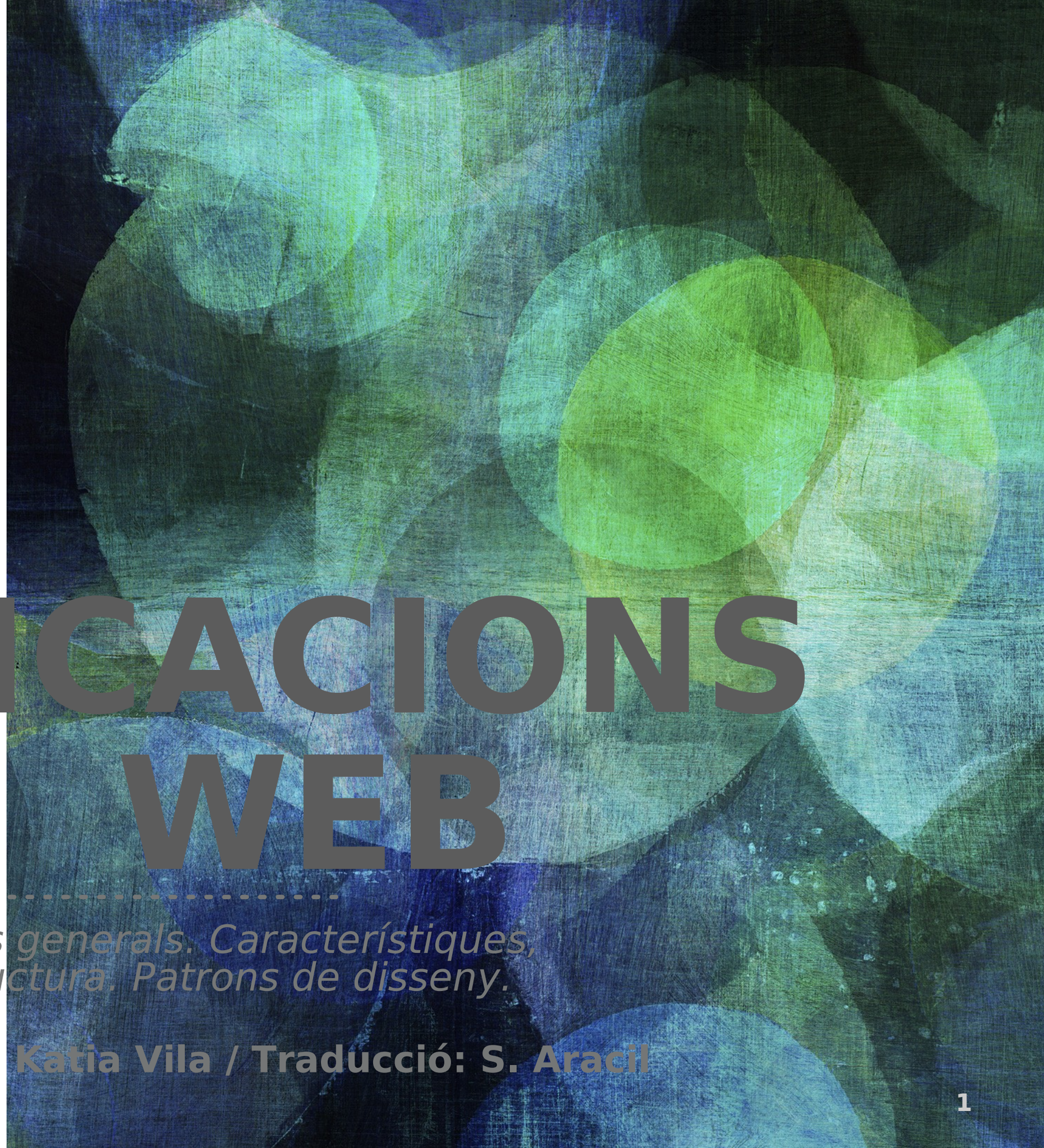


APLICACIONS WEB

***Unitat 1**. Conceptes generals. Característiques, funcionament i estructura. Patrons de disseny.*

Katia Vila / Traducció: S. Aracil

TIPUS D'APLICACIONS

Apps d'escriptori/desktop

Avantatges:

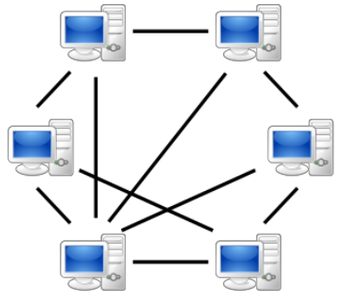
- Execució de forma local: major velocitat de processament.
- Major robustesa y estabilitat que les apps Web.
- Major rendiment: temps de resposta més ràpid.
- Majors garanties quant a seguretat.

Desavantatges:

- Accés limitat al dispositiu on estan instal·lades.
- Dependents de SO (en la seua majoria) i les capacitats del dispositiu (memòria, vídeo, etc.)
- Requereixen instal·lació/actualització personalitzada.
- Necessiten requeriments especials de programari i llibreries.

TIPUS D'APLICACIONS

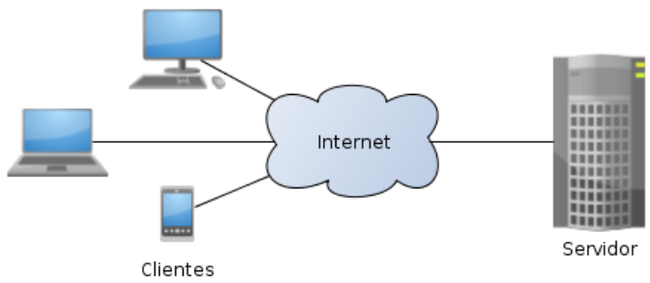
Apps amb connexió



P2P

- Xarxa d'ordinadors. Nodes que es comporten com a iguals entre si (actuen com a servidors o clients de cara a la resta nodes de la xarx

Client-Servidor



- Conjunt d'ordinadors (clients) que es connecten a un central (servidor), qui proporciona els serveis i la informació sol·licitada.

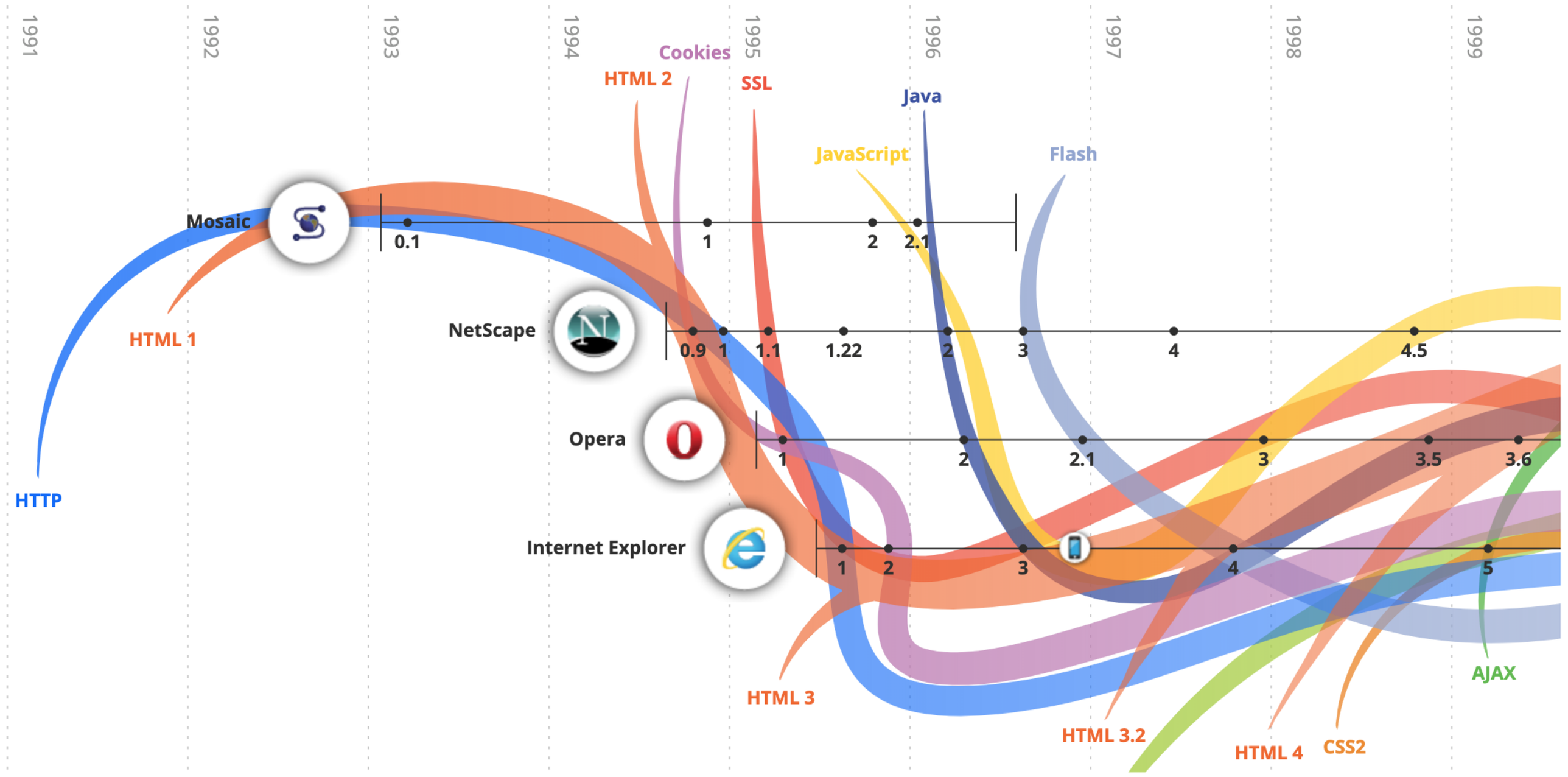
Aplicacions web

OBJECTIUS U1

- Identificar els avantatges i inconvenients de les aplicacion web enfront d'altres tipus d'aplicacions (apps d'escriptori).
- Reconéixer l'arquitectura de les aplicacions web.
- Identificar els elements de l'esquema de funcionament d'u aplicació web.
- Enunciar els protocols de comunicació emprats en les aplicacions web.
- Especificar els patrons de disseny programari emprat per a desenvolupament d'aplicacions web.

ARQUITECTURA D'UNA APLICACIÓ WEB

- Què és "**el Web**"? (evolució)



ARQUITECTURA D'UNA APLICACIÓ W

Elements d'una aplicació web

Client:

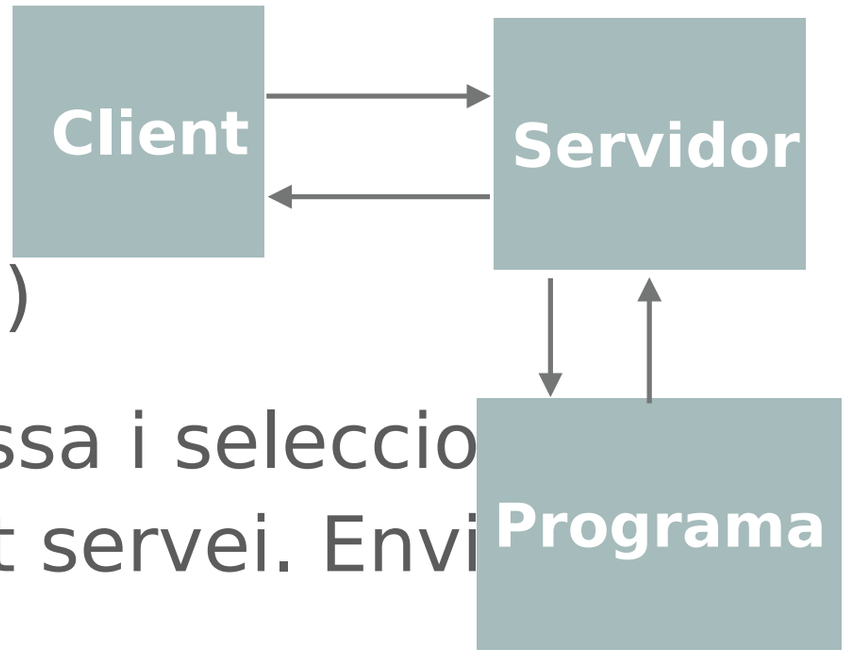
- On està l'usuari
- Utilitza un navegador web per a accedir a l'aplicació.

Servidor:

- On està situada l'aplicació.
- S'encarrega d'atendre les peticions dels clients i proporcionar-los la informació/servei sol·licitats.

ARQUITECTURA D’UNA APLICACIÓ W

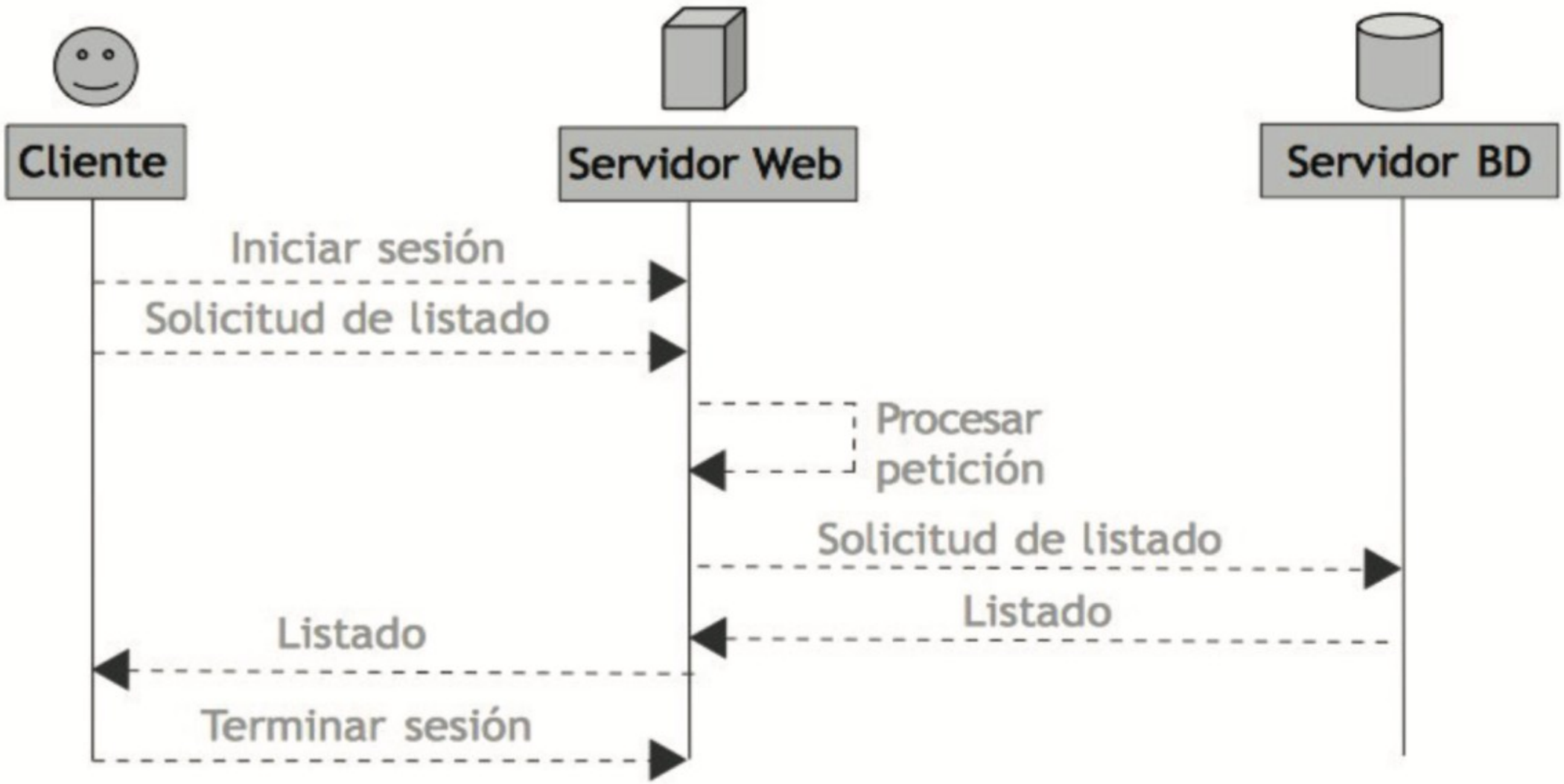
Funcionament d'una aplicació web



1. C: inicia sessió en el servidor.
2. C: sol·licitud al servidor (recurs/servici)
3. S: rep la sol·licitud del client, la processa i seleccio el programa que ha de donar-li aquest servei. Envi petició al programa.
4. P: processa la petició del servidor, prepara la resposta i li l’entrega al servidor.
5. S: envia la resposta al client.
6. C: acaba la sessió en el servidor o torna al pas 2 per a realitzar una nova petició.

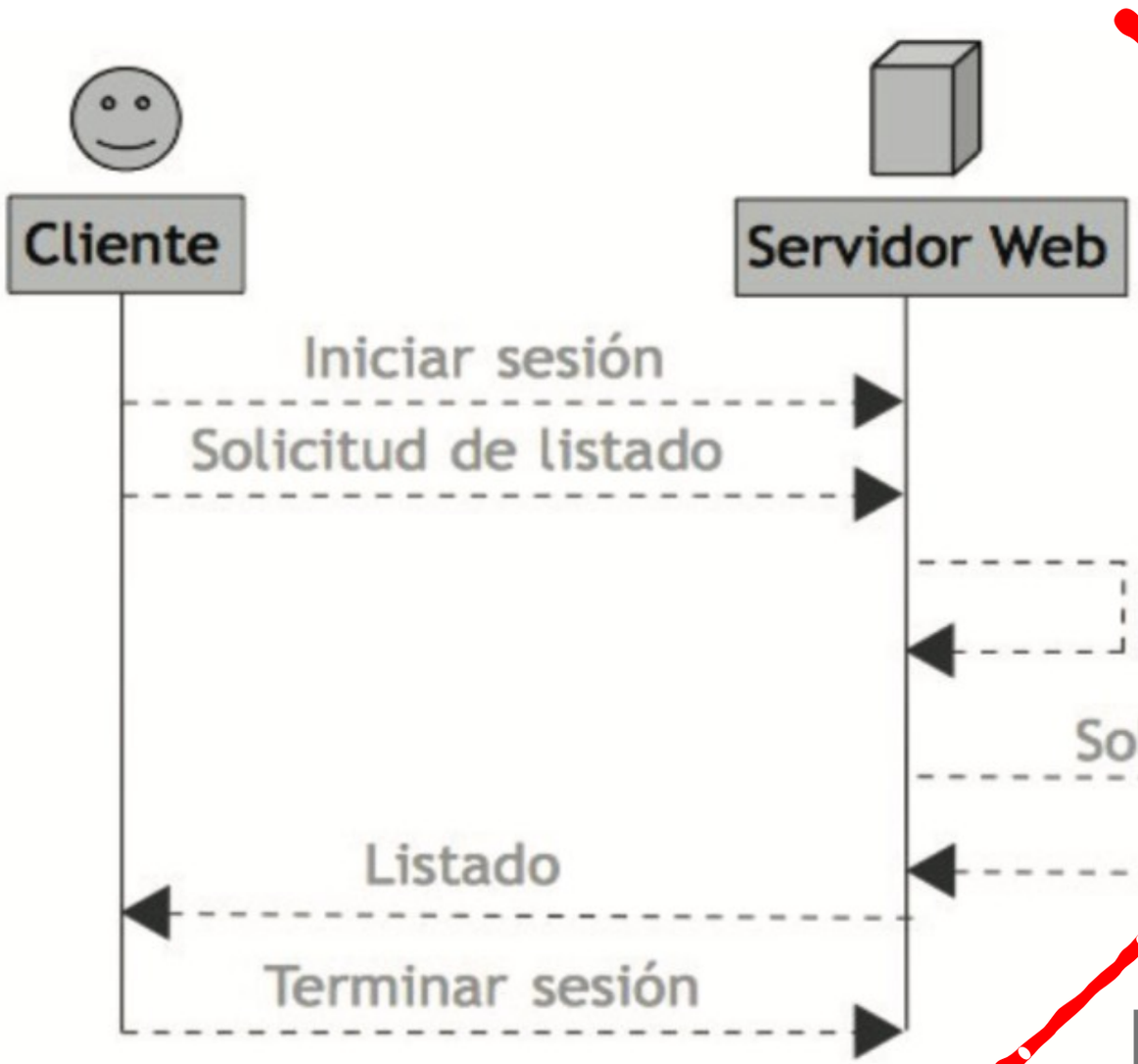
ARQUITECTURA MULTICAPA / MULTINIV

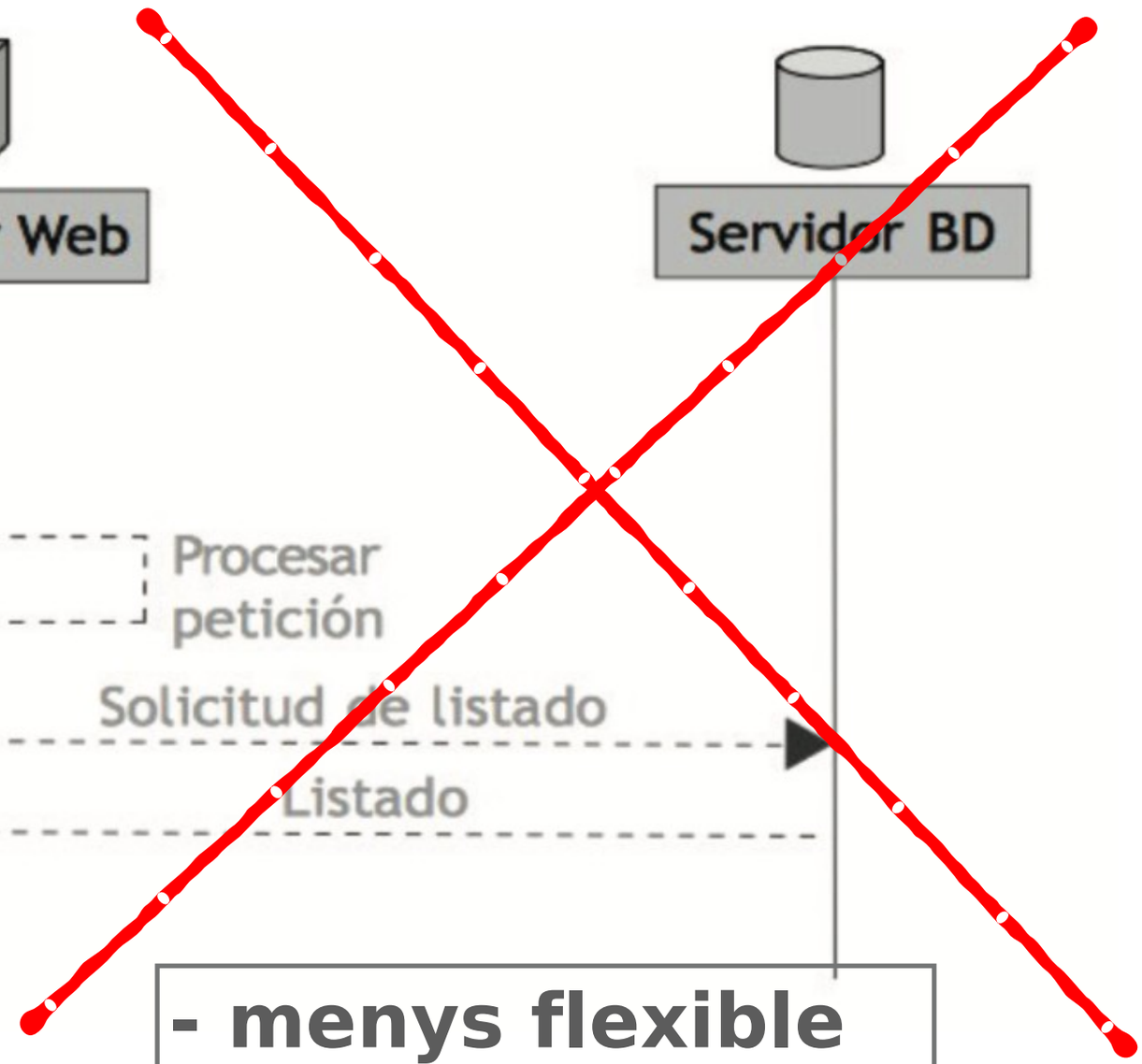
- Arquitectura de tres nivells



ARQUITECTURA MULTICAPA / MULTINIV

- Arquitectura de dos nivells



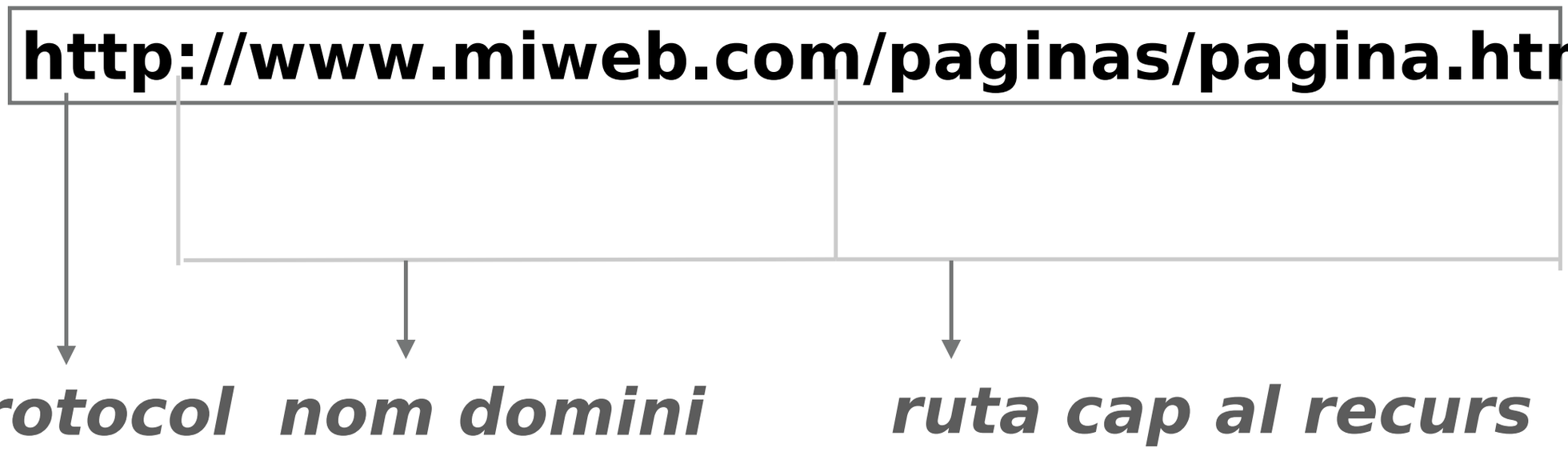


- menys flexible
- menys segura
- pitjor rendiment

URL

De quina forma sol·licita el client els recursos al servidor?

URL: *Uniform Resource Locator.* Manera d'identificar y locali
cada recurs d'una web.



Funcionament: el navegador obté la URL que ha escrit l'usuari, transforma el nom de domini en una adreça IP (segons el servidor DNS assignat) i envia al servidor indicat la petició.

PROTOCOLS

Protocols de comunicació entre client i servidor

- Què són els protocols?

Els més usats en aplicacions web

- **HTTP**: *HyperText Transfer Protocol* (1990). Transferència d'arxius (principalment HTML). Segueix l'esquema petició-resposta entre client-servidor.
- **HTTPS** (+*Secure*): versió segura d'HTTP. Encriptació de les dades. Ús: tractament d'informació confidencial, sistemes bancaris i de pagament com PayPal.
- **SMTP**(*Simple Mail Transfer Protocol*) o **POP3**(*Post Office Protocol*)/I (*Internet Message Access Protocol*):
  enviament i recepció de correu electrònic.
- **FTP** (*File Transfer Protocol*): transferència d'arxius.

LLENGUATGES

Llenguatges Clients: llenguatges a l'entorn del client

- **HTML** (*HyperText Markup Language*): per a crear i visualitzar les p
  web.
- **Javascript**: per a facilitar la interacció entre l'usuari i el navegado

Llenguatges servidor: llenguatges a l'entorn del servidor.

- **ASP .NET**: per a servidors de Microsoft i entorns Windows.
- **JSP** (*JavaServer Pages*): és similar a PHP, però usa el llenguatge de
  programació Java. Requereix un servidor web compatible amb
  contenidors servlet com a Apatxe Tomcat o Jetty.
- **PHP** (*HypertextPreprocessor*): programari lliure (llicència GNU). Pot s
  desplegat en la majoria dels servidors web i en tots els SO i plata
  sense cap cost.

PATRONS DE DISSENY DE PROGRAM

- **Patró de disseny**: conjunt de pautes a seguir, elements desenvolupar, jerarquies i ordre, que doten a una aplicació d'una estructura preestablida (garantia de funcionament).
- **Objectius**:
  - estandarditzar la forma en què es desenvolupen les app
  - elaborar elements o components reutilitzables entre diverses apps.

PATRÓ MVC

MVC: Model-Vista-Controlador

- Patró per excel·lència en l'actualitat (apps web o desktop)
- És molt concís i ben estructurat.
- Avantatges
- Permet aïllar el codi dels tres elements involucrats (model, vista i controlador)
- Treball de desenvolupament és més modular i divisible:
  - Desenvolupament de vistes: dissenyador web amb desconeix De programació en el servidor. **FRONT-END**
  - Desenvolupament dels controladors: programador de llenguat servidor (PHP, etc.) que desconeguen HTML. **BACK-END**

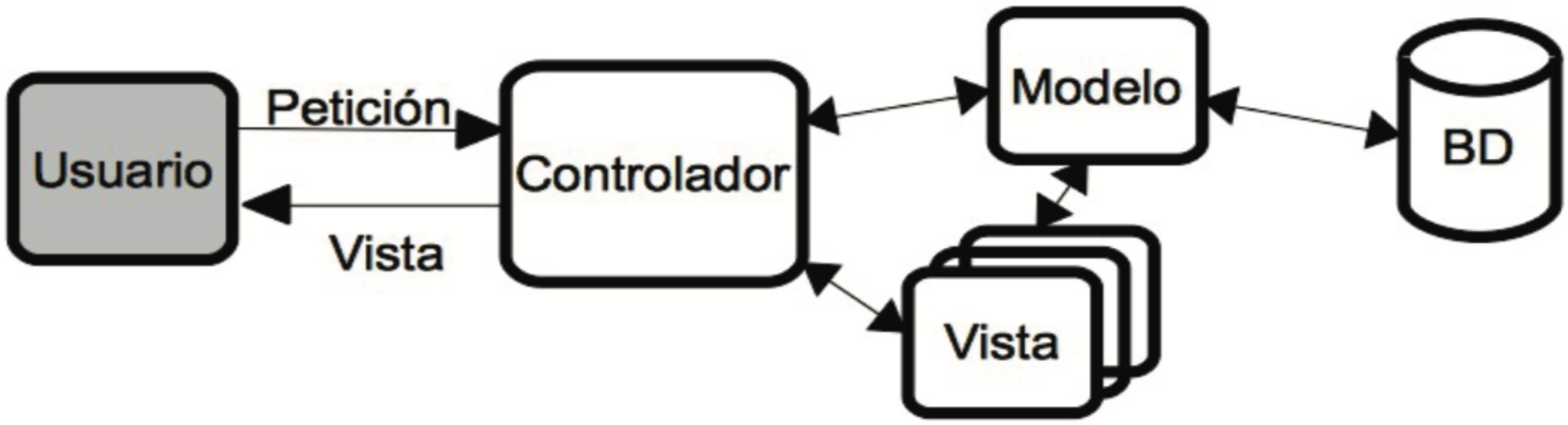
PATRÓ MVC

Components principals del patró MVC:

- **Model**: conjunt de totes les dades o informació que maneja la app **Què és**: variables u objectes extrets de la BD o sistema d'emmagatzematge. **Composició**: codi que conté les instruccions per a connectar amb la BD, recuperar la informació i emmagatzem variables.
- **Vista**: intermediari entre la app i l'*user*. **Què és**: les vistes per al *user*. **Composició**: pàgines, formularis, etc., que la app li mostra a l'*user* per a la interacció amb ell.
- **Controlador**(es): coordina el funcionament general de la app. **Composició**: fragments de codi per a la coord. **Funcionament**: reben i identifiquen les peticions dels users, accedeixen al model p actualitzar i recuperar dades, decideixen quina vista mostrar-li a l' continuació de l'acció que s'acaba de realitzar.

PATRÓ MVC

Funcionament del patró MVC



ALTRES PATRONS DE DISSENY DE PROGRAM

- Alternatives al patró MVC: **MVW** (*Model-View-Whatever*)
- **Similitud**: model i vista. **Diferència**: controlador (substitu per altres elements)

Activitat en classe:

- Analitzar els elements fonamentals dels altres patrons MVW i les seues principals diferències amb respecte a MVC
- Elaborar un diagrama que reflectisca el seu funcionament (
  - **MVVM** (Model-Vista-Vista-Model)
  - **MOVE** (Model-Operacions-Vista-Esdeveniments)
  - **MVP** (Model-Vista-Presentadors)

APLICACIONS WEB VS ESCRIPTORI

Activitat en classe:

- Exposar en classe, després d'una lectura ràpida, els avant
  i desavantatges de les apps web en comparació amb les a
  d'escriptori