

Creating config file /etc/php/7.4/mods-available/xmlreader.ini with new version

Creating config file /etc/php/7.4/mods-available/xmlwriter.ini with new version

Creating config file /etc/php/7.4/mods-available/xsl.ini with new version

Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.4) ...

Processing triggers for php7.4-cli (7.4.3-4ubuntu2.4) ...

monolinux@deadpool: /mnt/c/Workspaces\$ rm appstarter/

rm: cannot remove 'appstarter/': Is a directory

monolinux@deadpool: /mnt/c/Workspaces\$ rm -R appstarter/

monolinux@deadpool: /mnt/c/Workspaces\$ sudo composer create-project codeigniter4/appstarter

Do not run Composer as root/super user! See <https://getcomposer.org/root> for details

Creating a "codeigniter4/appstarter" project at "../appstarter"

Warning from <https://repo.packagist.org>: You are using an outdated version of Composer. Composer 2 is now available and you should upgrade. See <https://getcomposer.org>

Installing codeigniter4/appstarter (v4.1.1)

- Installing codeigniter4/appstarter (v4.1.1): Loading from cache

Created project in /mnt/c/Workspaces/appstarter

Loading composer repositories with package information

Warning from <https://repo.packagist.org>: You are using an outdated version of Composer. Composer 2 is now available and you should upgrade. See <https://getcomposer.org>

Updating dependencies (including require-dev)

Package operations: 41 installs, 0 updates, 0 removals

- Installing psr/log (1.1.3): Downloading (100%)
- Installing laminas/laminas-zendframework-bridge (1.2.0): Downloading (100%)
- Installing laminas/laminas-escaper (2.7.0): Downloading (100%)
- Installing kint-php/kint (3.3): Downloading (100%)
- Installing codeigniter4/framework (v4.1.1): Downloading (100%)
- Installing fakerphp/faker (v1.13.0): Downloading (100%)
- Installing mikey179/vfsstream (v1.6.8): Downloading (100%)
- Installing sebastian/version (3.0.2): Downloading (100%)
- Installing sebastian/type (2.3.1): Downloading (100%)
- Installing sebastian/resource-operations (3.0.3): Downloading (100%)
- Installing sebastian/recursion-context (4.0.4): Downloading (100%)
- Installing sebastian/object-reflector (2.0.4): Downloading (100%)
- Installing sebastian/object-enumerator (4.0.4): Downloading (100%)
- Installing sebastian/global-state (5.0.2): Downloading (100%)
- Installing sebastian/exporter (4.0.3): Downloading (100%)
- Installing sebastian/environment (5.1.3): Downloading (100%)
- Installing sebastian/diff (4.0.4): Downloading (100%)
- Installing sebastian/comparator (4.0.6): Downloading (100%)
- Installing sebastian/code-unit (1.0.8): Downloading (100%)
- Installing sebastian/cli-parser (1.0.1): Downloading (100%)
- Installing phpunit/php-timer (5.0.3): Downloading (100%)
- Installing phpunit/php-text-template (2.0.4): Downloading (100%)

# CURSO DE PHP

Con la técnica Blitzkrieg Bop



# TÁCTICA

Reuniendo a los mejores

# 1

Costos

Soporte

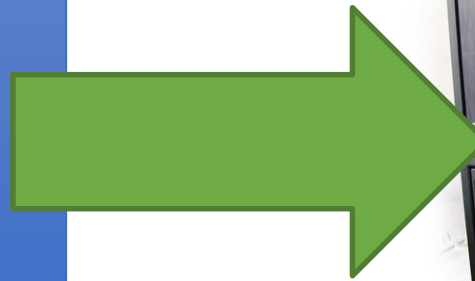
Documentación

SOFTWARE

Versiones

Comunidades

Licencias



ENTORNOS DE TRABAJO UNA SIMBIOSIS CON NUESTROS PROYECTOS |  
[La trinchera divergente \(junglancode.org\)](http://La trinchera divergente (junglancode.org))

# QUE SE INSTALARA ?

- LAMP
- VsCodium
- Composer
- Git
- Codeigniter
- MyCli/Dbeaver

- ~~Visual Studio~~ ó
- ~~VsCodium~~  
~~Apache~~
- **Mysql / MariaDB**
- **PHP**

[COMPOSER, PROFESIONALIZANDO A PHP | La trinchera divergente \(junglancode.org\)](#)

[CODEIGNITER 4.0, PENSEMOS EN EL FUTURO | La trinchera divergente \(junglancode.org\)](#)

<https://blog.junglancode.org/memorias/anecdotas/base-de-datos-con-mycli/>

# HISTORY

## • LAMP COMMANDS

- `sudo apt install lamp-server^`

- <https://askubuntu.com/questions/785440/how-to-install-lamp-server-on-ubuntu>

## • Vscodium

- `wget -qO - https://gitlab.com/paulcarroty/vscodium-deb-rpm-repo/raw/master/pub.gpg | gpg --dearmor | sudo dd of=/etc/apt/trusted.gpg.d/vscodium.gpg`
- `echo 'deb https://paulcarroty.gitlab.io/vscodium-deb-rpm-repo/debs/ vscodium main' | sudo tee --append /etc/apt/sources.list.d/vscodium.list`
- `sudo apt update && sudo apt install codium`

- <https://vscodium.com/>

## • Git

- `sudo apt install git`

- <https://git-scm.com/>

## • Composer

- `sudo apt install composer`

- <https://getcomposer.org/>

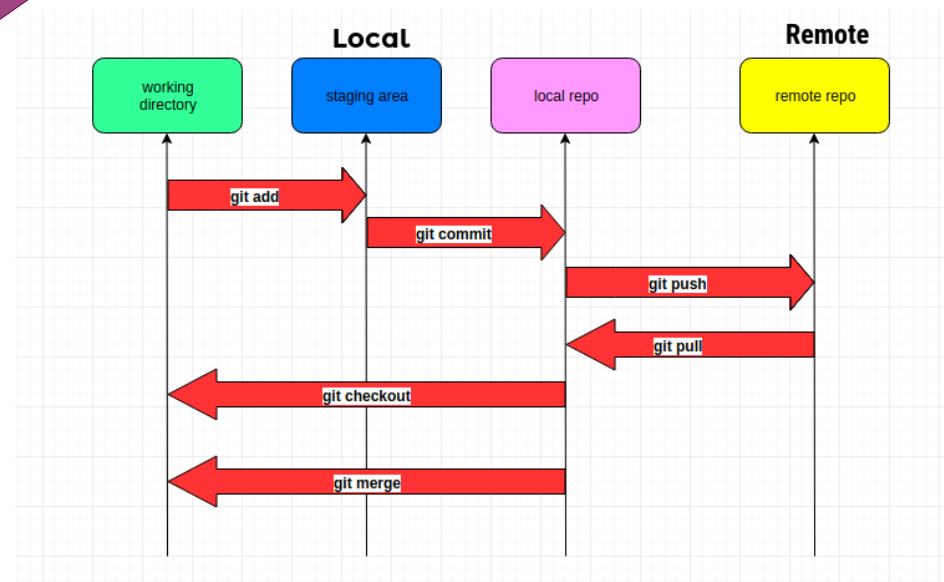
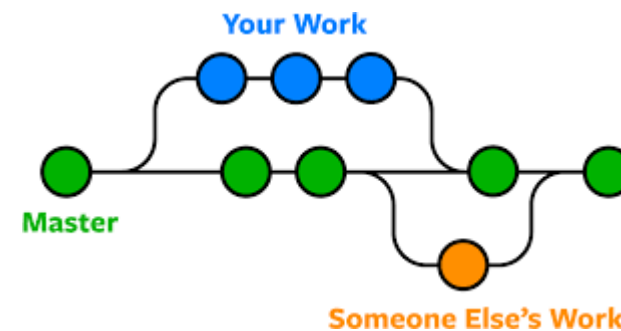
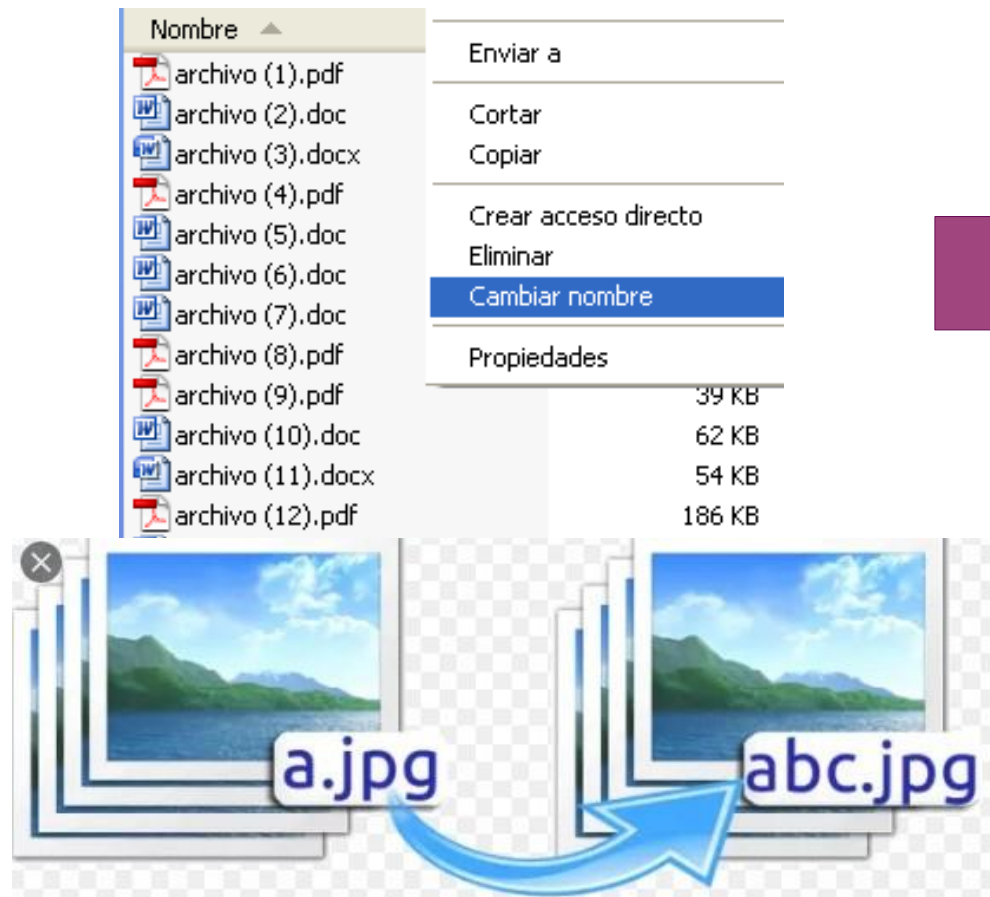
## • Codeigniter

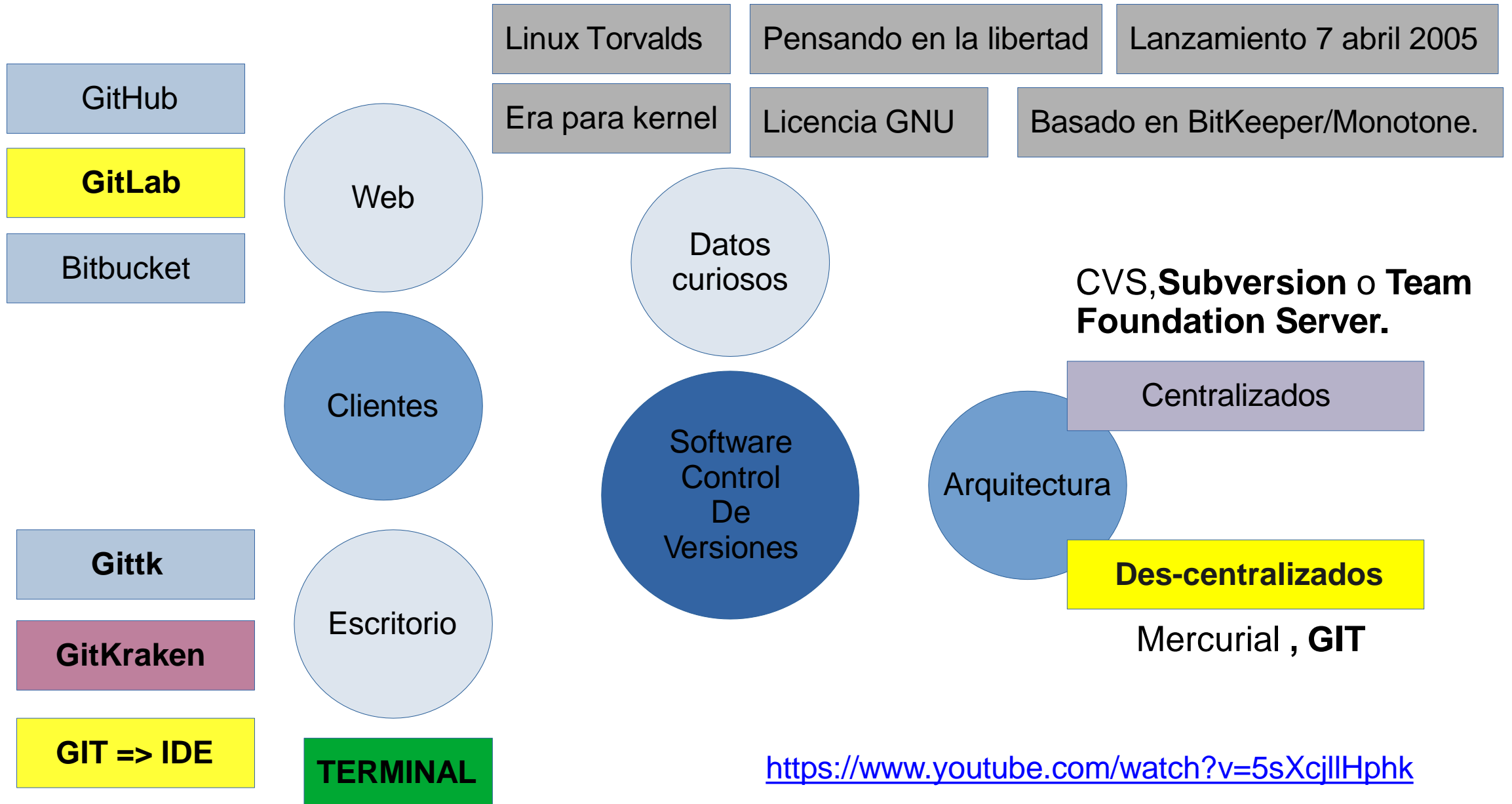
- `sudo composer create-project codeigniter4/appstarter project-root`

- [https://codeigniter.com/user\\_guide/installation/index.htm](https://codeigniter.com/user_guide/installation/index.htm)

# TÁCTICA

## Salvaguardando nuestro legado # 2









GitLab



GitHub



● Missing in GitHub

	Manage	5.5/8	<div><div></div></div>	4/8	<div><div></div></div>	<ul style="list-style-type: none"><li>Subgroups</li><li>Audit Events</li><li>Audit Reports</li><li>Compliance Management</li></ul>	<ul style="list-style-type: none"><li>Code Analytics</li><li>DevOps Reports</li><li>Value Stream Management</li><li>Insights</li></ul>
	Plan	6/8	<div><div></div></div>	1/8	<div><div></div></div>	<ul style="list-style-type: none"><li>Issue Tracking</li><li>Kanban Boards</li><li>Time Tracking</li><li>Epics</li></ul>	<ul style="list-style-type: none"><li>Roadmaps</li><li>Service Desk</li><li>Requirements Management</li><li>Quality Management</li></ul>
	Create	7.5/8	<div><div></div></div>	6/8	<div><div></div></div>	<ul style="list-style-type: none"><li>Source Code Management</li><li>Code Review</li><li>Wiki</li><li>Static Site Editor</li></ul>	<ul style="list-style-type: none"><li>Web IDE</li><li>Live Preview</li><li>Snippets</li><li>Design Management</li></ul>
	Verify	6/8	<div><div></div></div>	1/8	<div><div></div></div>	<ul style="list-style-type: none"><li>Continuous Integration</li><li>Code Quality</li><li>Code Testing and Coverage</li><li>Load Testing</li></ul>	<ul style="list-style-type: none"><li>Web Performance</li><li>Usability Testing</li><li>Accessibility Testing</li><li>Merge Trains</li></ul>
	Package	4.5/6	<div><div></div></div>	2/7	<div><div></div></div>	<ul style="list-style-type: none"><li>Package Registry</li><li>Container Registry</li><li>Helm Chart Registry</li><li>Dependency Proxy</li></ul>	<ul style="list-style-type: none"><li>Jupyter Notebooks</li><li>Git LFS</li><li>Dependency Firewall</li></ul>
	Secure	7/8	<div><div></div></div>	4/8	<div><div></div></div>	<ul style="list-style-type: none"><li>SAST</li><li>DAST</li><li>Fuzz Testing</li><li>Dependency Scanning</li></ul>	<ul style="list-style-type: none"><li>Container Scanning</li><li>License Compliance</li><li>Secret Detection</li><li>Vulnerability Management</li></ul>
	Release	7/8	<div><div></div></div>	4/8	<div><div></div></div>	<ul style="list-style-type: none"><li>Continuous Delivery</li><li>Pages</li><li>Review Apps</li><li>Advanced Deployments</li></ul>	<ul style="list-style-type: none"><li>Feature Flags</li><li>Release Orchestration</li><li>Release Evidence</li><li>Secrets Management</li></ul>
	Configure	4.5/7	<div><div></div></div>	1/7	<div><div></div></div>	<ul style="list-style-type: none"><li>Auto DevOps</li><li>Kubernetes Configuration</li><li>ChatOps</li><li>Runbooks</li></ul>	<ul style="list-style-type: none"><li>Serverless</li><li>Infrastructure as Code</li><li>Cluster Cost Optimization</li></ul>
	Monitor	5/8	<div><div></div></div>	1/8	<div><div></div></div>	<ul style="list-style-type: none"><li>Metrics</li><li>Alert Management</li><li>Incident Management</li><li>Logging</li></ul>	<ul style="list-style-type: none"><li>Tracing</li><li>Error Tracking</li><li>Product Analytics</li><li>Synthetic Monitoring</li></ul>
	Defend	1.5/3	<div><div></div></div>		<div></div>	<ul style="list-style-type: none"><li>Web Application Firewall</li><li>Container Host Security</li><li>Container Network Security</li></ul>	

## INSTALACION & COMANDOS

Estos podrían ser algunos de los comando más usados por los GitUser de la actualidad, haciendo de tu código más limpio y fácil de ejecutar con la herramienta de Git y Github.

GitHub para Windows

<https://windows.github.com>

GitHub para Mac

<https://mac.github.com>

Git para todas las Plataformas

<http://git-scm.com>

## SETUP

Configuración de la información del usuario la cual se utiliza en todos los repositorios locales

**git config --global user.name "[firstname lastname]"**

Establece un nombre que sea identificable para los crédito en los repositorios

**git config --global user.email "[valid-email]"**

Configurar una dirección de correo electrónico que se asociará con cada marcador de la historia

## SETUP & INIT

Configuración de la información del usuario, inicialización y clonar repositorios

**git init**

Inicializar un directorio para un repositorio Git

**git clone [url]**

Recuperar todo un repositorio desde un repositorio de Github a través de la URL

## STAGE & SNAPSHOT

Trabajar con instantáneas y el área de trabajo de Git

**git status**

Muestra las modificaciones de los archivos en el directorio de trabajo, organizando su próximo commit

**git add [file]**

Agregar un archivo para su próxima confirmación en el stage

**git reset [file]**

Quita un archivo sin perder los cambios en el directorio de trabajo

**git diff**

Muestra lo que está cambiado, pero no por etapas

**git commit -m "[mensaje descriptivo del commit]"**

Compromete su contenido en escena como un nuevo commit de instantáneas

## BRANCH & MERGE

Aislar el trabajo en las ramas, el cambio de contexto, y la integración de los cambios

**git branch**

En lista sus ramas creadas, aparecerá junto a la rama activa

**git branch [branch-name]**

Crea una nueva rama en el actual repositorio

**git checkout [branch-name]**

Cambiar a otra rama de trabajo y comprobar que en el directorio de trabajo

**git merge [branch]**

Fusiona la historia de la rama especificada en el actual

**git log**

Muestra todas las confirmaciones en la historia de las ramas actuales

## INSPECT & COMPARE

Examina los registros, diffs e información de objeto

**git log branchB...branchA**

Muestra las confirmaciones en branchA que no están en BranchB

**git log --follow [file]**

Muestra las confirmaciones que cambiaron los archivos, incluso a través de los cambios de nombre

**git diff branchB...branchA**

Muestra el diff de lo que es en branchA que no está en BranchB

**git show**

Muestra cualquier objeto en Git en formato legible por humanos

# ADD WORDS IN MY MIND Repositorios

Clonar

Push baby

Avientame el Pull

## Ramas

## Focking Merge

Issues

Features

## Releases

## Hotfix

Por que C\$"!%& estas en el  
master Cual es tu sha

Ya la cagamos regresa el

Commit





# TÁCTICA

## # 3

Conociendo el terreno



Mar 2021	Mar 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	15.33%	-1.00%
2	1	▼	Java	10.45%	-7.33%
3	3		Python	10.31%	+0.20%
4	4		C++	6.52%	-0.27%
5	5		C#	4.97%	-0.35%
6	6		Visual Basic	4.85%	-0.40%
	7		JavaScript	2.11%	+0.06%
	8		PHP	2.07%	+0.05%

Versión	Fecha de lanzamiento	Fin de soporte
7.2	30-11-2017	30-11-2020
7.3	6-12-2018	6-12-2021
7.4	28-11-2019	28-11-2022
8.0	26-11-2020	26-11-2023

<https://www.similartech.com/technologies/php>

<https://www.tiobe.com/tiobe-index/>

AHUEVO



# EL PANORAMA

3.- SESSIONES , GET , POST

2.- ESTRUCTURA DE CONTROL

4.- REQUIRES / INCLUDES

1.- COMO EJECUTARLO

**PHP**

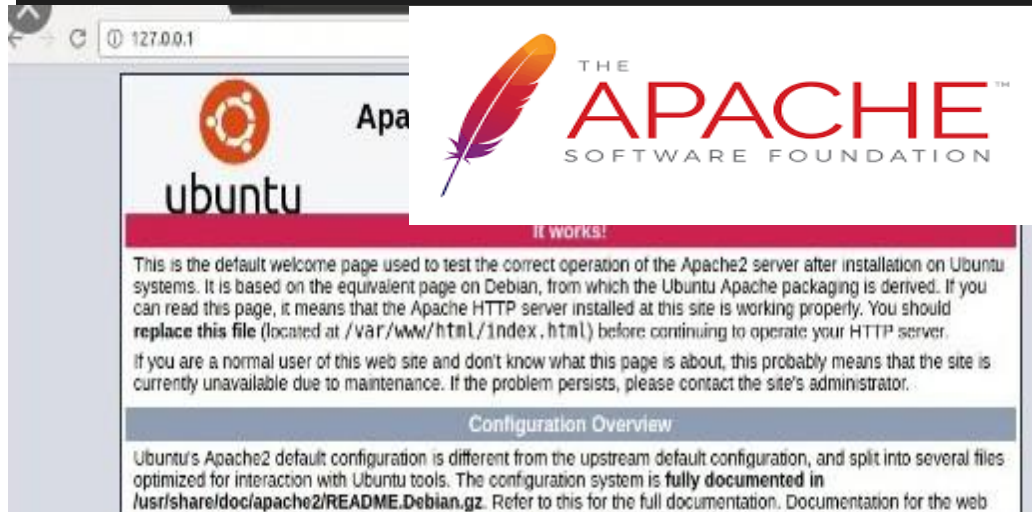
5.- ORIENTADA A OBJETOS

6.- MODELO – VISTA - CONTROLADOR

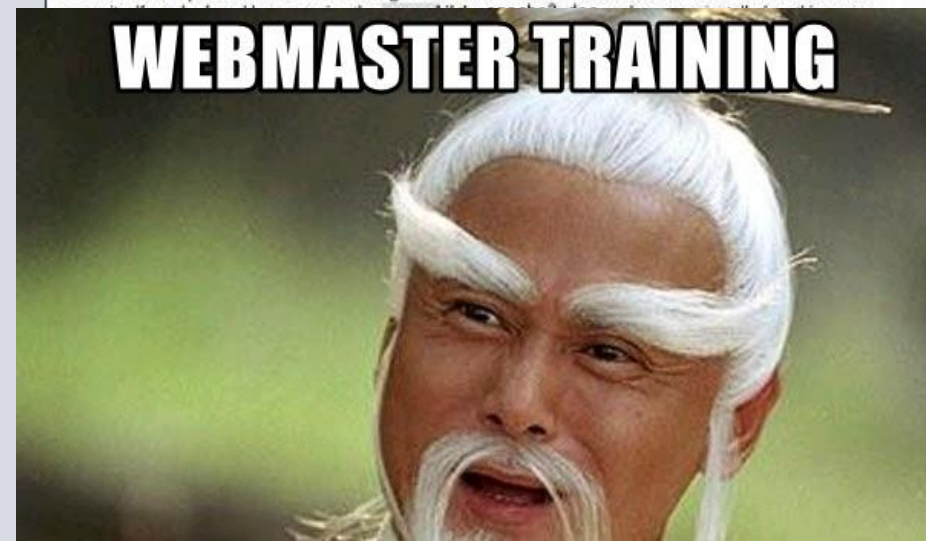
# COMO EJECUTO ESTA \$\_!&=12?

## LENGUAJE

```
jluís@localhost:~  
[jluís@localhost ~]$ php -f hola.php  
hola este es un script ejecutado desde[jluís@localhost ~]$
```



```
jluís@localhost:~ — sudo php -S localhost:666  
[jluís@localhost ~]$ sudo php -S localhost:666  
[Thu Mar 11 10:32:00 2021] PHP 7.4.15 Development Server (http://localhost:666) st
```



<https://parzibyte.me/blog/2018/05/18/ejecutar-codigo-php-desde-terminal/>

<https://httpd.apache.org/>

<https://blog.junglancode.org/memorias/php-servidor-web-embebido/>

# COMO \$ \_ ! & = 1 2 ? CONTROLLO ESTE LENGUAJE

1. if	9. continue
2. else	10. switch
3. elseif/else if	11. declare
4. while	12. return
5. do-while	13. include/include_once
6. for	14. require/require_once
7. foreach	15. goto
8. break	16. Sintaxis alternativas

[https://www.youtube.com/watch?v=svF4up3eT\\_Q](https://www.youtube.com/watch?v=svF4up3eT_Q)

[https://www.youtube.com/watch?v=XCG07zjDM\\_M](https://www.youtube.com/watch?v=XCG07zjDM_M)



# Variables predefinidas

PHP proporciona una gran cantidad de variables predefinidas para todos los scripts. Las variables representan de todo, desde [variables externas](#) hasta variables de entorno incorporadas, desde los últimos mensajes de error hasta los últimos encabezados recuperados.

## Tabla de contenidos

- [Superglobals](#) — Superglobals son variables internas que están disponibles siempre en todos los ámbitos
- [\\$GLOBALS](#) — Hace referencia a todas las variables disponibles en el ámbito global
- [\\$\\_SERVER](#) — Información del entorno del servidor y de ejecución
- [\\$\\_GET](#) — Variables HTTP GET
- [\\$\\_POST](#) — Variables POST de HTTP
- [\\$\\_FILES](#) — Variables de subida de ficheros HTTP
- [\\$\\_REQUEST](#) — Variables HTTP Request
- [\\$\\_SESSION](#) — Variables de sesión
- [\\$\\_ENV](#) — Variables de entorno
- [\\$\\_COOKIE](#) — Cookies HTTP
- [\\$php\\_errormsg](#) — El mensaje de error anterior
- [\\$http\\_response\\_header](#) — Encabezados de respuesta HTTP
- [\\$argc](#) — El número de argumentos pasados a un script
- [\\$argv](#) — Array de argumentos pasados a un script

## Variables predefinidas

### Superglobals

[\\$GLOBALS](#)

[\\$\\_SERVER](#)

[\\$\\_GET](#)

[\\$\\_POST](#)

[\\$\\_FILES](#)

[\\$\\_REQUEST](#)

» [\\$\\_SESSION](#)

[\\$\\_ENV](#)

[\\$\\_COOKIE](#)

[\\$http\\_response\\_header](#)

[\\$argc](#)

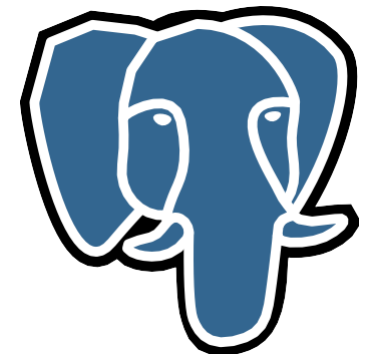
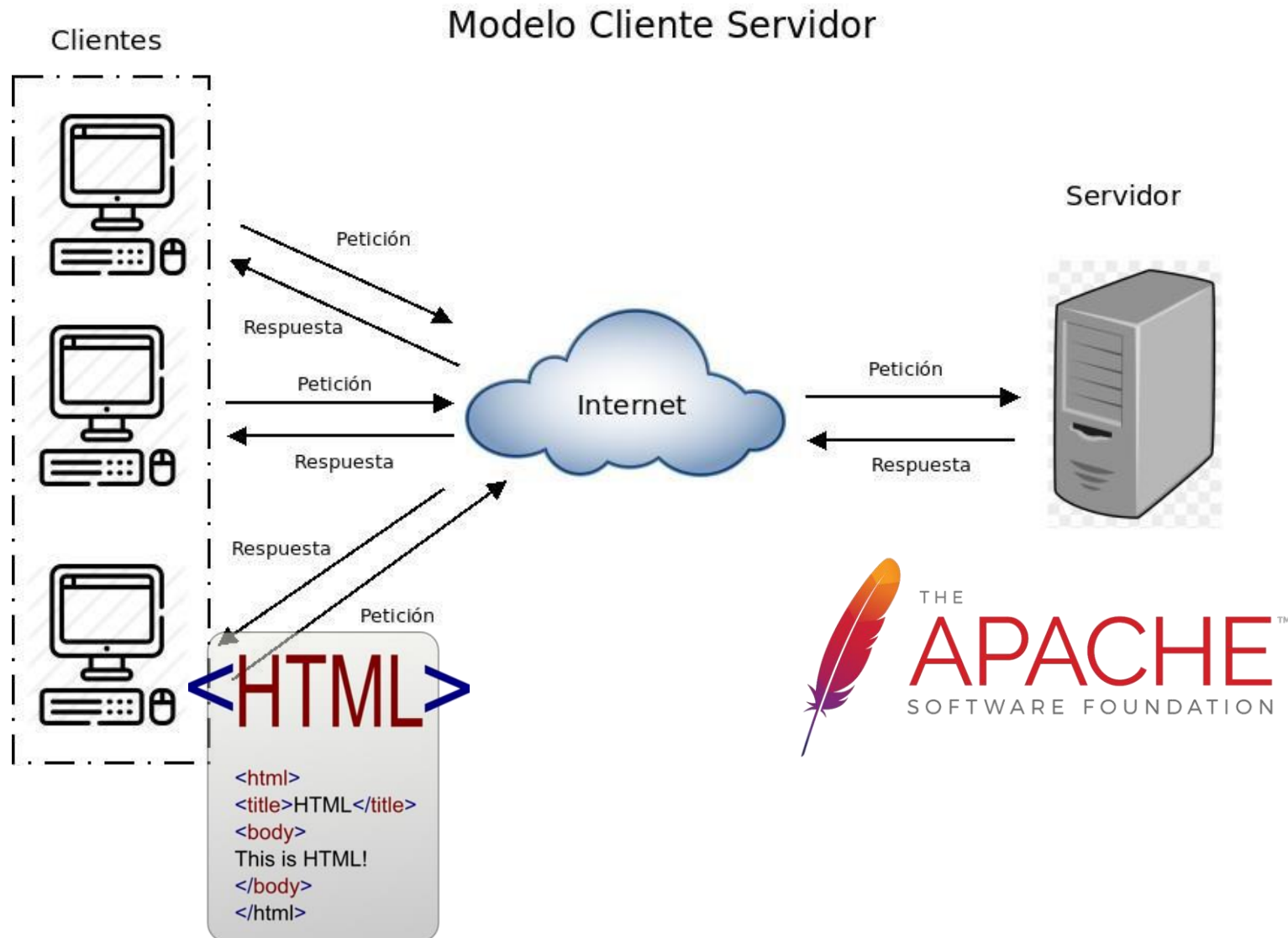
[\\$argv](#)

### Deprecated

[\\$php\\_errormsg](#)



# Entendamos como funciona este desmadre





# PROTOCOLO HTTP y HTTPS.

## NAVEGADOR



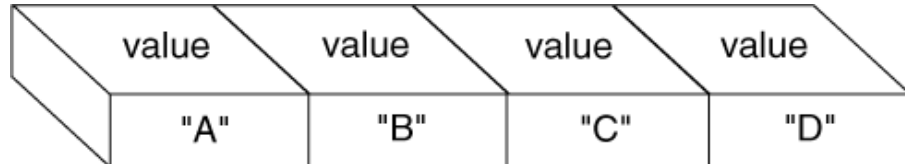
## GET

[www.ejemplo.com/index.htm?key1=value1&key2=value2&key3=value3...](http://www.ejemplo.com/index.htm?key1=value1&key2=value2&key3=value3...)

## POST

[www.ejemplo.com/index.htm](http://www.ejemplo.com/index.htm)

Viaja la información oculta



Associative Array

[https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)

<https://www.php.net/manual/es/reserved.variables.get.php>

<https://www.php.net/manual/es/reserved.variables.post.php>

<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>

# Ejemplo Practico

## Contact Form

Name: \*

First nameLast name

Email: \*

Phone: \*

Enviar Datos

Enviar Resultados

```
role_id' => $role_details['id'],
'resource_id' => $resource_details['id'],
);
if ( $this->rule_exists( $resource_details['id'], $role_details['id'] ) ) {
    if ( $access == false ) {
        // Remove the rule as there is currently no need for it
        $details['access'] = ! $access;
        $this->sql->delete( 'acl_rules', $details );
    } else {
        // Update the rule with the new access value
        $this->sql->update( 'acl_rules', array( 'access' => $access ) );
    }
}
foreach( $this->rules as $key=>$rule ) {
    if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {
        if ( $access == false ) {
            unset( $this->rules[ $key ] );
        } else {
            $this->rules[ $key ]['access'] = $access;
        }
    }
}
```

# SESIONES

Las **cookies sesión** son una **forma sencilla de almacenar datos** para usuarios de manera individual usando un **ID de sesión único**. Esto se puede usar para hacer **persistente la información** de estado entre ***peticiones de páginas..***

Y el lugar donde se almacenas la información es en un **ARRAY ASOCIATIVO** LLAMADO **\$\_SESSIONS**

```
session_start();  
$_SESSION[ 'username' ] = 'clevertechie';  
$_SESSION[ 'role' ] = 'admin';
```

[PHP: Introducción - Manual](#)

[PHP: Funciones de sesión - Manual](#)

[PHP: Sesiones y seguridad - Manual](#)

# INCLUDE/REQUIRE

## require

---

(PHP 4, PHP 5, PHP 7, PHP 8)

`require` es idéntico a [include](#) excepto que en caso de fallo producirá un error fatal de nivel **E\_COMPILE\_ERROR**. En otras palabras, éste detiene el script mientras que [include](#) sólo emitirá una advertencia (**E\_WARNING**) lo cual permite continuar el script.

Véase la documentación de [include](#) para más información.

## include

---

(PHP 4, PHP 5, PHP 7, PHP 8)

La sentencia `include` incluye y evalúa el archivo especificado.

La siguiente documentación también se aplica a [require](#).

Los archivos son incluidos con base en la ruta de acceso dada o, si ninguna es dada, el [include\\_path](#) especificado. Si el archivo no se encuentra en el [include\\_path](#), `include` finalmente verificará en el propio directorio del script que hace el llamado y en el directorio de trabajo actual, antes de fallar. El constructor `include` emitirá una [advertencia](#) si no puede encontrar un archivo, éste es un comportamiento diferente al de [require](#), el cual emitirá un [error fatal](#).

## require\_once

---

(PHP 4, PHP 5, PHP 7, PHP 8)

La sentencia `require_once` es idéntica a [require](#) excepto que PHP verificará si el archivo ya ha sido incluido y si es así, no se incluye (require) de nuevo.

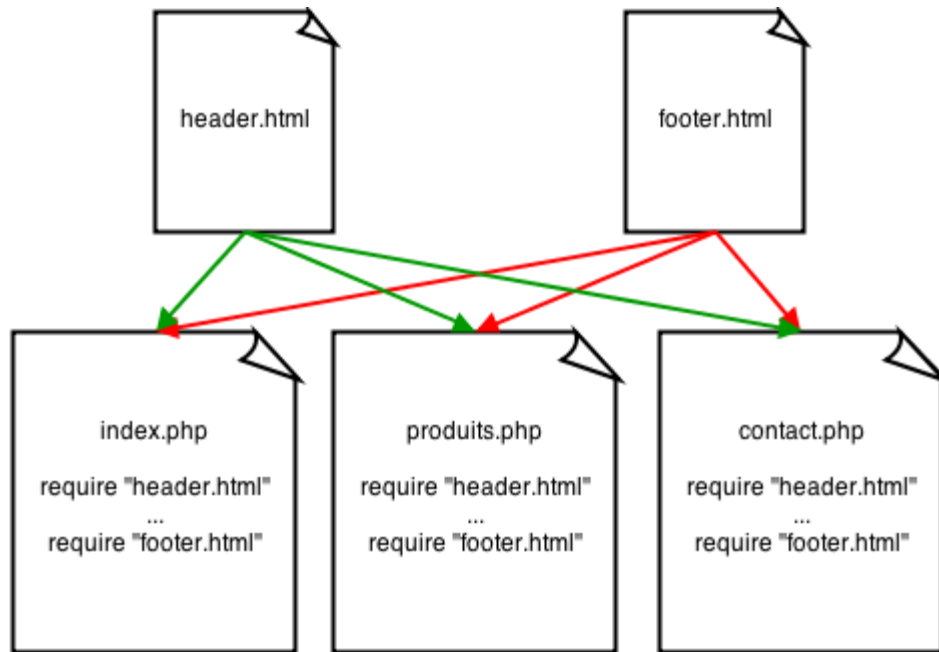
## include\_once

---

(PHP 4, PHP 5, PHP 7, PHP 8)

La sentencia `include_once` incluye y evalúa el fichero especificado durante la ejecución del script. Tiene un comportamiento similar al de la sentencia [include](#), siendo la única diferencia de que si el código del fichero ya ha sido incluido, no se volverá a incluir, e `include_once` devolverá **true**. Como su nombre indica, el fichero será incluido solamente una vez.

# COMO USAR ESTO



## DIFERENCIAS ENTRE

`include` ——— Y ——— `require`

Son constructores, por esa razón, no es necesario llamarlos con paréntesis.

“ Las diferencias entre `include` y `require` llegan cuando el archivo que se requiere, no se encuentra. ”

```
<?php
include 'archivo.php';
```

Solamente emitirá una advertencia, y continuará el script.

```
<?php
require 'archivo.php';
```

Emitirá un error fatal y detendrá el script.

“ **TIP:** Si el archivo que se requiere es crítico para que el script continúe correctamente, necesitarás usar `'require'` ”

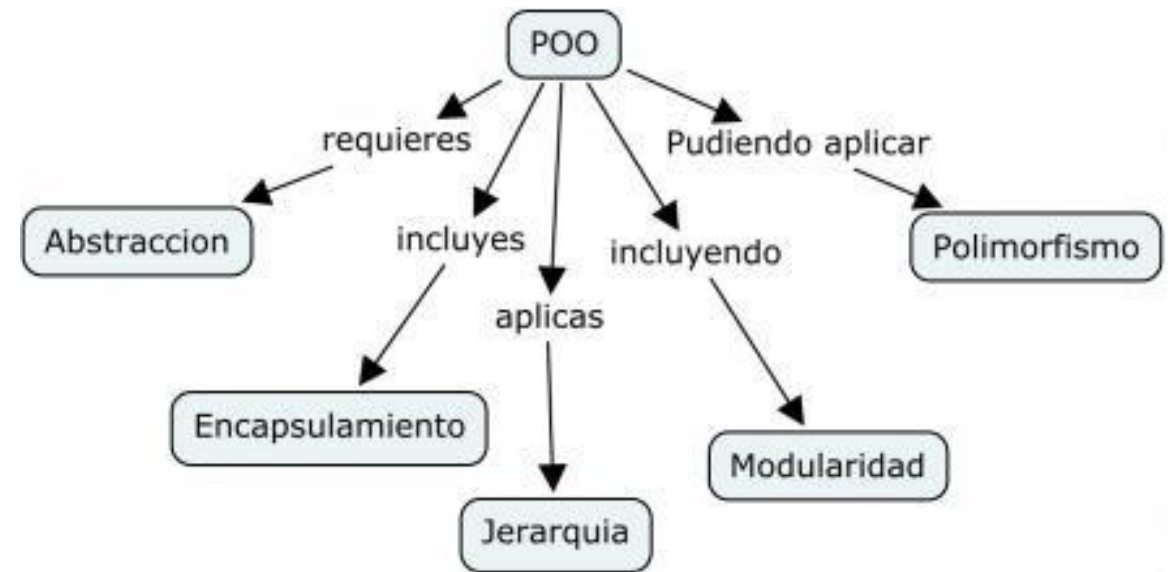
`include_once` y `require_once`, se comportan como `include` y `require`, con excepción de que solo lo incluyen si no a sido incluido previamente, de otra forma, lanzará error.

Aprende PHP y mucho más en...

# PROGRAMACIÓN ORIENTADA OBJETOS

Qué es la Programación Orientada a Objetos, un paradigma de programación que permite desarrollar aplicaciones complejas manteniendo un código más claro y manejable que otros paradigmas anteriores.

Principio de abstracción  
Principio de encapsulamiento  
Principio de modularidad  
Principio de paso de mensajes  
Principio de herencia  
Principio de polimorfismo





# Principio de Abstracción

Abstraer es suprimir u ocultar algunos detalles de un proceso o de un elemento, para resaltar algunos aspectos, detalles o estructuras.

La abstracción de objetos se refiere a la vista externa del objeto (interfaces).

La abstracción es la forma en que nuestra mente modela la realidad, formando los objetos.

# Encapsulamiento

## Proceso por el que se ocultan:

Las estructuras de datos

Los detalles de la implementación

Permite considerar a los objetos como "cajas negras", evitando que otros objetos accedan a detalles que NO LES INTERESA

Una vez creada la clase, las funciones usuarias no requieren conocer los detalles de su implementación

# Encapsulamiento

Toda clase tiene un conjunto de *atributos* y *métodos* asociados.

Todos ellos están *encapsulados* o contenidos dentro de la misma clase, de manera que son *miembros* de dicha clase.

Esos métodos y atributos pueden ser utilizados por *otras* clases *sólo si* la clase que los encapsula les brinda los *permisos* necesarios para ello.

# Encapsulamiento



A los métodos también se les puede aplicar distintos modificadores de acceso, por lo que un método puede ser también marcado como **private**.



La recomendación general es inicialmente hacer todo privado e irlo haciendo público conforme se va necesitando.



Entre menos métodos públicos tenga una clase es más fácil de entender.



No se recomienda tener atributos públicos en lo absoluto.

# Herencia



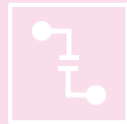
La herencia es un mecanismo que permite la definición de una clase a partir de la definición de otra ya existente.



La herencia permite compartir automáticamente métodos y atributos entre clases, subclases y objetos.



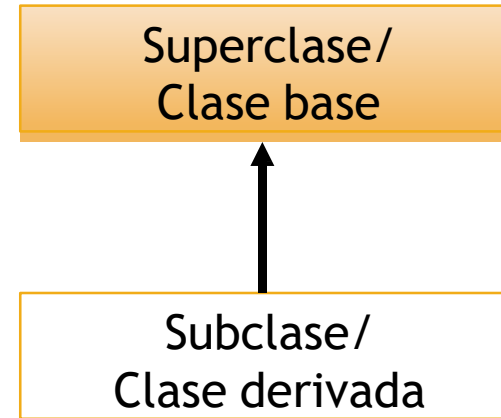
Permite reutilizar código creando nuevas clases a partir de las existentes (construidas y depuradas).



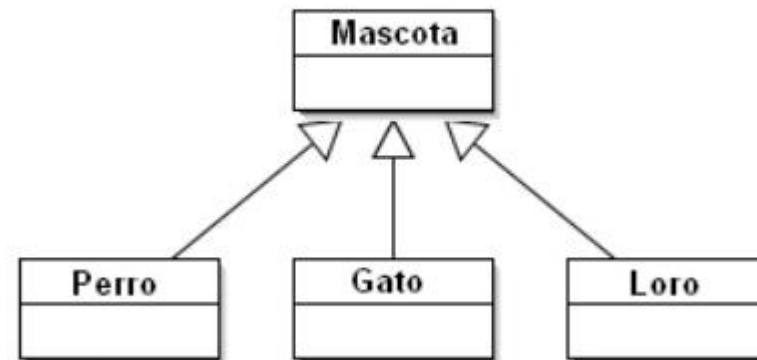
Compromete una relación de jerarquía (es-un).

# Herencia

En POO, la clase de la que se va a heredar se llama superclase o clase base. Mientras que la que hereda se le conoce como subclase o clase derivada.

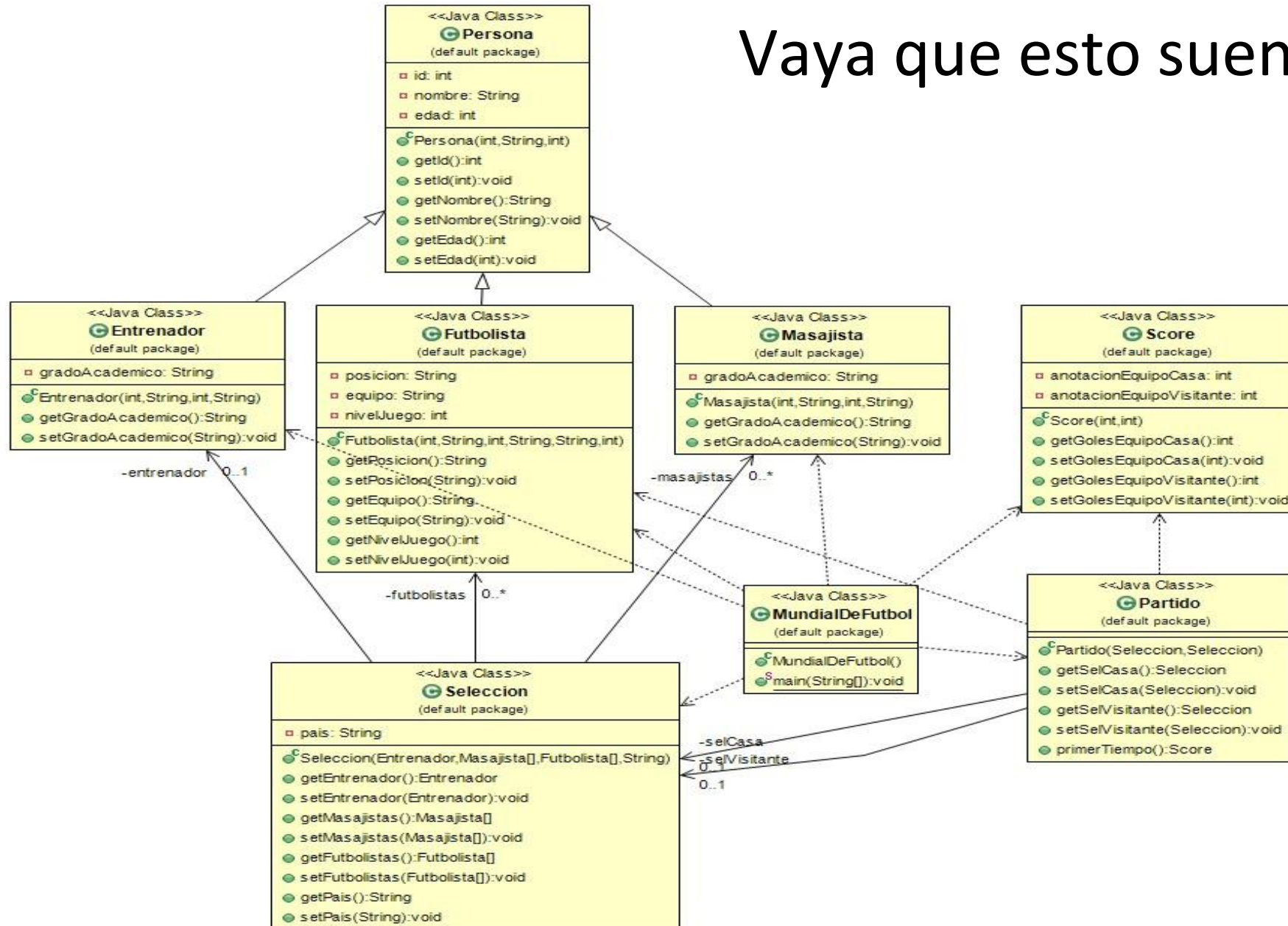


## HERENCIA





# Vaya que esto suena gran enigma



# Model-View-Controller

## Arquitectura de software

### Patrones de Diseño Gang of Four

#### Creacionales

- Abstract Factory
- Singleton
- Prototype
- Factory Builder

#### Estructurales

- Adapter
- Bridge
- Facade
- Flyweight
- Factory Builder

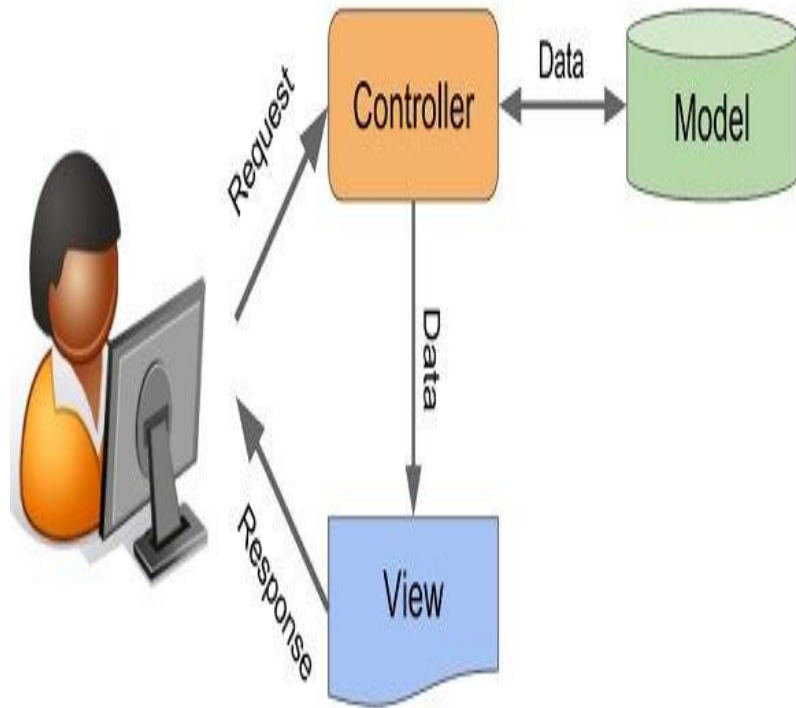
#### De Comportamiento

- Visitor
- Mediator
- Memento
- Observer
- Command
- Chain of responsibility
- State
- Strategy
- Iterator
- Template
- Interpreter

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

[PATRONES DE ARQUITECTURA Y DISEÑO DE SOFTWARE  
\(desarrollodepaginasweb.com.mx\)](http://desarrollodepaginasweb.com.mx)

# Model-View-Controller



**Los modelos** administran los datos de la aplicación y ayudan a aplicar las reglas de negocio especiales que la aplicación pueda necesitar.

**Las vistas** son archivos simples, con poca o ninguna lógica, que muestran la información al usuario.

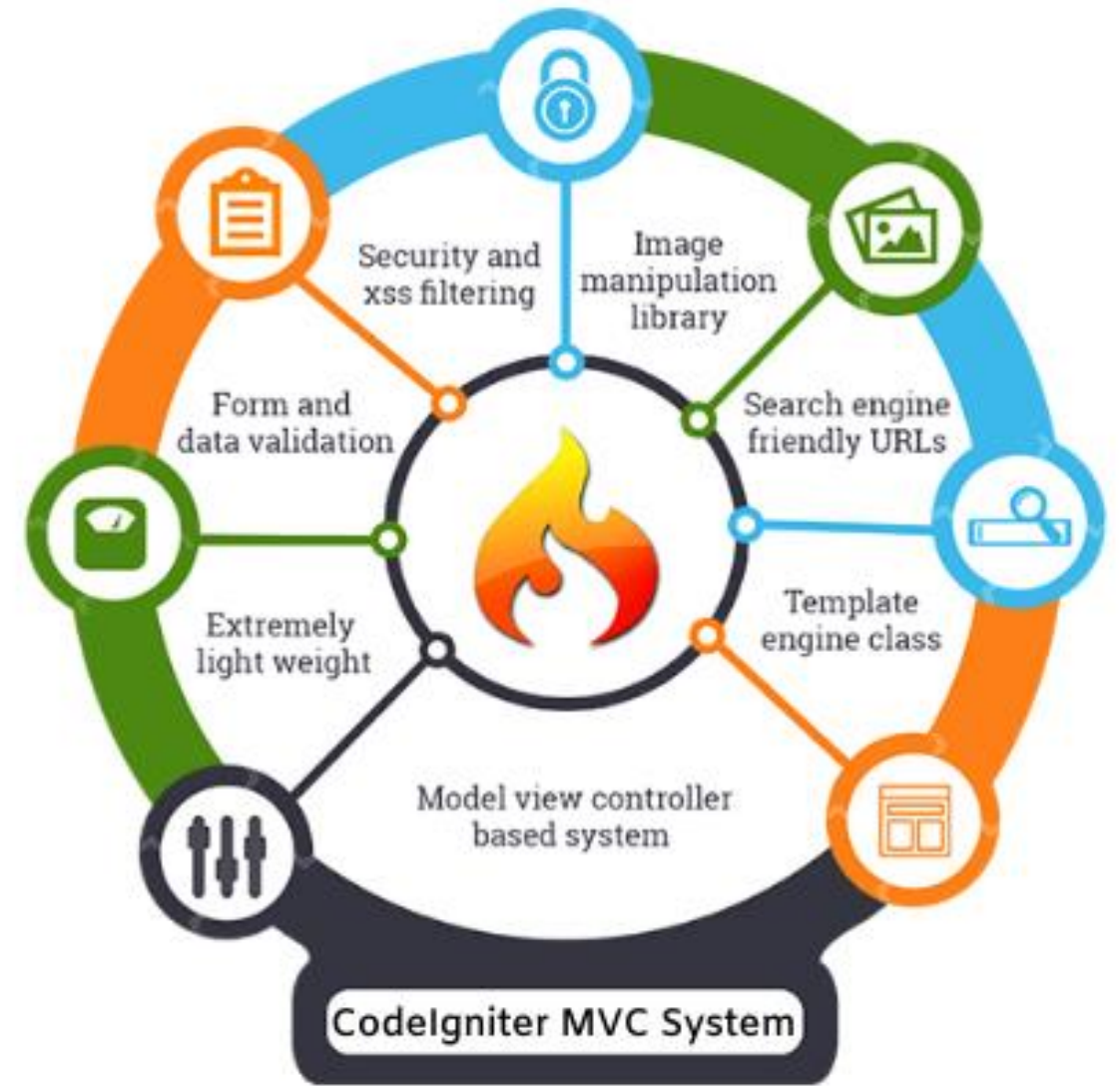
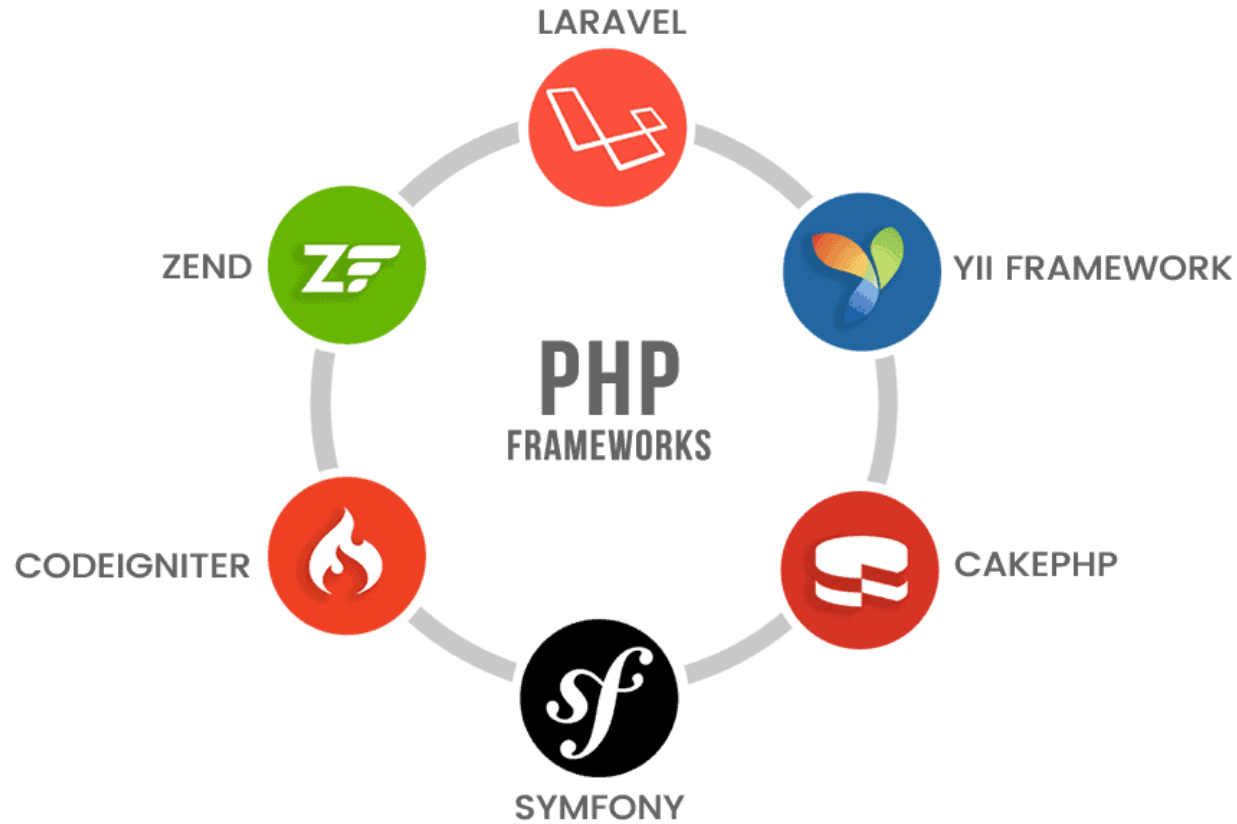
**Los controladores** actúan como código de pegamento, calculando referencias de datos de un lado a otro entre la vista (o el usuario que lo ve) y el almacenamiento de datos.

En su punto más básico, los controladores y modelos son simplemente clases que tienen un trabajo específico. No son los únicos tipos de clase que puede usar, obviamente, pero constituyen el núcleo de cómo se diseñará este marco de trabajo. Incluso tienen directorios designados en el directorio `/app` para su almacenamiento, aunque es libre de almacenarlos donde desee, siempre y cuando estén correctamente espacios de nombres. Lo discutiremos con más detalle a continuación.

[Models, Views, and Controllers — CodeIgniter 4.1.1 documentation](#)

# TÁCTICA #4

Esteroides para la victoria



Simple solutions over complexity

CodeIgniter encourages MVC, but does not force it on you.



# CODEIGNITER

Es un potente **framework PHP** no tan complicado y ligero , **diseñado para desarrolladores que necesitan un kit de herramientas simple y elegante** para crear aplicaciones web con todas las funciones.

```
/app
/Config      Stores the configuration files
/Controllers Controllers determine the program flow
/Database    Stores the database migrations and seeds files
/Filters      Stores filter classes that can run before and after controller
/Helpers      Helpers store collections of standalone functions
/Language     Multiple language support reads the language strings from here
/Libraries    Useful classes that don't fit in another category
/Models       Models work with the database to represent the business entities.
/ThirdParty   ThirdParty libraries that can be used in application
/Views        Views make up the HTML that is displayed to the client.
```

## Instarlo

```
composer create-project codeigniter4/appstarter project-root
```

```
composer create-project codeigniter4/appstarter --no-dev
```

[Application Structure — CodeIgniter 4.1.1 documentation](#)

[Composer Installation — CodeIgniter 4.1.1 documentation](#)



BASTA DE FILOSOFÍA Y AHORA SI  
PROGRAMEMOS

