Estudis	Grau en Comunicació Interactiva
Assignatura	104740 – Programació d'Aplicacions en Tecnologia Web
Pràctica 2	Aplicacions Backend I

Resultats d'aprenentatge

1 Analitzar una situació i identificar-ne els punts de millora.
4 Dissenyar pàgines web i aplicacions funcionals des del punt de vista tecnològic.
5 Idear aplicacions per a les pàgines web.
7 Identificar situacions que necessiten un canvi o millora.
9 Planificar i executar treballs acadèmics en l'àmbit de programació bàsica i avançada.
10 Ponderar els riscos i les oportunitats de les propostes de millora tant pròpies com alienes.
11 Presentar els treballs de l'assignatura en els terminis previstos i mostrar-ne la planificació individual o grupal aplicada.
12 Proposar nous mètodes o solucions alternatives fonamentades.

Temporització

9 hores lectives

Mode de desenvolupament

Grupal de 2/3 persones

Lliurament

Data de lliurament: **27/11/2023**

Tindrà un triple lliurament: per Campus Virtual, com a *release* a un repositori privat compartit de GitHub i a *alwaysdata*

Possibles recursos

1. SQL: https://www.w3schools.com/sql/. En especial:
    - Fer selects
    - Fer inserts
2. PHP: https://www.w3schools.com/php/default.asp. Més enllà dels fonaments de PHP:
    - Incloure arxius php
    - Validació de formularis
    - Treballar amb sessions
    - Treball amb MySQL . Se us recomana treballar amb la classe mysqli.
    - Redireccions amb PHP

Criteris d'Avaluació

	Pes	A	B	C	D	E
Funcionament	40					
Estructura i codi	20					
Ampliacions (mail...)	20					
Gantt/Trello	10					
Lliurament correcte	10					

Enunciat

Una vegada feta la capa de presentació anem a treballar en el *backend*. En aquesta primera fase ens centrarem únicament en els processos de validació (*login*) i de registre (*signin*).

Base de dades

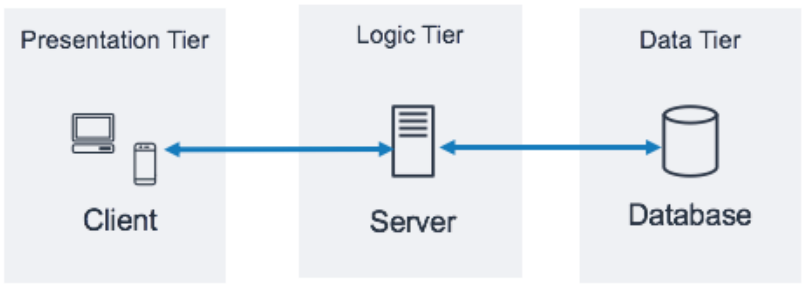
Creeu una base de dades MySQL anomenada *todoapp* i una taula anomenada *usuaris* amb els camps id (PK), nom, cognoms, alies, email i passwd. Us proposo aquesta definició per a cada camp:

Nom	Tipus	Característiques
id	int	Clave primària – Autoincremental
nom	varchar (30)	No null
cognoms	varchar (30)	No null
alies	varchar (30)	No null
email	varchar (50)	No null
passwd	varchar (255)	No null

Per tal de crear-la se us recomana que utilitzeu el phpmyadmin d'alwaysdata. Si ja sabeu fer-ho o voleu investigar, podeu utilitzar també l'eina *MySQL Workbench*.

Estructura del projecte

L'aplicació tindrà una arquitectura de 3 capes:



Com sabeu, la capa de presentació seria la que s'encarrega de les vistes de l'usuari, la capa lògica (*o de negoci*) qui gestiona els condicionants específics de l'aplicació i la capa de dades la que s'encarrega de "parlar" amb la base de dades.

A continuació, se us especifiquen els arxius mínims necessaris, una proposta de noms i funcionalitats i la capa en què treballaran:

Capa de presentació	
index.html	Aquesta plana serà l'inici de la nostra aplicació i ens mostrarà un formulari on poder-nos validar (email/pwd). Per altra banda, oferirà l'accés a la plana de registre (*registre.php*) per aquelles persones encara no registrades.
registre.html	Aquesta plana ens mostrarà un formulari on podrem introduir les dades per donar-nos d'alta dins de l'aplicació.

dashboard.php	Aquesta plana serà l'escriptori de l'aplicació on es mostraran i es gestionaran totes les tasques. Per ara, només haurà de tenir una petita capçalera on mostri l'à*lies* de l'usuari actiu i una opció de *logout*. ***En aquesta segona pràctica les funcionalitats de les tasques no han de ser funcionals***. Cas que algú accedeixi directament sense estar validat, farà una redirecció a la plana d'error (*error.php*) amb el missatge corresponent.
error.php	Aquesta plana mostrarà tots els errors que es vagin produint a l'aplicació. Per tenir missatges personalitzats utilitzarem variables de sessió entre planes (flash data).
Capa de negoci	
Usuari.php	Aquest arxiu serà una classe que permetrà gestionar tot el referent a l'usuari. Haurà d'oferir els mètodes necessaris com, per exemple, addUser, isValidPasswd, isValidUser...
login.php	Aquest arxiu està connectat amb l'*index.html* permetrà gestionar els processos de *login* Haurà de sanejar i validar tot el rebut de la capa la presentació. Si tot és correcte haurà d'instanciar un objecte Usuari i utilitzar els mètodes adequats. Si la validació és correcte farà una redirecció a *dashboard.php*. Si no farà una redirecció a la plana d'error (*error.php*) amb el missatge corresponent.
registre.php	Aquest arxiu està connectat amb *registre.html* i permetrà gestionar els processos de registre d'usuari. Haurà de sanejar i validar tot el rebut de la capa la presentació. Si tot és correcte haurà d'instanciar un objecte Usuari i utilitzar els mètodes adequats. Si el registre és correcte farà una redirecció a *dashboard.php* o a *index.php*. Si no farà una redirecció a la plana d'error (*error.php*) amb el missatge corresponent.
Capa de dades	
config.php	Es definiran totes les variables i constants necessàries per a accedir a la base de dades.
Database.php	Aquest arxiu serà una classe que permetrà gestionar l'accés a la BBDD i donarà servei a la capa de dades amb mètodes com ara . addUser, getUserPasswd, getUserByEmail,...
Helpers	
valida.php	Es definiran totes les funcions que permetre sanejar i validar els nostres continguts.

En aquesta pràctica serà necessari:

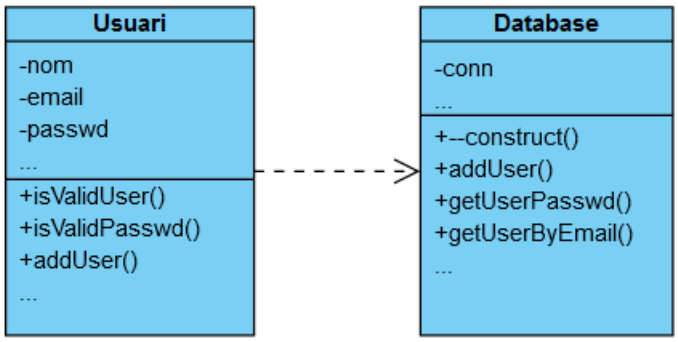
1. Que cada capa s'encarregui únicament de les seves responsabilitats.
2. Que creem una estructura d'arxius i directoris adequada i que tinguem clarament identificats els arxius que pertanyen a cada capa.
3. Que dividiu en arxius separats les diferents funcionalitats d'una determinada capa.

Programació

És important que tot el codi sigui el màxim de modular i clar possible. Us aconsello:

1. Que els noms siguin molt descriptius.
2. Que les classes estiguin en format PascalCase.
3. Que les funcions i mètodes estiguin en format *camelCase*.
4. Que tot estigui molt ben comentat.

A sota teniu el diagrama de classe UML. Com veieu, he utilitzat una dependència ja que la classe *Usuari* utilitzarà la classe *Database* per poder fer les seves tasques:



Com sempre, es valorarà molt positivament qualsevol millora com, per exemple:

1. Emmagatzematge de contrasenyes xifrades. Teniu informació a xifrar contrasenyes (funcions *password_hash(), pasword_verify()*).
2. Enviament d'un correu de benvinguda. Teniu informació a correus amb PHP.
3. Modificació de dades d'usuari.
4. Eliminació d'un compte d'usuari.
5. ...