

Laboratorio de Computación III

Clase XIII

Andrés Fortier

Repaso: Getters y setters

Surtidor
capacidadMaxima capacidadActual
<u>capacidadMaxima: litros</u> - initialize - capacidadMaxima: litros estaVacio litrosACargar

Surtidor
capacidadMaxima capacidadActual
<u>capacidadMaxima: litros</u> - initialize - capacidadMaxima - capacidadMaxima: litros - capacidadActual - capacidadActual: litros estaVacio litrosACargar

Surtidor>>initialize

capacidadActual := 0.

Surtidor>>initialize

self capacidadActual: 0.

Surtidor>>litrosACargar

^capacidadMaxima – capacidadActual.

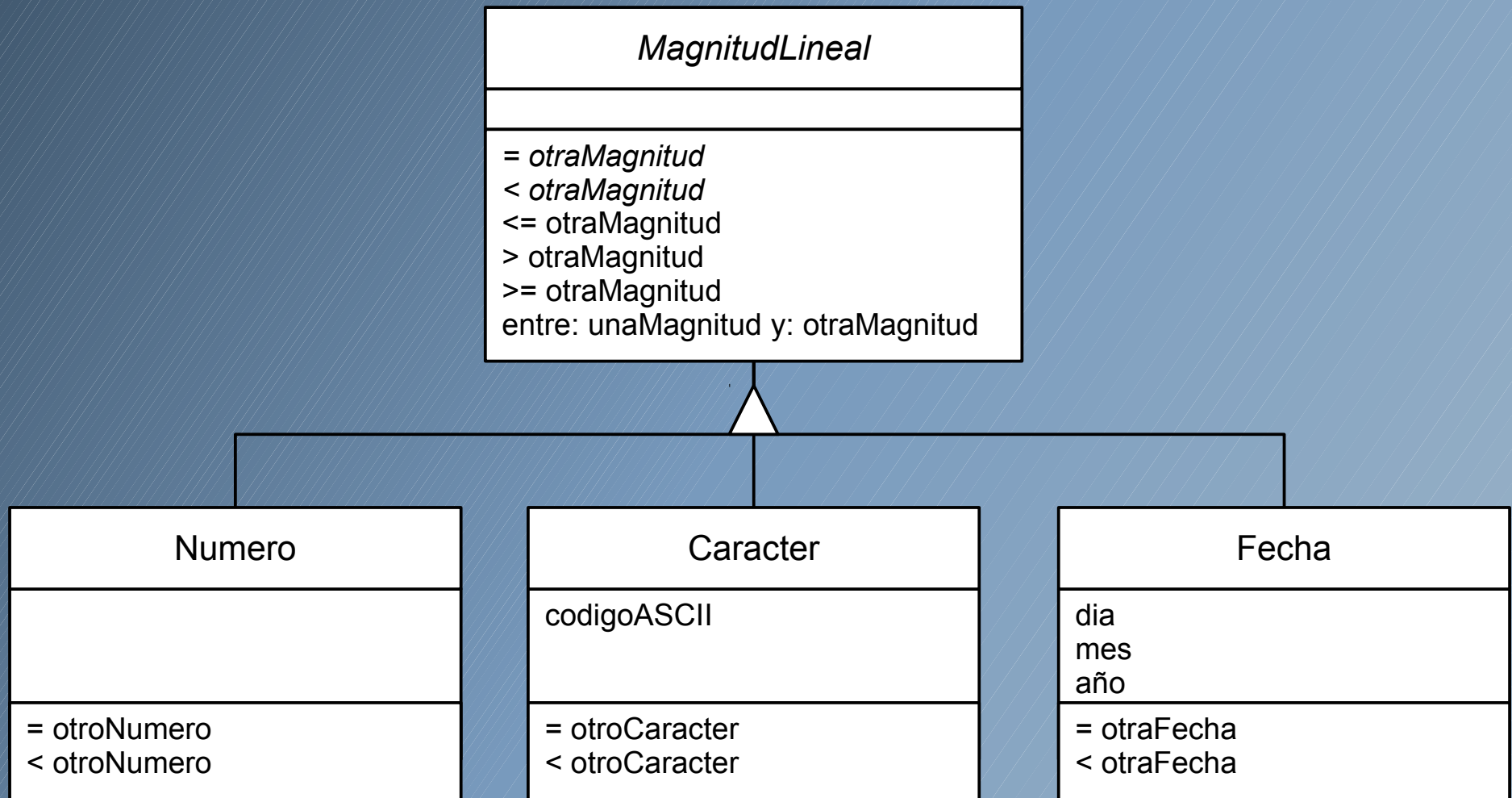
Surtidor>>litrosACargar

^self capacidadMaxima – self capacidadActual.

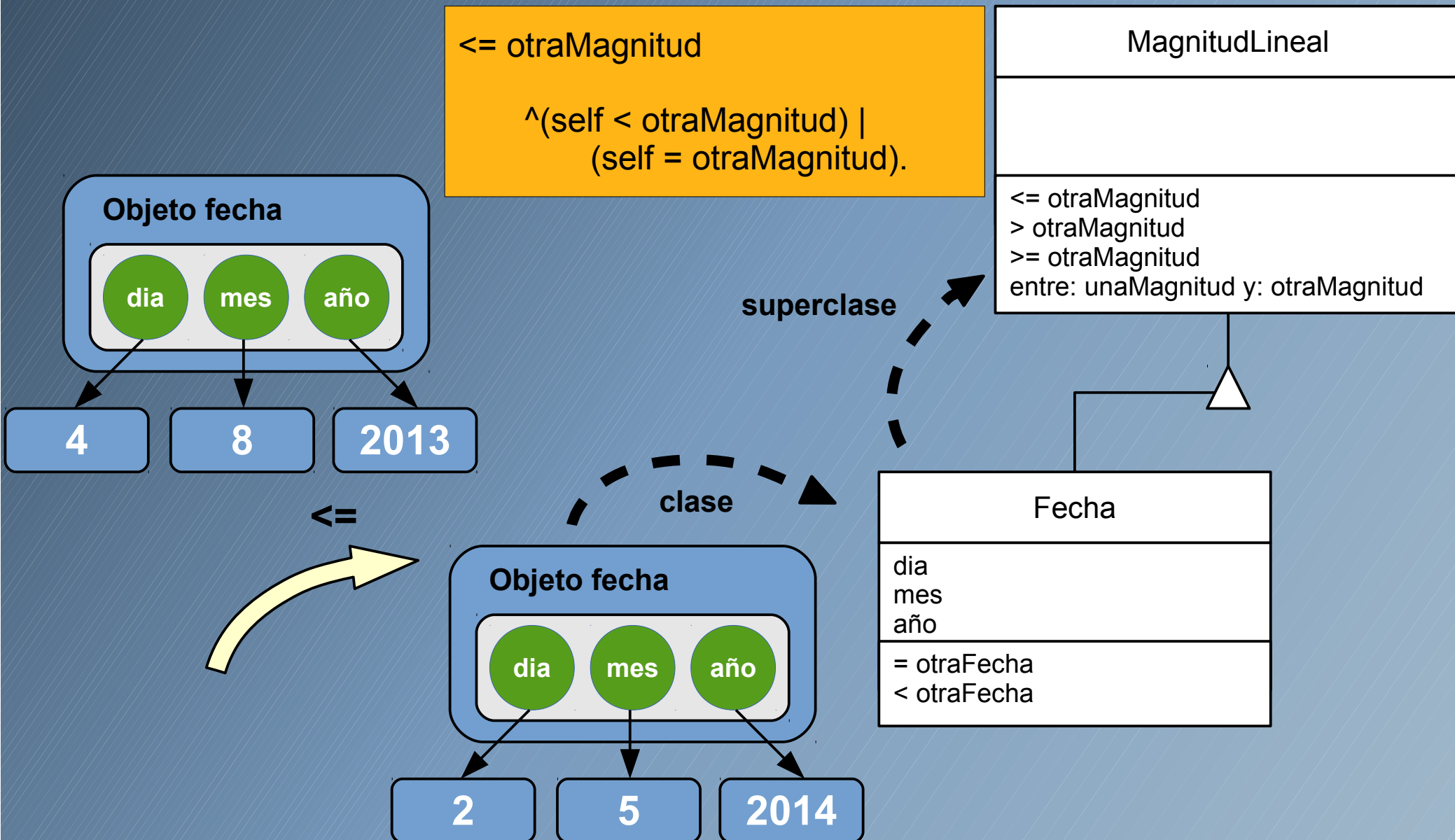
Repaso - Magnitudes lineales

- Cualquier magnitud que se pueda comparar linealmente puede utilizar la misma implementación.
- Sólo necesitamos definir dos conceptos:
 - Igualdad.
 - Ser menor a otra magnitud de mi clase.

Repaso: Magnitudes lineales



Repaso: Envío de mensaje



Entrega 2

- (+) Uso de constructores.
- (+) Uso de ;
- (+) Setters en protocolo *private*.
- (+) Buena cobertura de tests.

Entrega 2

- (-) Getters en protocolo *private* (?!).
- (-) Uso de paréntesis innecesarios.
- (-) Indentación de código.
- (-) Sobre-ingeniería en ciertos métodos.

Punto>>distanciaEnXa: otroPunto

Punto>>distanciaEnYa: otroPunto

Punto>>distanciaA: otroPunto

| deltaX deltaY |

deltaX:= self distanciaEnXa: otroPunto.
deltaY:= self distanciaEnYa: otroPunto.
^(deltaX squared + deltaY squared) sqrt.

Punto>>distanciaA: otroPunto

| deltaX deltaY |

deltaX:= self x - otroPunto x.
deltaY:= self y - otroPunto y.
^(deltaX squared + deltaY squared) sqrt.

^(deltaX squared + deltaY squared) sqrt.

Entrega 2

(-) Uso de setters privados en los tests en lugar de crear nuevos objetos

```
| punto |  
punto := Punto x:10 y:4.  
...  
self assert: ...  
...  
punto x:3; y:7.
```

```
| punto |  
punto := Punto x:10 y:4.  
...  
self assert: ...  
...  
punto := Punto x:3 y:7.
```


Entrega 2

Podrían omitir ciertas variables temporales:

```
...  
circulo := Circulo centro: (Punto x: 0 y: 0) radio: 2.  
...
```

en lugar de

```
...  
centro := Punto x: 0 y: 0.  
circulo := Circulo centro: centro1 radio: 2.  
...
```