

GRAU D'ENGINYERIA INFORMÀTICA

PROGRAMACIÓ II

Bloc 1:

Mòdul i abstracció de dades (2)

Laura Igual

Departament de Matemàtica Aplicada i Anàlisi

Facultat de Matemàtiques

Universitat de Barcelona

TIPUS ABSTRACTE DE DADES (TADS)

Tipus Abstracte de Dades (TADs)

- Estableixen les bases teòriques del mètode OO
- Criteris per obtenir descripcions apropiades dels objectes
 - Precises i no ambigües – Completes
 - No redundants
- Perill de l'especificació excessiva

Tipus Abstracte de Dades (TADs)

Definició:

- “Un TAD és una ens tancat i autosuficient, que no requereix d'un context específic per a poder ser utilitzat en un programa, la qual cosa garanteix portabilitat i reutilització del software i minimitza els efectes que puguin produir un canvi al interior del TAD”
 - Encapsulació de dades: ocultació de la representació del tipus de dades respecte de las aplicacions.

EXAMPLE:

Descomposició funcional descendent i
Descomposició orientada a objectes

Exemple 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,
 - Confirmació.

Exemple

- Sistema interactiu multi-panel
 - Patró general:
 - *Panell* (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:

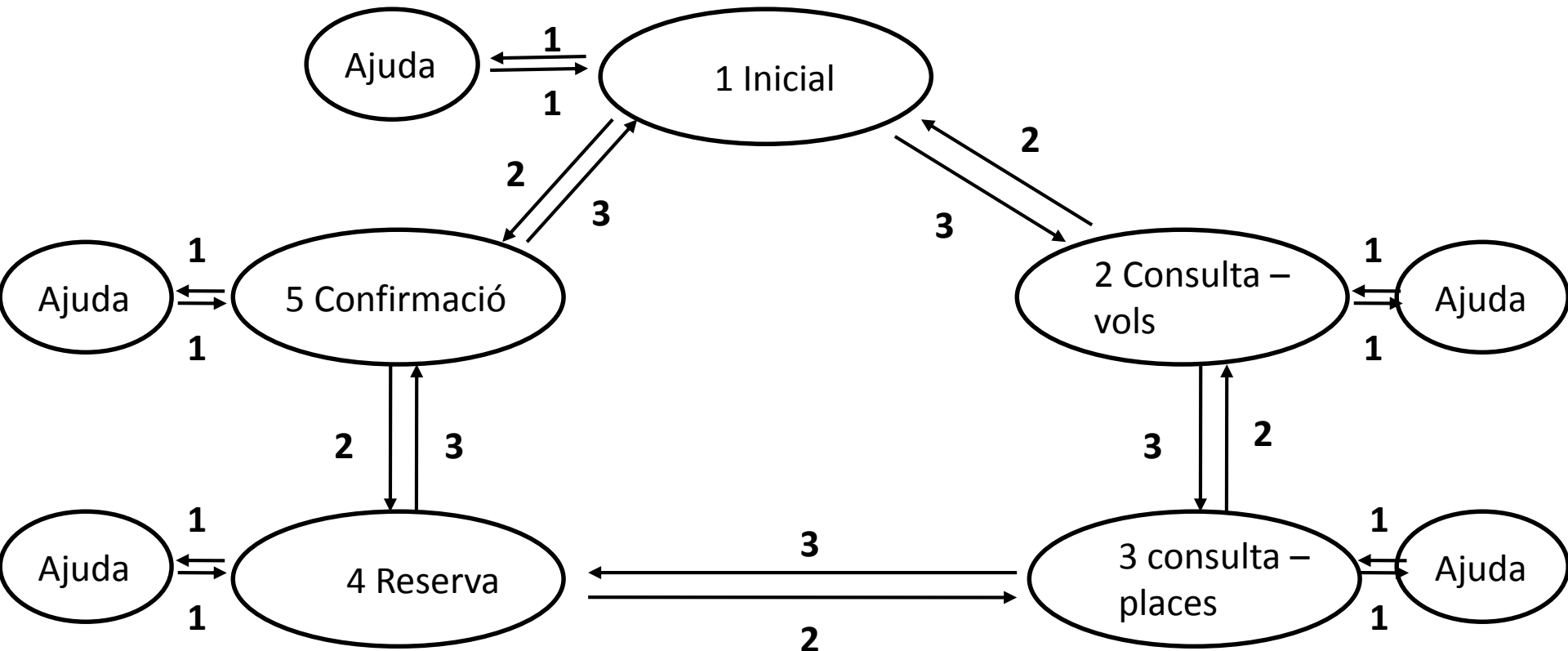
0 – Sortida

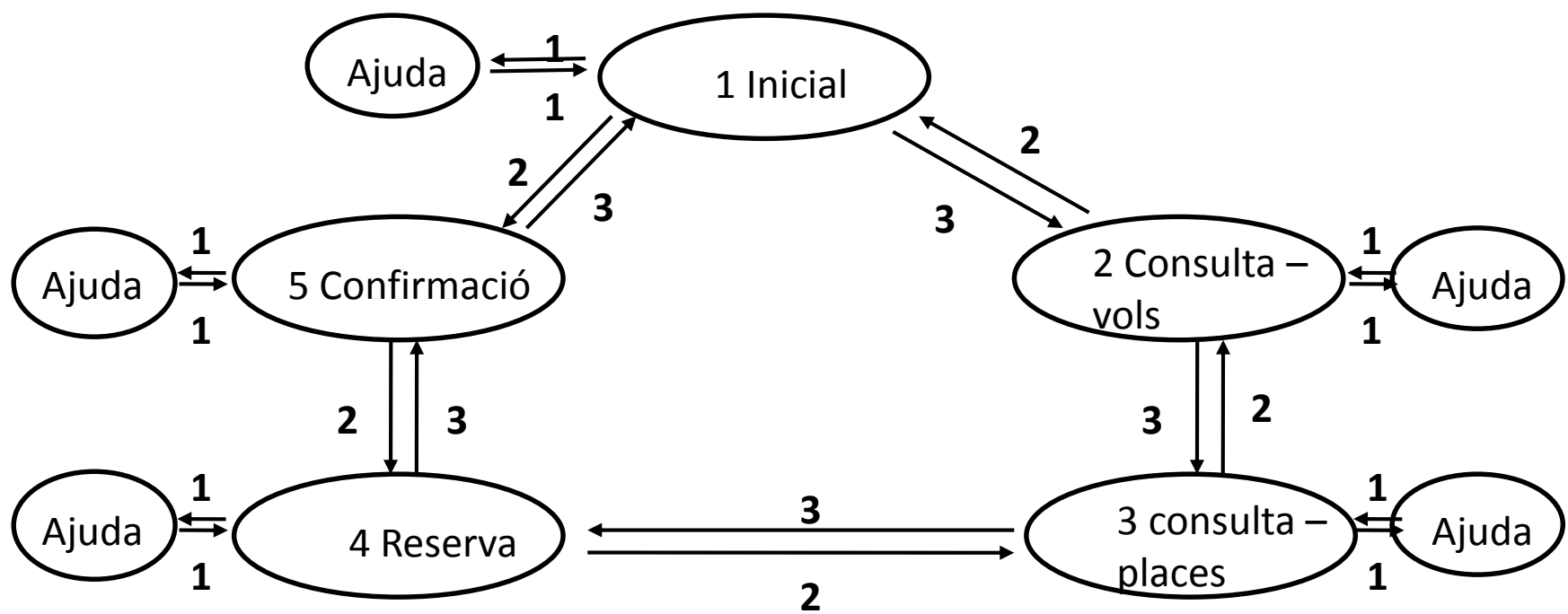
1 – Ajuda

2 – Següent petició

3 – Reserva plaça

Exemple: Un diagrama de transició





Taula de transició:

Panell / 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

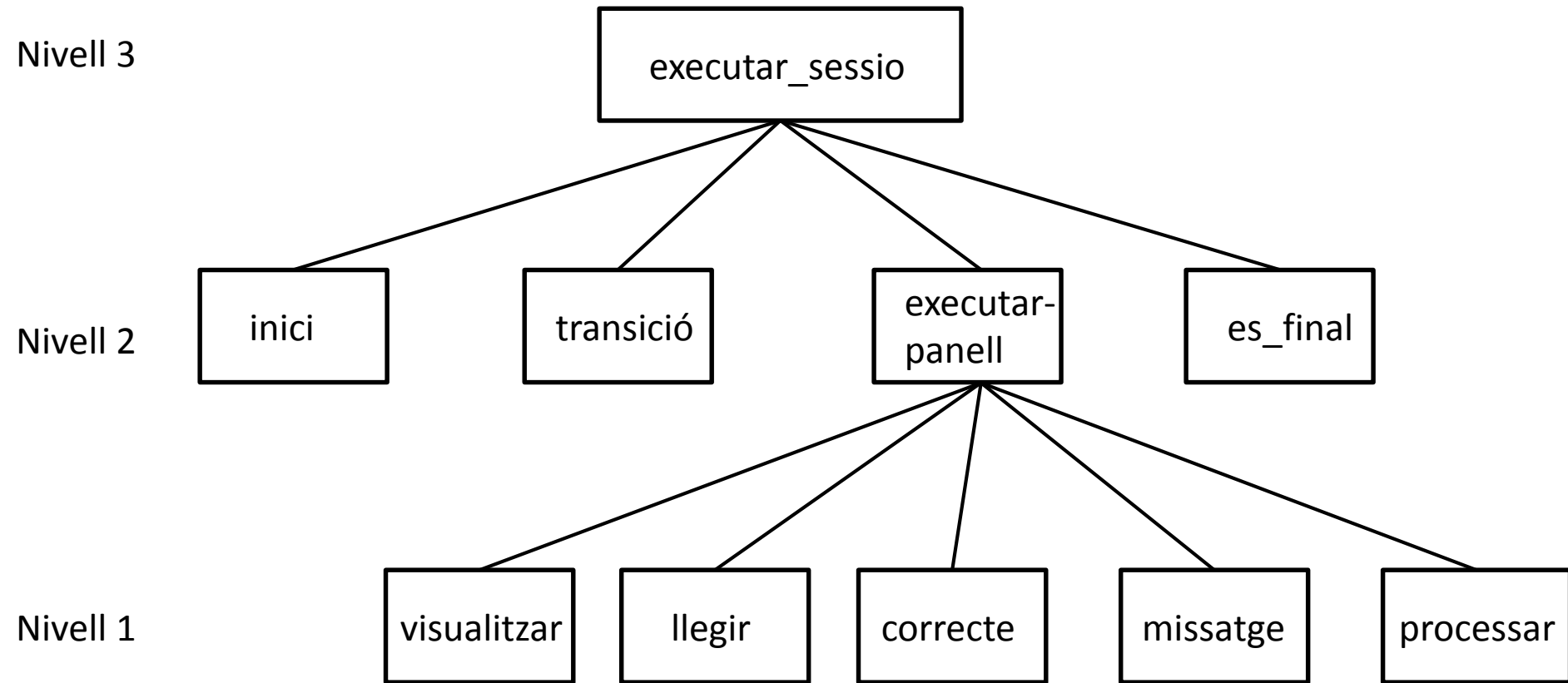
Exemple: Primer intent simple

- B_{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** B_{Consulta}
 C_1 : **goto** B_{Ajuda}
 C_2 : **goto** B_{Reserva}

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
    panell, seguent: INTEGER
  fer
    panell := inici
    repetir
      executar_panell(panell, → seguent)
      -- La rutina executar_panell actualitza
      el valor de següent, a més d'executar
      les accions associades al panell.
      panell := transicio(panell, seguent)
    fins es_final(panell) final
final
```

Exemple: Rutines

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

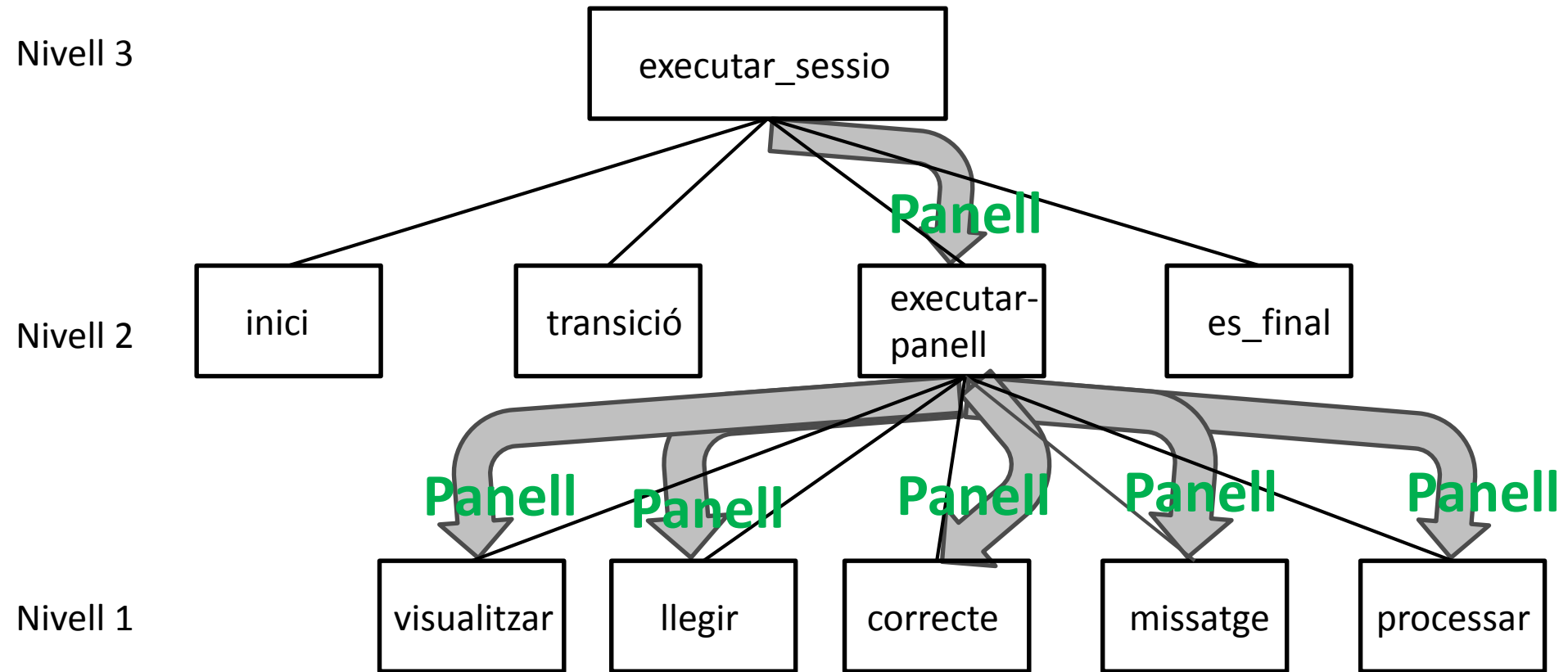
Exemple: Inconvenients de la solució

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

```
executar_panell (in p: PANELL; out p: ELECCIÓ)  
visualitzar    (in p: PANELL)  
llegir         (in p: PANELL; out r: RESPOSTA)  
correcte      (in p: PANELL; r: RESPOSTA) : BOOLEAN  
missatge      (in p: PANELL; r: RESPOSTA)  
proces        (in p: PANELL; r: RESPOSTA)
```

Intervenció de
PANELL

Exemple: El flux de dades



Exemple: Rutines

Totes les rutines del nivell 1 han de realitzar accions diferents depenent del panell p. Per tant, efectuaran una discriminació de la forma:

```
inspeccionar
```

```
    p
```

```
quan Inicial llavors
```

```
    ...
```

```
quan Consulta_sobre_vols llavors
```

```
    ...
```

```
    ...
```

```
final
```

Exemple:

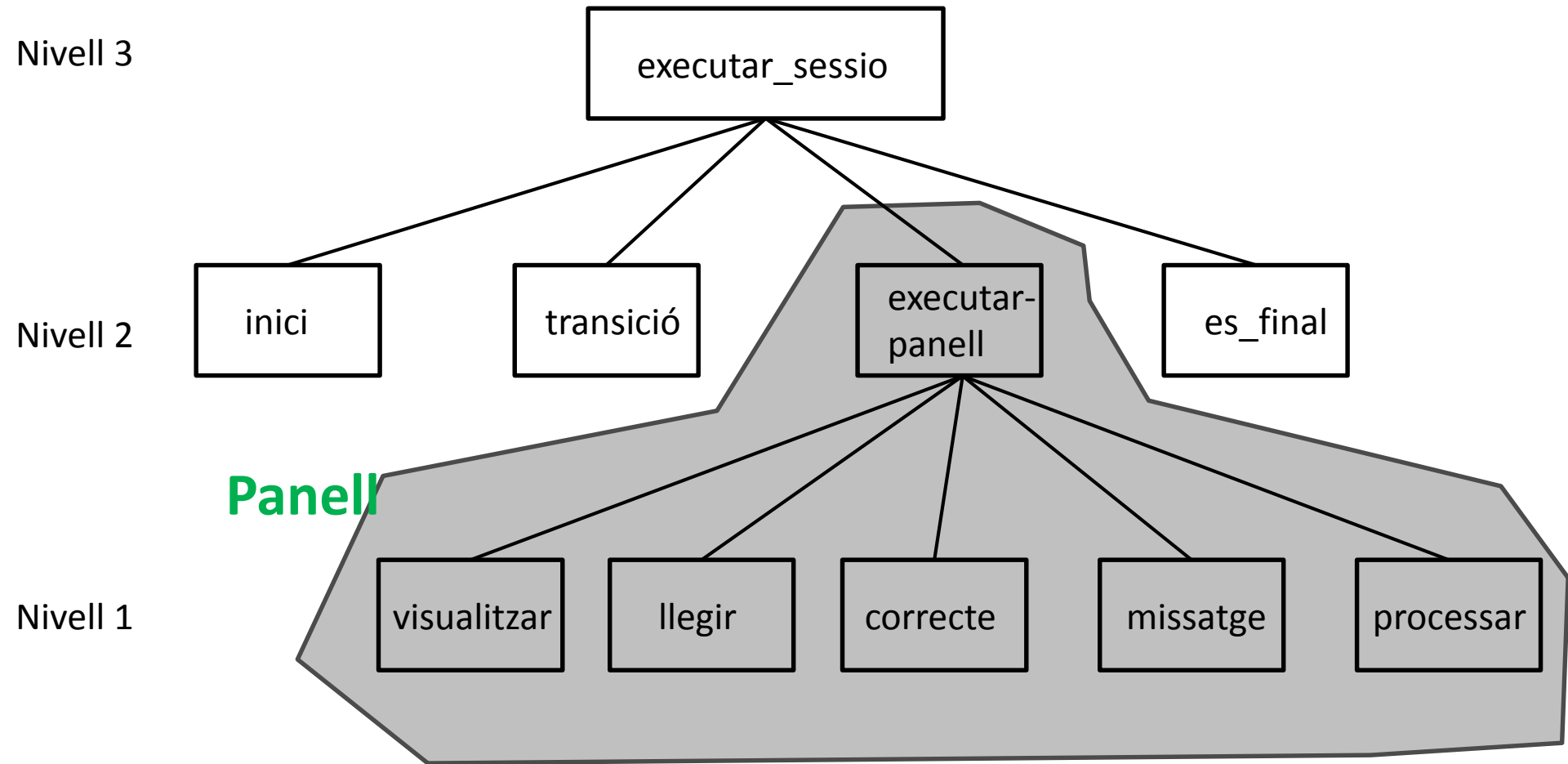
Arquitectura orientada a objectes

Llei d'inversió:

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

Exemple:

Característiques d'estat



Exemple: Classe

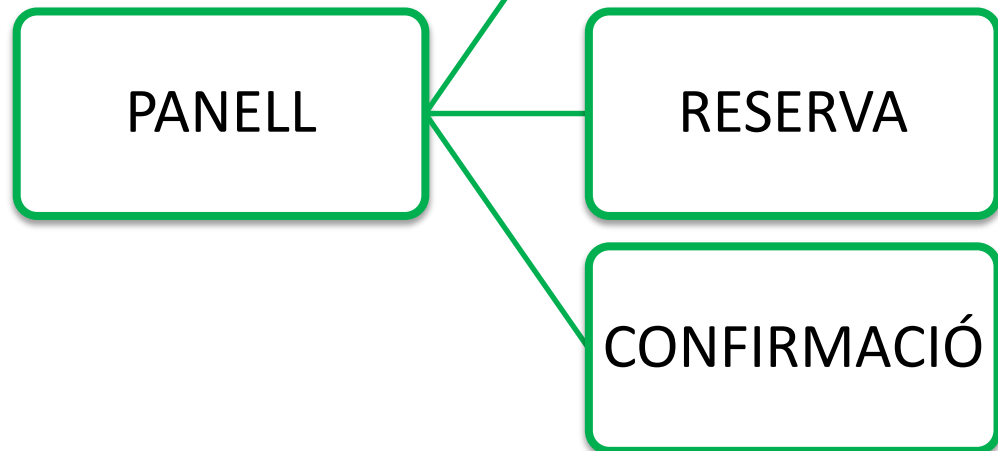
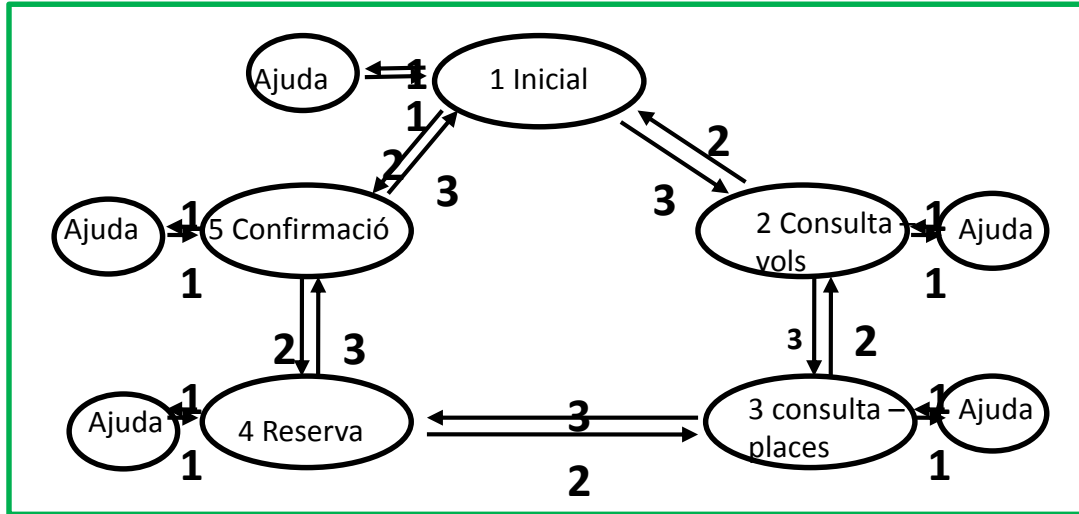
```
classe PANELL caracteristica  
  entrada: RESPOSTA  
  opció: INTEGER  
  executar es fer ... final  
  visualitzar es ...  
  llegir es ...  
  correcte: BOOLEAN es ...  
  missatge es ...  
  processar es ...  
final
```

atributs

Rutina igual per a tots els panells

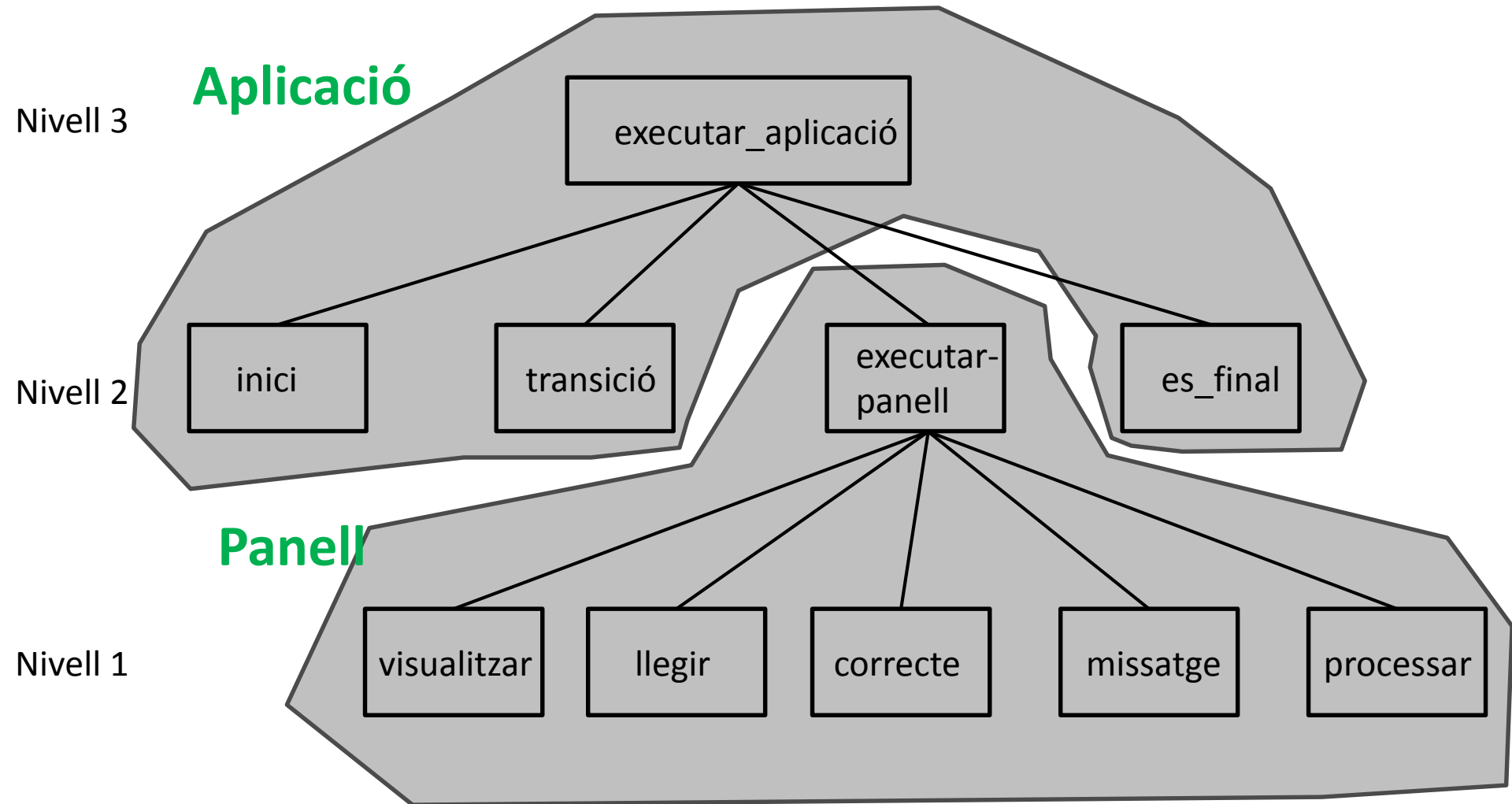
Rutines específiques de cada panell

Exemple: Herència



Exemple:

Característiques de PANELL 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 existents
- 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.