# Recursos compartidos y su especificación

*Lecturas: apuntes de especificación de la asignatura*

**Manuel Carro**

Universidad Politécnica de Madrid

22 de noviembre de 2007







## Del problema a la solución

Problema:

muchos datos a tener en cuenta simultáneamente

Solución:

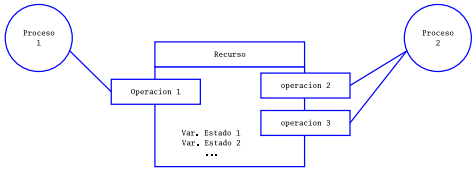
- Refinamiento progresivo
- Concentrarse en cada momento en pocos detalles:
  - Análisis de procesos y recursos
  - Estudio de condiciones de seguridad
  - Estudio de condiciones de vivacidad
- Algunos pasos automáticos
- Otros necesitan estudio especial





## Recursos compartidos

- Similares a *cápsulas* (Gehani)
- Tipo abstracto con:
  - Estado interno
  - Operaciones
  - Condiciones de sincronización
  - Exclusión mutua implícita




- Notación alto nivel
- Independiente de la implementación

## Especificación

- Nos centraremos en la especificación:
  - Formal (no ambigua, teoría de prueba)
  - Independiente del lenguaje
  - Clara y breve (al menos eso queremos)
- Muchas posibilidades
- Usaremos lógica primer orden
- Estructurada por recursos
- En cierto sentido, aumento de TAD de estructuras de datos





## Recursos y TADs

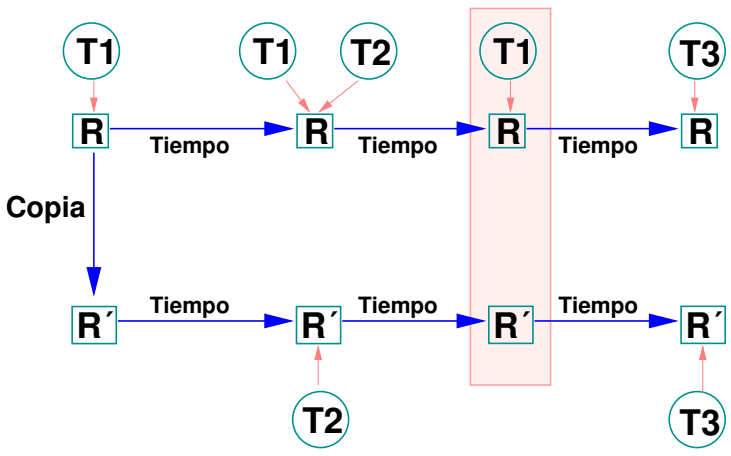
- **Semejanzas:**
  - Encapsulamiento datos
  - Reusabilidad
  - Razonamiento corrección recurso independiente de su uso
  - Razonamiento uso correcto sin su implementación
- **Diferencias:**
  - Estado recurso dependiente del mundo externo (tareas accediendo)
  - No noción igualdad, no copia
  - Notación aumentada
  - Algunos argumentos *privilegiados*: recurso único $\longrightarrow$ cambios propagados inmediatamente





## Historia de un recurso

¿Qué problemas hay para copiar un recurso?



## Especificación: visión general

**C-TADSOL** Nombre Recurso

**OPERACIONES**
**ACCIÓN** $\mathrm{Op}_1$: *Tipo_Recurso[es]* $\times \ldots$

**SEMÁNTICA**
**DOMINIO:**
**TIPO:** $Tipo\_Recurso = \ldots$
**INVARIANTE:** $\forall r \in Tipo\_Recurso \cdot Inv(r)$

**INICIAL($r$):** $Ini(r)$

**CPRE:** $P(r, a_1, \ldots, a_n)$
**$\mathrm{Op}_1$(r, $a_1$, ..., $a_n$)**
**POST:** $Q(r, a_1, \ldots, a_n)$





## Declaración de interfaz

- Nombre de operaciones + tipos y *modos* de argumentos
- Modos: no necesarios, ayudan a documentar: *[es], [e], [s]*
- Tipos:
  - Básicos: $\mathbb{B}$, $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$
  - Algebraicos: tuplas [ (Ta $\times$ Tb) ], constructores [ Nombre(Ta $\times$ Tb) ], unión de tipos [ (Ta | Tb) ]
  - Especiales: secuencias [ Secuencia(Ta) ], conjuntos [ Conjunto(Ta) ], tablas / *maps* / funciones parciales [ $Ta \nrightarrow Tb$ ]

| **Tipo** | **Ejemplo valor** |
|---|---|
| $\mathbb{N} \times \mathbb{B}$ | `(3, falso)` |
| $par(\mathbb{Z} \times \mathbb{R})$ | `par(-7, π)` |
| Secuencia($\mathbb{B}$) | ⟨`cierto, falso, falso, cierto`⟩ |
| Conjunto($\mathbb{R}$) | {`3.4, e,` log `2`} |
| $\mathbb{N} \nrightarrow \mathbb{B}$ | {`(2, true),(3,true),(4,false),(9,false)`} |





## Dominio: tipo + invariante

- Tipos: admitimos campos con nombre en tuplas ($\equiv$ registros / *records*)

  **TIPO:** $Tipo Complejo = Par(real: \mathbb{R} \times imaginario: \mathbb{R})$

- Invariante: restringe valores permitidos por el tipo

**TIPO:** $Tipo\_Irracional = \mathbb{R}$
**INVARIANTE:** $\forall i \in Tipo\_Irracional \bullet \nexists p, q \in \mathbb{Z} \cdot i = \frac{p}{q}$

**TIPO:** $Tipo\_Creciente = \text{Secuencia}(\mathbb{N})$
**INVARIANTE:** $\forall s \in Tipo\_Creciente \cdot (l = \text{Longitud}(s) \wedge \forall k, 1 \leq k \leq l - 1 \bullet s(k) < s(k+1))$

## Invariante

**INVARIANTE:** $I(r)$

*Magnitud o ley que se mantiene tras una transformación*

- Verdad antes de la **entrada** y tras la **salida** de cada operación
- Puede *violarse* dentro de una operación
- Muy importante:
  - Permite establecer propiedades generales
  - Permite trabajar contra esas propiedades





## Estado inicial recurso

**INICIAL($r$):** $In(r)$

- Determina valor inicial recurso
  (corrección en base a la historia del recurso)

- ¿Operación separada?
  - Sería algo especial
  - Sólo podría ser llamada una vez
  - Muchos lenguajes permiten especificar estado inicial





## Especificación de operaciones

**PRE:** $P(a_1, \ldots, a_n)$
**CPRE:** $C(r, a_1, \ldots, a_n)$
  **Op$_1$(r, a$_1$, . . . , a$_n$)**
**POST:** $Q(r, a_1, \ldots, a_n)$

- Precondición secuencial (**PRE**):
  - Condiciones no relacionadas con la concurrencia
- Precondición de concurrencia (**CPRE**):
  - Sincronización entre operaciones
- Postcondición (**POST**):
  - Cambio en el recurso / argumentos

## Especificación de operaciones (Cont.)

- División precondiciones: $P(a_1 \ldots a_n) \wedge C(r, a_1, \ldots, a_n)$
- Relacionadas con el recurso compartido: **CPRE**
  - Determinan sincronización
  - Pueden causar suspensión
  - Recurso cambiado por otras tareas
- Relacionadas con otros argumentos: **PRE**
  - Determinan corrección llamada
  - Pueden causar error de ejecución, excepciones, etc.



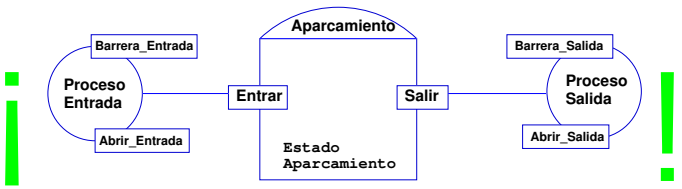


## Uso de aparcamiento

```
R_Aparcamiento: Tipo_Aparcamiento
```

```
loop  -- Entrada
    Barrera_Entrada;
    Entrar(R_Aparcamiento);
    Abrir_Entrada;
end loop;
```

```
loop  -- Salida
    Barrera_Salida;
    Salir(R_Aparcamiento);
    Abrir_Salida;
end loop;
```








## Aparcamiento

**ACCIÓN** Entrar: *Tipo_Aparcamiento[es]*
**ACCIÓN** Salir: *Tipo_Aparcamiento[es]*

**TIPO:** $Tipo\_Aparcamiento = 0..Max$

**INVARIANTE:** *cierto*
**INICIAL(a):** $a = 0$

**CPRE:** *cierto*
**Salir(a)**
**POST:** $a^{sal} = a^{ent} - 1$

**CPRE:** $a < Max$
**Entrar(a)**
**POST:** $a^{sal} = a^{ent} + 1$

¿$a < Max$ **necesaria?**

## Almacén de un dato

**ACCIÓN** Poner: *Tipo_Almacen[es]* × *Tipo_Dato[e]*
**ACCIÓN** Tomar: *Tipo_Almacen[es]* × *Tipo_Dato[s]*

**TIPO:** *Tipo_Almacén* = *(Dato: Tipo_Dato* × *HayDato:* $\mathbb{B}$*)*
**INVARIANTE:** *cierto*
**INICIAL(**b**):** $\neg b.HayDato$

**CPRE:** $b.HayDato$
**Tomar(b, e)**
**POST:** $e^{sal} = b^{ent}.Dato \wedge \neg b^{sal}.HayDato$

**CPRE:** $\neg b.HayDato$
**Poner(b, e)**
**POST:** $b^{sal}.Dato = e^{ent} \wedge b^{sal}.HayDato$

- Recordad implementación: sólo un semáforo
- Veremos prueba corrección seguridad





## Almacén un dato (Cont.)

- Separación estado recurso / dato almacenado
- Operaciones excluyentes:

  **CPRE:** $b.HayDato$
  **Tomar(b, e)**
  **POST:** $e^{sal} = b^{ent}.Dato \wedge \neg b^{sal}.HayDato$

  **CPRE:** $\neg b.HayDato$
  **Poner(b, e)**
  **POST:** $b^{sal}.Dato = e^{ent} \wedge b^{sal}.HayDato$

$$\mathbf{CPRE}(\mathbf{Tomar}(\mathbf{b}, \mathbf{e_1})) \wedge \mathbf{CPRE}(\mathbf{Poner}(\mathbf{b}, \mathbf{e_2})) \equiv$$
$$\neg b.HayDatos \wedge b.HayDatos \equiv$$
$$false$$

- Respetar acceso a dato asegura exclusión mutua

  $\Longrightarrow$ no necesario semáforo dedicado





## *Buffer*

**ACCIÓN** Poner: *Tipo_Buffer[es]* × *Tipo_Dato[e]*
**ACCIÓN** Tomar: *Tipo_Buffer[es]* × *Tipo_Dato[s]*

**TIPO:** *Tipo_Buffer* = Secuencia*(Tipo_Dato)*
**INVARIANTE:** $\forall b \in Tipo\_Buffer \bullet \text{Longitud}(b) \leq MAX$

**INICIAL(**b**):** $\text{Longitud}(b) = 0$

**CPRE:** $\text{Longitud}(b) > 0$
**Tomar(b, d)**
**POST:** $l = \text{Longitud}(b^{ent}) \wedge b^{ent}(1) = d^{sal} \wedge b^{sal} = b^{ent}(2..l)$

**CPRE:** $\text{Longitud}(b) < MAX$
**Poner(b, d)**
**POST:** $l = \text{Longitud}(b^{ent}) \wedge \text{Longitud}(b^{sal}) = l + 1 \wedge$
$b^{sal}(l+1) = d^{ent} \wedge b^{sal}(1..l) = b^{ent}$

# Productor/Consumidor de pares e impares

**ACCIÓN** Poner: $Tipo\_PI[es] \times \mathbb{N}[e]$
**ACCIÓN** Tomar: $Tipo\_PI[es] \times \mathbb{N}[s]$

**TIPO:** $Tipo\_PI = (Dato : \mathbb{N} \times \ HayDato : \mathbb{B})$

**INVARIANTE:** *cierto*

**INICIAL**(b)**:** $\neg b.HayDato$

**CPRE:** $\neg b.HayDato$
**Poner(b, d)**
**POST:** $b^{sal}.Dato = d^{ent} \wedge b^{sal}.HayDato$

**CPRE:** $b.HayDato \wedge (b.Dato \ mod \ 2 = 0)$
**TomarPar(b, d)**
**POST:** $b^{sal}.Dato = d^{ent} \wedge \neg b^{sal}.HayDato$

**CPRE:** $b.HayDato \wedge (b.Dato \ mod \ 2 = 1)$
**TomarImpar(b, d)**
**POST:** $b^{sal}.Dato = d^{ent} \wedge \neg b^{sal}.HayDato$





# Productor/Consumidor de pares e impares

Variante con una sola operación de tomar

**ACCIÓN** Poner: $Tipo\_PI[es] \times \mathbb{N}[e]$
**ACCIÓN** Tomar: $Tipo\_PI[es] \times \mathbb{N}[s] \times Tipo\_Paridad[e]$

**TIPO:** $Tipo\_PI = (Dato : \mathbb{N} \times \ HayDato : \mathbb{B})$
$Tipo\_Paridad = Par|Impar$
**INVARIANTE:** *cierto*

**INICIAL**(b)**:** $\neg b.HayDato$

**CPRE:** $\neg b.HayDato$
**Poner(b, d)**
**POST:** $b^{sal}.Dato = d^{ent} \wedge b^{sal}.HayDato$

**CPRE:** $b.HayDato \wedge ((b.Dato \ mod \ 2 = 0) \leftrightarrow (t = Par))$
**Tomar(b, d, t)**
**POST:** $b^{sal}.Dato = d^{ent} \wedge \neg b^{sal}.HayDato$