

GRAU D'ENGINYERIA INFORMÀTICA

# PROGRAMACIÓ II

## CURS 12-13

### Bloc 1:

## Mòdul i abstracció de dades (2)

**Laura Igual**

Departament de Matemàtica Aplicada i Anàlisi

Facultat de Matemàtiques

Universitat de Barcelona

# **EXAMPLE:**

## **Descomposició funcional descendent i Descomposició orientada a objectes**

Exemple extret del capítol 20 del llibre: “**Construcción de software orientado a objetos**”, Bertrand Meyer. Prentice Hall, 1998.

# Exemple

- Sistema de reserves per a una companyia Aérea
  - Stats: passos de processament
    - Identificació de l'usuari,
    - Consulta sobre vols,
    - Consulta sobre places,
    - Reserva.

# Exemple

- Sistema interactiu multi-panel
  - Patró general:
    - *Estat* (panel amb certes consultes)
    - *Transició* (selecció del següent pas a realitzar)

# Exemple: Panel

## – Consulta de vols –

Vol des de:

Barcelona

Destí:

París

Sortida prevista:

22 Maig

Arribada:

22 Maig

Companyia aèrea:

Requisits especials:

---

VOLS DISPONIBLES: 1

Vuelo: AA 42 Sortida 8:25 Arribada 10:05 Escala: -

Escolir una opció:

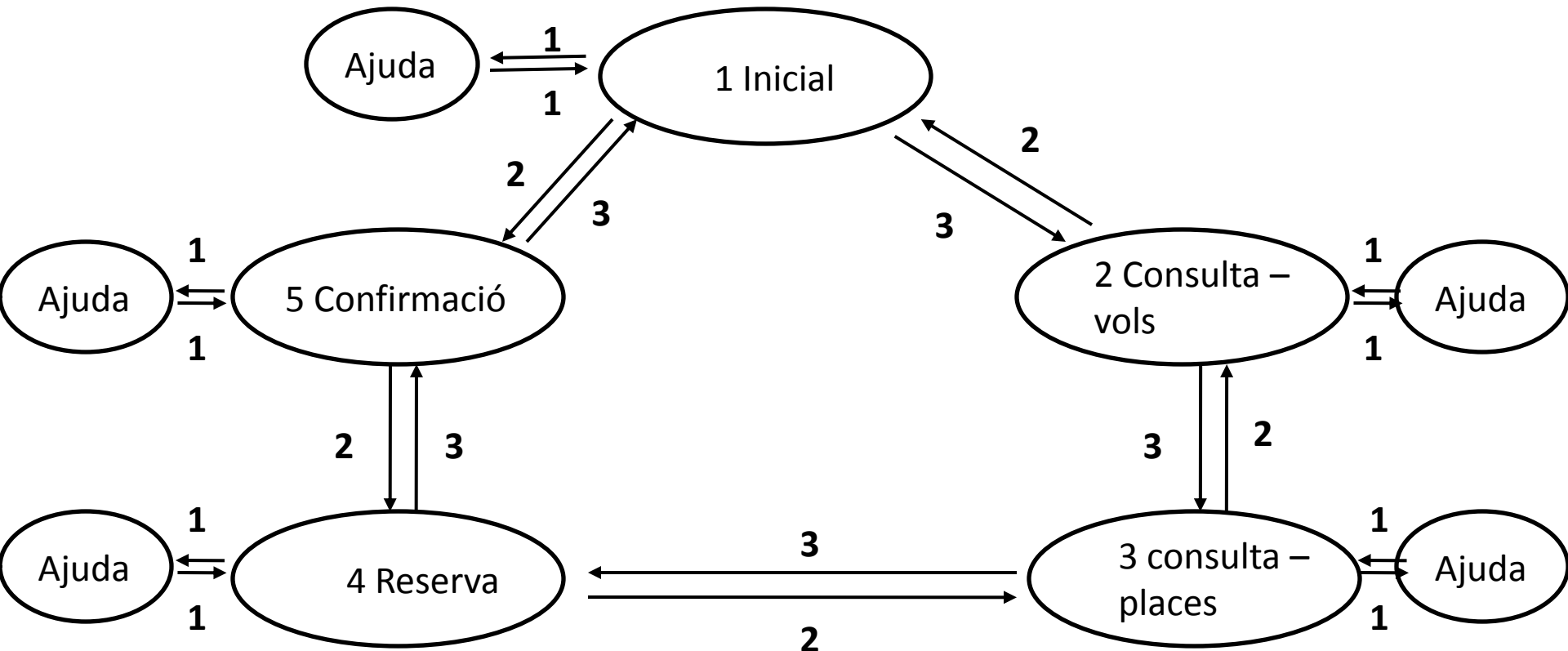
0 – Sortida

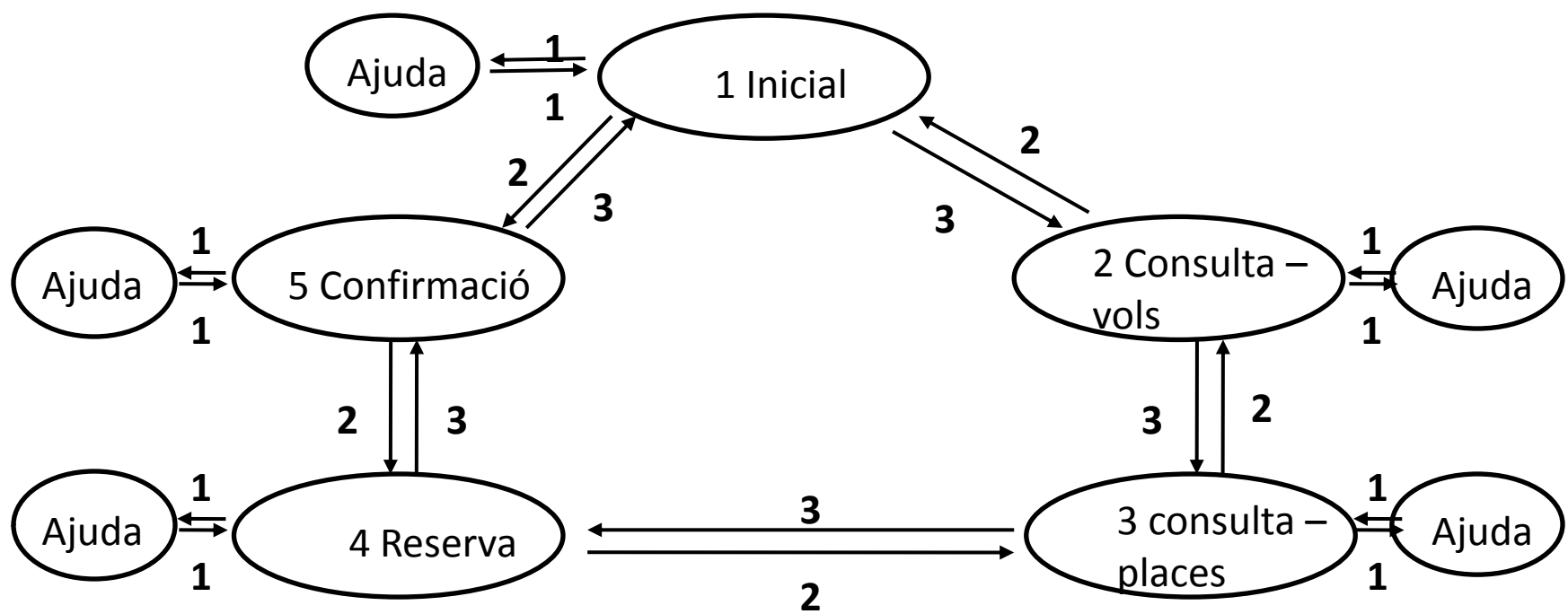
1 – Ajuda

2 – Següent petició

3 – Reserva plaça

# Exemple: Un diagrama de transició





**Taula de transició:**

Estat / Opció	0	1	2	3
1 (Inicial)	-1	0	5	2
2 (Vols)		0	1	3
3 (Places)		0	2	4
4 (Reserves)		0	3	5
5 (Confirm.)		0	4	1
0 (Ajuda)		Tornar		
-1 (Final)				

# Exemple: Esquemes del programa

- Primer intent simple
- Solució funcional descendent
- Solució orientada a objectes



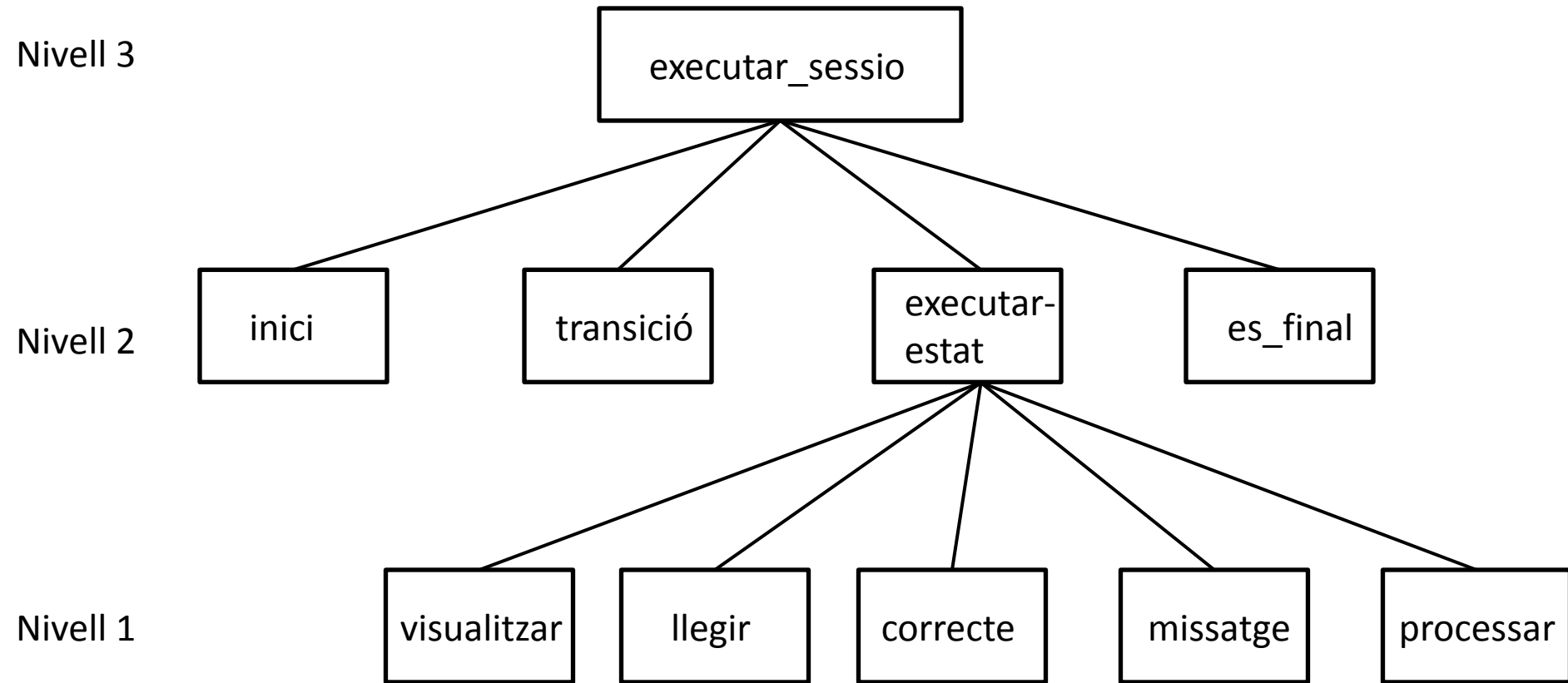
# Exemple: Primer intent simple

- $B_{\text{consulta}}$   
"Mostra el panel de Consultes de Vols"  
**repetir**  
"Llegir la resposta de l'usuari i escollir C com a següent pas"  
**si** "Hi ha un error a la resposta" llavors  
"Mostrar el misstage d'error apropiat" **final**  
**fins no** "error a la resposta" **final**  
"Processar la resposta"  
**cas** C **en**  
     $C_0$  : **goto** BConsulta  
     $C_1$  : **goto** B<sub>Ajuda</sub>  
     $C_2$  : **goto** B<sub>Reserva</sub>

Es repeteix aquest bloc per a cada estat.

# Exemple:

## Descomposició funcional descendent



# Exemple: Rutines

```
executar_sessio es
```

```
-- Executa una sessió completa del sistema  
interactiu
```

```
local
```

```
    estat, seguent: INTEGER
```

```
fer
```

```
    estat := inici
```

```
    repetir
```

```
        executar_estat(estat, → seguent)
```

```
-- La rutina executar_estat actualitza  
el valor de seguent, a més d'executar  
les accions associades al estat.
```

```
        estat := transicio(estat, seguent)
```

```
    fins es_final(estat) final
```

```
final
```

# Exemple: Rutines

```
executar_estat (in e: INTEGER; out op: INTEGER) es  
  -- Executa les accions associades al estat e,  
  tornant en  
  -- op l'opció escollida per l'usuari per al següent  
  estat.  
local  
  r: RESPOSTA; ok: BOOLEAN  
fer  
  repetir  
    visualitzar(e)  
    llegir(e, → r)  
    ok := correcte(e, r)  
    si no ok llavors missatge(e, r) final  
  fins que ok final  
  processar(e, r)  
  op := seguent_opcio(r)  
final
```

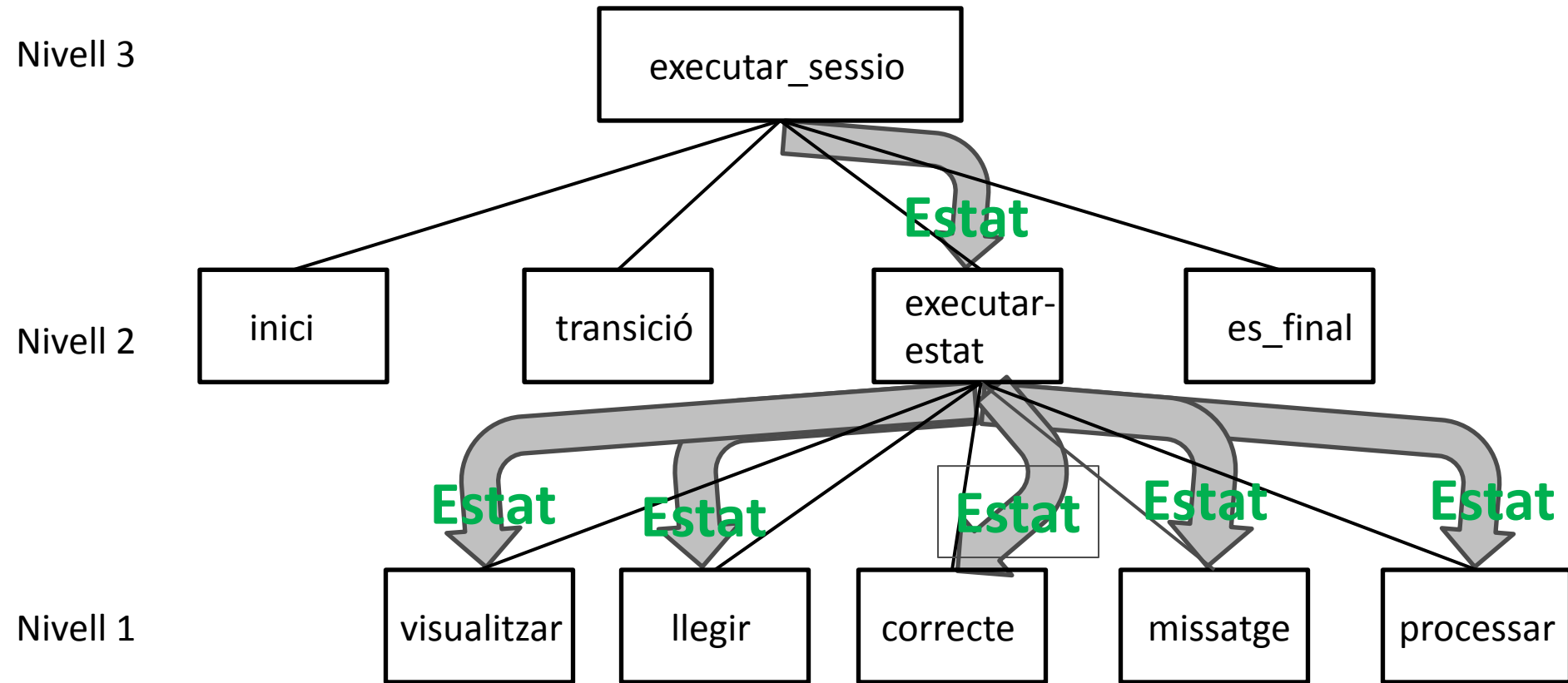
# Exemple: Inconvenients de la solució

- Genèric vs. específic
- Considerem les signatures de les rutines:

executar\_estat(in e: ESTADO; out e: ELECCIÓ)  
visualitzar (in e: ESTADO)  
llegir (in e: ESTADO; out e: RESPOSTA)  
correcte (in e: ESTADO; r: RESPOSTA): BOOLEAN  
missatge (in e: ESTADO; r: RESPOSTA)  
proces (in e: ESTADO; r: RESPOSTA)

Intervenció de  
estat

# Exemple: El flux de dades



# Exemple: Rutines

```
inspeccionar
    e
quan Inicial llavors
    ...
quan Consulta_sobre_vols llavors
    ...
...
final
```

# Exemple:

## Arquitectura orientada a objectes

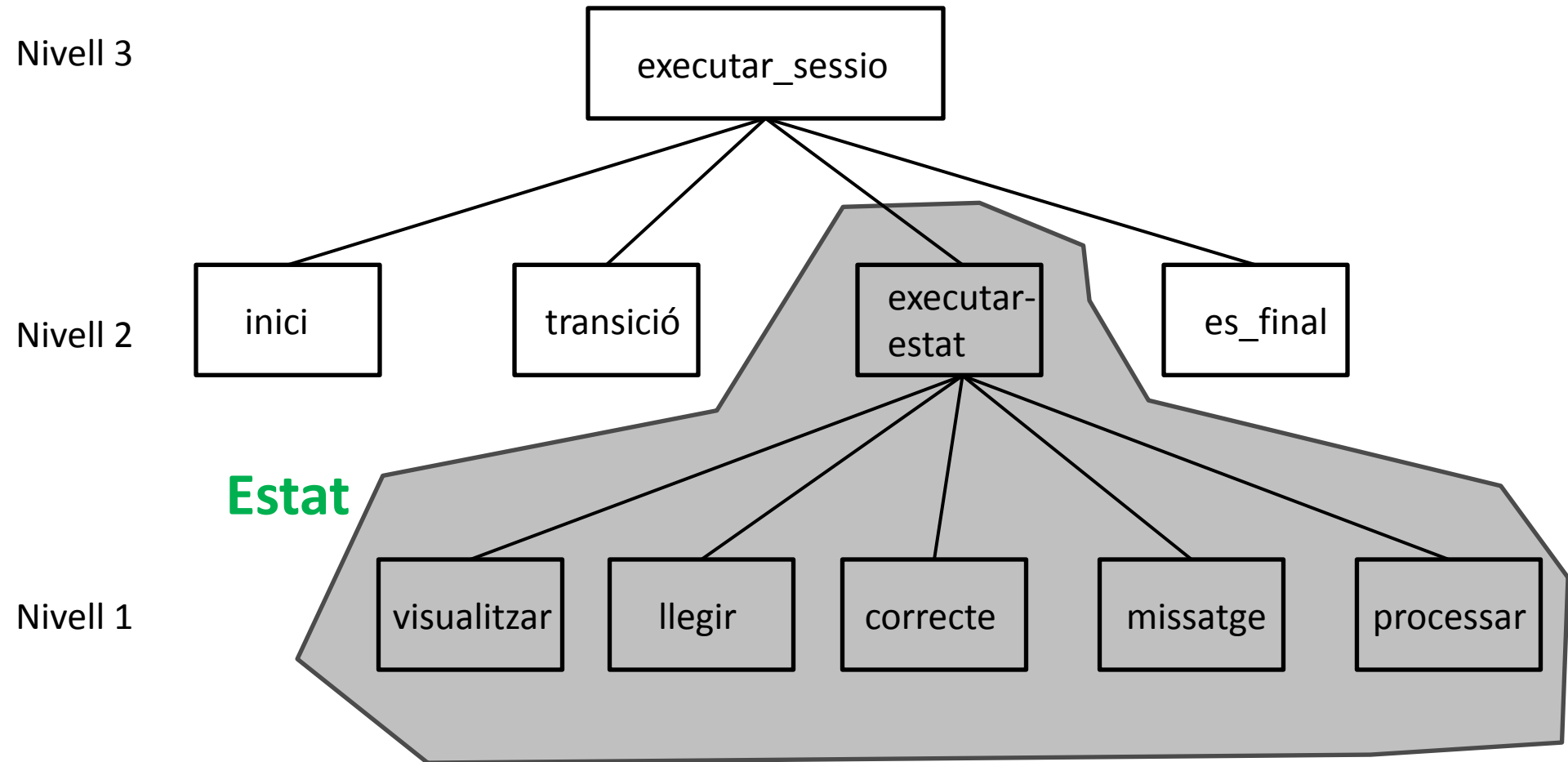
Llei d'inversió:

“Si les rutines intercanvien massa dades, posar les rutines en les dades.”



# Exemple:

## Característiques d'estat



# Exemple: Classe

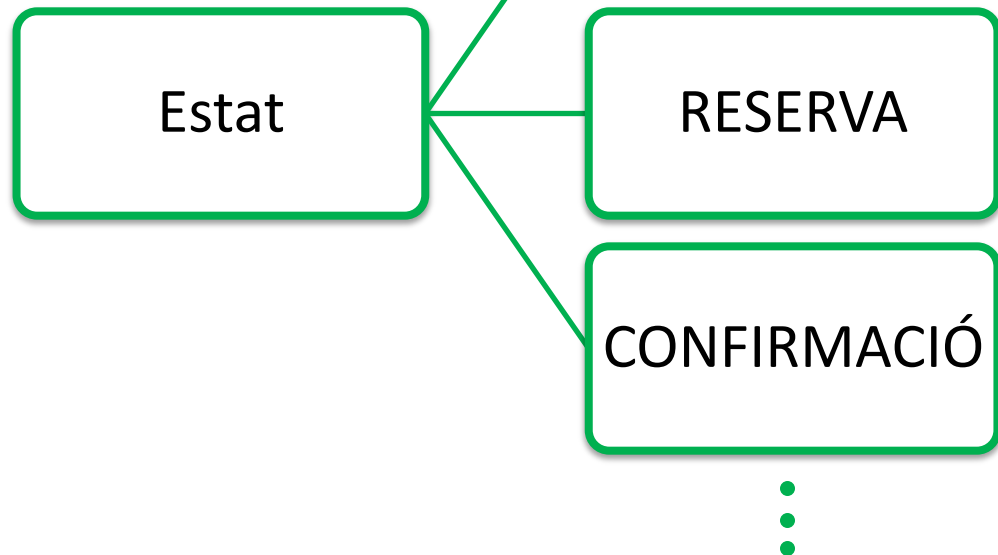
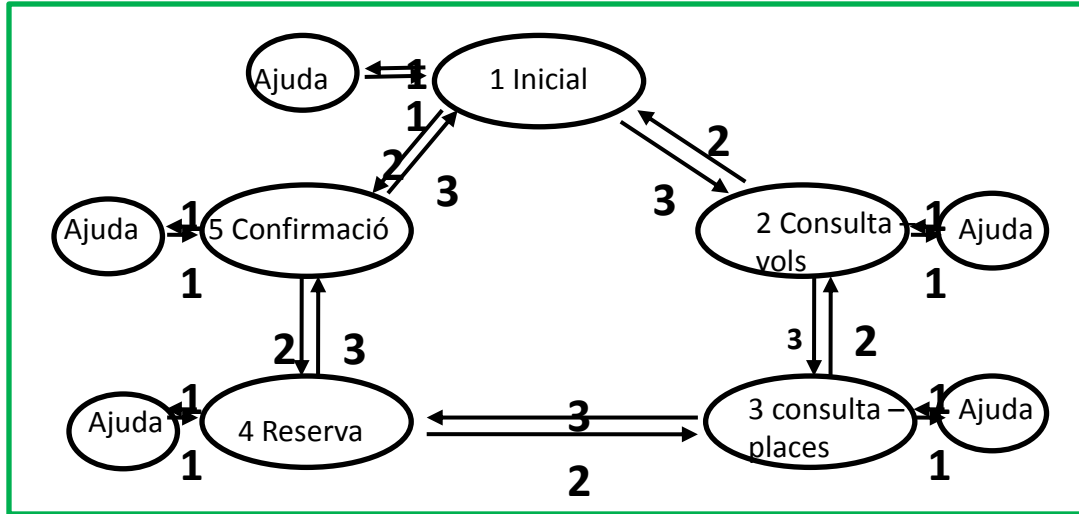
```
classe ESTADO caracteristica  
  entrada: RESPOSTA  
  opció: INTEGER  
  executar es fer ... final  
  visualitzar es ...  
  llegir es ...  
  correcte: BOOLEAN es ...  
  missatge es ...  
  processar es ...  
final
```

atributs

Rutina igual per a tots els estats

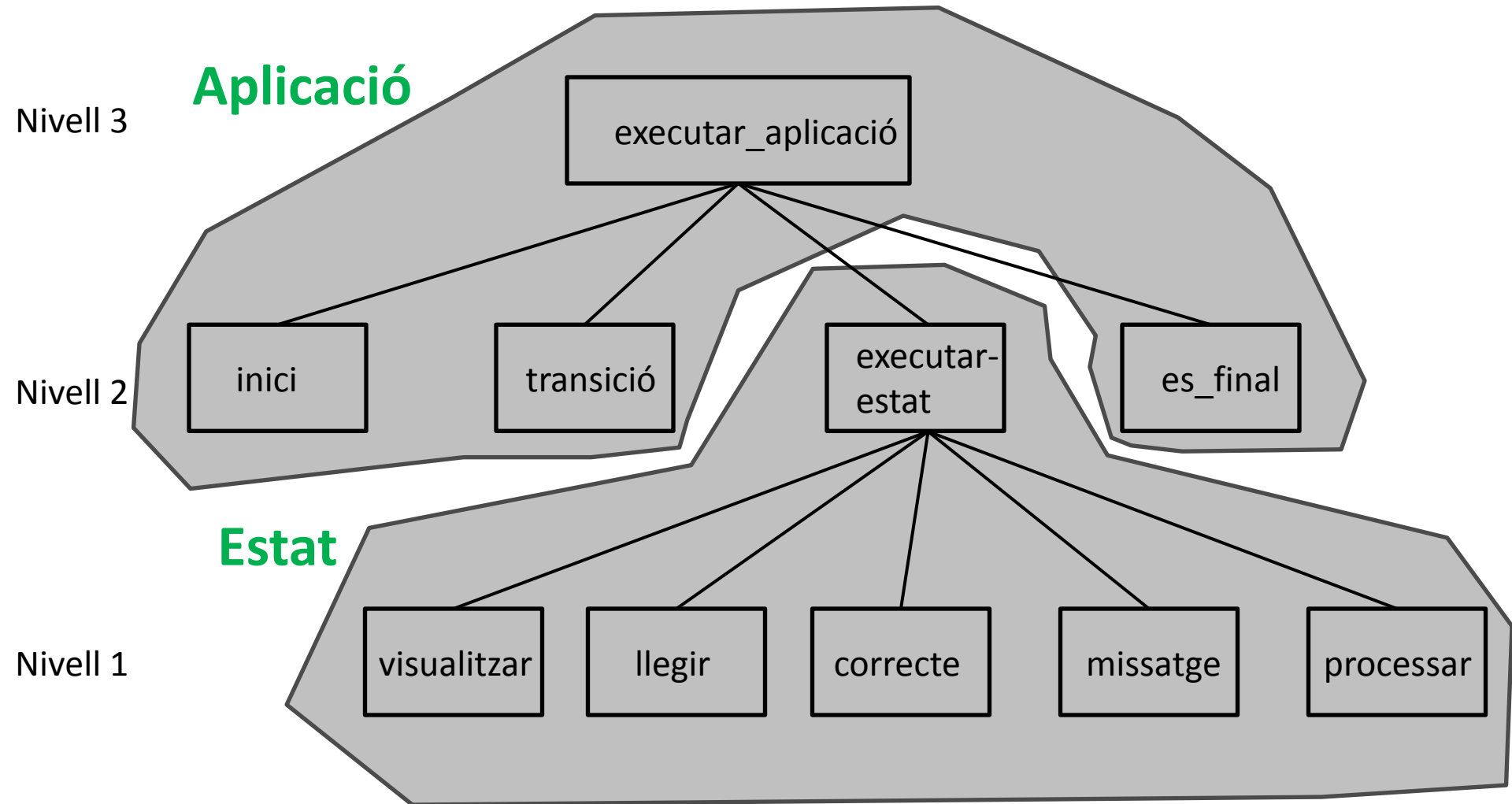
Rutines específiques de cada estat

# Exemple: Herència



# Exemple:

## Característiques d'ESTAT i APLICACIÓ



# Resum

- El disseny és el procés sobre el qual s'assenta la qualitat del software
- Per qualitat s'entén l'adequació del software als requisits exigits
- El disseny modular efectiu redueix la complexitat, facilita els canvis i produeix com a resultat una implementació més senzilla

# Referències

- Font utilitzada per l'Exemple:  
Capítol 20 del llibre de Bertrand Meyer, **“Construcción de software orientado a objetos”**, Prentice Hall, 1998.