



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

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

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

```

```

33 de: factores
34   factores isEmpty: [ ^ Unidad escalar ].
35   ( factores size = 1) ifTrue: [ ^ factores first ].
36   ^ 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 class = UnidadEscalar) ifTrue: [ ^ self productoEscalar: anUnidad ].
18   (anUnidad class = UnidadBasica) ifTrue: [ ^ self productoBasica: anUnidad ].
19   ^ self productoProducto: anUnidad.
20
21
22 productoEscalar: anUnidadEscalar
23   ^ self .
24
25 productoBasica: anUnidadBasica
26   ^ ProductoDeUnidades de: (OrderedCollection with: self with: anUnidadBasica).
27
28 productoProducto: aProductoDeUnidades
29   ^ ProductoDeUnidades de: ( aProductoDeUnidades factores) add: self ; yourself ).
30
31 = otraUnidad
32   ^ ( self class = otraUnidad class) and: [ self nombre = otraUnidad nombre].
33
34 printOn: aStream
35   ^ aStream nextPutAll: nombre.
36
37 "-----"
38
39 UnidadBasica class
40   instanceVariableNames: ''
41
42 deNombre: aString
43   ^ 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 class = UnidadEscalar) ifTrue: [ ^ self productoEscalar: anUnidad ].
9    (anUnidad class = UnidadBasica) ifTrue: [ ^ self productoBasica: anUnidad ].

```

```

10 ^ self productoProducto: anUnidad.
11
12 productoEscalar: anUnidadEscalar
13 ^ anUnidadEscalar.
14
15 productoBasica: anUnidadBasica
16 ^ anUnidadBasica .
17
18 productoProducto: aProductoDeUnidades
19 ^ aProductoDeUnidades.
20
21 = otraUnidad
22 ^ ( self class = otraUnidad class ).
23
24 printOn: aStream
25 ^ 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
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
2 cantidad
3   ^ self .
4
5 unidad
6   ^Unidad escalar .
7
8 kilogramo
9   ^Medida new cantidad: self unidad: Unidad kilogramo.
10
11 metro
12   ^Medida new cantidad: self unidad: Unidad metro.
13
14 segundo
15   ^Medida new cantidad: self unidad: Unidad segundo.
```