

# Arquitectura

## Interfícies d'usuari

Aleix Llusà Serra

Enginyeria de Sistemes TIC  
Universitat Politècnica de Catalunya

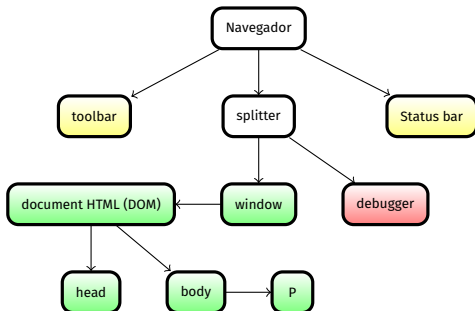
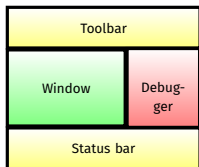
<http://epsem.upc.edu>

14 d'abril de 2021

Arquitectura en la implementació d'interfícies d'usuari gràfiques.  
Patrons de disseny (els més remarcables):

- Arbre de vistes
- Model-vista-controlador
- Observador
- Reactiu

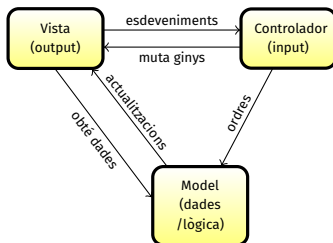
- Vista: objecte en una àrea de la pantalla, típicament una caixa rectangular
- Components d'una vista: ginys, widgets, controladors, elements, nodes...



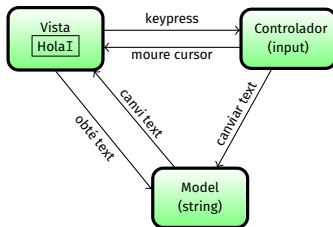
- Output: La interfície canvia la sortida mutant l'arbre. **Redraw algorithm.**
- Input: L'arbre controla com es processa l'entrada segons el focus a la pantalla.
- Layout: Com es posicionen les vistes a la pantalla.

# Patró model-vista-controlador (MVC)

- Separació de responsabilitats
- Desacoblament
  - Canviar cada part independentment
  - Reutilitzar les parts
  - Múltiples vistes del mateix model



# Exemple: caixa de text (textbox)

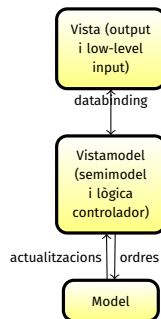


## Inconvenients MVC:

- Dificultat de separar la vista i el controlador
- La vista ha de mostrar affordances pels controls, feedback en prémer controls
- Alguns estats de la interfície no tenen un objecte clar en el model  
P.ex. la selecció de text (highlight, cursor, keypress)

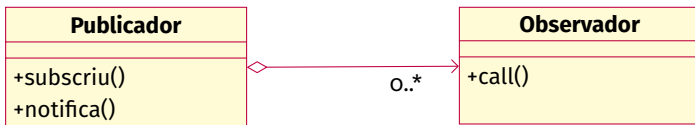
## Algunes alternatives:

- Model-view (MV): vista i controlador acoblats
- Model-vista-vistamodel (MVVM)
- Resources i views (**Pyramid**)
- Model-template-view (**Django**)



## Gang of Four

- També anomenats listener, event handler, publish/subscriber...
- Gestionen les accions d'input de l'arbre de vistes





- No reactiu  $a:=b+c$  (a només canvia quan s'executa la sentència)
- Llenguatges de descripció de hardware:  $b \leq$  NOT a;
- Interfícies d'usuari, p.ex. Svelte:

```
<script>  
let a = 1;  
let b = 2;  
</script>
```

```
<input type="number" bind:value={a}>  
<input type="number" bind:value={b}>
```

```
<p>{a} + {b} = {a + b}</p>
```