Patrón de diseño MVC

+

PHP y POO

Moisés Espíndola Oropeza
www.creasati.com.mx
zaer00t@gmail.com
@zaer00t

PHP
HTML
CSS
JAVASCRIPT

Requisitos

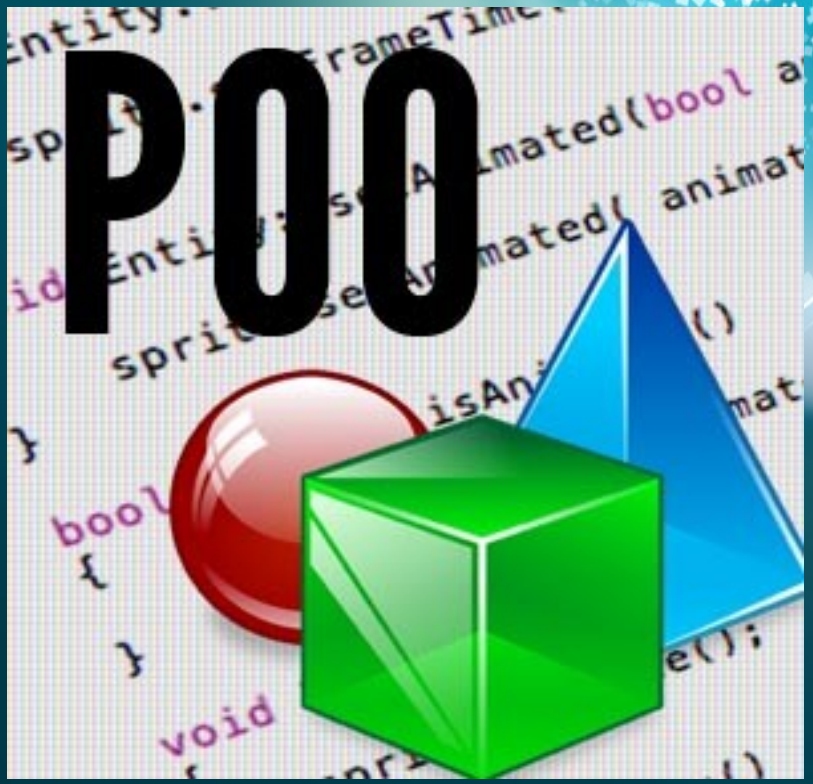
Hacer software no es fácil

Diseñar software orientado a objetos es difícil, y diseñar software orientado a objetos reutilizable es todavía más difícil

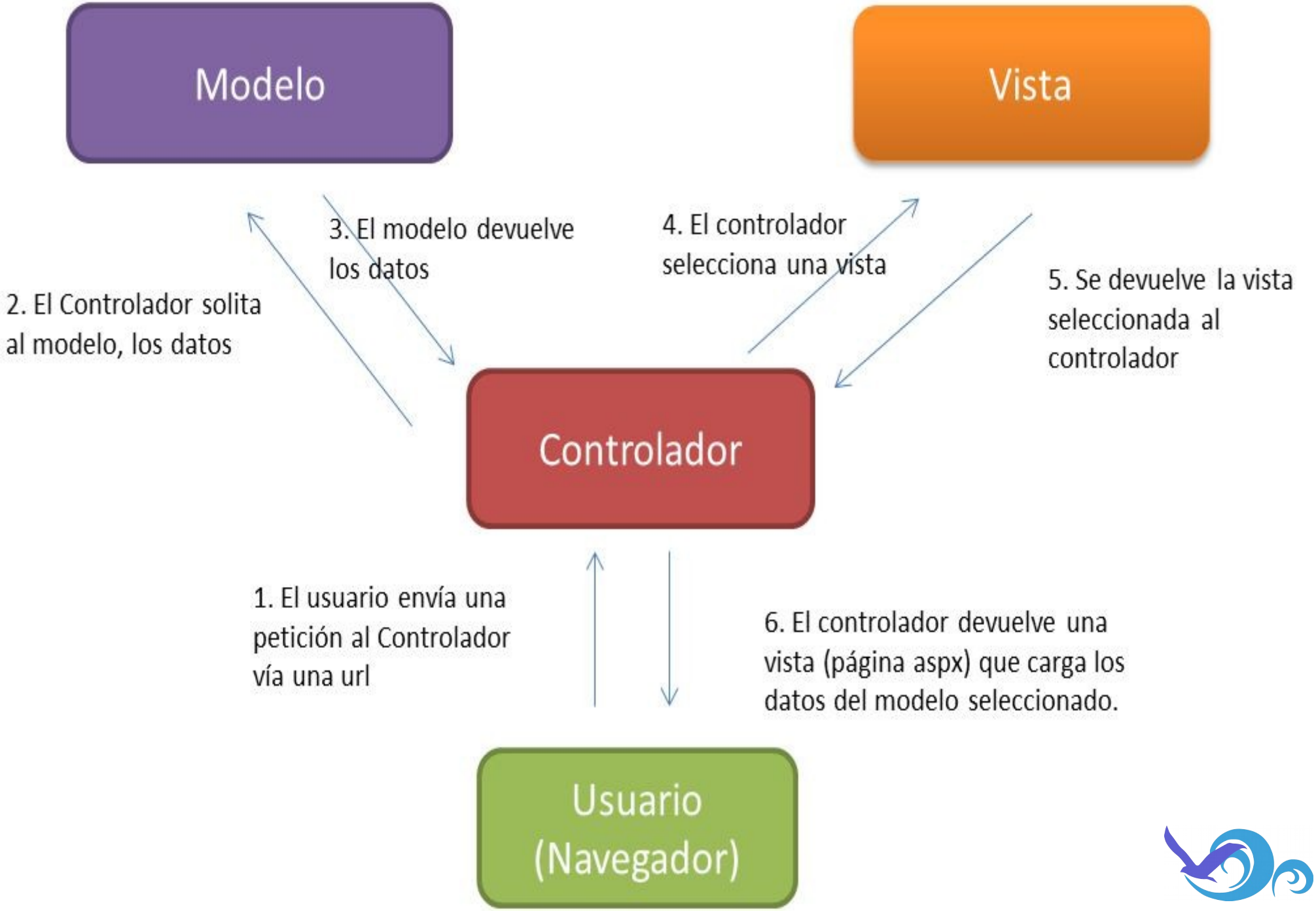
...y un software capaz de evolucionar tiene que ser reutilizable (al menos para las versiones futuras)

Diseñar para el cambio

- **El software cambia**
- **Para anticiparse a los cambios en los requisitos hay que diseñar pensando en qué aspectos pueden cambiar**
- **Los patrones de diseño están orientados al cambio**



Ejemplo de patrón de diseño



Let's Get Ready to Rhumble

A PROGRAMAR!

Estructura de directorios.

El directorio principal: “app” contiene la estructura del sitio.

- “assets” elementos frontend
- “classes” modulos y nucleo de la app.
- “controllers”
- “views”

```
zaer@titan:~/www/crt$ tree -d
.
└── app
    ├── assets
    │   ├── css
    │   ├── img
    │   └── js
    ├── classes
    │   ├── agua
    │   │   └── imageworkshop
    │   └── core
    │       └── php_mailer
    ├── controllers
    └── views
        ├── Blog
        ├── promos
        └── widgets
            └── Blog

16 directories
zaer@titan:~/www/crt$
```

Configuración app.php

app.php

```php
if (APP_PROFILE == 1)
{
    define("APP_HOST_URL", "http://" . $_SERVER["HTTP_HOST"]);
    define("APP_ROOT_PATH", $_SERVER['DOCUMENT_ROOT']);

    define("APP_DB_HOST", "localhost");
    define("APP_DB_USER", "[redacted]");
    define("APP_DB_PASSWORD", "[redacted]");
    define("APP_DB_CATALOG", "[redacted]");
}
define("APP_NAME","www.creasati.com.mx");
define("APP_DEFAULT_CONTROLLER", "InicioController");
define("APP_ASSETS", APP_HOST_URL . "/app/assets");
define("APP_IMG_URL", APP_HOST_URL . "/app/assets/img");
define("APP_CSS_URL", APP_HOST_URL . "/app/assets/css");
define("APP_JS_URL", APP_HOST_URL . "/app/assets/js");
define("APP_BIN_PATH", APP_ROOT_PATH . "/app");
define("APP_CLASS_PATH", APP_BIN_PATH . "/classes");
define("APP_VIEW_PATH", APP_BIN_PATH . "/views");
define("APP_CONTROLLER_PATH", APP_BIN_PATH . "/controllers");
define("APP_WIDGET_PATH", APP_BIN_PATH . "/views/widgets");
define("APP_MAILS_PATH", APP_BIN_PATH . "/mails");
define("APP_IMG_PATH", APP_BIN_PATH . "/assets/img");
define("APP_DEBUG", true);
```

Peticion al Controlador Principal

index.php

Line: 23

```
17  require_once("app/app.php");
18  $class_name = APP_DEFAULT_CONTROLLER; //Controlador por defecto a ser invocado
19  $function_name = "index";»  //metodo del controlador
20  $function_params = array();
21  $params = array();
22
23  if (isset($_GET["url"]))
24      $params = explode("/", $_GET["url"]);
25
26  if(isset($params[0]))
27  {
28      $tmp = ucfirst($params[0]) . "Controller";
29
30      if(file_exists(APP_CONTROLLER_PATH . "/" . $tmp . ".php" ))
31      {
32          $class_name = $tmp;
33          App::load_controller($class_name);
34          if(isset($params[1]))
35          {
36              if(method_exists($class_name, $params[1]))
37              {
38                  $function_name = $params[1];
39                  for($i = 2; $i < count($params); $i++)
40                  {
```

¿Como funciona?

http://web.com/blog/entradas/php


```php
class BlogController implements Controller
{
    public static function index($params)
    {
        $db = new Database();
        $db->connect();
        $articulos = Miblog::leer_todos($db);
        App::load_view("Blog/index",array('articulos'=>$articulos));
    }
}
```

Interfaces

- Ampliando las referencias de PHP, incorporando con el concepto de interfaces.

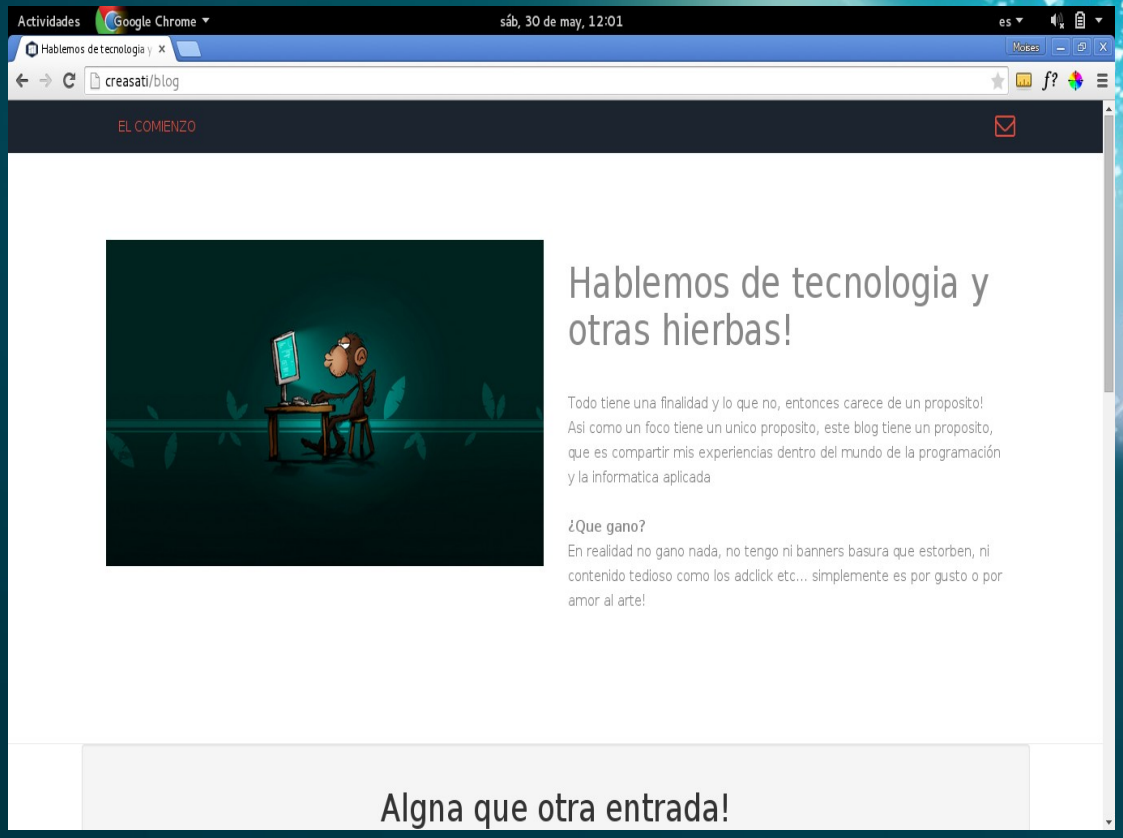
```
interface Controller {

    public static function index($params);
}
```

El Manual Oficial de PHP describe las interfaces de la siguiente forma:

“Las interfaces de objetos permiten crear código con el cual especificar qué métodos deben ser implementados por una clase, sin tener que definir cómo estos métodos son manipulados. Las interfaces son definidas utilizando la palabra clave interface, de la misma forma que con clases estándar, pero sin métodos que tengan su contenido definido. Todos los métodos declarados en una interfaz deben ser public, ya que ésta es la naturaleza de una interfaz.

VISTAS

Generalmente, en la práctica, no somos los programadores quienes nos hemos de encargar de la GUI. Es tarea que corresponde a diseñadores Web o gráficos, según aplique.



REFERENCIAS

Patron MVC con PHP

- Github
- http://github.com/zaer00t
- POO y PHP
- http://php.net
- Su servidor
- http://creasati.com.mx
- Twitter: @zaer00t