
18
19   kilogramo
20     ^UnidadBasica deNombre:'kilogramo'.
21
22   segundo
23     ^UnidadBasica deNombre:'segundo'.

```

### 1.1.1. Producto de Unidades

```

1 Unidad subclass: #ProductoDeUnidades
2   instanceVariableNames: 'factores'
3   classVariableNames: ''
4   poolDictionaries: ''
5   category: 'PLP-TP-2015C1'
6
7   factores
8     ^ factores
9
10   init : aCollection
11     factores := aCollection .
12
13   * anUnidad
14     ^anUnidad productoProducto: self.
15
16   productoProducto: aProductoDeUnidades
17     ^ ProductoDeUnidades de: ((aProductoDeUnidades factores) addAll: (self factores) ; yourself ).
18
19   productoBasica: anUnidadBasica
20     ^ ProductoDeUnidades de: ((OrderedCollection with: anUnidadBasica) addAll: (self factores) ; yourself ).
21
22   productoEscalar: anUnidadEscalar
23     ^ self .
24
25   = otraUnidad
26     ^ ( self class = otraUnidad class) and: [ self factores asBag = otraUnidad factores asBag ].
27
28   printOn: aStream
29     (1 to: (factores size - 1)) do: [ :each | (factores at:each) printOn: aStream . aStream nextPutAll: ' ' ].
30     ^(factores at:(factores size)) printOn: aStream.
31
32 "-----"

```

```

33
34 ProductoDeUnidades class
35     instanceVariableNames: ''
36
37 de: factores
38     factores ifEmpty: [ ^ Unidad escalar ].
39     (factores size = 1) ifTrue: [ ^ factores first ].
40     ^ self new init: factores

```

### 1.1.2. Unidad Básica

```

1  Unidad subclass: #UnidadBasica
2      instanceVariableNames: 'nombre'
3      classVariableNames: ''
4      poolDictionaries: ''
5      category: 'PLP-TP-2015C1'
6
7  initNombre: aString
8      nombre := aString.
9
10 hash
11     ^ nombre hash
12
13 nombre
14     ^ nombre
15
16 * anUnidad
17     ^anUnidad productoBasica: self .
18
19 productoEscalar: anUnidadEscalar
20     ^ self .
21
22 productoBasica: anUnidadBasica
23     ^ ProductoDeUnidades de: (OrderedCollection with: anUnidadBasica with: self ).
24
25 productoProducto: aProductoDeUnidades
26     ^ ProductoDeUnidades de: ( (aProductoDeUnidades factores) add: self ; yourself ).
27
28 = otraUnidad
29     ^ ( self class = otraUnidad class) and: [ self nombre = otraUnidad nombre].
30
31 printOn: aStream
32     ^ aStream nextPutAll: nombre.
33
34 "-----"!
35
36 UnidadBasica class
37     instanceVariableNames: ''
38
39 deNombre: aString
40     ^ self new initNombre: aString.

```

### 1.1.3. Unidad Escalar

```

1  Unidad subclass: #UnidadEscalar
2      instanceVariableNames: ''
3      classVariableNames: ''
4      poolDictionaries: ''
5      category: 'PLP-TP-2015C1'
6
7  * anUnidad
8      ^anUnidad productoEscalar: self .

```

```

9
10 productoEscalar: anUnidadEscalar
11   ^ anUnidadEscalar.
12
13 productoBasica: anUnidadBasica
14   ^ anUnidadBasica .
15
16 productoProducto: aProductoDeUnidades
17   ^ aProductoDeUnidades.
18
19 = otraUnidad
20   ^ ( self class = otraUnidad class ).
21
22 printOn: aStream
23   ^ aStream nextPutAll: 'escalar' .

```

## 1.2. Medida

```

1 Object subclass: #Medida
2   instanceVariableNames: 'cantidad unidad'
3   classVariableNames: ''
4   poolDictionaries: ''
5   category: 'PLP-TP-2015C1'
6
7   cantidad: anInteger unidad: anUnidad
8     cantidad := anInteger.
9     unidad := anUnidad.
10
11   cantidad
12     ^ cantidad.
13
14   unidad
15     ^ unidad.
16
17   + otraMedida
18     ( self unidad = otraMedida unidad ) ifTrue: [ ^ Medida new cantidad: ( self cantidad + otraMedida cantidad ) unidad: ( self unidad + otraMedida unidad ) ].
19     ^ super + otraMedida.
20
21   - otro
22     ^ self + ((-1) * otro).
23
24   * otraMedida
25     ^ self cantidad: cantidad * (otraMedida cantidad) unidad: unidad * (otraMedida unidad).
26
27   metro
28     ^ Medida new cantidad: ( self cantidad ) unidad: (( self unidad ) * (Unidad metro)).
29
30   kilogramo
31     ^ Medida new cantidad: ( self cantidad ) unidad: (( self unidad ) * (Unidad kilogramo)).
32
33   segundo
34     ^ Medida new cantidad: ( self cantidad ) unidad: (( self unidad ) * (Unidad segundo)).
35
36   = medida2
37     ^ ( self class = medida2 class ) and: [(cantidad = medida2 cantidad) and: (unidad = medida2 unidad)].
38
39   printOn: aStream
40     aStream nextPutAll: cantidad asString.
41     aStream nextPutAll: ' '.
42     ^unidad printOn: aStream.

```

### 1.3. Number

```
1 cantidad
2   ^self .
3
4 unidad
5   ^Unidad escalar .
6
7 kilogramo
8   ^Medida new cantidad: self unidad: Unidad kilogramo.
9
10 metro
11   ^Medida new cantidad: self unidad: Unidad metro.
12
13 segundo
14   ^Medida new cantidad: self unidad: Unidad segundo.
```