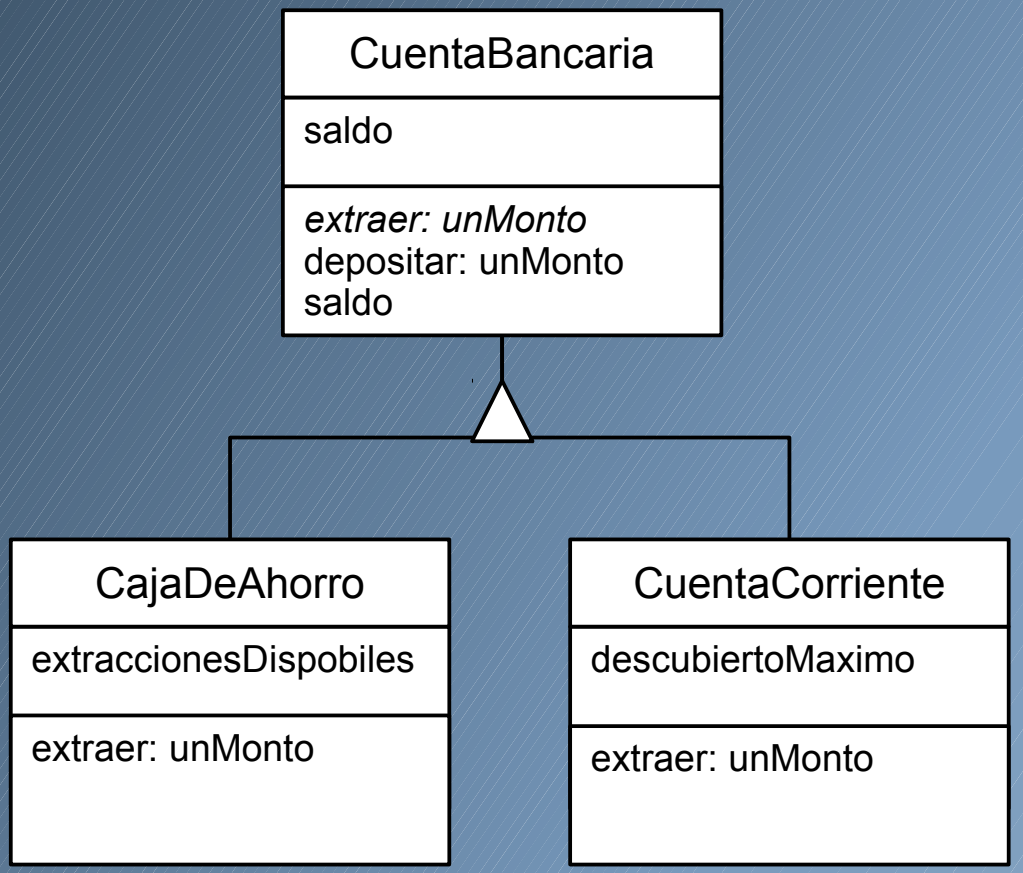
Programación III

Clase XV

Andrés Fortier

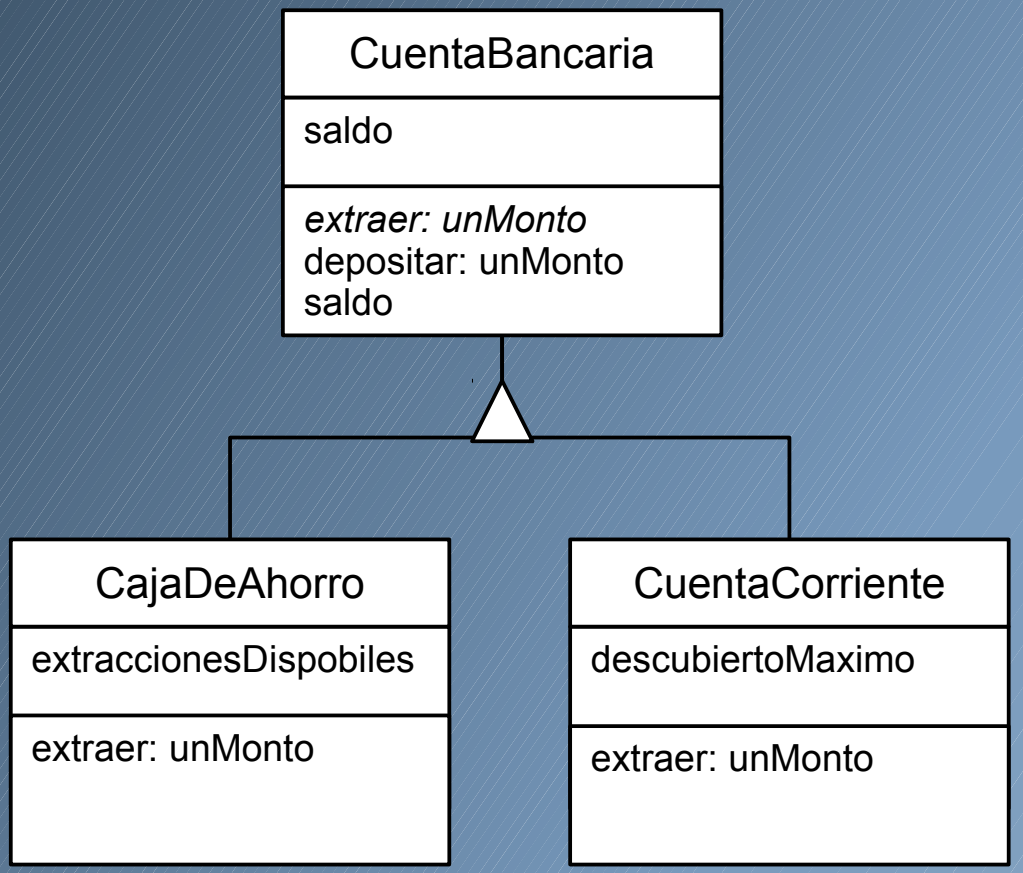
Cuentas bancarias




```
extraer: unMonto

        saldo + descubiertoMaximo >= unMonto
                ifTrue:[saldo := saldo - unMonto]
```

```
extraer: unMonto

        (saldo >= unMonto) & (extraccionesDisponibles > 0)
        ifTrue:[
                saldo := saldo – unMonto.
                extraccionesDisponibles := extraccionesDisponibles – 1.
                ]
```

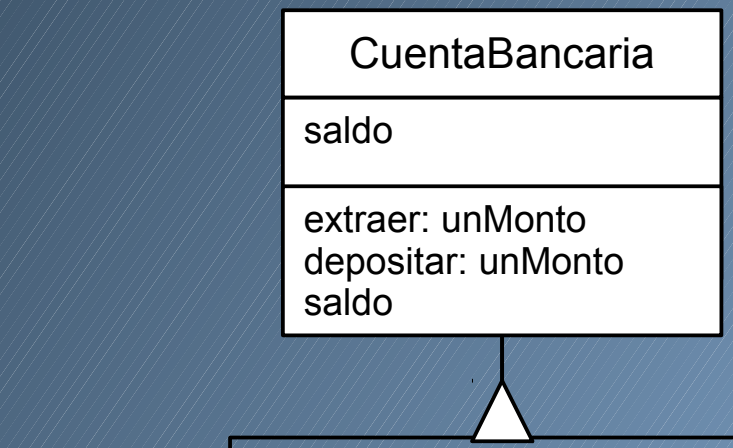
Cuentas bancarias



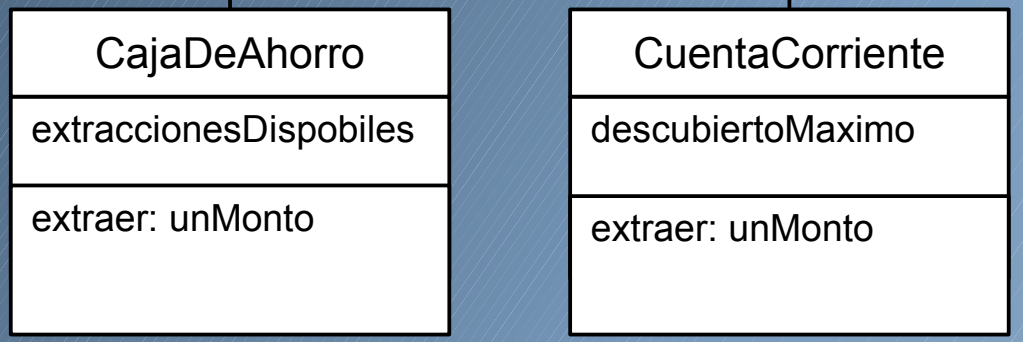

```
extraer: unMonto

        saldo + descubiertoMaximo >= unMonto
                ifTrue:[saldo := saldo – unMonto].
```

```
extraer: unMonto

        (saldo >= unMonto) & (extraccionesDisponibles > 0)
        ifTrue:[
                saldo := saldo – unMonto.
                extraccionesDisponibles := extraccionesDisponibles – 1.
                ]
```

Cuentas bancarias – uso de super



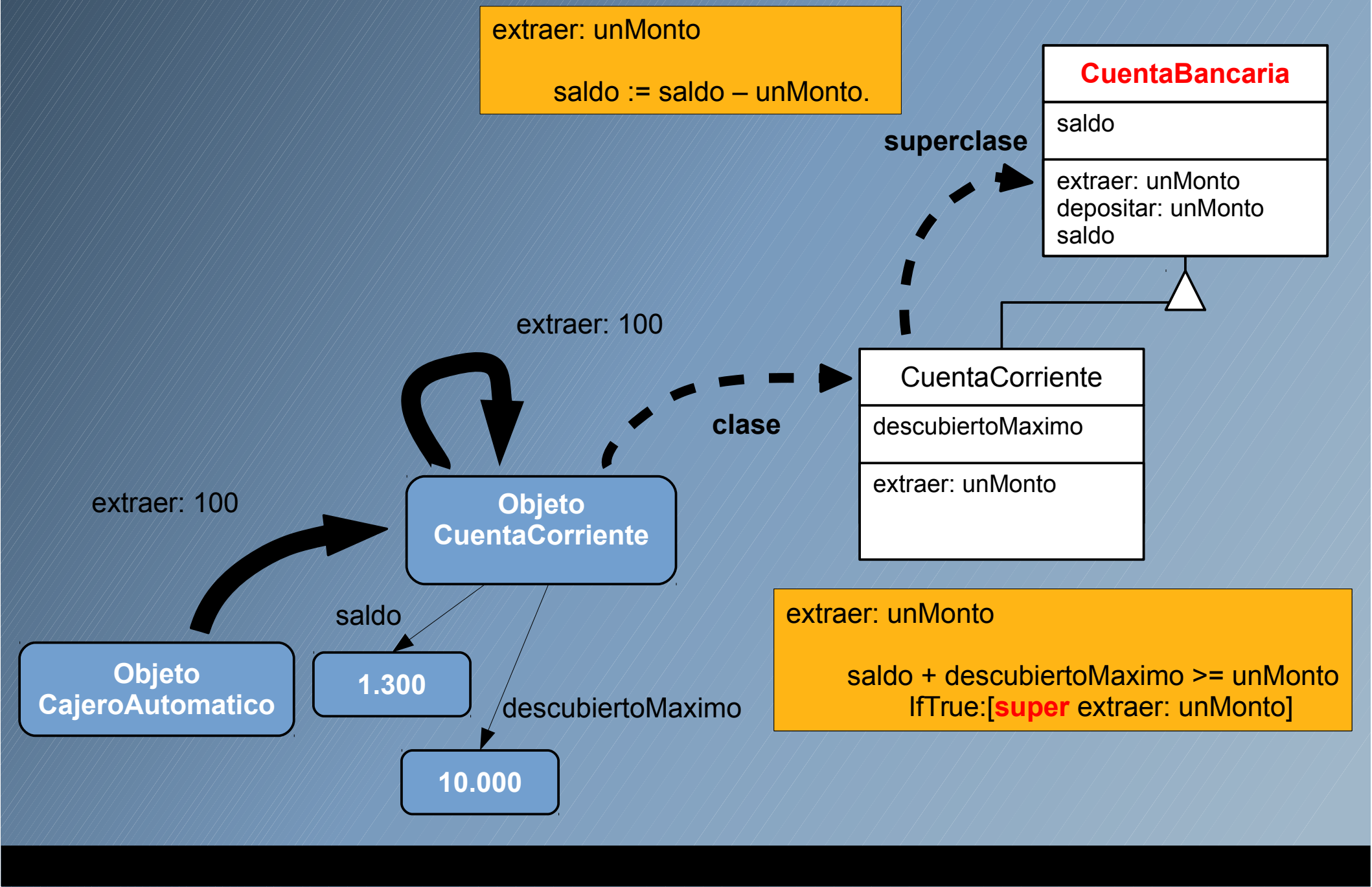

```
extraer: unMonto

        saldo := saldo – unMonto.
```




```
extraer: unMonto

        saldo + descubiertoMaximo >= unMonto
                ifTrue: [
                        super extraer: unMonto
                ]
```

```
extraer: unMonto

        (saldo >= unMonto) & (extraccionesDisponibles > 0)
        IfTrue:[
                super extraer: unMonto.
                extraccionesDisponibles := extraccionesDisponibles – 1.
                ]
```

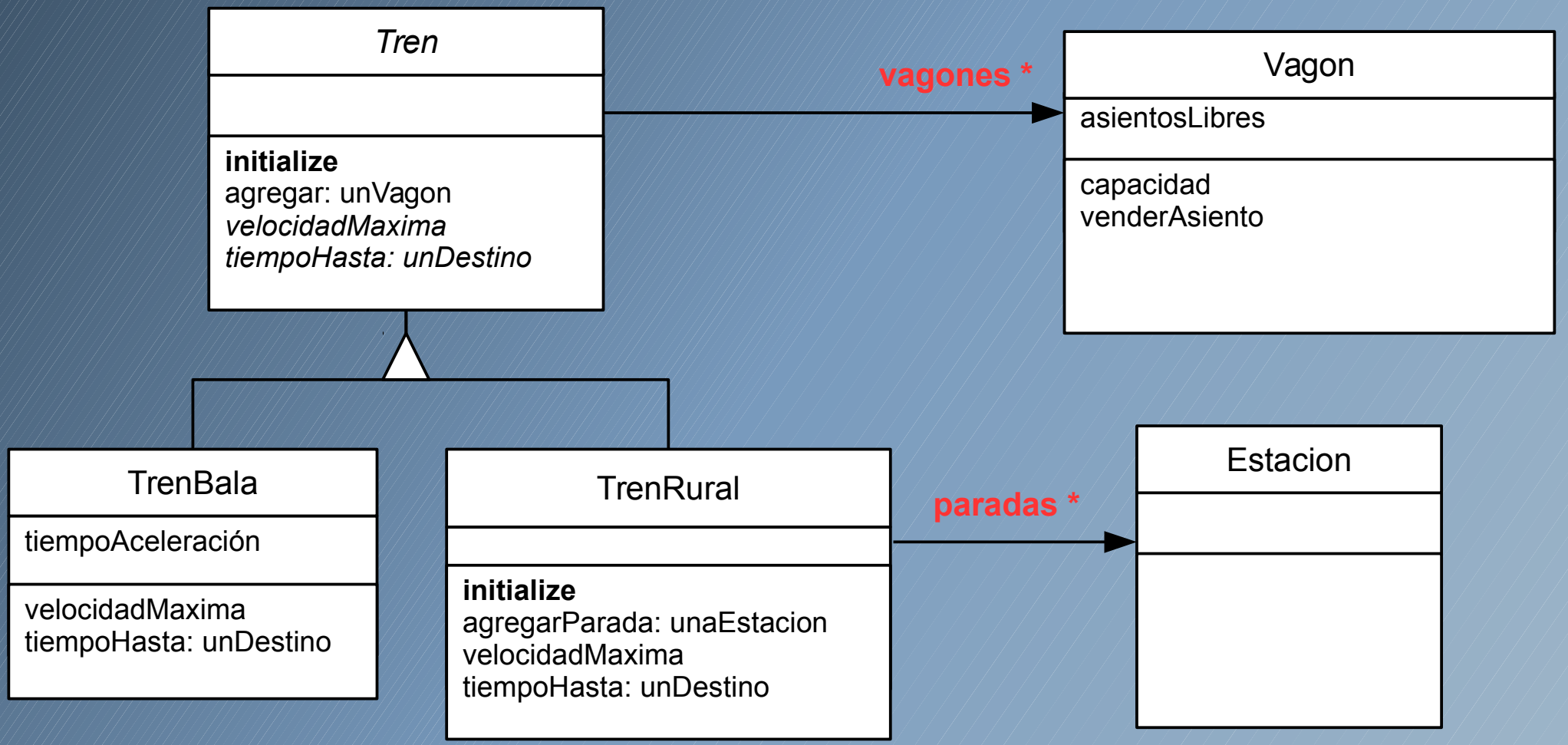
Super - envío de mensaje



super

- La pseudovariable *self* la usamos para enviarnos un mensaje a nosotros mismos.
- *super* es similar en el sentido que el receptor del mensaje sigue siendo el mismo objeto.
- Lo que cambia es a partir de qué clase se inicia la búsqueda del método.
  - La superclase de la clase en la que está definido el método.

Supongamos el siguiente modelo

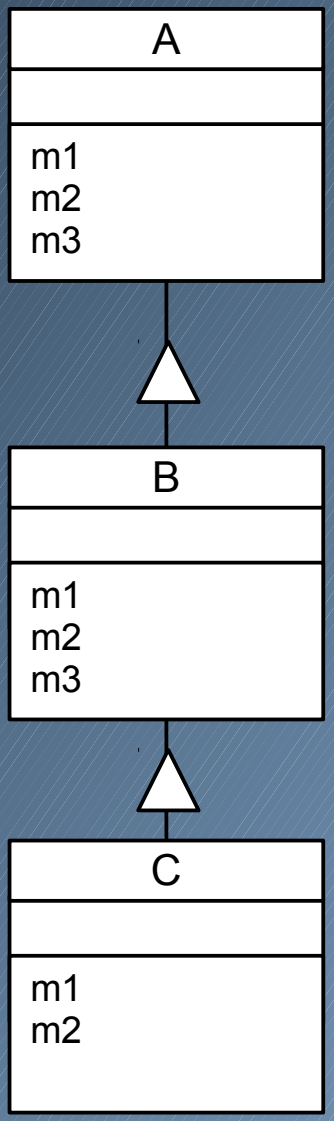


Inicialización - super

```
Tren >> inicializar

   vagones := OrderedCollection new.
```

```
TrenRural >> inicializar

   super inicializar.
   paradas := OrderedCollection new.
```

“super” diversión (cuac!)



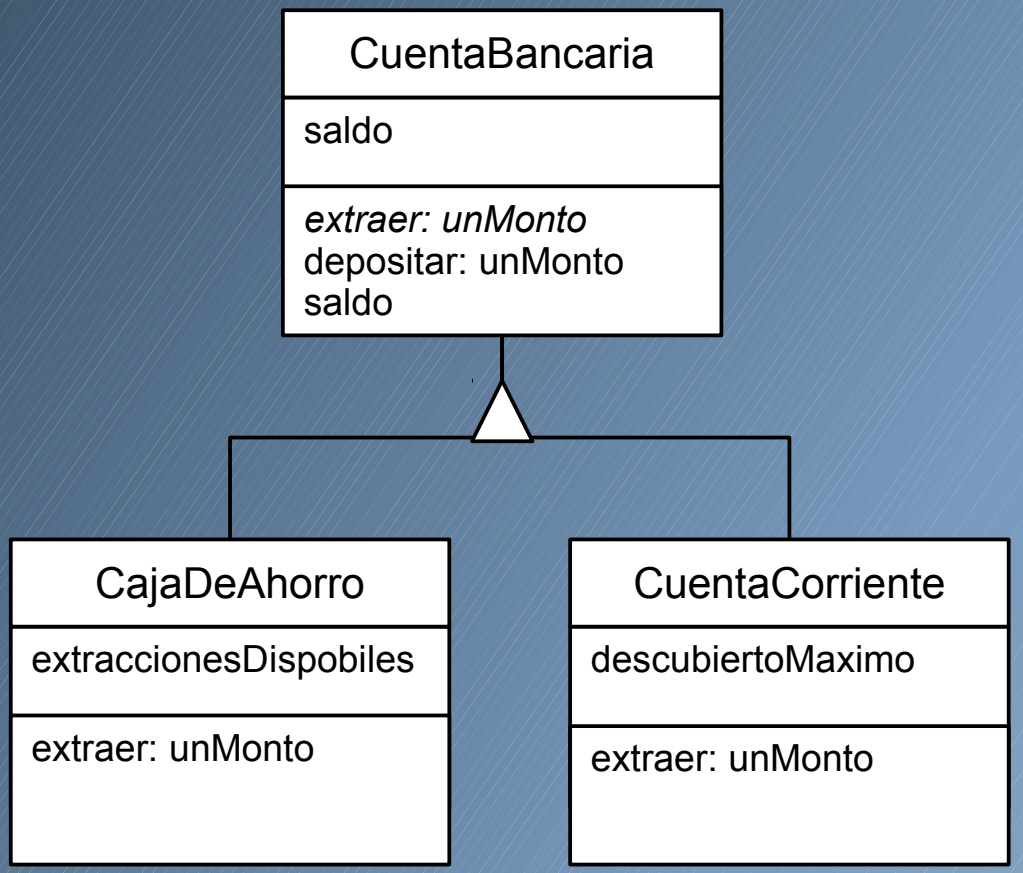
A	B	C
m1 ^10.	m1 ^2.	m1 ^super m2.
m2 ^25.	m2 ^super m2 - 5.	m2 ^44.
m3 ^self m1 + self m2.	m3 ^self m2 + super m3.	

```
| b c |
b := B new.
b m2.
b m3.
c := C new.
C m3.
```

super

- ¿Alguna superpregunta?
- Volvamos al ejemplo de cuentas bancarias para analizar posibles soluciones alternativas.

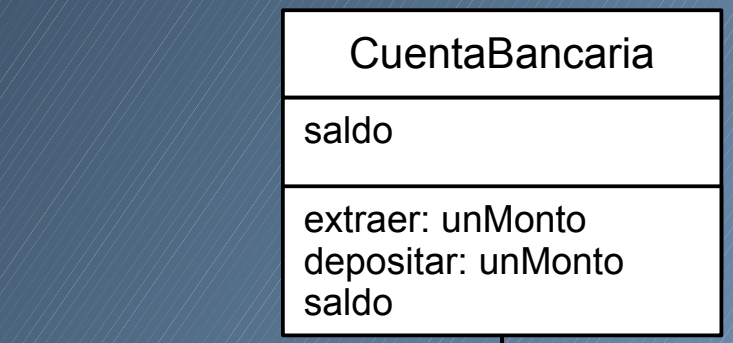
Cuentas bancarias



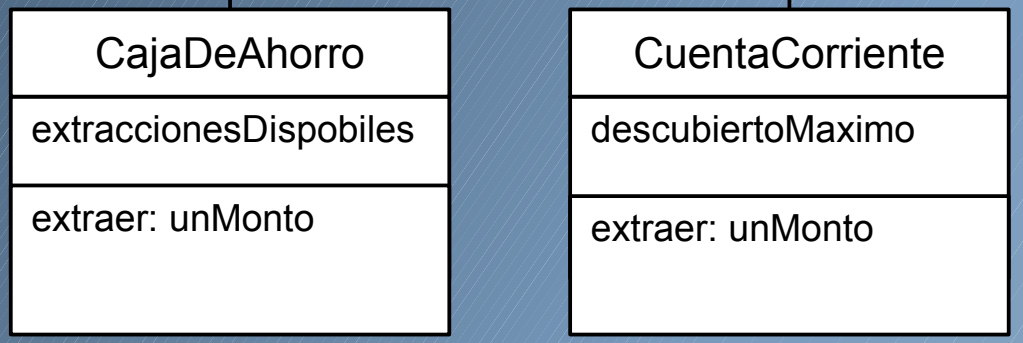

```
extraer: unMonto

        saldo + descubiertoMaximo >= unMonto
                ifTrue:[saldo := saldo - unMonto]
```

```
extraer: unMonto

        (saldo >= unMonto) & (extraccionesDisponibles > 0)
        ifTrue:[
                saldo := saldo – unMonto.
                extraccionesDisponibles := extraccionesDisponibles – 1.
                ]
```

Cuentas bancarias – opción 1



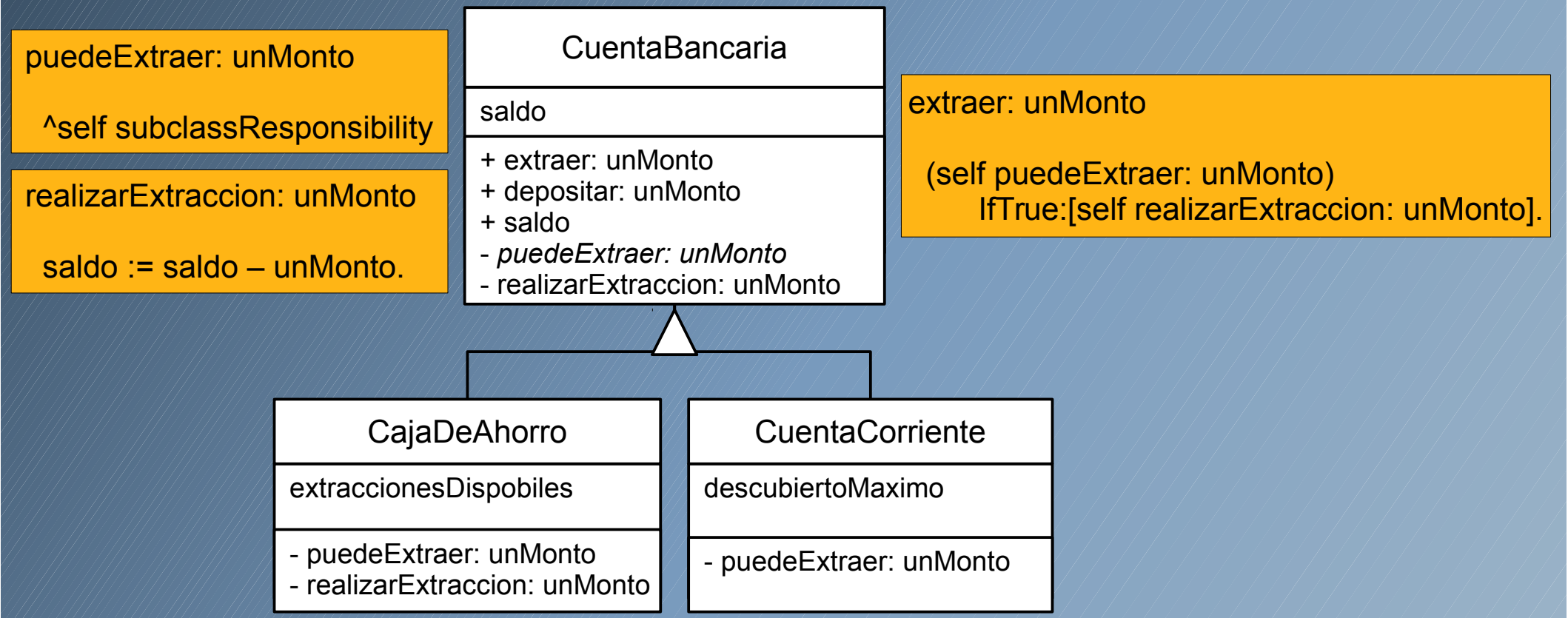

```
extraer: unMonto

        saldo := saldo – unMonto.
```




```
extraer: unMonto

        saldo + descubiertoMaximo >= unMonto
                IfTrue:[super extraer: unMonto]
```

```
extraer: unMonto

        (saldo >= unMonto) & (extraccionesDisponibles > 0)
        IfTrue:[
                super extraer: unMonto.
                extraccionesDisponibles := extraccionesDisponibles – 1.
                ]
```

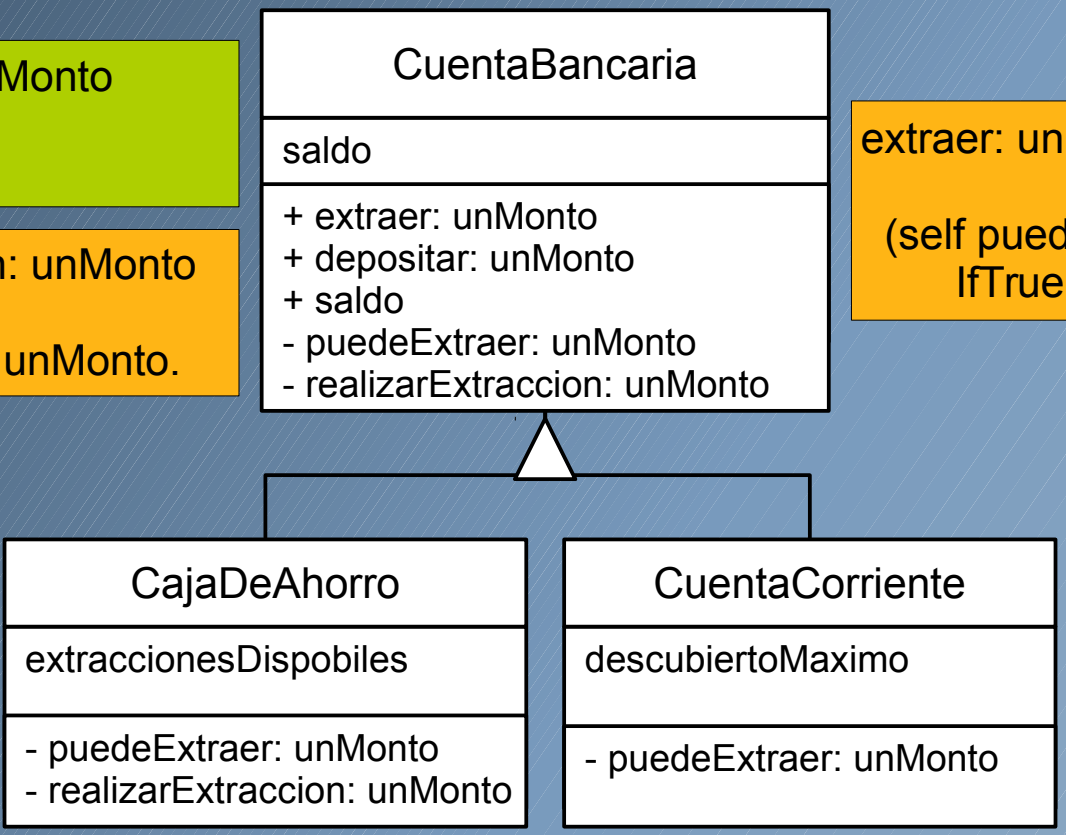
Cuentas bancarias – opción 2



Cuentas bancarias – opción 2 – variación 1

```
puedeExtraer: unMonto

  ^true
```

```
realizarExtraccion: unMonto

  saldo := saldo – unMonto.
```



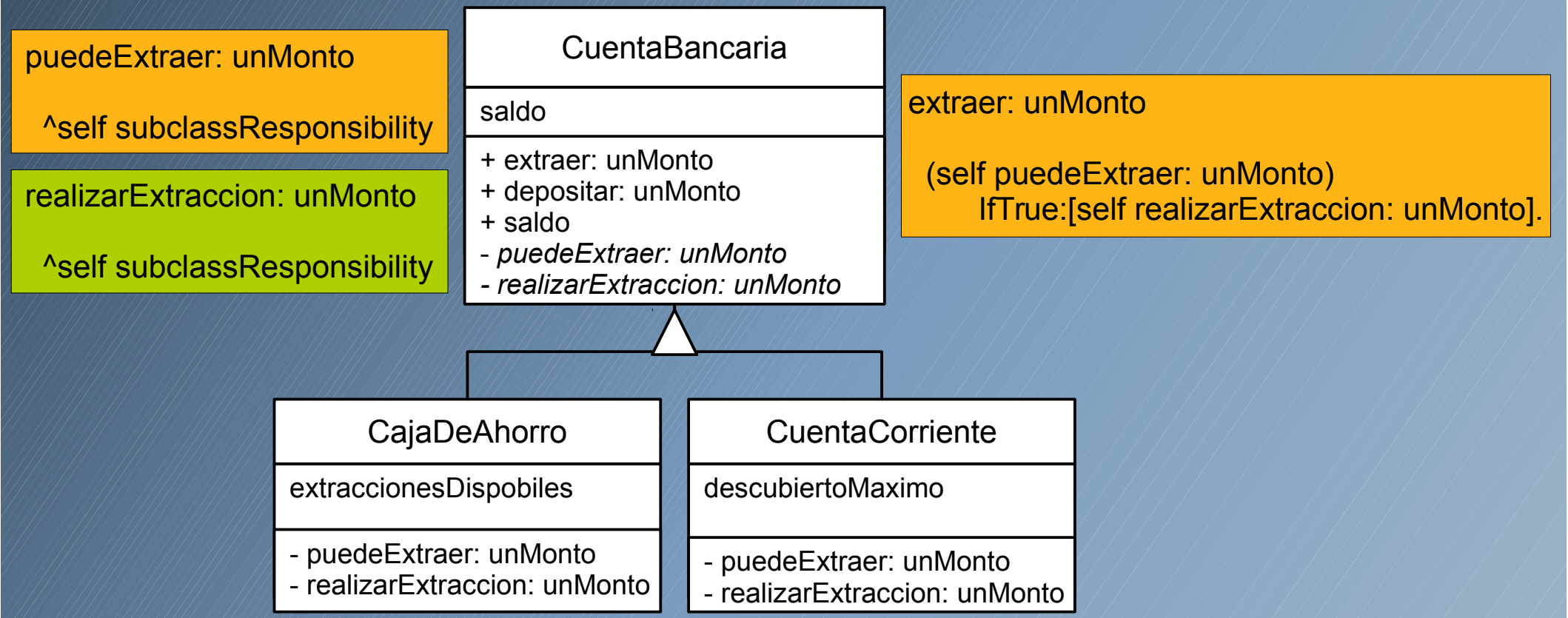

```
extraer: unMonto

  (self puedeExtraer: unMonto)
        IfTrue:[self realizarExtraccion: unMonto].
```

```
puedeExtraer: unMonto

  ^(saldo >= unMonto) & (extraccionesDisponibles > 0).
```

```
realizarExtraccion: unMonto

  super realizarExtraccion: unMonto.
  extraccionesDisponibles := extraccionesDisponibles – 1
```

```
puedeExtraer: unMonto

  ^saldo + descubiertoMaximo >= unMonto.
```

Cuentas bancarias – opción 2 – variación 2




```
puedeExtraer: unMonto

  ^(saldo >= unMonto) & (extraccionesDisponibles > 0).
```

```
realizarExtraccion: unMonto

  saldo := saldo - unMonto.
  extraccionesDisponibles := extraccionesDisponibles – 1
```

```
puedeExtraer: unMonto

  ^saldo + descubiertoMaximo >= unMonto.
```

```
realizarExtraccion: unMonto

  saldo := saldo - unMonto.
```

Cerrando

- A partir de ahora vamos a sumar herramientas.
- No hay una única forma de hacer las cosas.