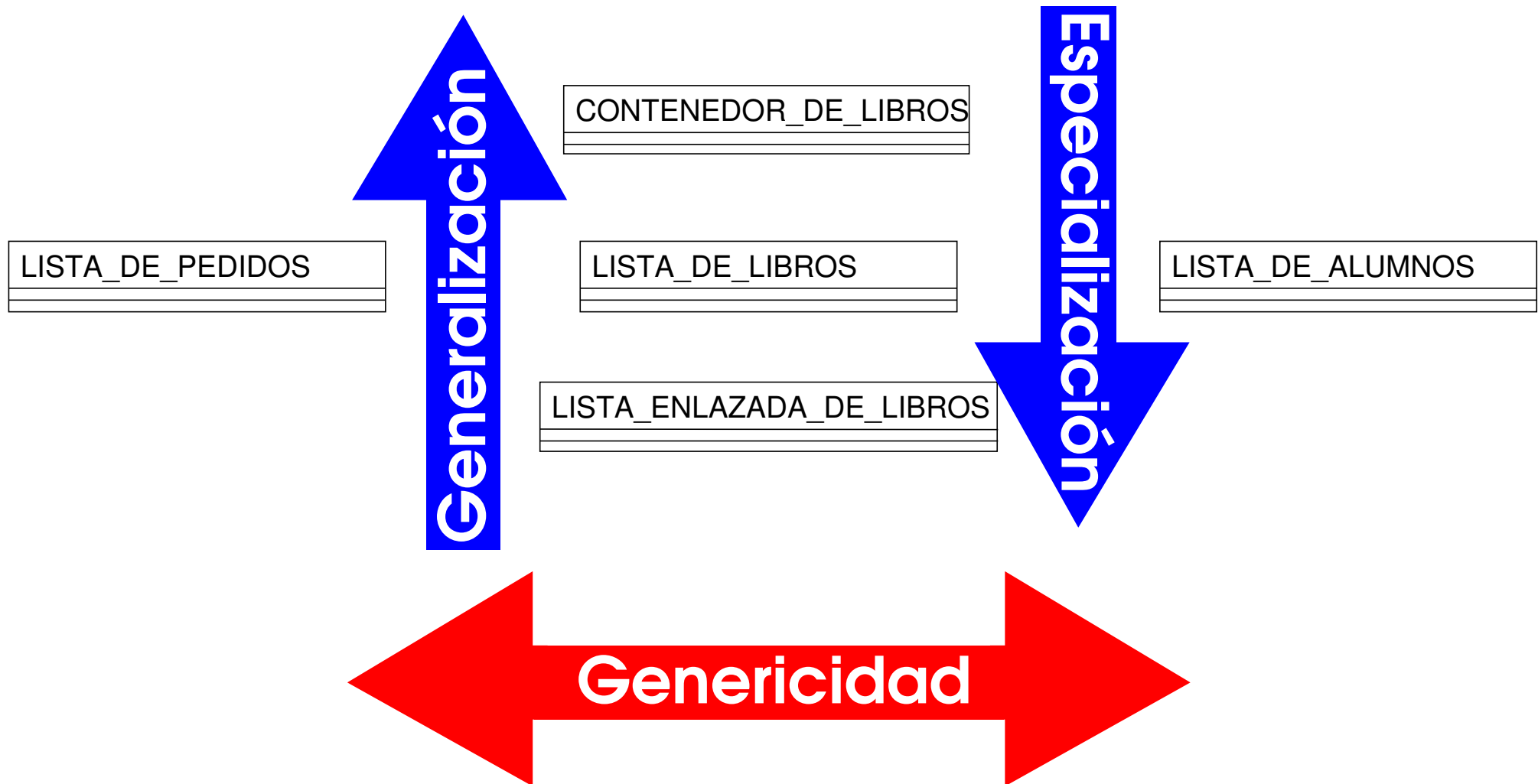


Programación III
I.T. Informática de Sistemas
Genericidad.



Prof. Félix Prieto Arambillet
Departamento de Informática
Universidad de Valladolid
Curso 2003/2004

Dimensiones de la “generalidad”



- Utilizamos un **parámetro genérico**, G en este caso
- Para instanciar la clase debemos proporcionar como **parámetro actual** un tipo, por ejemplo `a:PILA[INTEGER]`
- De una clase genérica se derivan tantos tipos como parámetros actuales podamos proporcionar.
- El parámetro puede aparecer en el código de los métodos, definiendo el tipo de las entidades utilizadas
- Debemos reconsiderar las reglas que controlan el la sustitución de **argumentos formales** por **argumentos actuales** en el paso de mensajes

```
class PILA[G]
feature
  contador: INTEGER
  -- Número de elementos
  vacia: BOOLEAN is
  -- ¿Está vacía?
  do ... end
  llena: BOOLEAN is
  -- ¿Está llena?
  do ... end
  cima: G is
  -- Elemento de la cima de la pila.
  do ... end
  mete( x:G) is
  -- Introduce x por la cima
  do ... end
  saca is
  -- Quita el elemento de la cima
  do ... end
end -- Clase PILA
```

Genericidad y mensajes (I)

Regla 1 Sea c una clase no genérica y $f(a:T) : U$ **is** ... la definición de un método que aparece en ella. La llamada $x.f(d)$ que aparece en un método de la clase B es correcta si:

- x es de tipo c (o compatible por herencia).
- f está exportada a B (explícita o implícitamente).
- d es de tipo T (o compatible por herencia).

En este caso el resultado de f es de tipo U .

Genericidad y mensajes (II)

Regla 2 Sea c una clase genérica con parámetro formal G y $h(a:G):G$ **is** ... la definición de un método que aparece en ella. La llamada $y.h(e)$ que aparece en un método de la clase B es correcta si existe un tipo V , parámetro actual de c tal que :

- y es de tipo $C[V]$ (o compatible por herencia).
- h está exportada a B (explícita o implícitamente).
- e es de tipo V (o compatible por herencia).

En este caso el resultado de h es de tipo V .

Utilización de clases genéricas

ejemplo **is**

local

```
pa: PILA[ALUMNO]; pp: PILA[PEDIDO]
```

```
p: PEDIDO; a: ALUMNO;
```

do

— Código que crea las instancias adecuadas

—

— Llamadas válidas

```
pa.mete(a); pp.mete(p); a:=pa.cima
```

— Llamadas no válidas

```
pp.mete(a); pa.mete(p); p:=pa.cima
```

Restricciones asociadas a la genericidad

Regla 3 Si x es una entidad cuyo tipo G es parámetro formal de una clase genérica las operaciones permitidas sobre x son:

- *Uso en el lado derecho o izquierdo de una asignación, en la que el otro lado también es de tipo G*
- *Uso en el lado derecho o izquierdo de una expresión lógica en la que el otro lado es también de tipo G*
- *Uso como parámetro actual para una rutina correspondiendo a un parámetro formal declarado de tipo G o **ANY**.*
- *Uso como objetivo de una característica de **ANY**.*