



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

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 Object subclass: #Unidad
2   instanceVariableNames: ''
3   classVariableNames: ''
4   poolDictionaries: ''
5   category: 'PLP-TP-2015C1'
6
7 "-----"
8
9 Unidad class
10   instanceVariableNames: ''
11
12 escalar
13   ^UnidadEscalar new.
14
15 metro
16   ^UnidadBasica deNombre:'metro'.
17
18 kilogramo
19   ^UnidadBasica deNombre:'kilogramo'.
20
21 segundo
22   ^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 * anUnidad
8   ^anUnidad productoProducto: self.
9
10 productoEscalar: anUnidadEscalar
11   ^self .
12
13 productoBasica: anUnidadBasica
14   ^ ProductoDeUnidades de: ((OrderedCollection with: anUnidadBasica) addAll: (self factores); yourself ).
15
16 productoProducto: aProductoDeUnidades
17   ^ ProductoDeUnidades de: ((aProductoDeUnidades factores) addAll: (self factores); yourself ).
18
19 init : aCollection
20   factores := aCollection .
21
22 factores
23   ^ factores .
24
25 printOn: aStream
26   (1 to: (factores size - 1)) do: [ :each | (factores at:each) printOn: aStream . aStream nextPutAll: ' ' ].
27   ^(factores at:(factores size)) printOn: aStream.
28
29 = otraUnidad
30   ^ (self class = otraUnidad class) and: [ self factores asBag = otraUnidad factores asBag ].
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
19     cantidad + otraMedida cantidad) unidad: ( self unidad)].
20   ^ super + otraMedida.
21
22 - otro
23   ^ self + ((-1) * otro).
24
25 * otraMedida
26   ^ self cantidad: cantidad * (otraMedida cantidad) unidad: unidad * (otraMedida unidad).
27
28 metro
29   ^ Medida new cantidad: ( self cantidad) unidad: (( self unidad) * (Unidad metro)).
30
31 kilogramo
32   ^ Medida new cantidad: ( self cantidad) unidad: (( self unidad) * (Unidad kilogramo)).
33
34 segundo
35   ^ Medida new cantidad: ( self cantidad) unidad: (( self unidad) * (Unidad segundo)).
36
37 = medida2
38   ^ ( self class = medida2 class) and: [(cantidad = medida2 cantidad) and: (unidad = medida2 unidad)].
39
40 printOn: aStream
41   aStream nextPutAll: cantidad asString.
42   aStream nextPutAll: ' '.
43   ^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.
```