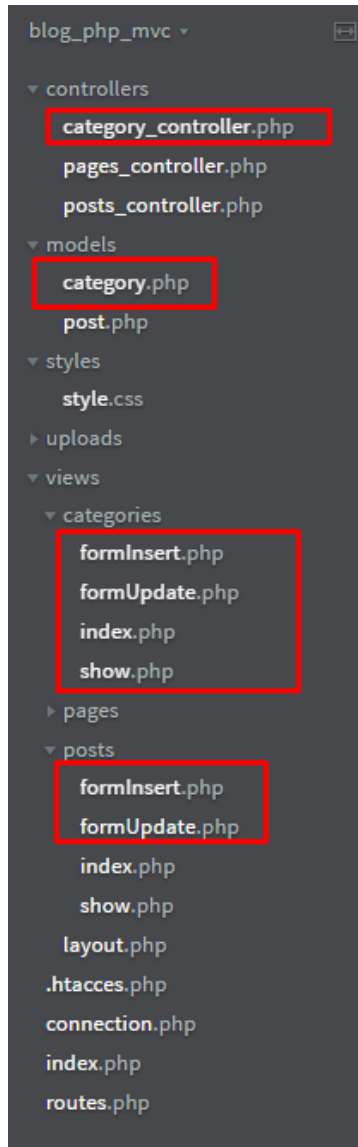


M07 UF2: PRÀCTICA 2

- Que he afegit?

Bàsicament, he completat el CRUD de la taula posts i he afegit una nova taula amb altre crud.



He creat tots aquests arxius per poder realitzar tot el funcionament del CRUD, sobretot per a la nova taula de categories.

L'esquema i els procediments són exactament els mateixos que es mostraven al pdf del tutorial, així que tot té una estructura semblant.

Per poder completar el CRUD per a la taula posts he hagut d'afegir 2 fitxers nous: El formInsert i el formUpdate. Aquests dos són formularis per poder afegir els paràmetres necessaris per crear o modificar un post.

Apart d'això, he creat noves funcions dins de posts_controller i de post per poder fer que funcioni correctament.

- Arxius posts_controller y post

Aquests dos arxius han rebut grans canvis i s'han afegit moltes funcions noves.

```

▼ public function formInsert(){ //Aquesta funcio només truca a l'arxiu del formulari post
    require_once('views/posts/formInsert.php');
}
▼ public function insert(){ //Aquesta funció realitza un insert amb els paràmetres del formulari i torna a la llista de
posts
    $insert=Post::insert();
    call('posts', 'index');
}
▼ public function delete(){ //funcio que esborra el post seleccionar a l'index
    $delete=Post::delete();
    call('posts', 'index');
}
▼ public function formUpdate(){ //Funcio que truca al formulari per actualitzar els posts
▼     if (!isset($_GET['id'])) {
        return call('pages', 'error');
    }
    // utilizamos el id para obtener el post correspondiente
    $post = Post::find($_GET['id']);
    require_once('views/posts/formUpdate.php');
}
▼ public function update(){ //Funcio que actualitza els parametres d'un post i torna a index.php
    $update=Post::update();
    call('posts', 'index');
}
}
?>

```

Aquestes funcions, dins de post_controller, implementant el crud juntament amb l'arxiu post.php.

```

42
43 ▼ public static function insert() { //Funcio per insertar un nou post
44     $db = Db::getInstance();
45
46     $author=$_POST['author']; //declarem les variables
47     $content=$_POST['content'];
48     $title=$_POST['title'];
49     $categoria=$_POST['categoria'];
50     $foto=htmlspecialchars(strip_tags($_FILES['foto']['tmp_name'])); //Per a la foto, treiem els caràcters especials
51
52     $foto=!empty($_FILES["foto"]["name"])
53         ? sha1_file($_FILES['foto']['tmp_name']) . "-" . basename($_FILES["foto"]["name"]) : "";
54
55     $req = $db->prepare("INSERT into posts(author,content, title, foto, categoria) VALUES (:author, :content,:title,
:foto, :categoria)"); //Preparam la comanda
56     $req->bindParam(':author', $author);
57     $req->bindParam(':content', $content);
58     $req->bindParam(':title', $title);
59     $req->bindParam(':foto', $foto);
60     $req->bindParam(':categoria', $categoria);
61 ► if($req->execute()){...}
140
141 // if unable to create the product, tell the user
142 ▼ else{
143     echo "<div class='alert alert-danger'>No s'ha pogut crear el producte</div>";
144 }
145
146 }
147
148 ▼ public static function update(){ //Funcio per actualitzar un post
149     $author=$_POST['author']; //Fem el mateix procés que a insert: Declarem variables
150     $content=$_POST['content'];
151     $title=$_POST['title'];
152     $fecha = new DateTime();
153     $fecha=$fecha->getTimestamp(); //Posarem la data d'última modificació com el mateix instant en el que s'executa
aquesta funció
154     $id=$_POST['id'];
155     $categoria=$_POST['categoria'];
156     $foto=htmlspecialchars(strip_tags($_FILES['foto']['tmp_name']));
157
158     $foto=!empty($_FILES["foto"]["name"])
159         ? sha1_file($_FILES['foto']['tmp_name']) . "-" . basename($_FILES["foto"]["name"]) : "";
160

```

L'arxiu post té bastant més contingut (tot explicat dins del codi), i ha sigut bastant difícil fer que tot funcioni correctament.

Totes aquestes funcions fan que el CRUD funcioni de manera correcta.

[Home](#) [Veure Posts](#) [Insertar Post](#) [Insertar Categoria](#) [Veures categories](#)

Listat de categories:

Andrés Baco Baltanal

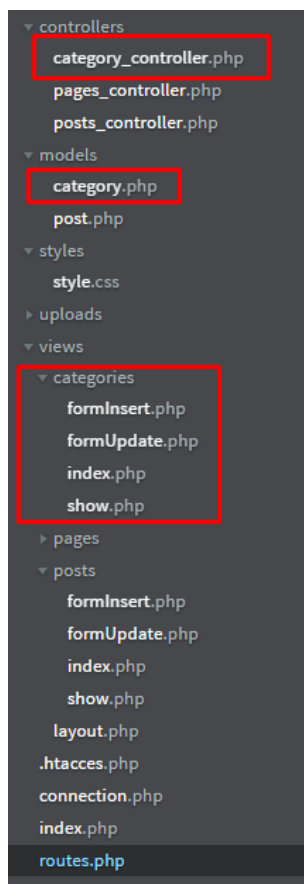
Categoria	Data d'alta			
Article	2018-11-22	Ver contenido	Eliminar	Modificar
manga	2018-12-20	Ver contenido	Eliminar	Modificar

- CRUD per a la taula categories

El CRUD que he realitzat per a la nova taula és codi pràcticament copiat del CRUD de posts. Només he hagut de canviar les variables i modificar els formularis. Encara que és gairebé igual a la taula posts, he hagut d'afegir bastants arxius nous i modificar alguns ja creats.

```
$controllers = array( 'pages' => ['home', 'error'],
//S'han afegit noves accions. Les accions form son els formularis que s'utilitzaran per fer els
inserts i els updates
'posts' => ['index', 'show', 'formInsert', 'insert', 'delete', 'formUpdate', 'update'],
//he creat el nou controlador anomenat categoria per utilitzar i manipular una segona taula.
'category' => ['index', 'show', 'formInsert', 'insert', 'delete', 'update', 'formUpdate']);
```

El principal canvi és que he afegit un nou controlador amb les mateixes accions que a POSTS.



A continuació, he creat tots aquests arxius per a la manipulació de la taula categories, que en realitat són codis copiats de la taula posts, ja que les funcions que es realitzen en les dos taules són les mateixes.

- Dificultats

Deixant de banda les tasques que no he pogut realitzar (URLS amigables, paginació, etc.) la major dificultat que he tingut ha sigut la fotografia de la taula posts. Crec que no sóc l'únic que s'ha quedat la majoria del temps per fer aquesta pràctica intentant fer que la foto s'afegeixi correctament i després es pugui veure. Al final ho vaig aconseguir:

Data d'alta: 2018-11-28 19:49:14.508188

Ultima modificacio: 0000-00-00 00:00:00.000000

UsuariController	PersonalController
alta() baixa() modificar() consultarDades()	alta() baixa() modificar() consultarDades()

Foto:

Andrés Baco Baltanal