

Código limpio

revelando la intención del código



**EXTREMADURA
DIGITAL
DAY**

**EVENTO
EXTREMEÑO DE
NEGOCIO,
MAKERS Y
DESARROLLO**

Manuel Pijierro Sa - @mpijierro

Gracias a...



an NTT DATA Company



KEEP CODIN



vabadus



Gracias a...



Wordpress
Extremadura



Drupal Cáceres



Wordpress Badajoz



PHP Cáceres



CáceresJS



GDG Cáceres



ExtrePython



Caceres DeVops

Comunidades

Sobre mi



Manuel Pijierro Sa



mandev.es/cv/



Reference Leader Software Development at **Singular**



[@mpijierro](https://twitter.com/mpijierro)



<https://github.com/mpijierro>



<https://medium.com/@mpijierro>

Sobre la charla

Talk is cheap. Show me the code

Torvalds, Linux (25-08-2000)

Sobre el código limpio

Correcto

Simple

Testeable

Extensible

Sobre el código limpio

No existe una definición exacta

Tiene algo de subjetivo

Proceso de mejora continua

Sobre el código limpio

"Code is like humor. When you have to explain it, it's bad"

Cory House, @housecor

"It is not enough for code to work"

Robert C. Martin, @unclebobmartin

Mantenibilidad



Mantenibilidad

LEGIBLE

COMPENSIBLE

Aspectos del código que mejoran
y facilitan su legibilidad y
comprensibilidad

1) Sobre los nombres



1) Sobre los nombres

Evita nombres que no revelen la intención

```
private function isAssoc(array $arr)
{
    if ([] === $arr) {
        return false;
    }

    return array_keys($arr) !== range( low: 0, high: count($arr) - 1);
}
```

```
function obtener_listado_productos(&$parametros = array())
{
    $this->info = array();

    $res = $this->ci->productos_model->obtener_listado_productos($parametros);

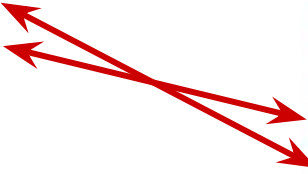
    if ($res->num_rows) {
        //...
        //....
    }
}
```

1) Sobre los nombres nombres

Evita usar varias palabras para un mismo concepto

```
Equipos_library
├── f info:array
├── f ci:CI_Base
├── f equipos:array
├── m Equipos_library():Equipos_library
├── m obtener_listado_equipos([&parametros : array
├── m obtener_equipo([&parametros : array = array())
├── m alta_equipo([&parametros : array = array()):voi
├── m eliminar_equipo([&parametros : array = array())
├── m construir_select_equipos([&parametros : array
├── m formatear_equipo([&parametros : array = array
├── m obtener_datos():array
```

```
Viajes_destinos_library
├── f id_monte
├── f ci:CI_Base
├── f es_script:bool
├── f mensaje_error:string
├── m Viajes_destinos_library():Viajes_destinos_libra
├── m asociar_viaje_destino_seleccionado_del_monte
├── m asociar_viaje_destino_producto():bool
├── m actualizar_destinos_viajes():void
├── m asociar_destino_al_monte(id_viaje, id_destino,
├── m borrar_viaje_destino(id_viaje):void
├── m crear_destino_producto():bool
├── m obtener_destinos_de_viaje(id_viaje : int):array
├── m obtener_productos_de_destinos_de_viaje(&pa
├── m obtener_viaje_destino(id_viaje : int):array
├── m asociar_viaje_destino(id_viaje : int, id_destino :
```



1) Sobre los nombres nombres

Una palabra por concepto usando estándares y convenciones

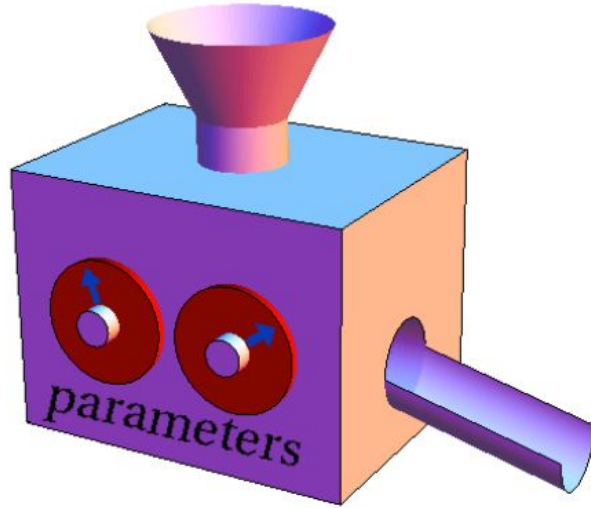
Nombres de variables

- begin / end
- locked / unlocked
- next / previous
- opened / closed
- up / down
- first / last
- min / max
- old / new
- source / target

Nombre de funciones

- add / remove
- begin / end
- create / destroy
- first / last
- show / hide
- increment / decrement
- insert / delete
- lock / unlock
- get / set
- start / stop

2) Sobre los parámetros



2) Sobre los parámetros

Usa operandos y no opciones

```
public function updateDeveloper ($name, $email, $active = false){  
    if ($active){  
        $this->developer->save($name, $email);  
    }  
    else{  
        $this->developer->save($name, '');  
    }  
}
```



2) Sobre los parámetros

Usa operandos y no opciones

```
public function updateActiveDeveloper ($name, $email){  
    $this->developer->save($name, $email);  
}
```



```
public function updateInactiveDeveloper ($name){  
    $this->developer->save($name, '');  
}
```

2) Sobre los parámetros

Objetos de dominio y 'data clumps'

```
Class Gps
{
    private $latitude = '';
    private $longitude = '';

    public function __construct($latitude, $longitude)
    {
        $this->latitude = $latitude;
        $this->longitude = $longitude;
    }

    public function move ($latitude, $longitude){
        return new self($latitude, $longitude);
    }
}
```



2) Sobre los parámetros

Objetos de dominio y 'data clumps'

```
Class Position
{
    private $latitude = '';
    private $longitude = '';

    public function __construct(string $latitude, string $longitude)
    {
        $this->latitude = $latitude;
        $this->longitude = $longitude;
    }

    public function latitude():string
    {
        return $this->latitude;
    }

    public function longitude():string
    {
        return $this->longitude;
    }

    public function isNorth():bool
    {
        //...
    }

    //...
}
```

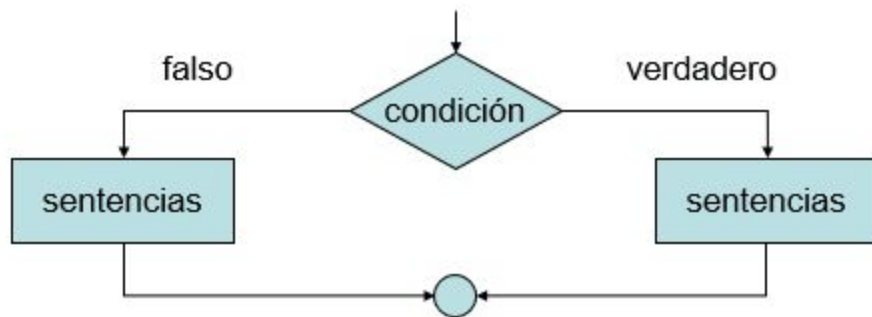
```
Class Gps
{
    private $position;

    public function __construct(Position $position)
    {
        $this->position = $position;
    }

    public function move (Position $otherPosition): Gps{
        return new self($otherPosition);
    }
}
```



3) Sobre las condiciones



3) Sobre las condiciones

Utiliza condicionales semánticos fáciles de leer y comprender

```
public function __invoke (){  
    //....  
    if ($foo == $var){  
        //...  
    }  
    //...  
}
```



```
public function __invoke (){  
    //....  
    if ($this->hasLimit()){  
        //...  
    }  
    //...  
}
```



```
private function hasLimit(){  
    return $this->foo == $this->var;  
}
```

3) Sobre las condiciones

Utiliza condicionales semánticos fáciles de leer y comprender

```
//...  
//...  
if ((($parametros['filtros']['id_cliente']) AND ($viaje['cliente_transporte']['id'] == $parametros['filtros']['id_cliente']))) {  
    return true;  
}  
//...  
//...
```



```
//...  
//...  
$filtramosPorCliente = $parametros['filtros']['id_cliente'];  
$clienteTieneTransporte = $viaje['cliente_transporte']['id'] == $parametros['filtros']['id_cliente'];  
  
if ($filtramosPorCliente AND $clienteTieneTransporte) {  
    return true;  
}  
//...  
//...
```



3) Sobre las condiciones

Un solo nivel de indentación por método

```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```



3) Sobre las condiciones

Un solo nivel de indentación por método



```
function _obtener_url_categoria() {  
    $cat_listado=$this->categoria_library->obtener_subtipo_seleccionado();  
    if (!$cat_listado) {  
        $cat_listado=$this->categoria_library->obtener_tipo_seleccionado();  
        if (!$cat_listado) {  
            $cat_listado=$this->categoria_library->obtener_subcategoria_seleccionada();  
            if (!$cat_listado) {  
                $cat_listado=$this->categoria_library->obtener_categoria_seleccionada();  
                return $cat_listado['url'];  
            } else {  
                return $cat_listado['url'];  
            }  
        } else {  
            return $cat_listado['url'];  
        }  
    } else {  
        return $cat_listado['url'];  
    }  
}
```



3) Sobre las condiciones

Un solo nivel de indentación por método: cláusulas de guarda

```
function _obtener_url_categoria()
{
    if ($categoria = $this->categoria_library->obtener_subtipo_seleccionado())
        return $categoria['url'];
    }

    if ($categoria = $this->categoria_library->obtener_tipo_seleccionado()) {
        return $categoria['url'];
    }

    if ($categoria = $this->categoria_library->obtener_subcategoria_seleccionada()) {
        return $categoria['url'];
    }

    if ($categoria = $this->categoria_library->obtener_subcategoria_seleccionada()) {
        return $categoria['url'];
    }

    $categoria = $this->categoria_library->obtener_categoria_seleccionada();
    return $categoria['url'];
}
```



3) Sobre las condiciones

Un solo nivel de indentación por método



```
function do_login()
{
    if ( ! $this->dx_auth->is_logged_in() ) {
        $val = $this->form_validation;

        if ( $val->run() AND $this->dx_auth->login($val->set_value('username'), $val->set_value('password'), $val->set_value('remember')) ) {
            $url_redireccion = $this->input->post('url');

            if ( $url_redireccion == '' ) {
                $url_redireccion = '/';
            }

            redirect($url_redireccion, method: 'location');
        } else {
            if ( $this->dx_auth->is_banned() ) {
                $this->session->set_flashdata('error_login', "El usuario ".$val->set_value('username')." ha borrado su cuenta o está baneado.");

                return false;
            } else {
                $this->session->set_flashdata('error_login', "Nombre de usuario o contraseña incorrectos.");

                return false;
            }
        }
    } else {
        redirect( uri: '/', method: 'location');
    }
}
```

3) Sobre las condiciones

Un solo nivel de indentación por método



```
function do_login()
{
    if ($this->usuarioEstaLogueado()) {
        $this->redireccionarA( url: '/');
    }

    if ($this->usuarioHaceLogin()) {
        $this->redireccionarA($this->urlRedireccion());
    }

    $this->configurarMensajeErrorLogin();

    return false;
}

private function redireccionarA(string $url){
    redirect($url, method: 'location');
}
```

```
private function usuarioEstaLogueado(){
    return $this->dx_auth->is_logged_in();
}

private function usuarioHaceLogin (){
    $val = $this->form_validation;

    return ($val->run() AND $this->dx_auth->login($val->set_value('username'), $val->set_value('password')));
}

private function urlRedireccion (){
    $url_redireccion = $this->input->post('url');

    if ($url_redireccion == '') {
        $url_redireccion = '/';
    }

    return $url_redireccion;
}

private function configurarMensajeErrorLogin(){
    if ($this->usuarioEstaBaneado()) {
        $this->session->set_flashdata('error_login', "El usuario ".$val->set_value('username')." ha sido baneado.");
    }
    else{
        $this->session->set_flashdata('error_login', "Nombre de usuario o contraseña incorrectos.");
    }
}

private function usuarioEstaBaneado(){
    return $this->dx_auth->is_banned();
}
```

4) Sobre las funciones

Aprovecha para contar una historia en cada método público...

```
class StoreResourceHandler
{
    private $command;
    private $resource;
    //..
    public function __invoke(StoreResourceCommand $command)
    {
        $this->initializeCommand($command);

        $this->fillResource();

        $this->fillSource();

        $this->create();

        $this->attachAuthorTags();

        $this->attachEventTags();

        $this->attachTechnologyTags();

        $this->attachResourceTags();
    }
}
```



4) Sobre las funciones

...pero no cuentes historias de miedo



```
function obtener_listado_montes($parametros = array())
{
    $lista = array();

    if ( ! isset($parametros['obtener_select_montes'])) {
        $parametros['obtener_select_montes'] = false;
    }

    $parametros['tipos_monte'] = $this->ci->input->post('tipos_monte');
    $parametros['energeticos'] = $this->ci->input->post('energeticos');
    $parametros['finalizados'] = $this->ci->input->post('finalizados');

    $res = $this->ci->montes_model->obtener_listado_montes($parametros);

    if ($res->num_rows) {
        $res = $res->result_array();

        foreach ($res AS $monte) {
            $this->formatear_datos_monte($monte);

            if ($monte['id_monte_gerencia']) {
                $monte['monte_gerencia'] = $this->ci->monte_gerencia_model->obtener_uno($monte['id_monte_gerencia']);
            }

            if (isset($parametros['obtener_tipo_monte'])) {
                $monte['tipo_monte'] = $this->obtener_tipo_monte($monte['id_monte']);
                $monte['es_proveedor'] = (isset($monte['tipo_monte'][ID_MONTE_PROVEEDOR])) ? true : false;
                $monte['es_parque'] = (isset($monte['tipo_monte'][ID_MONTE_TIPO_PARQUE])) ? true : false;
            }

            $monte['proveedores'] = $this->ci->montes_proveedores_library->obtener_proveedores_de_un_monte($monte['id_monte']);
            $lista[$monte['id_monte']] = $monte;
        }
    }

    if ($parametros['energeticos']) {
        foreach ($lista AS $id => $monte) {
            $lista[$id]['info_cultivo'] = $this->obtener_cultivo_energetico($monte['id_monte']);
            $lista[$id]['info_cultivo']['toneladas_extraidas'] = $this->ci->viajes_model->return_peso_viajes_energeticos($lista[$id]['info_cultivo']['id_cultivo']);
        }
    }

    $parametros['listado_montes'] = $lista;
    $this->info = $lista;

    if ($parametros['obtener_select_montes']) {
        $this->construir_select_montes($parametros);
    }

    return $this->info;
}
```

4) Sobre las funciones

Pequeñas, menos es más



```
public function __invoke(UpdateSubscriptionAccessCommand $command)
{
    $this->initialize($command);
    $this->configureStatus();
    $this->obtainSubscription();
    $this->checkCompanyHasActiveSubscription();
    $this->obtainCompany();
    $this->updateStatusAccessKey();
}

private function initialize(UpdateSubscriptionAccessCommand $command)
{
    $this->command = $command;
}

private function configureStatus()
{
    $this->status = $this->accessKeyStatusFactory->buildStatusFromString($this->command->status());
}

private function obtainSubscription()
{
    $this->subscription = Subscription::findOrFail($this->command->subscriptionId());
}

private function checkCompanyHasActiveSubscription()
{
    if (! $this->subscription->isActive(Carbon::now())) {
        throw new SubscriptionIsNotActiveException();
    }
}

private function obtainCompany()
{
    $this->company = $this->subscription->company;
}
```

5) Sobre los principios

Evita las sorpresas: Principio de la Mínima Sorpresa



```
private function associateUser(CreateCompanyCommand $command)
{
    $this->user = $this->company->users()->create([
        'username' => $command->getUsername(),
        'email' => $command->getUserEmail(),
        'name' => $command->getName(),
        'surname_first' => $command->getSurnameFirst(),
        'surname_second' => $command->getSurnameSecond(),
        'gender' => $command->getGender(),
        'password' => $command->getPassword(),
        'old_id' => null,
    ]);

    $this->user->assignRole(Role::ROLE_COMPANY);

    activity()->causedBy($this->user)->log(trans( key: 'activity.user_register'));

    $this->configUserVerification();

    event(new Registered($this->user));
}
```


5) Sobre los principios

Evita las sorpresas: Principio de la Mínima Sorpresa



```
public function __invoke(CreateCompanyCommand $command)
{
    $this->createUser($command);

    $this->assignRole();

    $this->logActivity();

    $this->configUserVerification();

    $this->throwEvent();
}

private function createUser(CreateCompanyCommand $command)|)
{
    $this->user = $this->company->users()->create([
        'username' => $command->getUsername(),
        'email' => $command->getUserEmail(),
        'name' => $command->getName(),
        'surname_first' => $command->getSurnameFirst(),
        'surname_second' => $command->getSurnameSecond(),
        'gender' => $command->getGender(),
        'password' => $command->getPassword(),
        'old_id' => null,
        'uuid' => Uuid::generate()->string,
    ]);
}

private function assignRole (){
    $this->user->assignRole(Role::ROLE_COMPANY);
}

private function logActivity (){
    activity()->causedBy($this->user)->log(trans( key: 'activity.user_register'));
}

private function throwEvent (){
    event(new Registered($this->user));
}
```

5) Sobre los principios

Evita las sorpresas: Principio de la Mínima Sorpresa



```
public function addFakeRegisterInFiles($slrFilePath, $fileName, $fakeRegisters) {  
    shell_exec( cmd: 'rm -rf ' . $this->getPath().$fileName.'*');  
    shell_exec( cmd: 'mkdir ' . $this->getPath().$fileName);  
    shell_exec( cmd: 'cp ' . $this->getPath().$fileName.'file[1-5].txt ' . $this->getPath().$fileName);  
  
    $file = 1;  
    foreach ($fakeRegisters as $fake) {  
        $line = $this->generateLineFakeRegister($fake).$this->addBreakLine();  
        File::append($this->getPath().$fileName.'/file'.$file.'.txt', $line);  
        if(file_exists ( filename: $this->getPath().$fileName.'/file' . ($file+1) . '.txt' )) {  
            $file++;  
        }  
    }  
  
    shell_exec( cmd: 'cat ' . $this->getPath().$fileName.'/file[1-5].txt >> ' . $slrFilePath);  
    shell_exec( cmd: 'rm -rf ' . $this->getPath().$fileName.'/*');  
}
```

5) Sobre los principios

Evita las sorpresas: Principio de la Mínima Sorpresa



```
public function addFakeRegisterInFiles($slrFilePath, $fileName, $fakeRegisters) {  
    $this->clearFolder($fileName);  
    $this->createFolder($filename);  
    $this->copyFiles($filename);  
    $this->addFakeRegisterInFiles($fileName, $fakeRegisters);  
    $this->writeFile($filename);  
    $this->clearFolder($fileName);  
}  
  
private function clearFolder (string $filename){  
    return shell_exec( cmd: 'rm -rf ' . $this->getPath().$fileName.'*');  
}  
  
private function createFolder (string $filename){  
    return shell_exec( cmd: 'mkdir ' . $this->getPath().$fileName);  
}  
  
private function copyFiles (string $filename){  
    return shell_exec( cmd: 'cp ' . $this->getPath().$file[1-5].txt ' . $this->getPath().$fileName);  
}  
  
private function addFakeRegisters ($filename, $fakeRegisters){  
    $file = 1;  
    foreach ($fakeRegisters as $fake) {  
        $line = $this->generateLineFakeRegister($fake).$this->addBreakLine();  
        File::append($this->getPath().$fileName.'/file.'.$file.'.txt', $line);  
        if(file_exists ( filename: $this->getPath().$fileName.'/file'.($file+1).'.txt' )) {  
            $file++;  
        }  
    }  
}  
  
private function writeFile (string $filename, string $slrFilePath){  
    return shell_exec( cmd: 'cat ' . $this->getPath().$fileName.'/file[1-5].txt >> ' . $slrFilePath);  
}
```

5) Sobre los principios

Evita las sorpresas: Principio de la Mínima Sorpresa



```
interface Filesystem {  
  
    public function clearFolder (string $path);  
    public function createFolder (string $path);  
    public function writeFile (string $path, string $filename);  
    //..  
}
```

```
class FilesystemGateway implements Filesystem{  
    public function clearFolder($path)  
    {  
        //...local  
    }  
  
    public function createFolder($path)  
    {  
        //...local  
    }  
  
    public function writeFile($path, $name)  
    {  
        //...local  
    }  
}
```

```
class S3Gateway implements Filesystem{  
    public function clearFolder($path)  
    {  
        //...Amazon  
    }  
  
    public function createFolder($path)  
    {  
        //...Amazon  
    }  
  
    public function writeFile($path, $name)  
    {  
        //...Amazon  
    }  
}
```

5) Sobre los principios

Evita las sorpresas: Principio de la Mínima Sorpresa



```
class Foo {  
  
    private $filesystem;  
  
    public function __construct(Filesystem $filesystem)  
    {  
        $this->filesystem = $filesystem;  
    }  
  
    public function addFakeRegisterInFiles($slrFilePath, $fileName, $fakeRegisters) {  
  
        $this->clearFolder($fileName);  
  
        $this->createFolder($filename);  
  
        $this->copyFiles($filename);  
  
        $this->addFakeRegisterInFiles($fileName, $fakeRegisters);  
  
        $this->writeFile($filename);  
  
        $this->clearFolder($fileName);  
    }  
  
    private function crearFolder (string $path){  
        $this->filesystem->clearFolder($path);  
    }  
    //..  
}
```

6) Sobre los comentarios

El código es el único que siempre dice la verdad



```
/**
 * @return mixed
 */
public function getCreatedAt()
{
    if (empty($this->createdAt)) {
        return Carbon::now();
    }

    return $this->createdAt;
}
```



```
/*
 * Construir un select con el listado de matriculas de un viaje
 * Entrada
 * $parametros['listado_matriculas']
 * $parametros['nombre_options']
 */

function es_post_viaje_conductor()
{
    if ($_POST['viaje_conductor']) {
        return true;
    }

    return false;
}
```

6) Sobre los comentarios

El código es el único que siempre dice la verdad



```
foreach ($res AS $producto) {  
  
    // Obtener viajes desde este monte-producto para ver las toneladas que se llevan extraídas.  
    $data['id_monte_producto'] = $producto['id_registro'];  
    $producto['tn_extraídas'] = $this->ci->viajes_model->obtener_toneladas_extraídas($producto['id_registro']);  
  
    // Formatear datos del monte-producto  
    $this->formatear_datos_producto_monte($producto);  
  
    // Obtener datos del producto origen  
    $producto['producto_origen'] = array();  
    if ($producto['id_producto']) {  
        $producto['producto_origen'] = $this->ci->productos_library->obtener_producto($producto);  
    }  
  
    // Obtener datos del destino  
    if ($producto['id_destino_producto']) {  
        $datos = $this->ci->destinos_library->obtener_datos_destino_producto($producto);  
        $producto['datos_destino'] = $datos['datos_destino'];  
        $producto['producto_destino'] = $datos['producto_destino'];  
    }  
}
```


6) Sobre los comentarios

El código es el único que siempre dice la verdad



```
foreach ($res AS $producto) {  
    $this->obtenerViajes();  
    $this->obtenerMonte();  
    $this->obtenerProducto();  
    $this->obtenerDestino();  
}
```


7) Simetría

El código debe parecerse en cualquier parte del proyecto

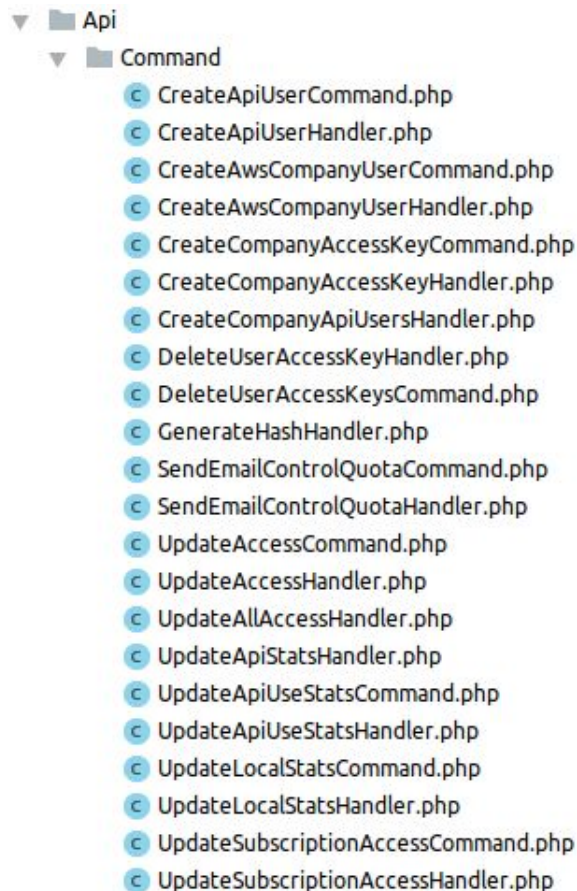
Simetría sintáctica

Simetría semántica

Simetría sistémica

7) Simetría

Simetría sistémica



7) Simetría

Simetría sistémica

```
public function __invoke(UpdateApiUseStatsCommand $command)
{
    $this->initialize($command);
    $this->checkDayHasNotBeenProcessed();
    $this->downloadLogs();
    $this->processLogs();
    $this->beginTransaction();
    $this->resetStatsByDay();
    $this->writeStatsInLocal();
    $this->endTransaction();
    $this->writeStatsInApi();
    $this->markProcessAsSuccessful();
}
```

```
public function __invoke(CreateAwsCompanyUserCommand $command)
{
    $this->initialize($command);
    $this->generateUsername();
    $this->checkUsernameExists();
    $this->createApiUserInAws();
    $this->generateApiUserInSlr();
    $this->attachToGroup();
}
```

```
public function __invoke(UpdateSubscriptionAccessCommand $command)
{
    $this->initialize($command);
    $this->configureStatus();
    $this->obtainSubscription();
    $this->checkCompanyHasActiveSubscription();
    $this->obtainCompany();
    $this->updateStatusAccessKey();
}
```

De tu parte queda...

Entrenar

De tu parte queda...

Entrenar
Concentrarse

De tu parte queda...

Entrenar
Concentrarse
Ser consciente

De tu parte queda...

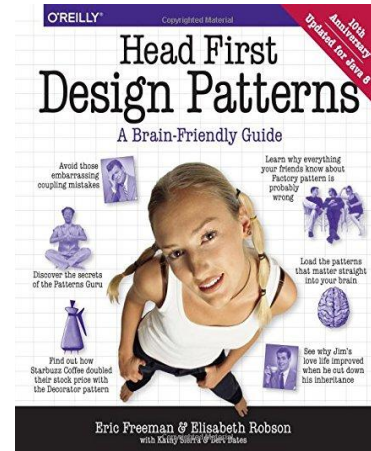
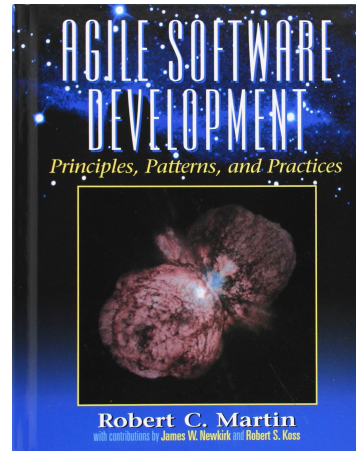
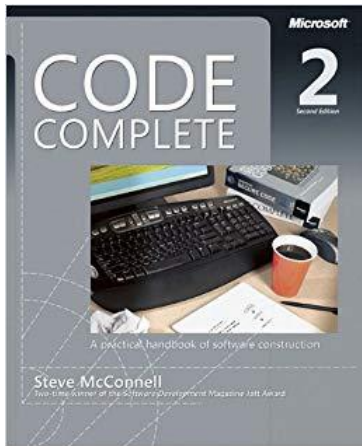
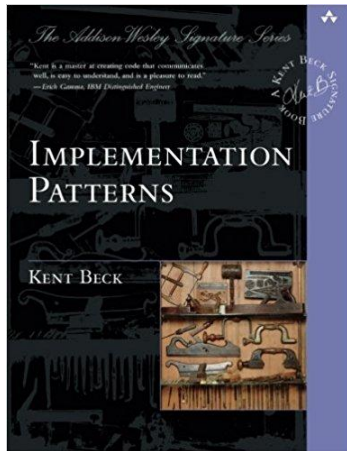
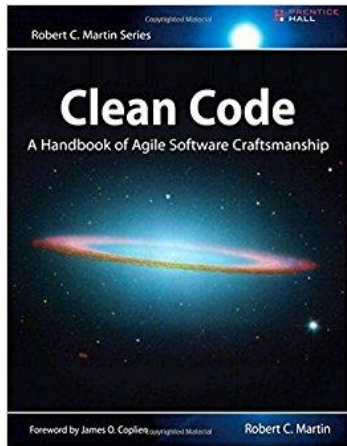
Entrenar

Concentrarse

Ser consciente

Ser responsable

Bibliografía recomendada



Ahora te toca a ti

¿preguntas? ¿comentarios? ¿sugerencias?

Fin

Gracias por vuestra atención.