MANUAL IMPLEMENTACION ORM

Dados los incontables problemas que me ocasoinó implementar el OMR de Illuminate en un proyecto de php es que decido dejar este manual como base de procedimiento rápido para su implementación a posteriori.

1) crear una carpeta de proyecto en htdocs

3) desde la terminal instalar composer si no se tiene -> sudo apt install composer

4)posicionados en la carpeta de desarrollo comenzar a instalar las dependencias que necesitamos:

1- **composer require raveren/kint** ->nos instala un paquete que nos brinda la funcion global d($objeto) para visulizarla de forma super agradable mejorando a var
2- **composer require Illuminate/database** -> instala el ORM que utilizaremos para gestionar la base de datos.

Todo esto nos auto genera el archivo composer.json necesario como manifiesto de dependencias dentro del proyecto.

5) Crear carpetas y archivos necesarios del proyecto

1- Para cada archivo hay que usar el require ‘vendor/autoload.php’ como cabecera, para poder usar el auto cargado de namespaces.

6) se crea un archivo de conexion a la base de datos,es buena idea crear una carpeta llamada database.php con toda esta configuracion y posteriormetente exportar a donde se necesite.

1- en este archivo es necesario incluir el siguiente codigo en el cual se instancia la base de datos, lo que permite tener acceso a sus metodos de conexion.

```php
<?php
use Illuminate\Database\Capsule\Manager as Capsule;

$capsule = new Capsule;

$capsule->addConnection([
    'driver'    => 'mysql',
    'host'      => 'localhost',
    'database'  => 'database',
    'username'  => 'root',
    'password'  => 'password',
    'charset'   => 'utf8',
    'collation' => 'utf8_unicode_ci',
    'prefix'    => '',
]);


// Make this Capsule instance available globally via static
methods... (optional)
$capsule->setAsGlobal();

// Setup the Eloquent ORM... (optional; unless you've used
setEventDispatcher())
$capsule->bootEloquent();
```

2- se deben modificarlos valores adecu[andols a la base de datos usada

7) se crea dentro de la carpeta model a las clases de entidades que vayamos a utilizar en archivos independientes, la forma general de esto es:

```php
<?php

use Illuminate\Database\Eloquent\Model;

class MiClase extends Model {

    //Declaramos obigatoriamente esta variable table, la cual contiene el nombre de la tabla asociada en la DB con esta clase modelo
    protected $table= 'mi_clase';

    /declaramos demas atributos de la clase. No olvidar que cada vez que se crea una tabla en la base de datos para asociarla, hay que agregar los campo
    //created_at y updated_at*/

/*El siguiente metodo sirve tanto  para declarar atributos de la clase asi como para designar a solo los atributos que seran enviados a la tabla asociada dentro de la base de datos */
    protected $fillable=['atributo1','atributo2',...];


}
```

Operaciones en las bases de datos

Mostrar todos los registros de una clase asociada a la calse Model:

```php
$usuarios = Usuario::get();
```

Mostrar un registro especifico (considerando que se trata de un array) y un atributo especifico

```php
d($usuarios[0]->nombre);
```

Buscar un registro especifico por id

```php
$usuario = Usuario::find(2);

d($usuario);
```

Insertar un registro desde una clase modelo, definiendo el array asociativo correspondiente a los $Fillable declarados

```php
Usuario::create([
    'nombre' => 'DIEGO',
    'apellido' => 'GONZALES',
    'email' => 'EMAIL'
]);
```

Hacer un Update: primero se buca con find el registro a modificar, luego con el metodo update se modifican los campos que se quieren cambiar, y finalmente con save se ejecuta

```php
$usuario = Usuario::find(3);

$usuario->update([
    'email' => 'diego@gmail.com'
]);

$usuario->save();
```

Eliminar un registro por el id

```php
Usuario::destroy(2);
```