



Desenvolupament Web en Entorn Servidor

UD 1 - Arquitectures i eines

2n CFGS Desenvolupament Aplicacions Web





BLOC 2 - Framework del servidor

UD1 - Architectures i eines de programació

UD2 - Databases

UD3 - Creació d'API

UD4 - Desenvolupament del Back Office



Objectius

Després d'acabar aquesta unitat, l'alumne hauria de ser capaç de:

- De conèixer les diferents arquitectures i eines que intervenen en un entorn Servidor **d'una manera fluïda**



UD1 – Architectures i eines de programació

1. [Introducció](#)
2. [Capes de l'estructura Web](#)
3. [Programació en entorn client i en entorn servidor](#)
4. [El paradigma Client-Servidor](#)
5. [Front-end vs Back-end](#)
6. Les eines
 - [El llenguatge PHP](#)
 - [La base de dades MySQL](#)
 - [El servidor web Apache](#)
7. [Frameworks](#)
 - Laravel



UD1: ... - Introducció

Les **arquitectures web** defineixen la manera com les pàgines d'un lloc web estan estructurades i enllaçades entre si.

Les aplicacions web es basen en model **client-servidor**, que utilitza un navegador com a client i que es connecta interactivament a un servidor.

Un lloc web simplement ens ofereix **pàgines estàtiques**.

Una aplicació web presenta **contingut dinàmic** construït a partir d'una petició amb paràmetres, del comportament de l'usuari o de consideracions de seguretat,



UD1: ... - Capes de l'estructura web

1. Capa de presentació (Client)

Aquesta capa es la que veu l'usuari. L'hi presenta una interfície gràfica del recurs demanat i serveix per recollir la seva interacció. Està situada al client. La programació en aquesta capa es centra en el format de la informació enviada pel servidor i la captura de les accions fetes pel client.

2. Capa de la lògica de negoci (Servidor)

Es la capa que coneix i gestiona les funcionalitats que esperam del sistema o aplicació web. Habitualment es on es reben les peticions del usuari i des d'on s'envien les respostes després del processament apropiades de la informació proporcionada pel client.

3. Capa de persistència (B. Dades)

La capa on resideixen les dades i l'encarregada d'accedir a aquestes dades, normalment, està formada per un o mes sistemes gestors de bases de dades



UD1: ... - Capes de l'estructura web

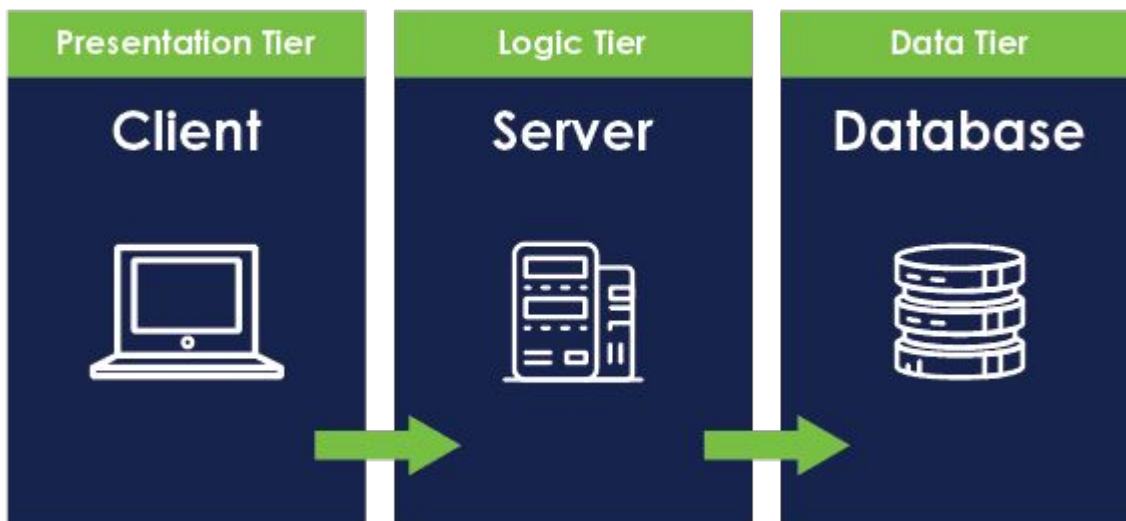
Una aplicació web s'estructura comunament com una aplicació de tres nivells o capes:

- Navegadors web (presentació)
- Motor utilitzant una tecnologia de contingut web dinàmic (lògica de l'aplicació)
- Base de dades (persistència)

Facilita el millor rendiment, flexibilitat i seguretat.



UD1: ... - Capes de l'estructura web





UD1: ... - Programació en entorn client i en entorn servidor

La programació del costat del client s'executa en el navegador, generalment programat en Javascript.

La programació del costat del servidor és una tecnologia on la sol·licitud d'un usuari es correspon a l'execució d'un script directament al servidor web per:

1. Generar pàgines web dinàmiques.
 2. Esperant sol·licituds de programes client per després processar-les, generalment mitjançant arxius en format JSON.
- S'utilitza generalment per proporcionar llocs web interactius que interactuen amb bases de dades o altres magatzems de dades.



UD1: ... – Programació en entorn client i en entorn servidor

Per què la programació del costat del servidor?

- **Es escalable:**

L'arquitectura de 3 capes basada en la web és fàcil de fer créixer simplement afegint més recursos a dita arquitectura.

- **És fàcil personalitzar:**

Sobre la base dels requisits de l'usuari, els drets d'accés, o consultes en magatzems de dades.

- **Clients de baix cost:**

No necessita clients cars: simplement un navegador basat en un llenguatge adient, per exemple Javascript.



UD1: ... - El paradigma Client-Servidor

L'arquitectura client-servidor és un model de disseny de programari on les tasques o serveis es divideixen entre dos tipus d'entitats: el **client**: que sol·licita serveis o recursos i el **servidor**, que les proporciona.

Dit d'una altra manera: el client envia peticions al servidor, que les processa i retorna una resposta.

Aquest model és comú en aplicacions web: per una banda tenim un navegador que actua com a client i per l'altra tenim un servidor web que gestiona les sol·licituds.



UD1: ... - El paradigma Client-Servidor

Avantatges

- La centralització: hi ha un control centralitzat. Fàcil gestió.
- La millora i Escalabilitat: Es poden fer canvis fàcilment amb només actualitzar el servidor. També es poden afegir nous recursos fent els canvis necessaris en el servidor.
- Accessibilitat: Des de diverses plataformes, el servidor es pot accedir de forma remota.
- Seguretat: Es poden definir els drets d'accés i seguretat en el moment de la posada a punt del servidor.
- **Separació de tasques:** El client i el servidor tenen rols diferents i específics. El client és responsable de la interfície d'usuari i de la recollida de dades, mentre que el servidor gestiona les operacions de fons, com l'emmagatzematge de dades o el càlcul.



UD1: ... - El paradigma Client-Servidor

Desavantages

- La congestió a la xarxa: Massa peticions dels clients poden portar a la congestió.
- Si el servidor falla, tot el sistema cau.
- Cost: És molt car d'instal·lar i gestionar aquest tipus de computació (costat del servidor).
- Es necessiten informàtics per mantenir els servidors i altres detalls tècnics de la xarxa.



UD1: ... - Front-end vs Back-end

En un entorn de desenvolupament web, l'arquitectura client-servidor és fonamental per al funcionament de les aplicacions i serveis en línia.

Aquesta arquitectura defineix com els components del sistema interactúen per processar, emmagatzemar i mostrar informació a l'usuari final.



UD1: ... - Front-end vs Back-end

El Funcionament en un entorn web està basat en 2 entorns:

- Client (Front-end) i
- servidor (Back-end).



UD1: ... - Front-end vs Back-end

- **Client (Front-end):**

El client, sovint representat pel navegador web de l'usuari, és responsable de la presentació i la interacció amb l'usuari.

El client envia sol·licituds al servidor mitjançant protocols com HTTP o HTTPS, normalment sol·licitant pàgines HTML, CSS, JavaScript o dades en format JSON o XML.

Els llenguatges i tecnologies utilitzats en el client inclouen HTML per a l'estructura, CSS per al disseny i JavaScript per a la interactivitat i l'actualització dinàmica del contingut.



UD1: ... - Front-end vs Back-end

- **Servidor (Back-end):**

El servidor processa les sol·licituds dels clients, accedeix a la base de dades si cal, i retorna una resposta adequada. Aquesta resposta pot ser una pàgina HTML, un fitxer, o dades que el client ha de processar.

El servidor sovint executa aplicacions desenvolupades amb tecnologies com PHP (Laravel), Python (Django, Flask), Ruby (Ruby on Rails), Node.js, o Java (Spring).

Aquestes aplicacions gestionen la lògica de negoci, la seguretat i la interacció amb les bases de dades.

El servidor també pot gestionar serveis d'autenticació, autorització i altres funcions essencials com la validació de dades i la generació de continguts dinàmics.



UD1: ... - Front-end vs Back-end

Com s'ha dit, altres termes que apareixen en la part del servidor són:

- **Base de dades:**

Sovint, el servidor està connectat a una base de dades que emmagatzema la informació necessària per a l'aplicació web. Les bases de dades poden ser SQL (com MySQL, PostgreSQL) o NoSQL (com MongoDB).

Quan un client sol·licita dades específiques (per exemple, els detalls d'un producte en una botiga en línia), el servidor fa una consulta a la base de dades, obté la informació necessària, i la retorna al client.



UD1: ... – Front-end vs Back-end

- **APIs i serveis web:**

En molts casos, el servidor exposa una API (Application Programming Interface) que permet als clients accedir a funcionalitats específiques de l'aplicació web. Les APIs RESTful són comunes, i permeten als clients interactuar amb el servidor mitjançant operacions com GET, POST, PUT, i DELETE.

Aquestes APIs poden ser consumides no només pels navegadors web, sinó també per aplicacions mòbils o altres serveis web, facilitant la integració amb sistemes externs.



UD1: ... - Front-end vs Back-end

- **Capa de seguretat:**

En un entorn web, la seguretat és clau. El servidor implementa mesures de seguretat per assegurar que només els usuaris autoritzats poden accedir a certs recursos. Això inclou l'ús de certificats SSL/TLS per xifrar les comunicacions, l'autenticació d'usuaris, i la protecció contra atacs comuns com el SQL injection o el Cross-Site Scripting (XSS).



UD1: ... - Front-end vs Back-end

- **Exemple pràctic:**

Suposem una aplicació qualsevol, per exemple, un usuari vol veure un producte:

- El client (el navegador) envia una sol·licitud al servidor per veure les dades del producte.
- El servidor processa la sol·licitud, consulta la base de dades per obtenir la informació del producte (nom, preu, descripció, imatges, etc.), i genera una pàgina HTML que inclou aquesta informació. Aquesta pàgina es retorna al client, que la mostra a l'usuari.
- També podria retornar un arxiu JSON amb les dades del producte i ser el client qui, en rebre l'arxiu, el mostri en el navegador

En aquest procés, el model client-servidor permet una clara separació de responsabilitats, on el servidor se centra en la gestió de la lògica i les dades, mentre que el client es focalitza en la presentació i la interacció amb l'usuari. Això facilita el desenvolupament, la mantenibilitat i l'escalabilitat de l'aplicació web.



UD1: ... - Front-end vs Back-end

En resum ...

Ambdós termes són utilitzats en el desenvolupament web. Descriuen les dues àrees d'una aplicació web.

Front-end

Es refereix a la part d'una aplicació que interactua directament amb els usuaris. És la cara visible d'una aplicació i s'executa al navegador de l'usuari.

Les tecnologies principals que es fan servir són:

- HTML, CSS i JavaScript



UD1: ... - Front-end vs Back-end

Back-end

Es refereix a la part d'una aplicació que s'executa al servidor i s'encarrega de processar la lògica de negoci, la gestió de les dades i la comunicació amb bases de dades o altres serveis externs. El back-end s'ocupa de:

- La autenticació i autorització d'usuaris
- La manipulació de dades.
- La lògica de negoci i la generació de respostes que s'envien al front- end

Les tecnologies utilitzades poden variar segons les preferències i el propòsit de l'aplicació, Les més comuns són:

- Llenguatges de programació: **PHP**, Python, Ruby, Java, C#, ...
- Bases de dades: **MySQL**, PostgreSQL, MongoDB, ...
- Servidors web: **Apache**, Nginx, IIS, etc.



UD1: ... - El llenguatge PHP

PHP és un llenguatge interpretat de propòsit general àmpliament utilitzat especialment dissenyat per al desenvolupament web que es pot incrustar dins del codi HTML.

Generalment s'executa en un servidor web, pren el codi PHP com a entrada i crea pàgines HTML com a sortida. També és capaç de crear serveis web (comunment anomenats micro-serveis) en forma d'API per resoldre les peticions de la part client.

Es pot implementar a la majoria de servidors web i a gairebé tots els sistemes operatius i plataformes sense cap cost.



UD1: ... - La base de dades MySQL

MySQL és un sistema de gestió de bases de dades relacional de codi obert, àmpliament utilitzat en aplicacions web per emmagatzemar i gestionar dades.

Quan s'utilitza amb PHP, un llenguatge de programació per al desenvolupament web, MySQL serveix com a backend per emmagatzemar dades que PHP pot recuperar, processar i mostrar dinàmicament en pàgines web.

Aquesta combinació és popular per crear aplicacions web interactives i dinàmiques.



UD1: ... - El servidor web Apache

Apache Server és un servidor web de codi obert que serveix contingut web als navegadors dels usuaris.

Aquest conjunt permet crear aplicacions web completes, on Apache s'encarrega de les sol·licituds dels usuaris, PHP de la lògica del servidor i MySQL de l'emmagatzematge de dades.



UD1: ... - El servidor web Apache

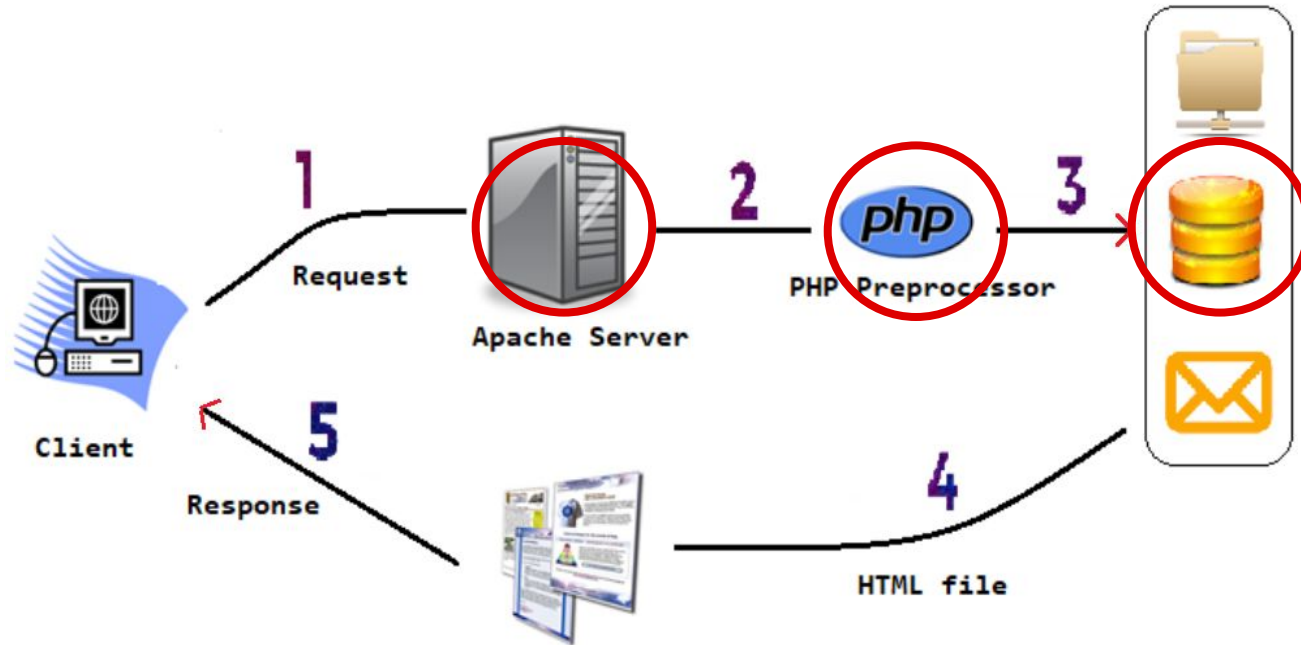
Quan es combina amb PHP i MySQL, Apache Server forma part del popular stack de desenvolupament LAMP (Linux, Apache, MySQL, PHP), utilitzat per crear aplicacions web dinàmiques.

- Apache i PHP: Apache Server pot processar scripts PHP incrustats en pàgines web. Quan un usuari sol·licita una pàgina PHP, Apache delega la tasca al motor de PHP per executar el codi i generar HTML dinàmic que es retorna al navegador.
- Apache i MySQL: En una aplicació web, PHP s'utilitza per interactuar amb una base de dades MySQL, enviant consultes per recuperar o emmagatzemar dades. Apache Server gestiona aquestes sol·licituds, permetent que PHP accedeixi a MySQL i retornant els resultats als usuaris a través de pàgines web.



UD1: ... - LAMP

PHP execution Phases





UD1: ... – Frameworks

Un framework és un conjunt d'eines, llibreries que proporciona una estructura predefinida per al desenvolupament de programes o aplicacions.

Facilita el treball dels desenvolupadors en oferir solucions reutilitzables per problemes comuns, com ara la gestió de dades, el control de flux o la interacció amb la interfície d'usuari.

Un framework estableix una base sobre la qual es pot construir una aplicació, estalviant temps i assegurant una arquitectura més organitzada i mantenible.

Exemples de frameworks en entorn servidor són Laravel per PHP, Spring per Java, o React per JavaScript.



UD1: ... – Frameworks – Laravel

Laravel és un Framework basat en PHP multiplataforma que és molt útil per desenvolupar aplicacions web.

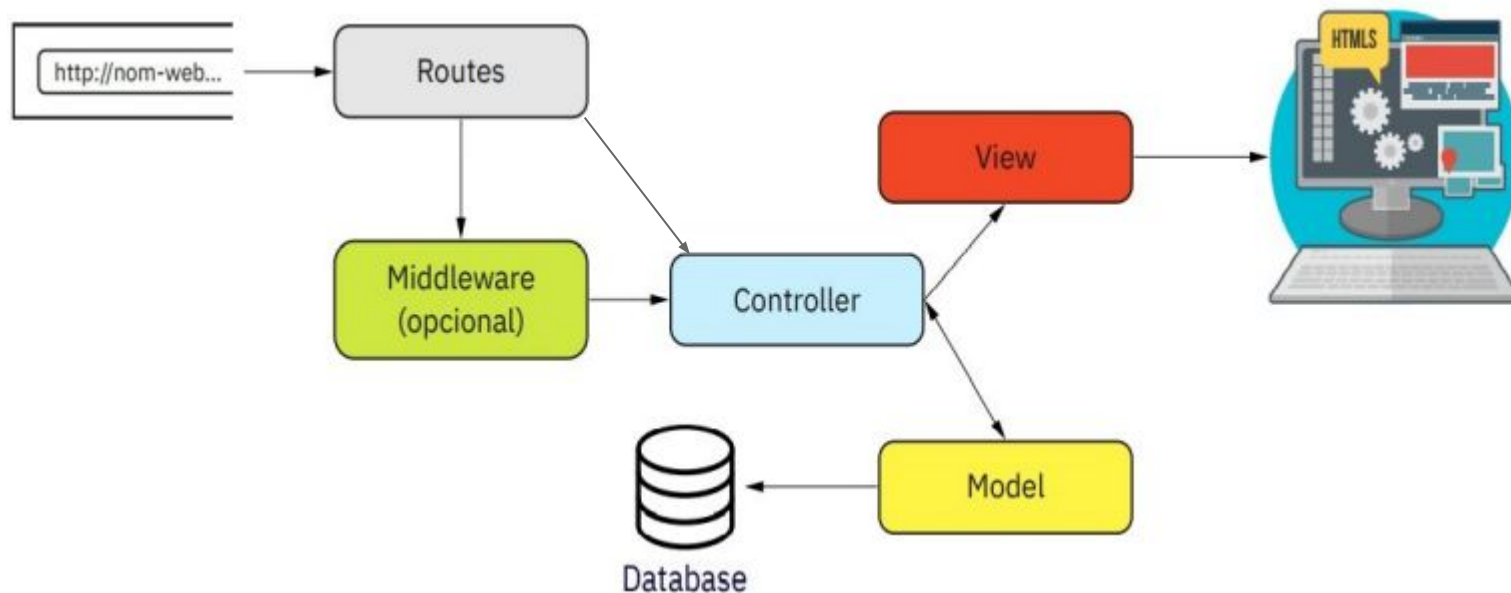
Incorpora algunes eines que faciliten el desenvolupament d'aplicacions:

- **Artisan Console:** inclou comandes per generar codi, executar migracions, gestionar la base de dades, etc.
- **Eloquent ORM:** facilita la creació, lectura, actualització i esborrat (CRUD) de dades en la base de dades.
- **Blade Templating Engine:** permet la creació de plantilles HTML amb una sintaxi clara i fàcil d'utilitzar.
- **Migrations:** permet crear, modificar i compartir l'esquema de la base de dades de manera estructurada i versionada.
- **Seeders i factories:** facilita la generació de dades inicials per a proves i desenvolupament.
- **Tinker:** permet provar i interactuar amb el codi de l'aplicació directament des de la línia de comandes.
- ...



UD1: ... - Introducció

Laravel utilitza el patró MVC (Model, Vista, Controlador) que és un patró arquitectònic de programari que separa una aplicació en tres capes.



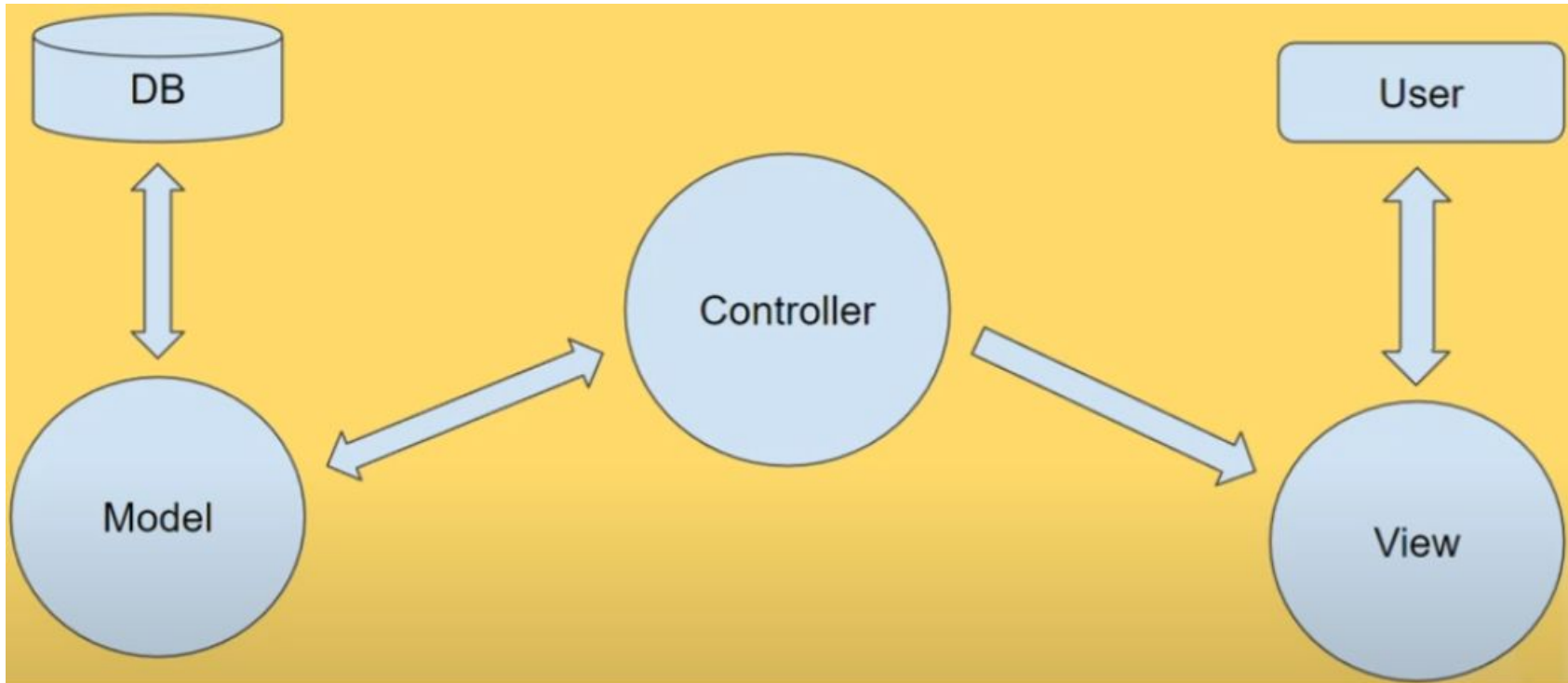


UD1: ... - Introducció a l'arquitectura Model-Vista-Controlador

El MVC ('Model-Vista-Controlador') és una arquitectura que s'utilitza en el desenvolupament d'aplicacions web o d'escriptori. En aquesta, cada component realitza una tasca de manera separada, la qual cosa facilita el desenvolupament d'aplicacions en equip.

- 'Controlador': aplica la lògica de càlcul de l'aplicació.
- 'Vista': representa la interfície entre el programa i l'usuari: mostra la informació i la recull.
- 'Model': és l'encarregat de gestionar les dades: proveir-les, validar-las, formatejar-les ...

UD1: ... - Introducció a l'arquitectura Model-Vista-Controlador





UD1: ... - Introducció

El punt d'entrada per a totes les sol·licituds a una aplicació Laravel és l'arxiu `'public/index.php'`. El servidor web es configura de manera que totes les sol·licituds són dirigides a aquest arxiu.

En essència, el que fa és el següent:

1. L'arxiu es carrega amb totes les dependències generades.
2. Generar instància del kernel de laravel i registra els components principals.
3. Maneja la sol·licitud, genera i envia una resposta.
4. Finalitza la sol·licitud.



Recursos de la UD

Additional resources

- <https://aitor-medrano.github.io/dwes2122/index.html>
- <https://www.apuntesinformaticafp.com/modulos/dwes.html>



Bibliografia

- **Implantació d'Aplicacions Web**
CFGS – Administració de sistemes informàtics i xarxes
Institut d'Ensenyaments a Distància de les Illes Balears
 - <https://iedib.net/avirtual/course/view.php?id=263>
Usuari: '*visitant*'; Passwd: '*iedib*'
Revisar els Lliuraments 3, 4 i 5



Autor i Llicència:

Autor: Bartomeu Vives

Sou Lliures de:

Compartir — copiar i redistribuir el material en qualsevol mitjà i format

Adaptar — remesclar, transformar i crear a partir del material

El llicenciador no pot revocar aquestes llibertats, sempre que seguiu els termes de la llicència.

Amb els termes següents:

Reconeixement — Heu de reconèixer l'autoria de manera apropiada, proporcionar un enllaç a la llicència i indicar si heu fet algun canvi. Podeu fer-ho de qualsevol manera raonable, però no d'una manera que suggereixi que el llicenciador us dóna suport o patrocina l'ús que en feu.

NoComercial — No podeu utilitzar el material per a finalitats comercials.

CompartirIgual — Si remescleu, transformeu o creeu a partir del material, heu de difondre les vostres creacions amb la mateixa llicència que l'obra original.

No hi ha cap restricció addicional — No podeu aplicar termes legals ni mesures tecnològiques que restringeixin legalment a altres de fer qualsevol cosa que la llicència permet

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>