 Universidad Tecnológica Tula Tepeji

“PHP”

Desarrollado por: Adriana Guadalupe Alberto Abad

Grupo: 3TIIRD-G1

Fecha: 09/07/2020

# Índice

**[Índice 2](#_Toc45229920)**

[**INTRODUCCION 3**](#_Toc45229921)

[**Que es PHP 4**](#_Toc45229922)

[**Variables 4**](#_Toc45229923)

[**Que es una Superglobals 9**](#_Toc45229924)

[**Condicionales 13**](#_Toc45229925)

[**Bucles 15**](#_Toc45229926)

[**Patrón MVC para aplicaciones web con PHP 17**](#_Toc45229927)

[**Reflexión 19**](#_Toc45229928)

[**Bibliografía 20**](#_Toc45229929)

# INTRODUCCION

En esta investigación se basa sobre que es PHP variables, descripciones y los ejemplos de cada uno de ellos tomando en cuenta que al igual de la Super Globals como los GETS y POST al igual mostrando los tipos de condicionales que patrones obtienes de las capas basándose en cómo se aplican en PHP.

# Que es PHP

PHP es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) de uso general ya que se adapta especialmente al [desarrollo web](https://es.wikipedia.org/wiki/Desarrollo_web) dado que Fue creado inicialmente por el [programador](https://es.wikipedia.org/wiki/Programador) danés-canadiense [Rasmus Lerdorf](https://es.wikipedia.org/wiki/Rasmus_Lerdorf) en 1994.​ En la actualidad, la [implementación de referencia](https://es.wikipedia.org/wiki/Implementaci%C3%B3n_de_referencia) de PHP es producida por The PHP Group. PHP originalmente significaba Personal Home Page, pero ahora significa el inicializo 1​ recursivo PHP: Hipertexto Reproceso. ​

Ya que el código de PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un Damon o como un ejecutable de interfaz de entrada común.

# Variables

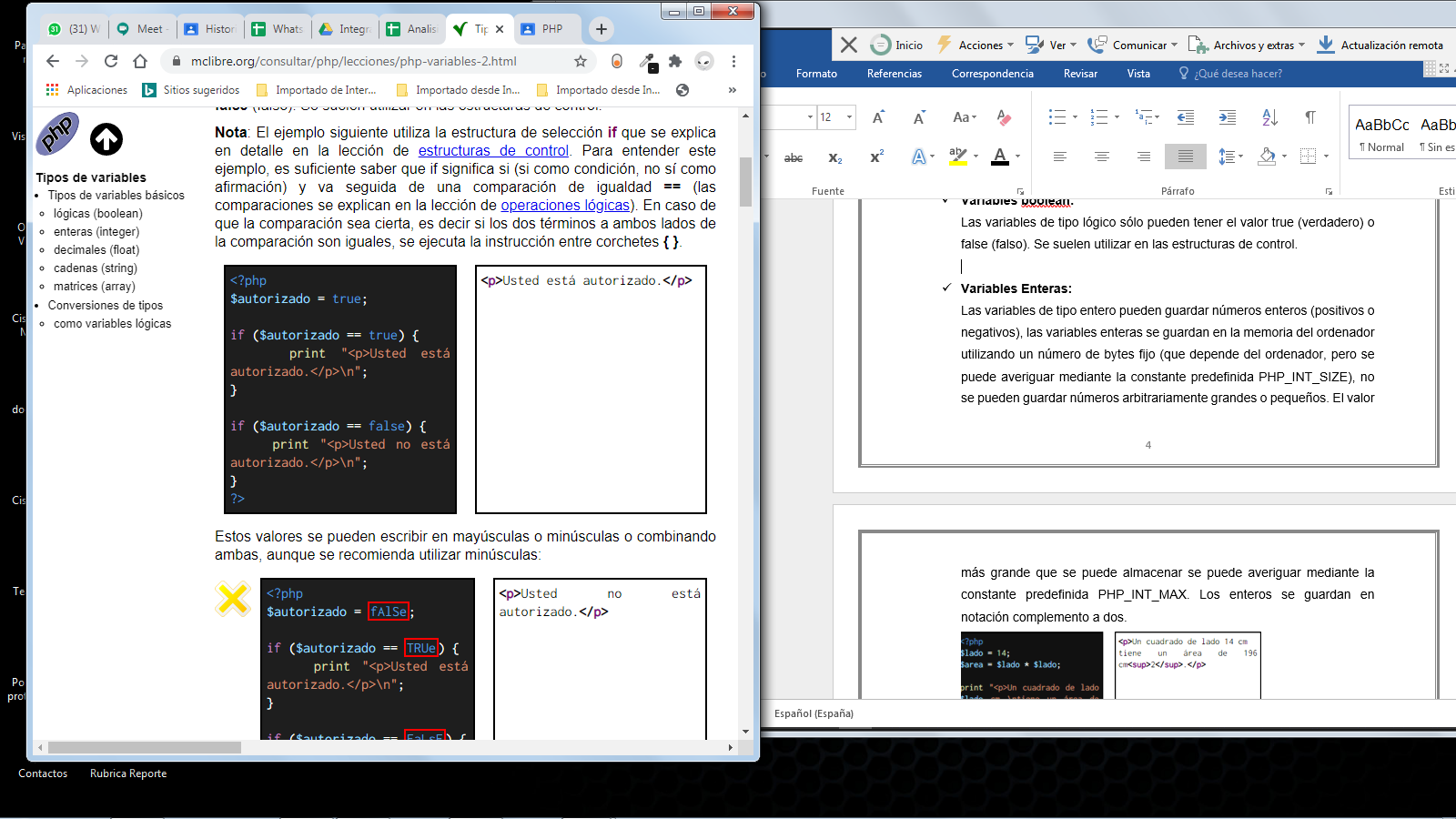
Los tipos de variables básicos son los siguientes:

* lógicas o booleanas (boolean)
* enteros (integer)
* decimales (float)
* cadenas (string)
* matrices (arrays)

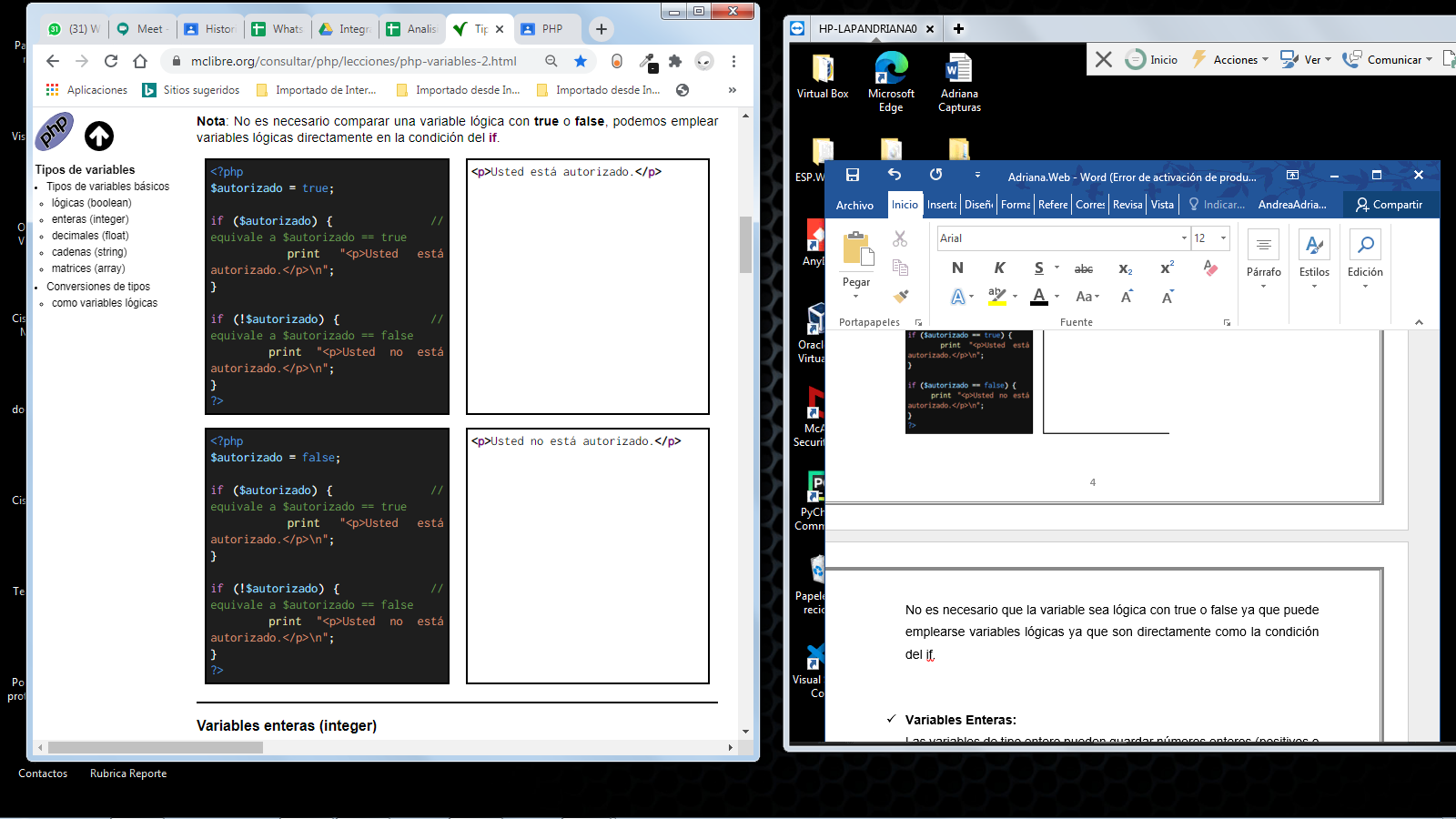
Otros tipos de variables:

* objetos (object)
* recursos (resource)
* nulo (null)
* **Variables boolean:**

Las variables de tipo lógico sólo pueden tener el valor true (verdadero) o false (falso). Se suelen utilizar en las estructuras de control.

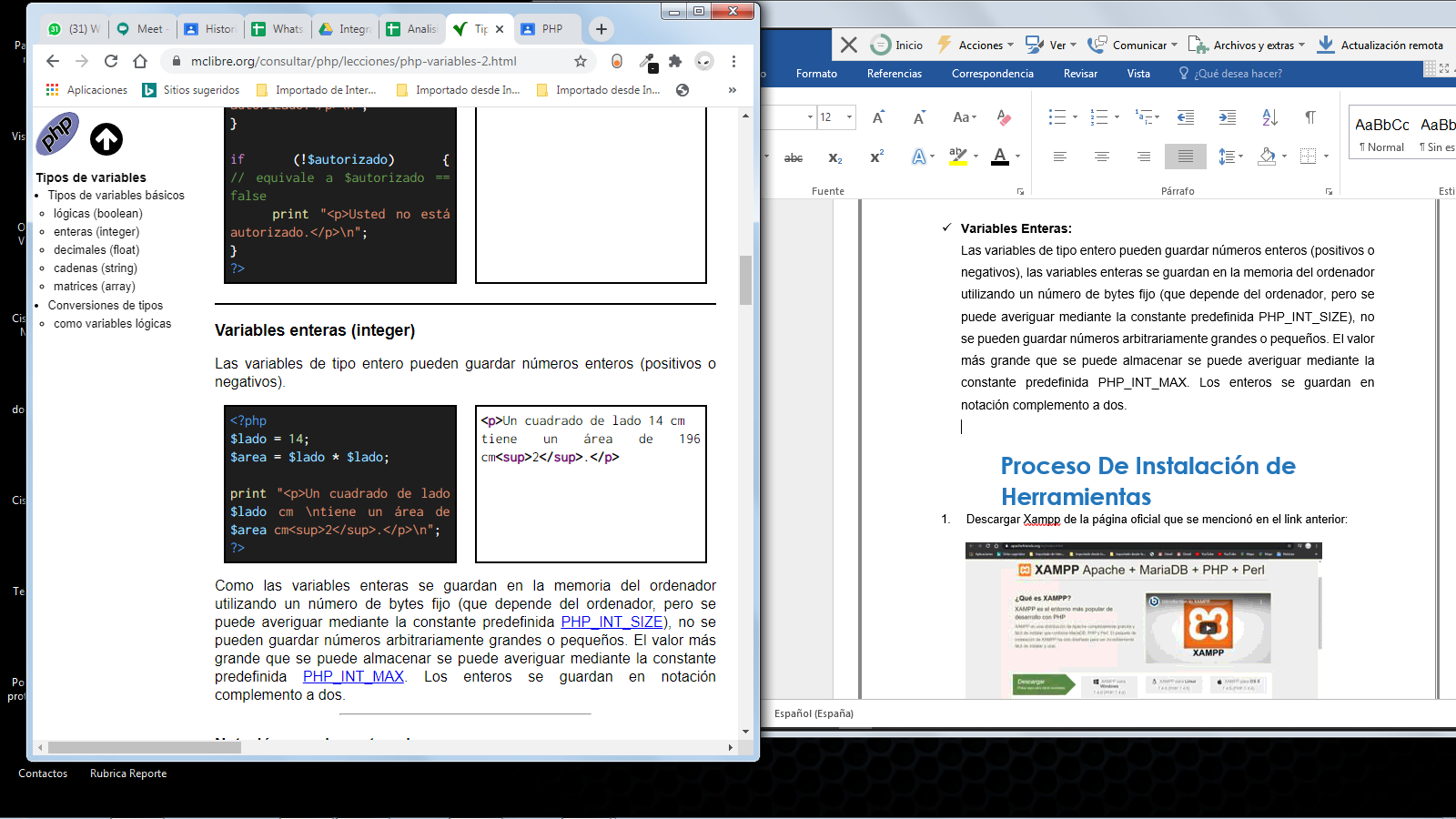


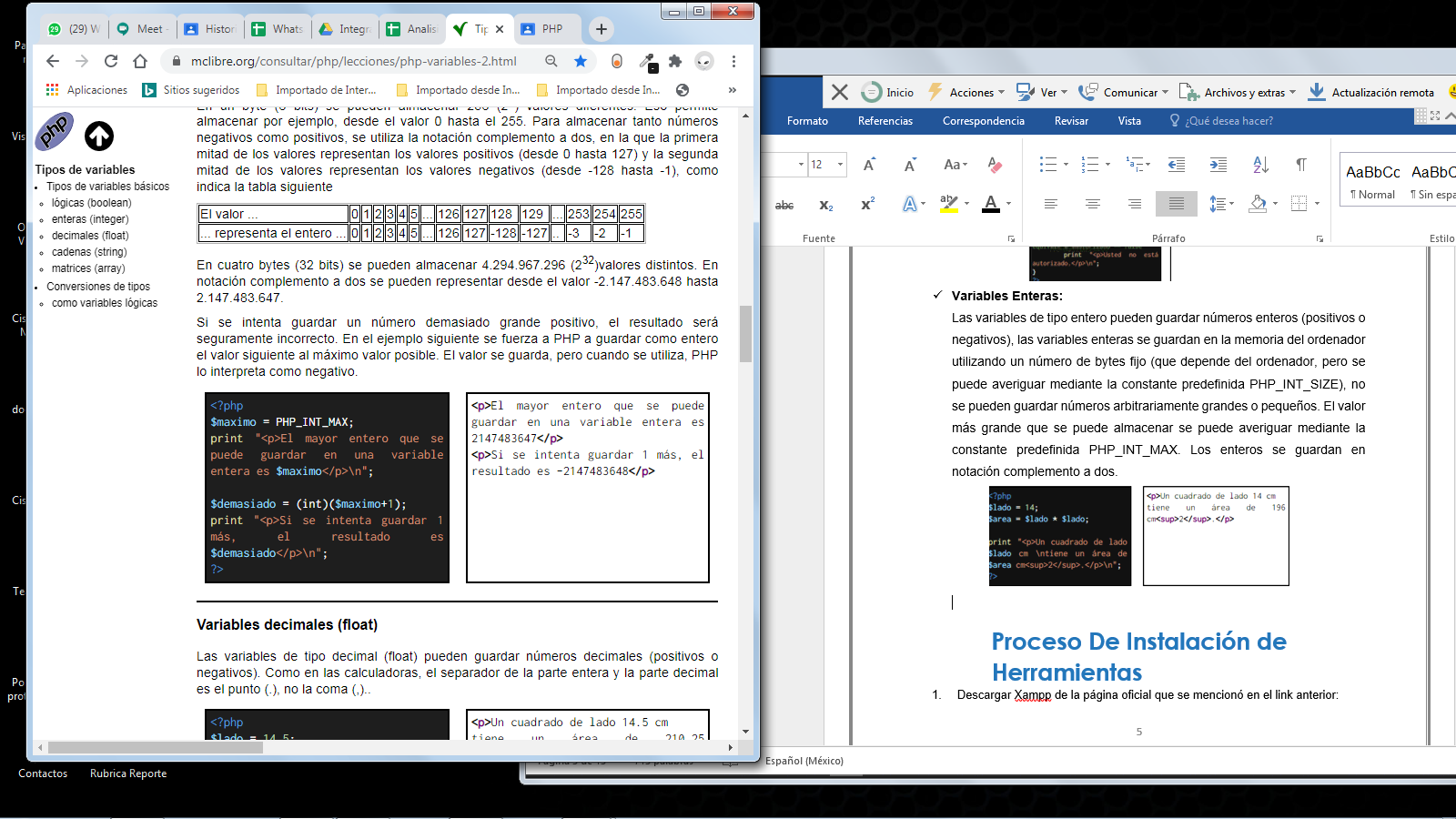
No es necesario que la variable sea lógica con true o false ya que puede emplearse variables lógicas ya que son directamente como la condición del if.



* **Variables Enteras:**

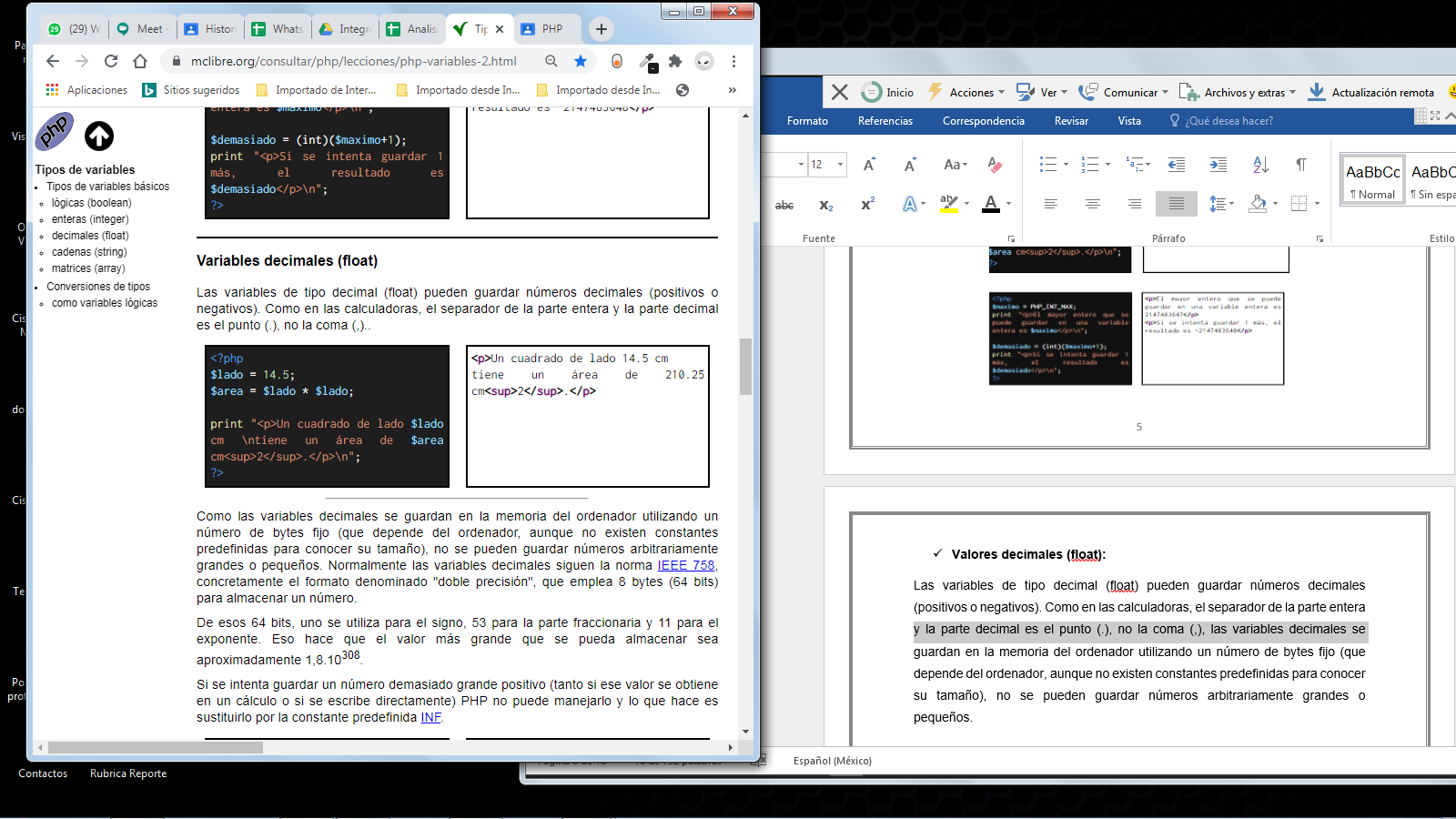
Las variables de tipo entero pueden guardar números enteros (positivos o negativos), las variables enteras se guardan en la memoria del ordenador utilizando un número de bytes fijo (que depende del ordenador, pero se puede averiguar mediante la constante predefinida PHP\_INT\_SIZE), no se pueden guardar números arbitrariamente grandes o pequeños. El valor más grande que se puede almacenar se puede averiguar mediante la constante predefinida PHP\_INT\_MAX. Los enteros se guardan en notación complemento a dos.

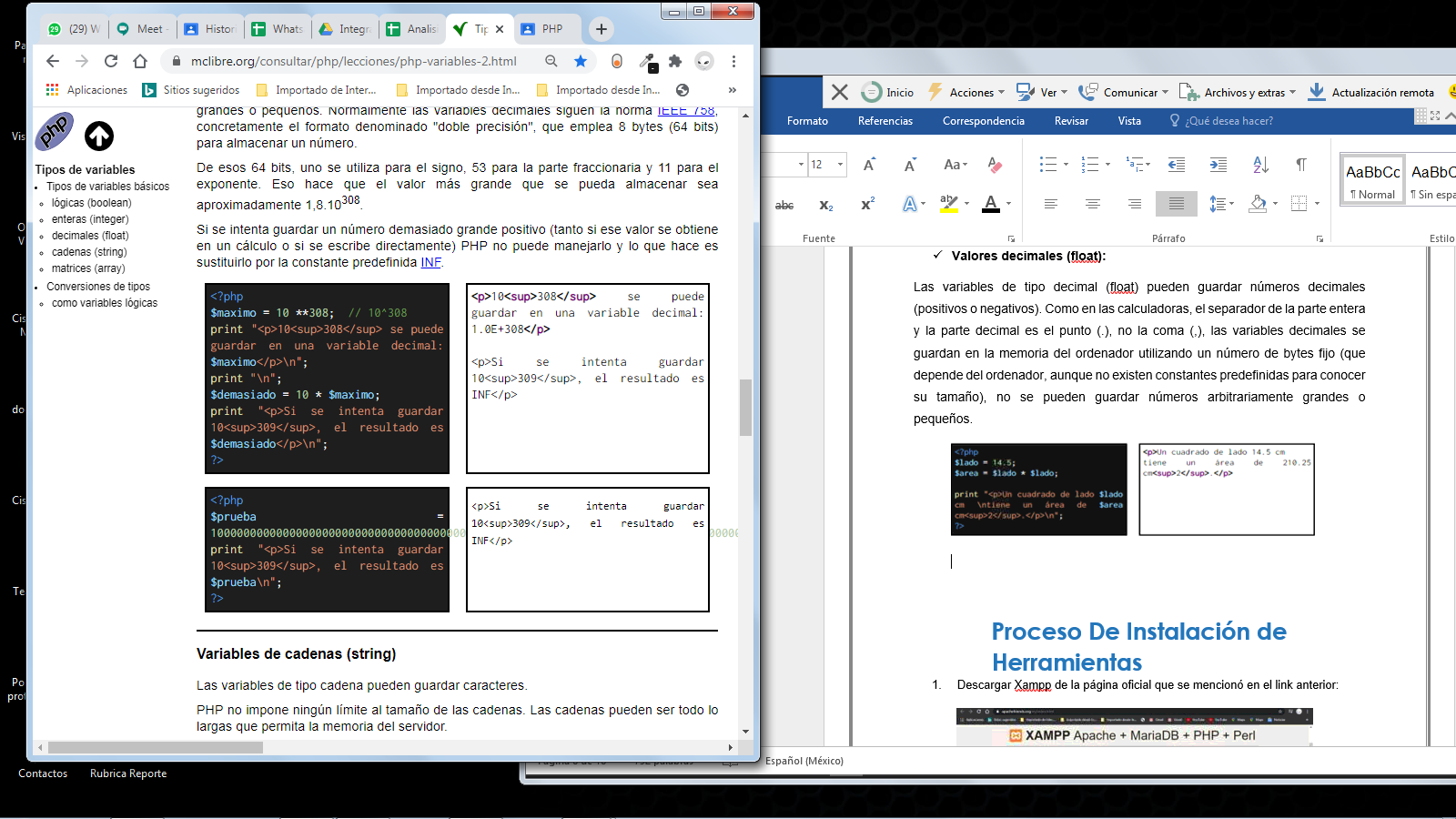




* **Valores decimales (float):**

Las variables de tipo decimal (float) pueden guardar números decimales (positivos o negativos). Como en las calculadoras, el separador de la parte entera y la parte decimal es el punto (.), no la coma (,), las variables decimales se guardan en la memoria del ordenador utilizando un número de bytes fijo (que depende del ordenador, aunque no existen constantes predefinidas para conocer su tamaño), no se pueden guardar números arbitrariamente grandes o pequeños.

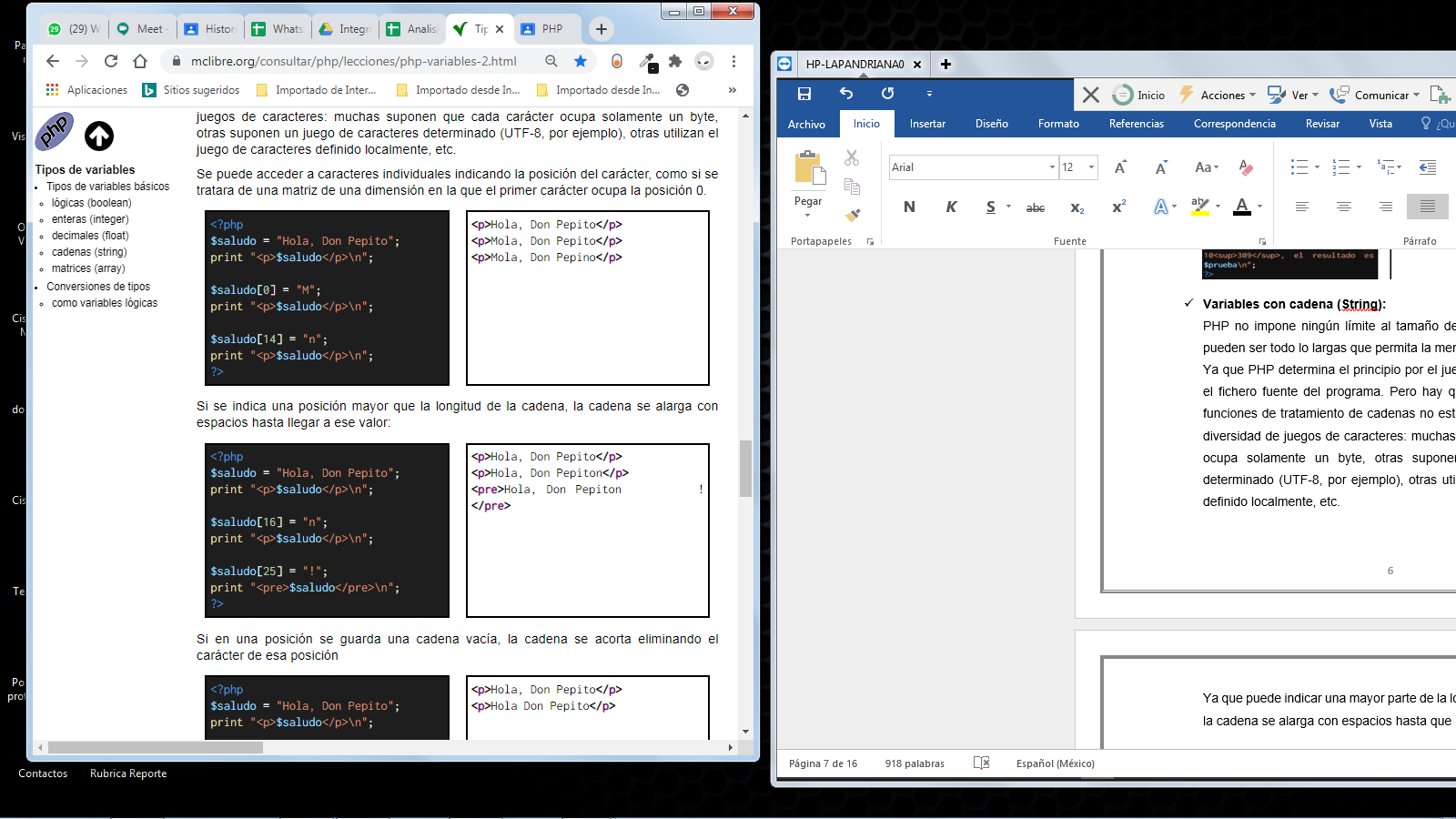




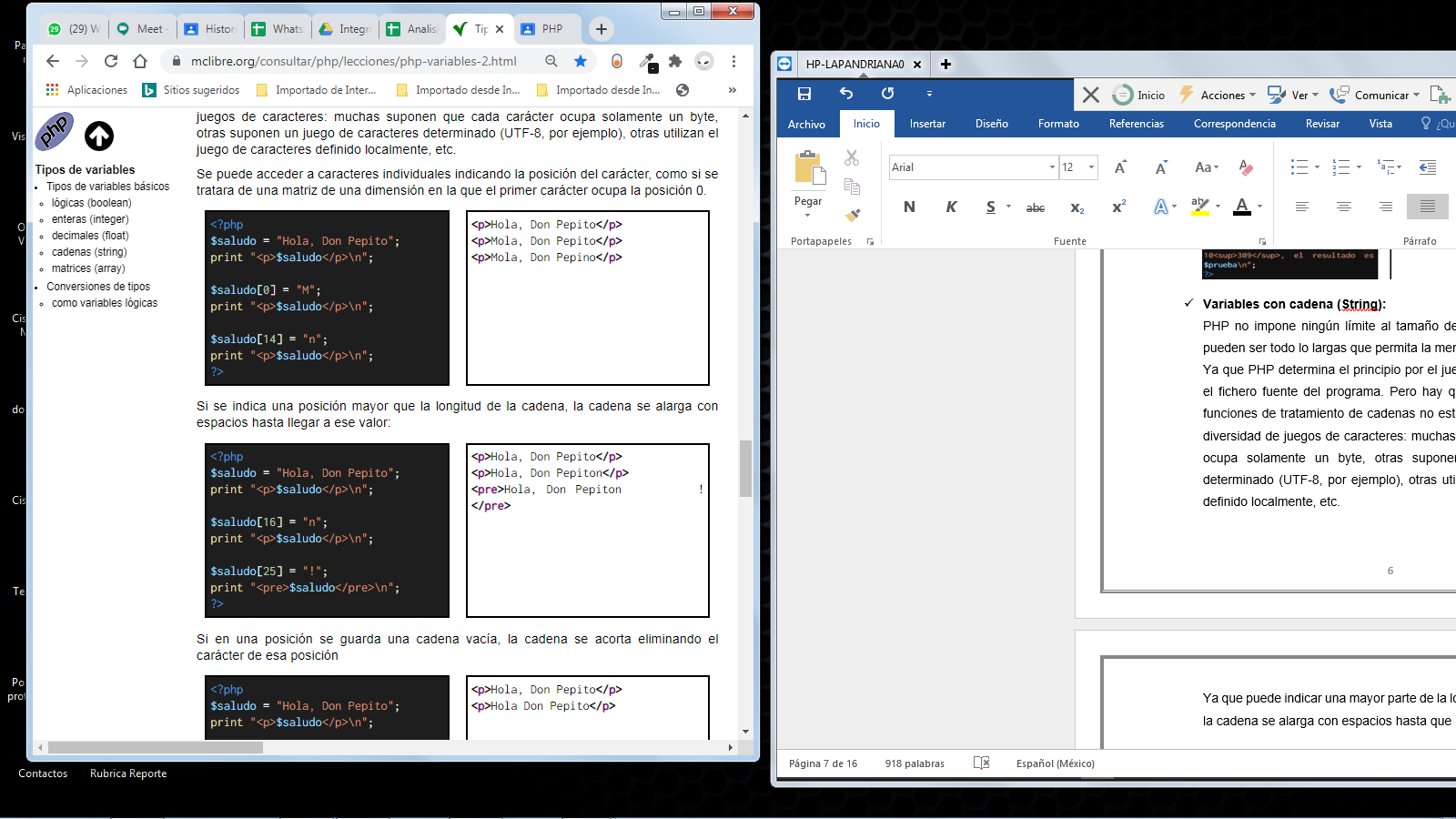
* **Variables con cadena (String):**

PHP no impone ningún límite al tamaño de las cadenas. Las cadenas pueden ser todo lo largas que permita la memoria del servidor.

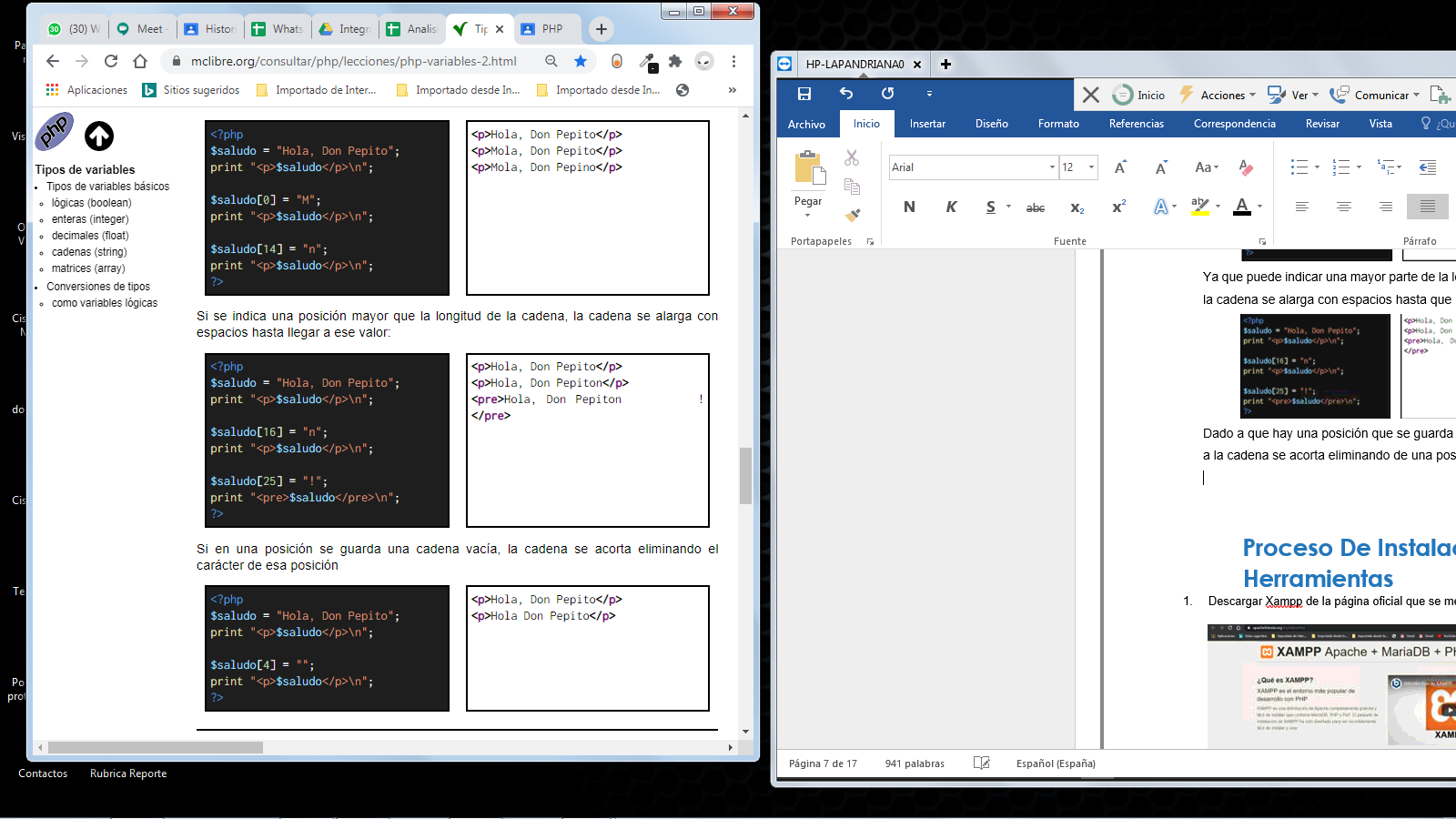
Ya que PHP determina el principio por el juego de caracteres que utiliza el fichero fuente del programa. Pero hay que tener en cuenta que las funciones de tratamiento de cadenas no están preparadas para tratar la diversidad de juegos de caracteres: muchas suponen que cada carácter ocupa solamente un byte, otras suponen un juego de caracteres determinado (UTF-8, por ejemplo), otras utilizan el juego de caracteres definido localmente, etc.



Ya que puede indicar una mayor parte de la longitud de la cadena, ya que la cadena se alarga con espacios hasta que llega a otro valor.



Dado a que hay una posición que se guarda en una cadena vacía ya que a la cadena se acorta eliminando de una posición.

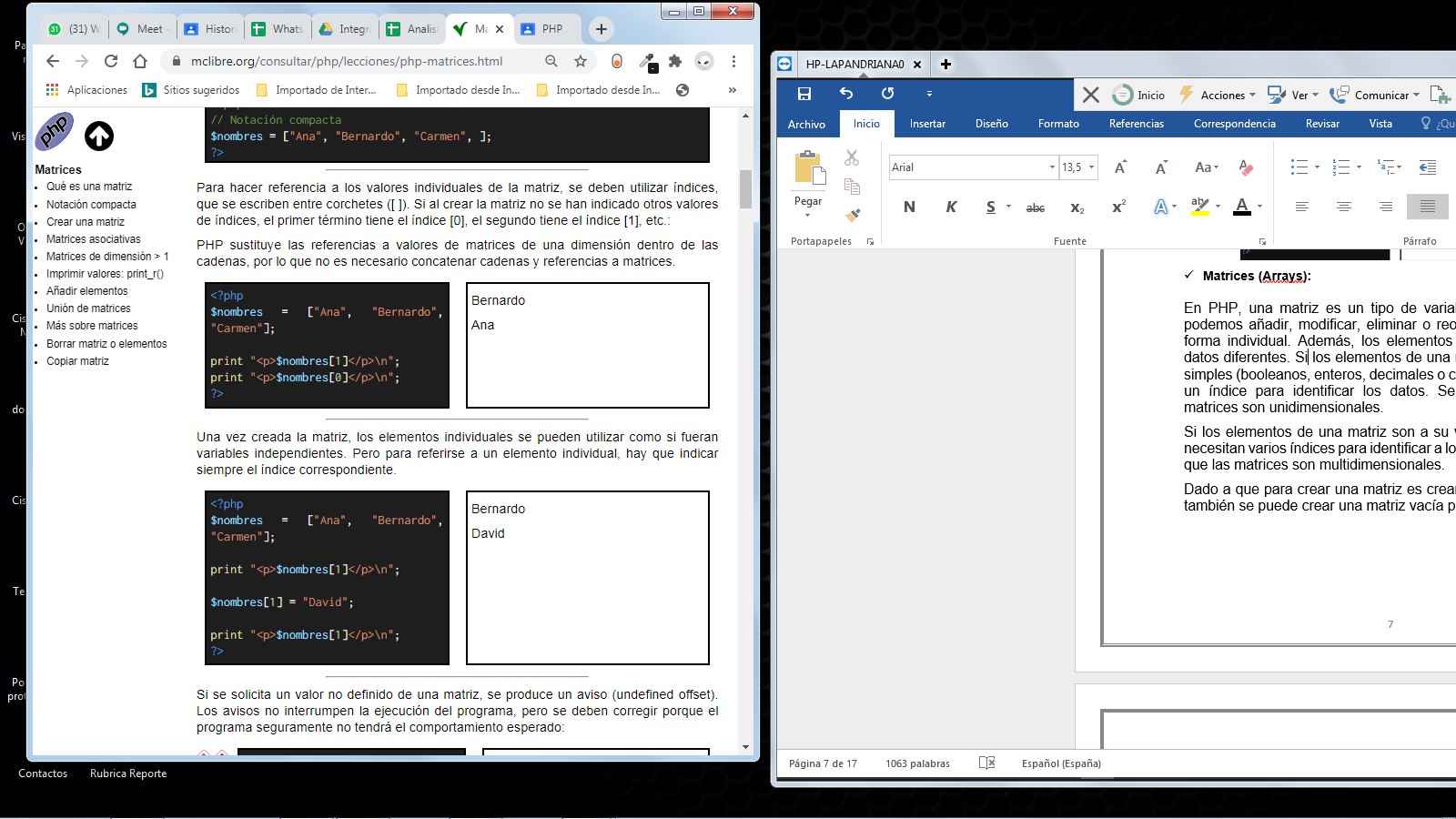


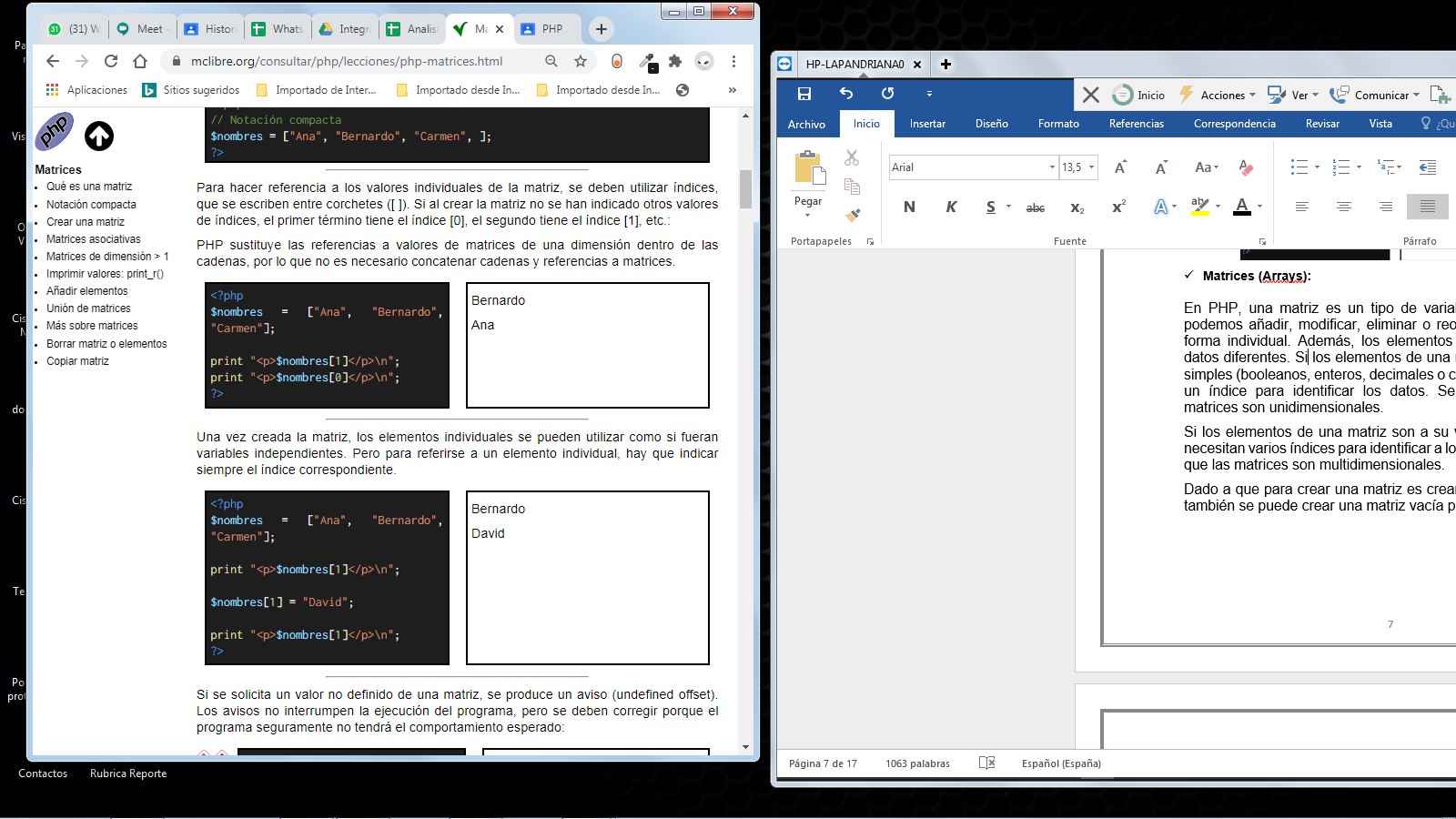
* **Matrices (Arrays):**

En PHP, una matriz es un tipo de variable muy flexible, ya que podemos añadir, modificar, eliminar o reordenar los elementos de forma individual. Además, los elementos pueden ser de tipos de datos diferentes. Si los elementos de una matriz son datos de tipos simples (booleanos, enteros, decimales o cadenas), sólo se necesita un índice para identificar los datos. Se dice entonces que las matrices son unidimensionales.

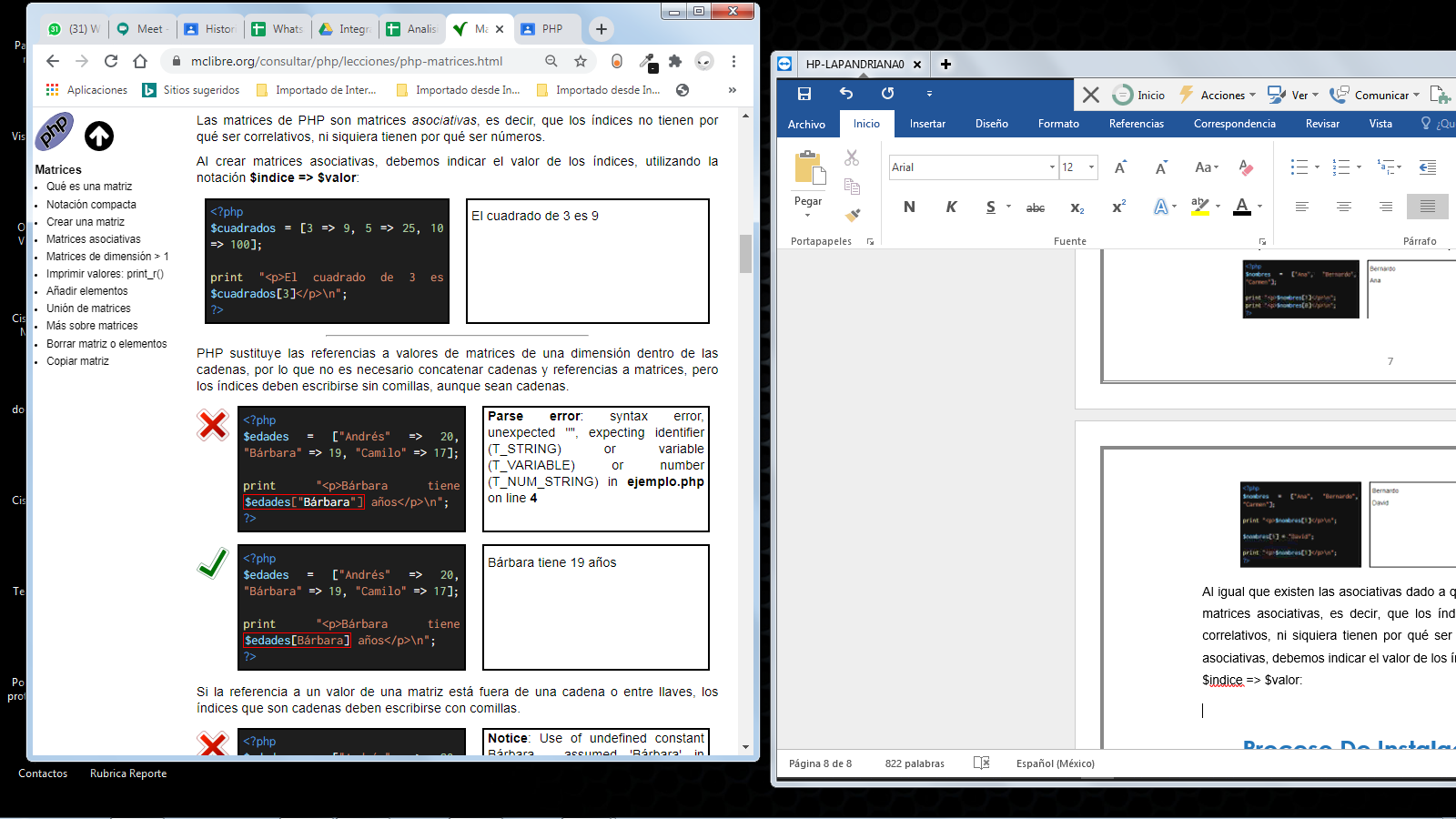
Si los elementos de una matriz son a su vez también matrices, se necesitan varios índices para identificar a los datos. Se dice entonces que las matrices son multidimensionales.

Dado a que para crear una matriz es creando los corchetes ya que también se puede crear una matriz vacía para añadir elementos



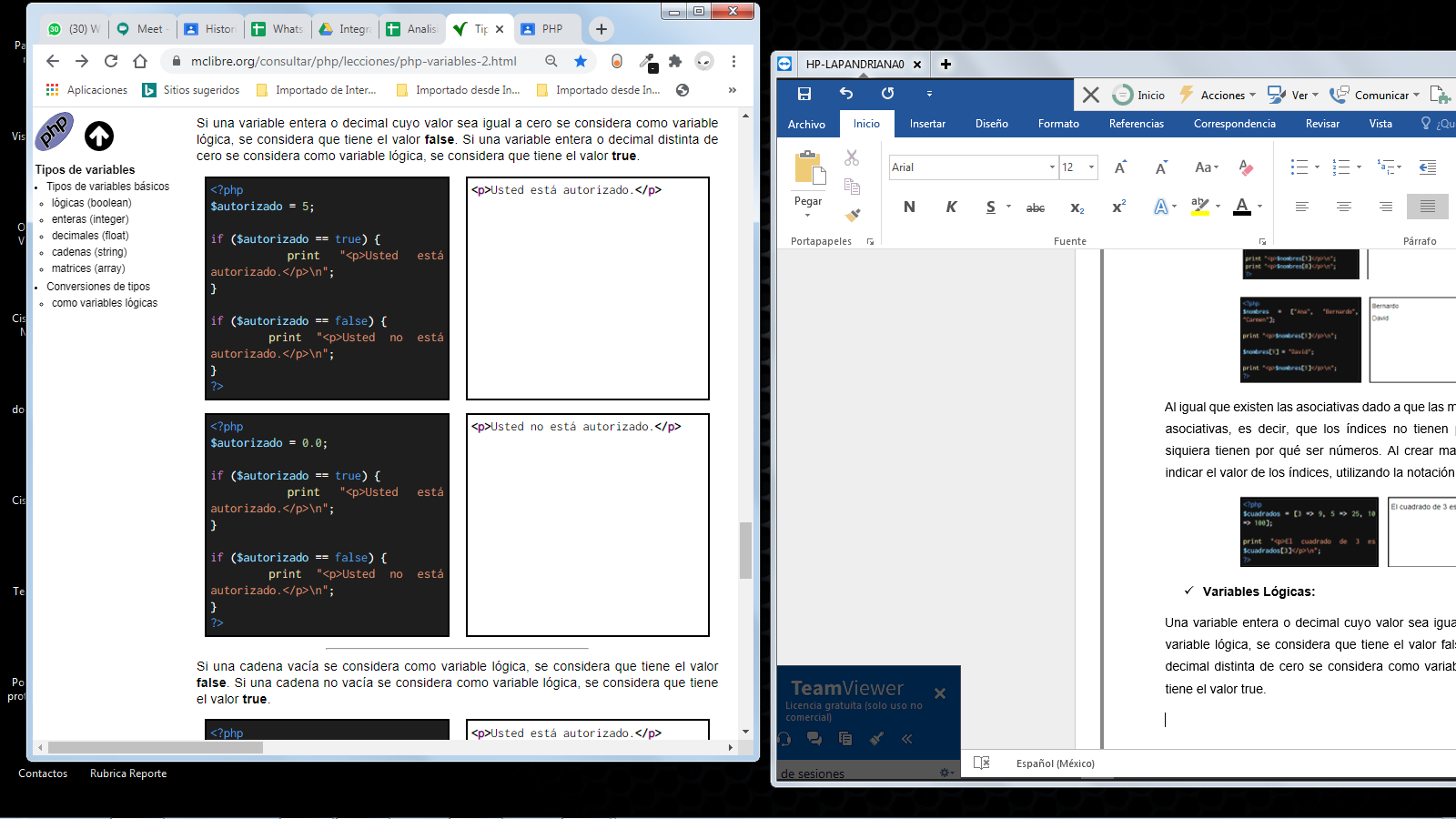


Al igual que existen las asociativas dado a que las matrices de PHP son matrices asociativas, es decir, que los índices no tienen por qué ser correlativos, ni siquiera tienen por qué ser números. Al crear matrices asociativas, debemos indicar el valor de los índices, utilizando la notación $indice => $valor:



* **Variables Lógicas:**

Una variable entera o decimal cuyo valor sea igual a cero se considera como variable lógica, se considera que tiene el valor false. Si una variable entera o decimal distinta de cero se considera como variable lógica, se considera que tiene el valor true.



Si una cadena vacía se considera como variable lógica, se considera que tiene el valor false. Si una cadena no vacía se considera como variable lógica, se considera que tiene el valor true



# Que es una Superglobals

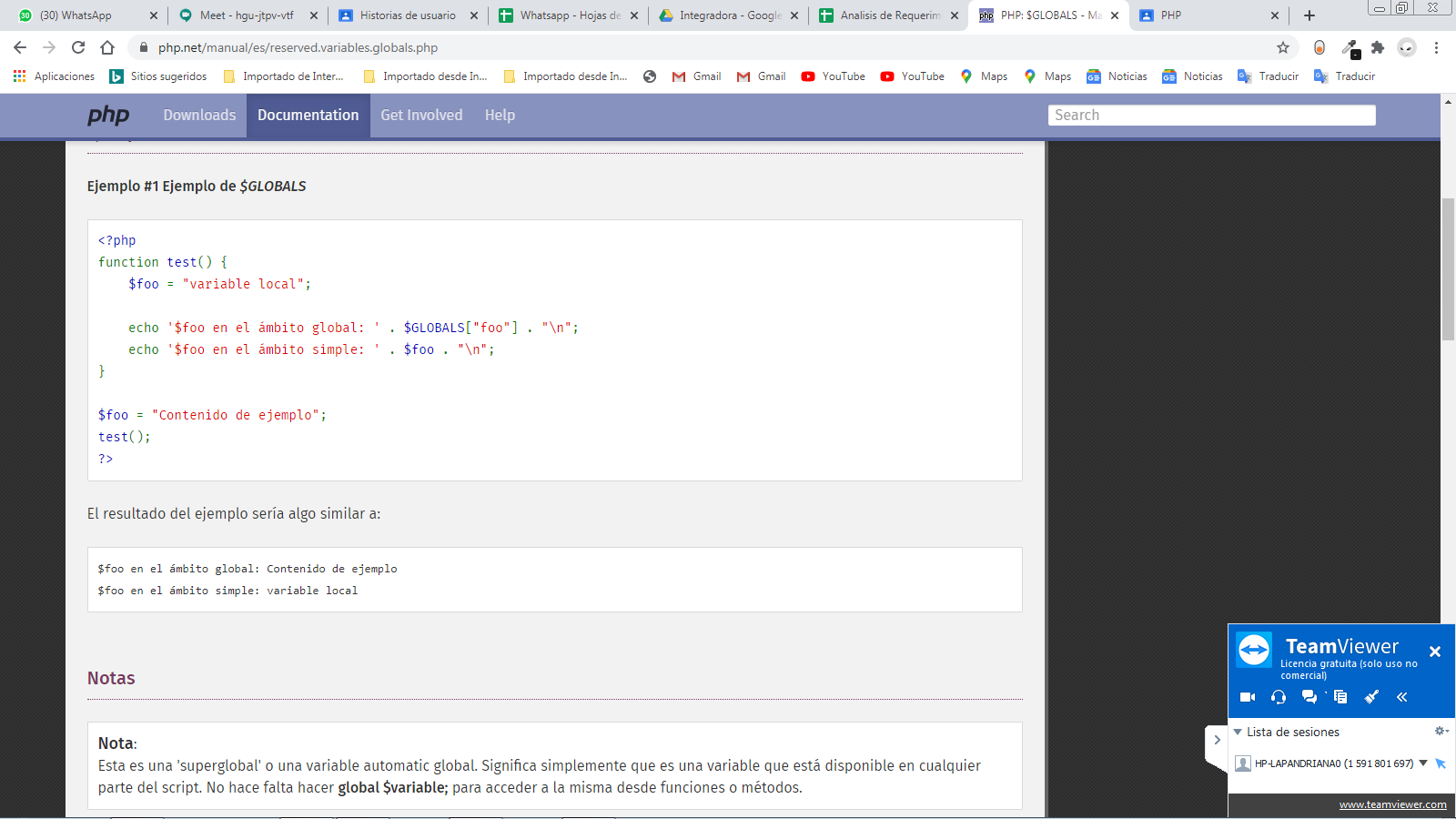
Una Superglobals son variables internas que están disponibles siempre en todos los ámbitos.

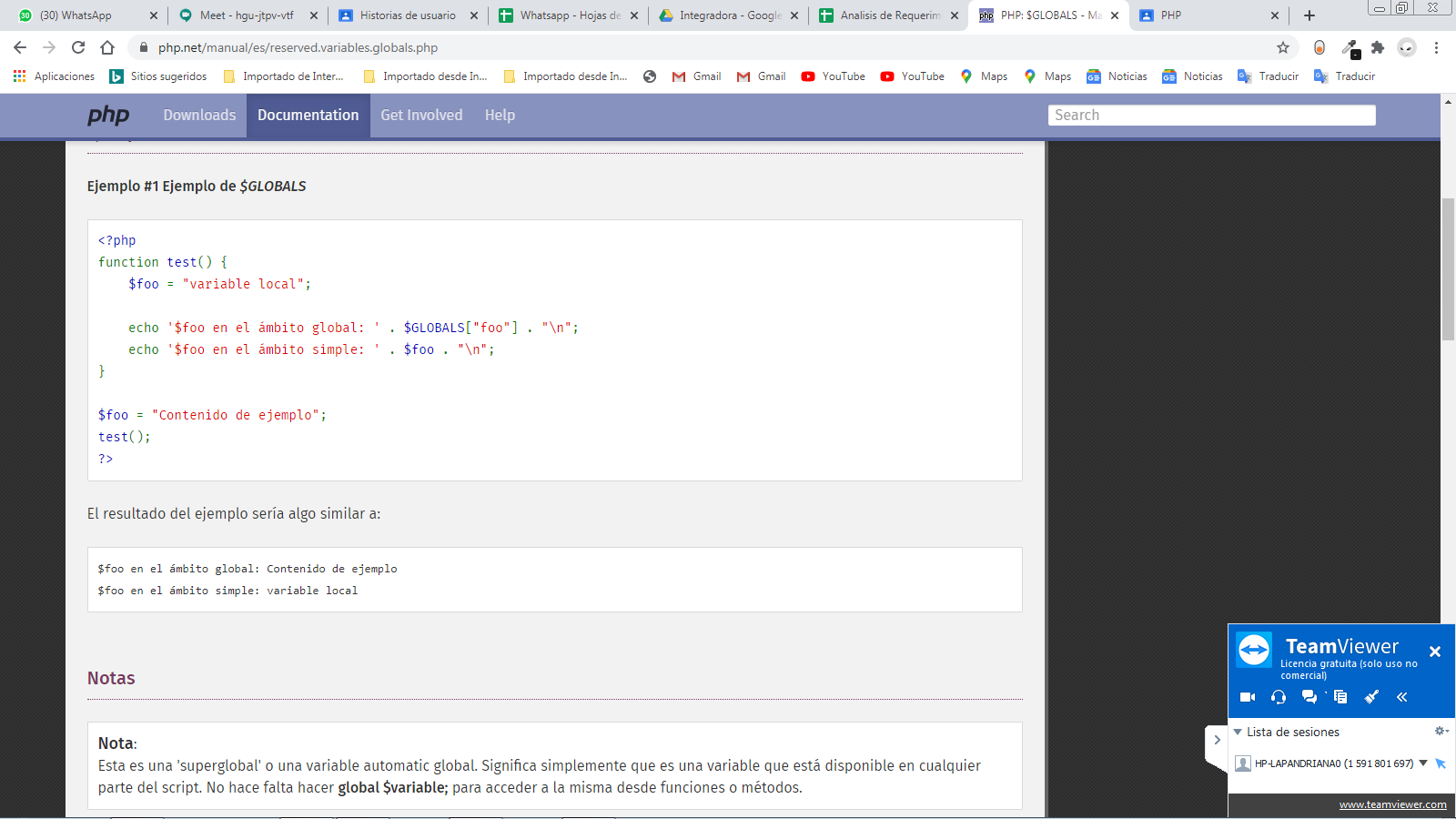
Algunas variables predefinidas en PHP son "superglobales", lo que significa que están disponibles en todos los ámbitos a lo largo del script. No es necesario emplear global $variable; para acceder a ellas dentro de las funciones o métodos.

Las variables superglobals son:

* [$GLOBALS](https://www.php.net/manual/es/reserved.variables.globals.php)
* [$\_SERVER](https://www.php.net/manual/es/reserved.variables.server.php)
* [$\_GET](https://www.php.net/manual/es/reserved.variables.get.php)
* [$\_POST](https://www.php.net/manual/es/reserved.variables.post.php)
* [$\_FILES](https://www.php.net/manual/es/reserved.variables.files.php)
* [$\_COOKIE](https://www.php.net/manual/es/reserved.variables.cookies.php)
* [$\_SESSION](https://www.php.net/manual/es/reserved.variables.session.php)
* [$\_REQUEST](https://www.php.net/manual/es/reserved.variables.request.php)
* [$\_ENV](https://www.php.net/manual/es/reserved.variables.environment.php)
* **$GLOBALS**

Hace referencia a todas las variables disponibles en el ámbito global, dado a que es una array asociativo que contiene las referencias a todas a las variables án definidas en el ámbito global del script. Los nombres de las variables son las claves del array.





* **$\_Server:**

Es un array que contiene información, tales como cabeceras, rutas y ubicaciones de script. Las entradas de este array son creadas por el servidor web. No hay garantía que cada servidor web proporcione alguna de estas entradas, existen servidores que pueden omitir algunas o proporcionar otras no recogidas aquí. Un gran número de estas variables se encuentran recogidas así que al menos debe esperar encontrar estas entradas.

* PHP\_SELF'

El nombre del archivo de script ejecutándose actualmente, relativa al directorio raíz de documentos del servidor. Por ejemplo, el valor de $\_SERVER['PHP\_SELF'] en un script ejecutado en la dirección http://example.com/foo/bar.php será /foo/bar.php.

* '[argv](https://www.php.net/manual/es/reserved.variables.argv.php)'

Array de los argumentos enviados al script. Cuando se ejecuta el script en línea de comando se obtiene acceso a los parámetros de línea de comando con un estilo parecido a cómo sería en C. Cuando se ejecuta el script mediante el método GET, contendrá la cadena de la consulta.

* '[argc](https://www.php.net/manual/es/reserved.variables.argc.php)'

Contiene el número de parámetros de línea de comando enviados al script (si se ejecuta en línea de comando).

* 'GATEWAY\_INTERFACE'

Número de revisión de la especificación CGI que está empleando el servidor, por ejemplo 'CGI/1.1'.

* 'SERVER\_ADDR'

La dirección IP del servidor donde se está ejecutando actualmente el script.

* 'SERVER\_NAME'

El nombre del host del servidor donde se está ejecutando actualmente el script. Si el script se ejecuta en un host virtual se obtendrá el valor del nombre definido para dicho host virtual.

* 'SERVER\_SOFTWARE'

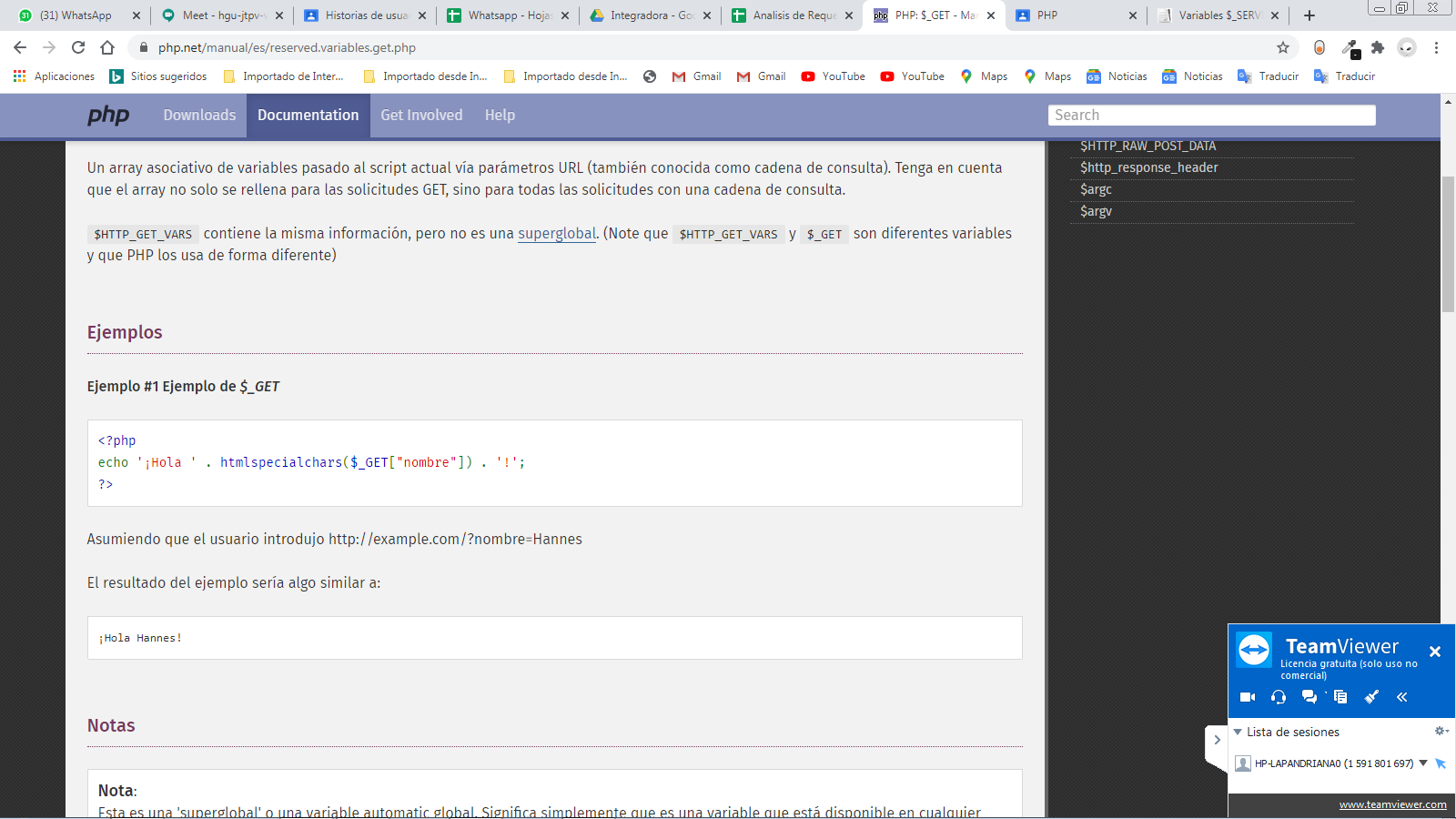
Cadena de identificación del servidor dada en las cabeceras de respuesta a las peticiones.

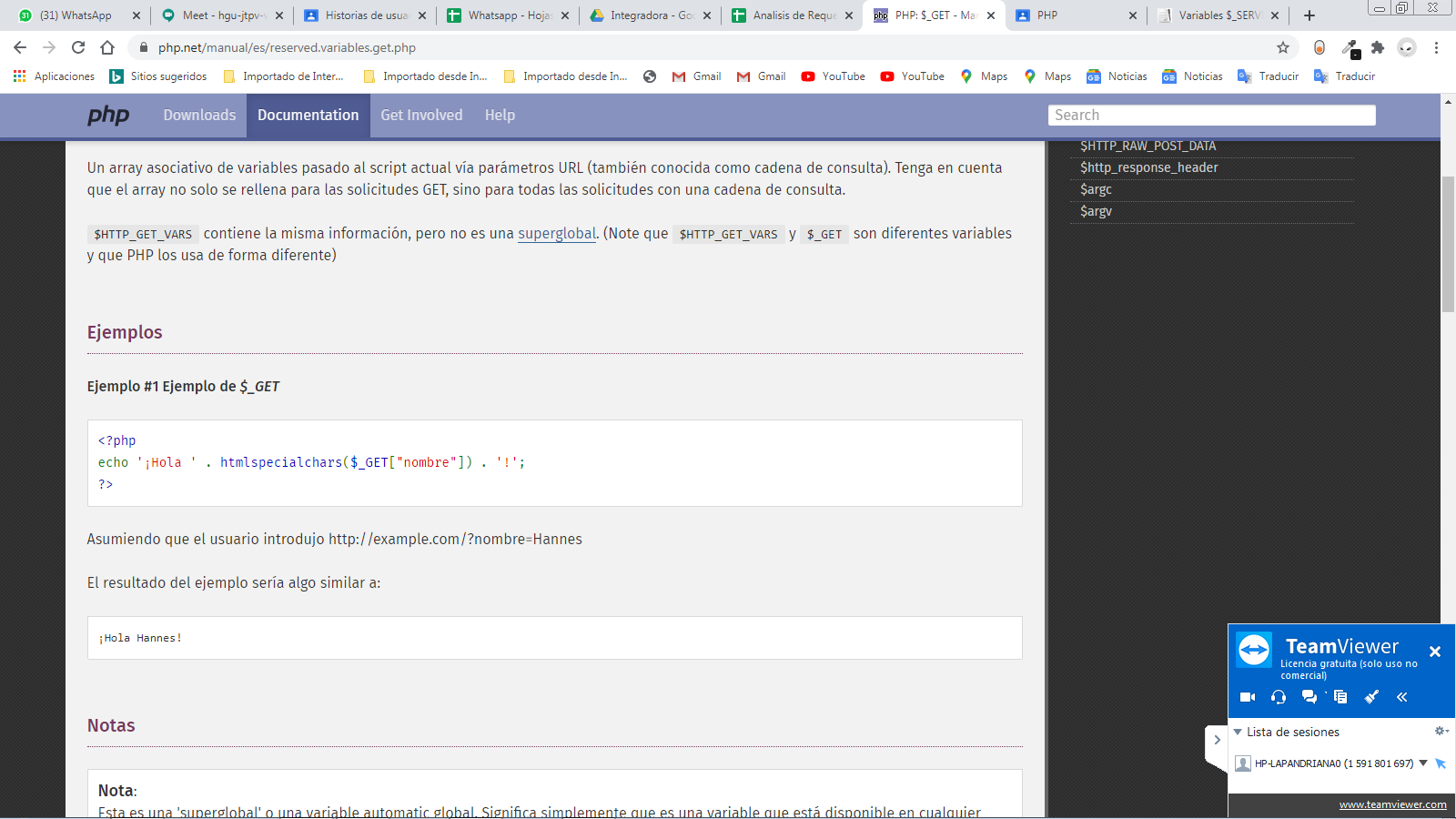
* **$Get\_:**

Un array asociativo de variables pasado al script actual vía parámetros URL (también conocida como cadena de consulta). Tenga en cuenta que el array no solo se rellena para las solicitudes GET, sino para todas las solicitudes con una cadena de consulta.

$HTTP\_GET\_VARS contiene la misma información, pero no es una [superglobal](https://www.php.net/manual/es/language.variables.superglobals.php).

$HTTP\_GET\_VARS y $\_GET son diferentes variables y que PHP los usa de forma diferente).

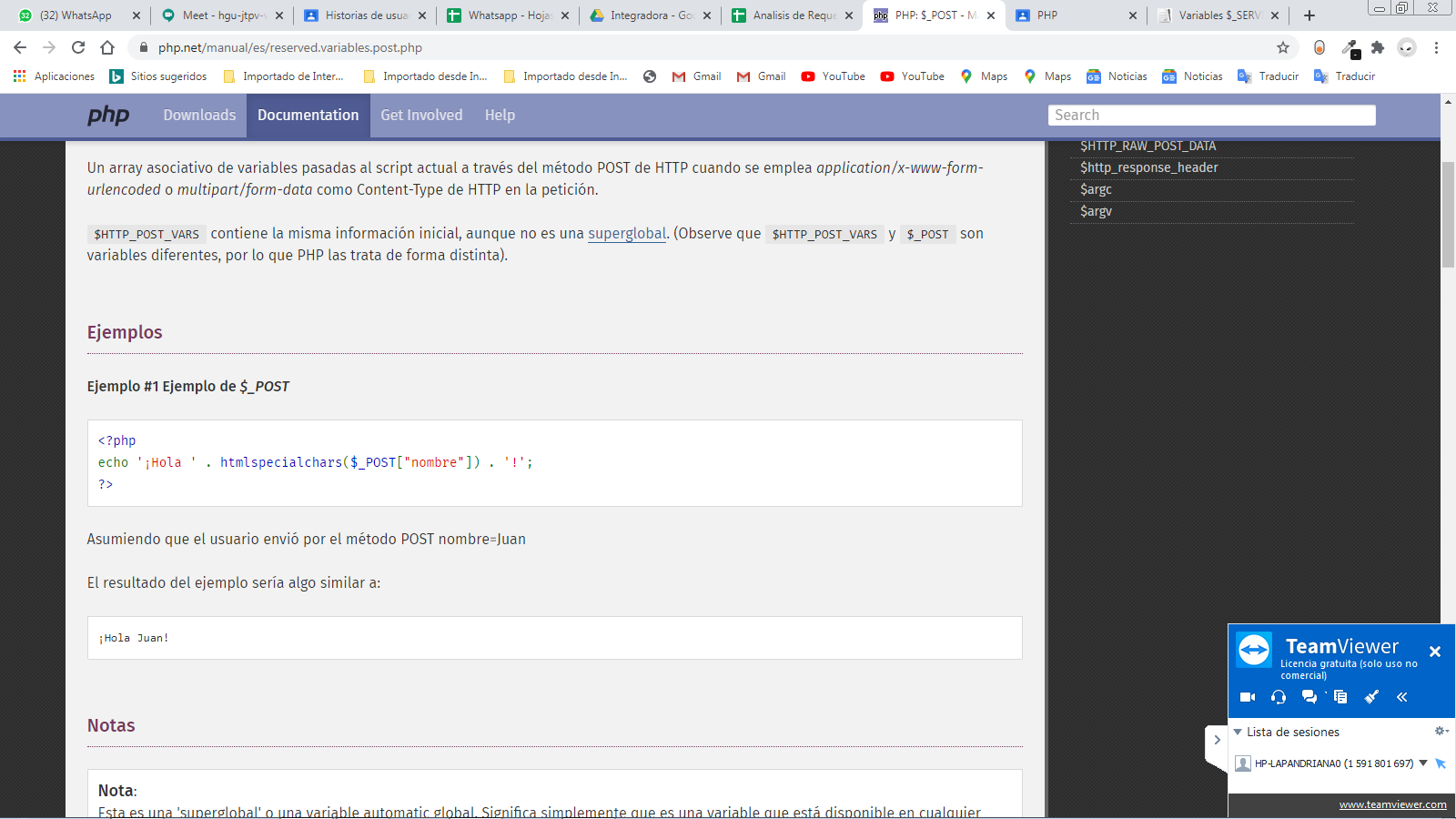


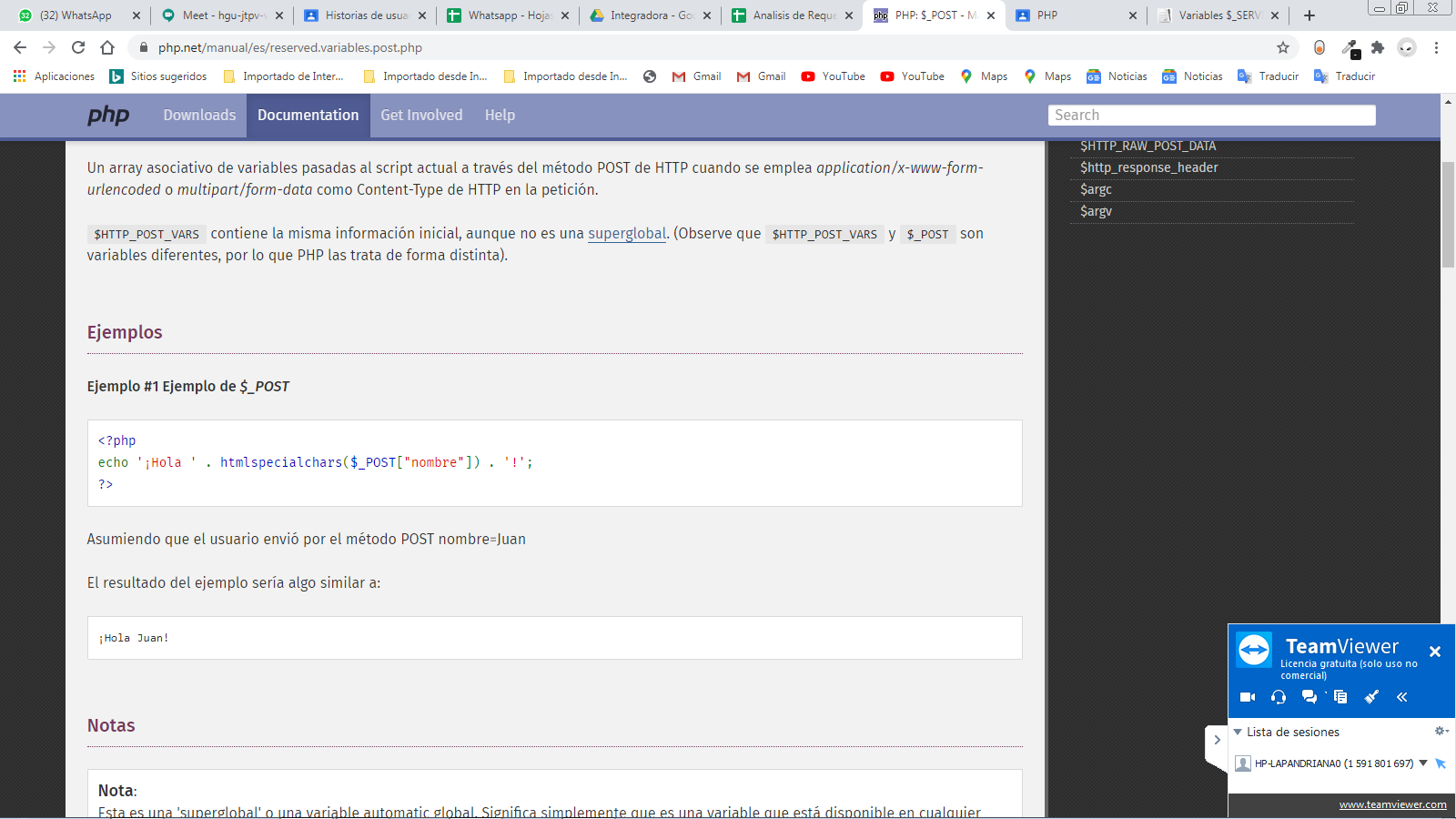


* **$Post:**

Un array asociativo de variables pasadas al script actual a través del método POST de HTTP cuando se emplea application/x-www-form-urlencoded o multipart/form-data como Content-Type de HTTP en la petición.

$HTTP\_POST\_VARS contiene la misma información inicial, aunque no es una [superglobal](https://www.php.net/manual/es/language.variables.superglobals.php).  $HTTP\_POST\_VARS y $\_POST son variables diferentes, por lo que PHP las trata de forma distinta.

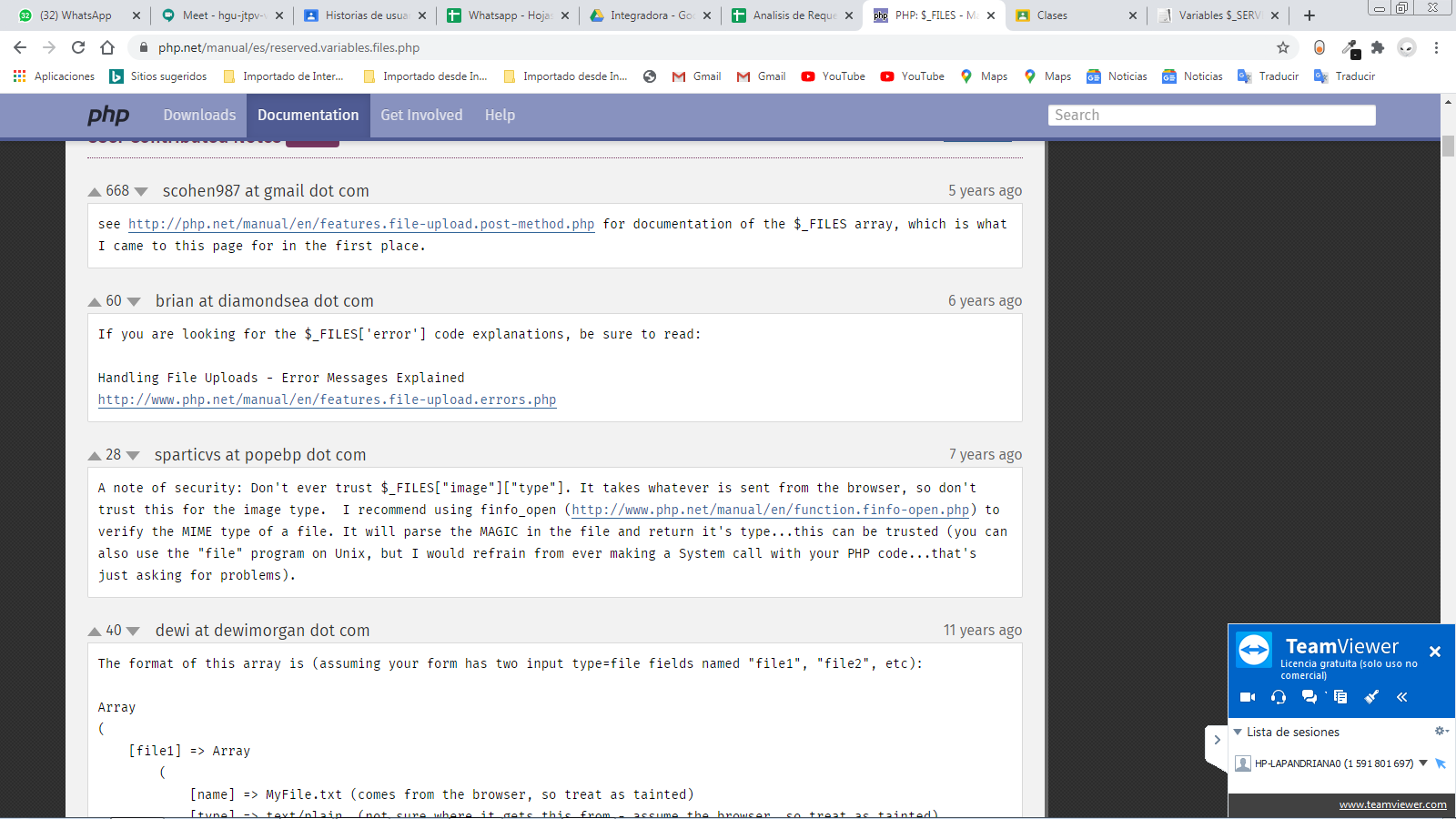




* **$Files:**

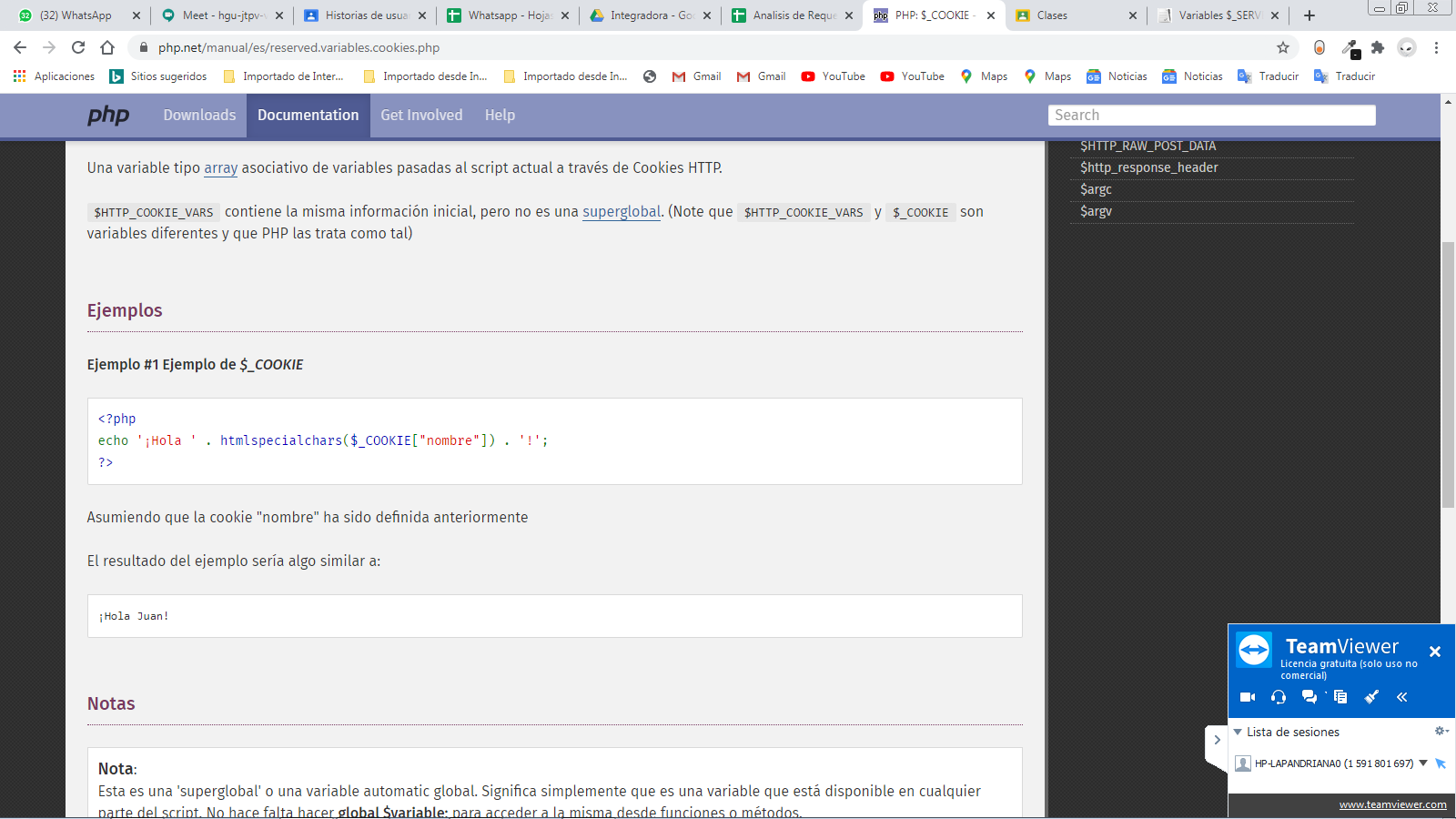
Un [array](https://www.php.net/manual/es/language.types.array.php) asociativo de elementos subidos al script en curso a través del método POST. La estructura de este array se resume en la sección [Subidas con el método POST](https://www.php.net/manual/es/features.file-upload.post-method.php).

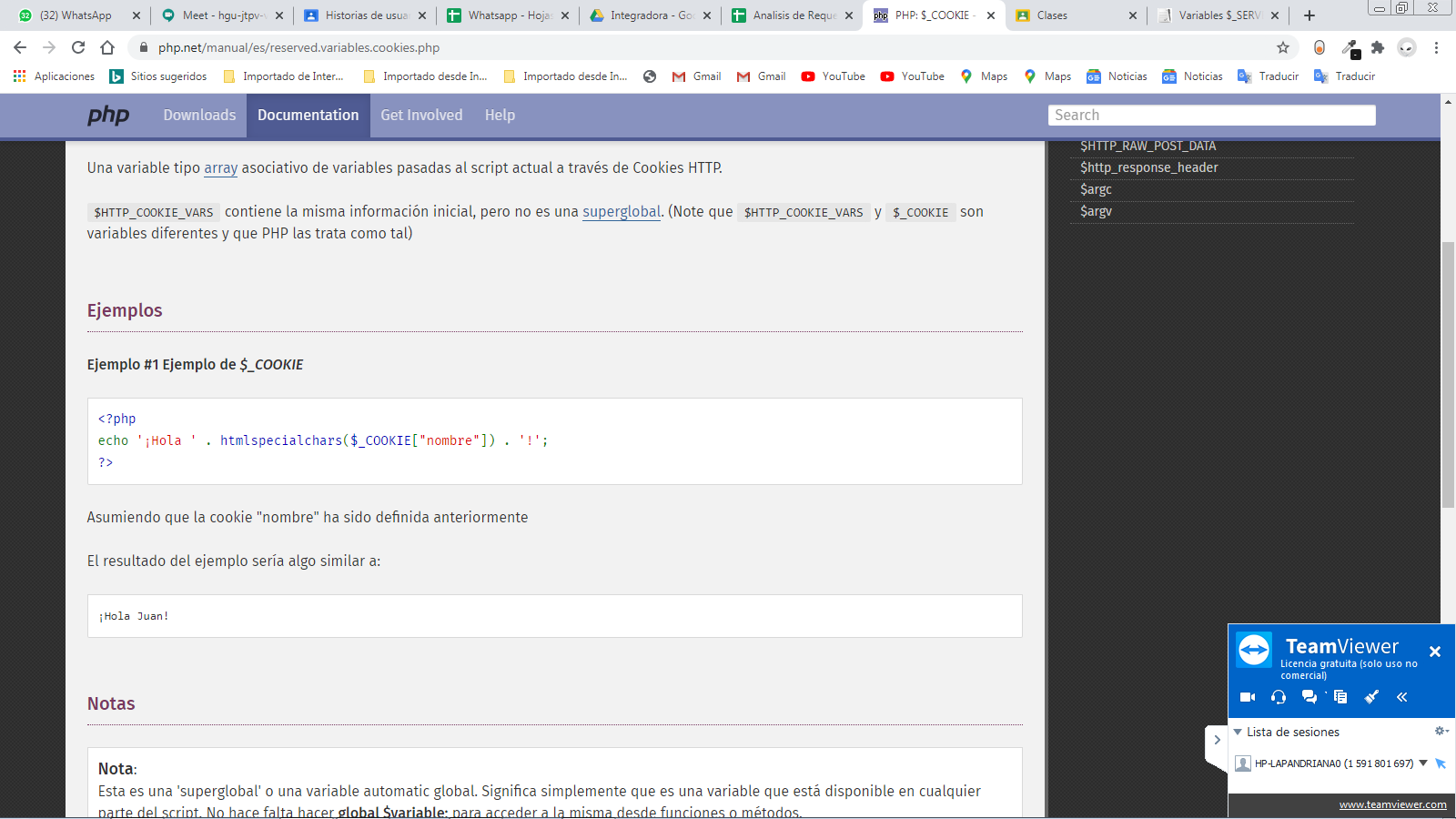
$HTTP\_POST\_FILES contiene la misma información inicial, pero no es [superglobal](https://www.php.net/manual/es/language.variables.superglobals.php).  $HTTP\_POST\_FILES y $\_FILES son variables diferentes y que PHP las trata como tal).



* **Cookie:**

Una variable tipo [array](https://www.php.net/manual/es/language.types.array.php) asociativo de variables pasadas al script actual a través de Cookies HTTP. $HTTP\_COOKIE\_VARS contiene la misma información inicial, pero no es una [superglobal](https://www.php.net/manual/es/language.variables.superglobals.php). $HTTP\_COOKIE\_VARS y $\_COOKIE son variables diferentes y que PHP las trata como tal.





* **$Session:**

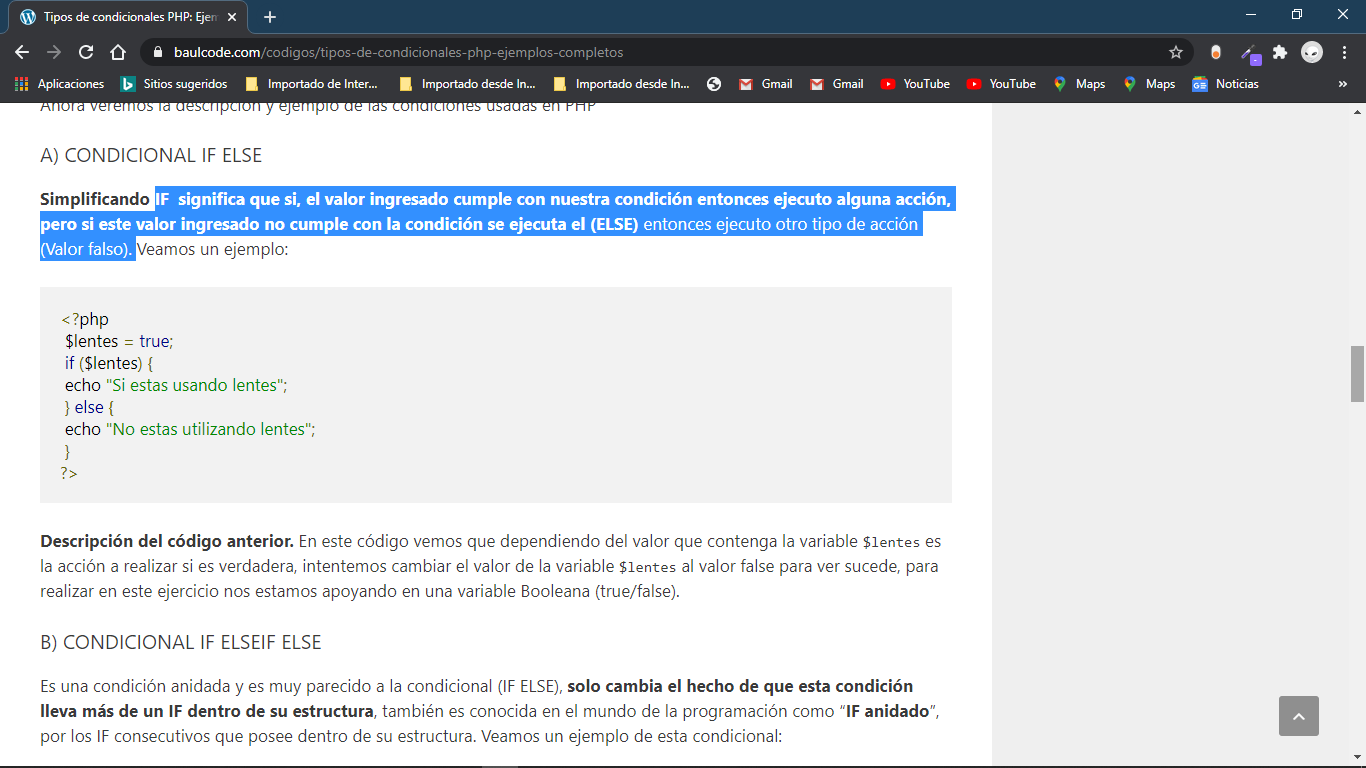
Es un array asociativo que contiene variables de sesión disponibles para el script actual. Ver la documentación de [Funciones de sesión](https://www.php.net/manual/es/ref.session.php) para más información sobre su uso. $HTTP\_SESSION\_VARS contiene la misma información inicial pero no es una [superglobal](https://www.php.net/manual/es/language.variables.superglobals.php) $HTTP\_SESSION\_VARS y $\_SESSION son diferentes variables y PHP las trata de forma distinta.

# Condicionales

Las condiciones ya que cuando tengas un valor “x” tú decides hacia donde debe de ir ese elemento, aquí les mostrare una lista de las principales condiciones existentes en PHP:

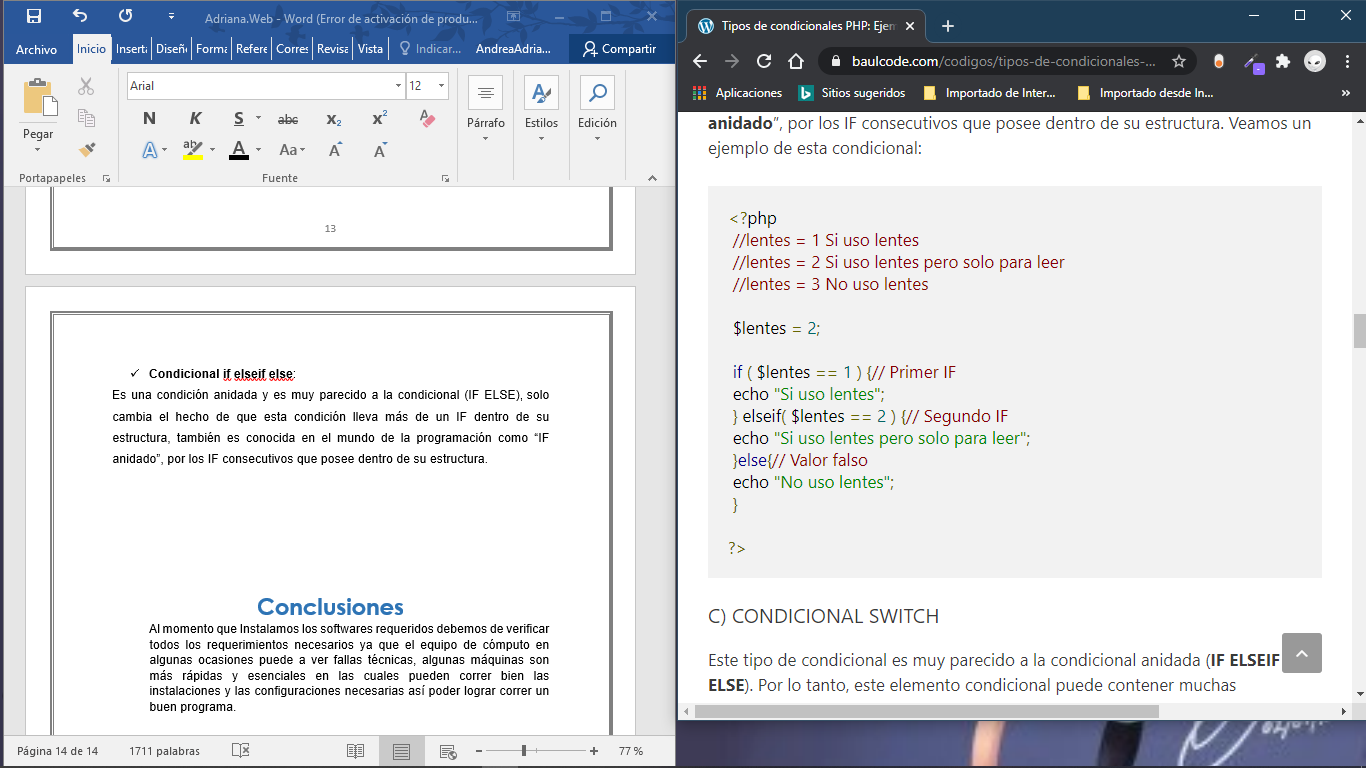
* Condicional If else
* La condicional If elseif else
* Instrucción switch
* Condicional Ternaria
* **Condicional if else:**

IF significa que si, el valor ingresado cumple con nuestra condición entonces ejecuto alguna acción, pero si este valor ingresado no cumple con la condición se ejecuta el (ELSE) entonces ejecuto otro tipo de acción (Valor falso).



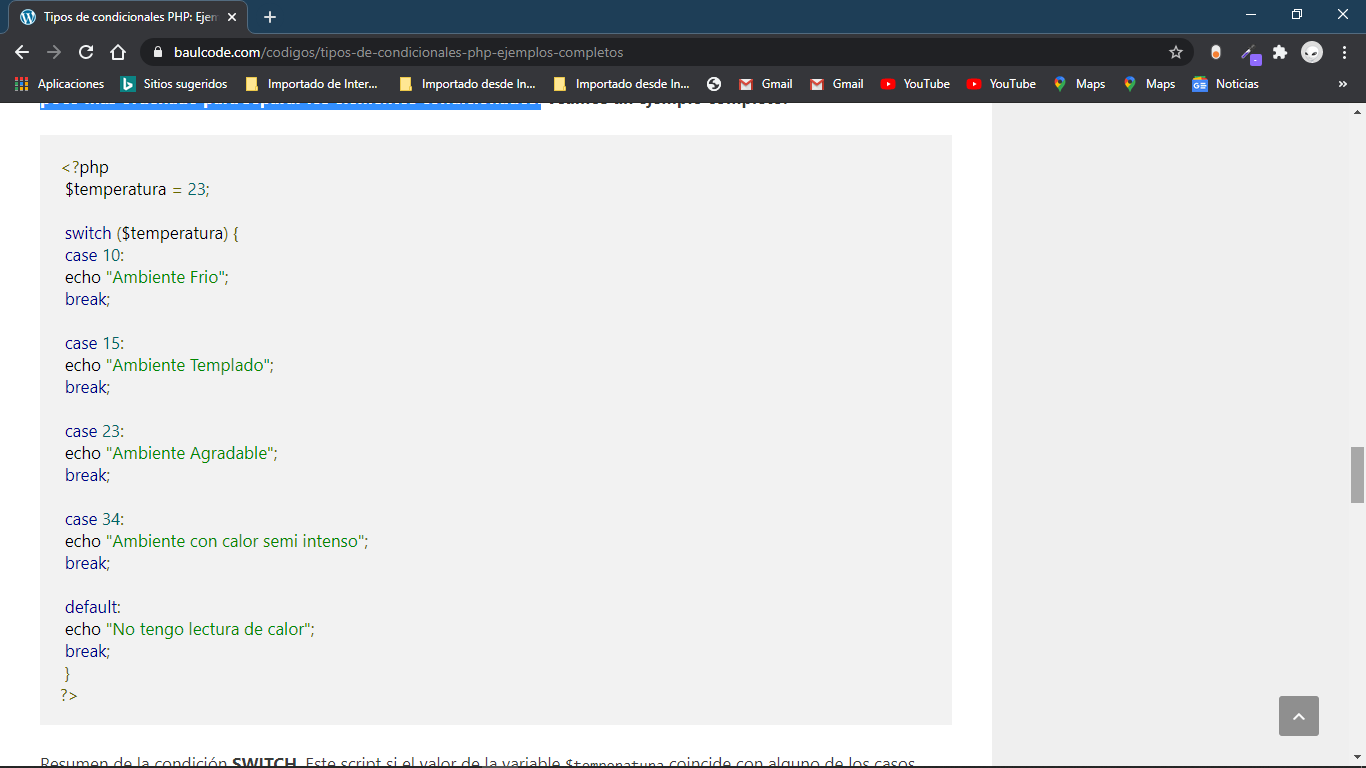
* **Condicional if elseif else**:

Es una condición anidada y es muy parecido a la condicional (IF ELSE), solo cambia el hecho de que esta condición lleva más de un IF dentro de su estructura, también es conocida en el mundo de la programación como “IF anidado”, por los IF consecutivos que posee dentro de su estructura.



* **Condicional con Switch:**

Este tipo de condicional es muy parecido a la condicional anidada (IF ELSEIF ELSE). Por lo tanto, este elemento condicional puede contener muchas condiciones dentro de ella de manera muy sencilla apoyadas de una variable, es un poco más ordenado para separar los elementos condicionados.

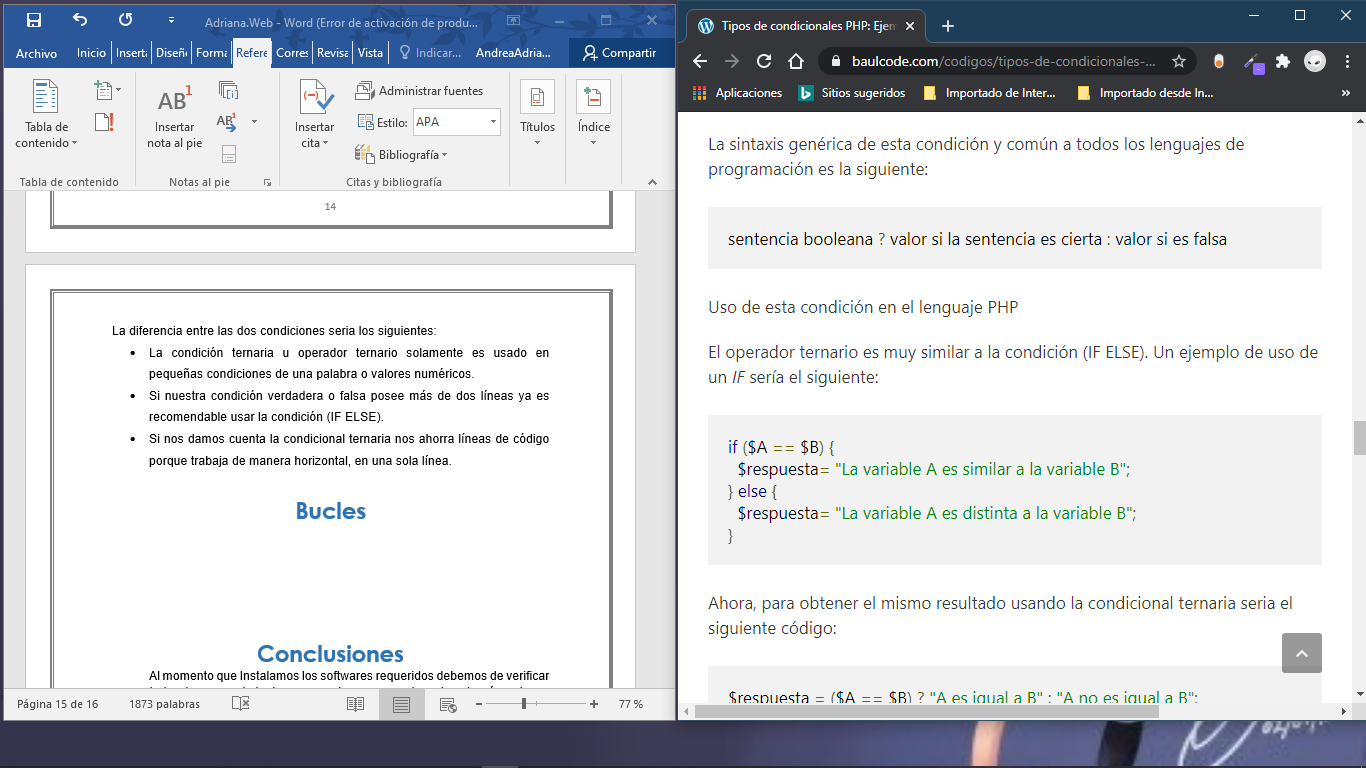


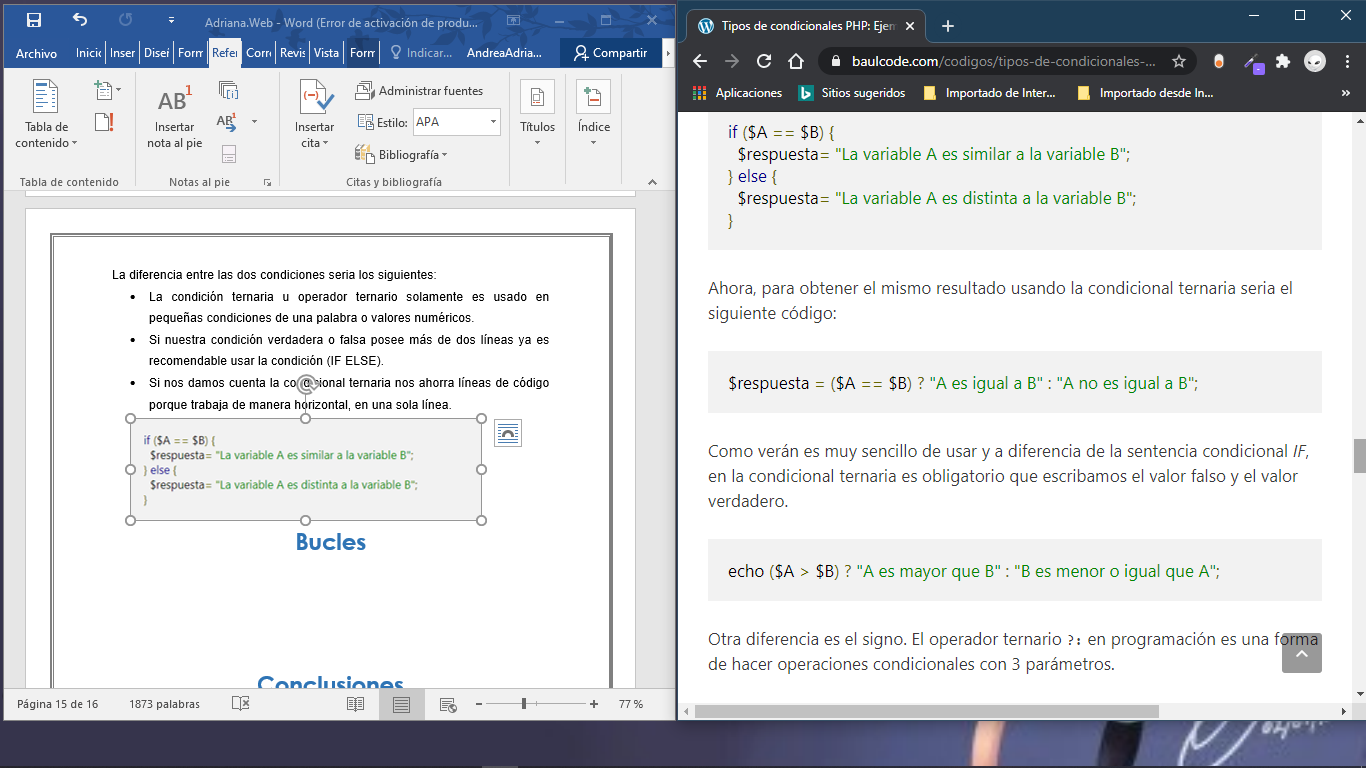
* **Condicional Ternaria:**

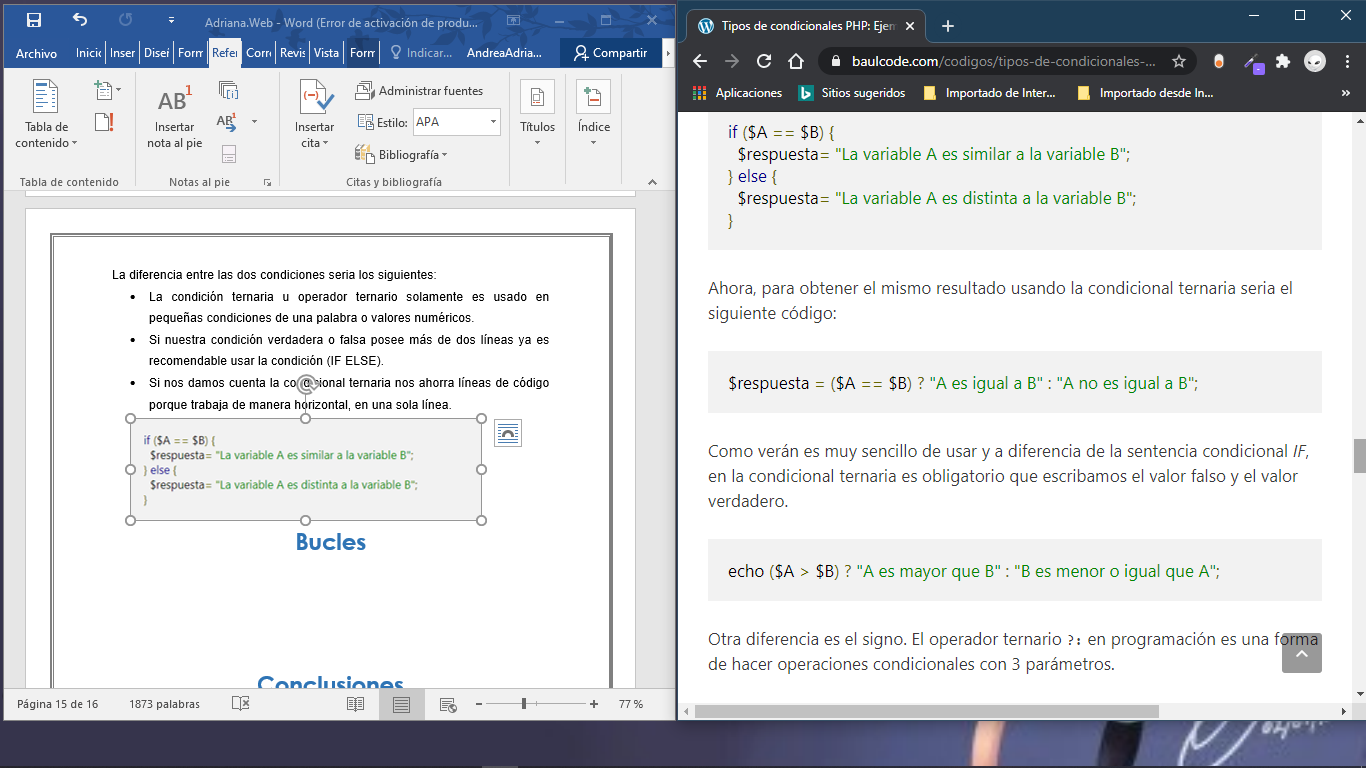
La condicional ternaria existe en muchos lenguajes de programación y sirve para que ejecute una sentencia si se cumple la condición u otra si no se cumple. Por lo tanto, es muy similar a la condición (IF), salvo algunas diferencias que posee.

La diferencia entre las dos condiciones seria los siguientes:

* La condición ternaria u operador ternario solamente es usado en pequeñas condiciones de una palabra o valores numéricos.
* Si nuestra condición verdadera o falsa posee más de dos líneas ya es recomendable usar la condición (IF ELSE).
* Si nos damos cuenta la condicional ternaria nos ahorra líneas de código porque trabaja de manera horizontal, en una sola línea.







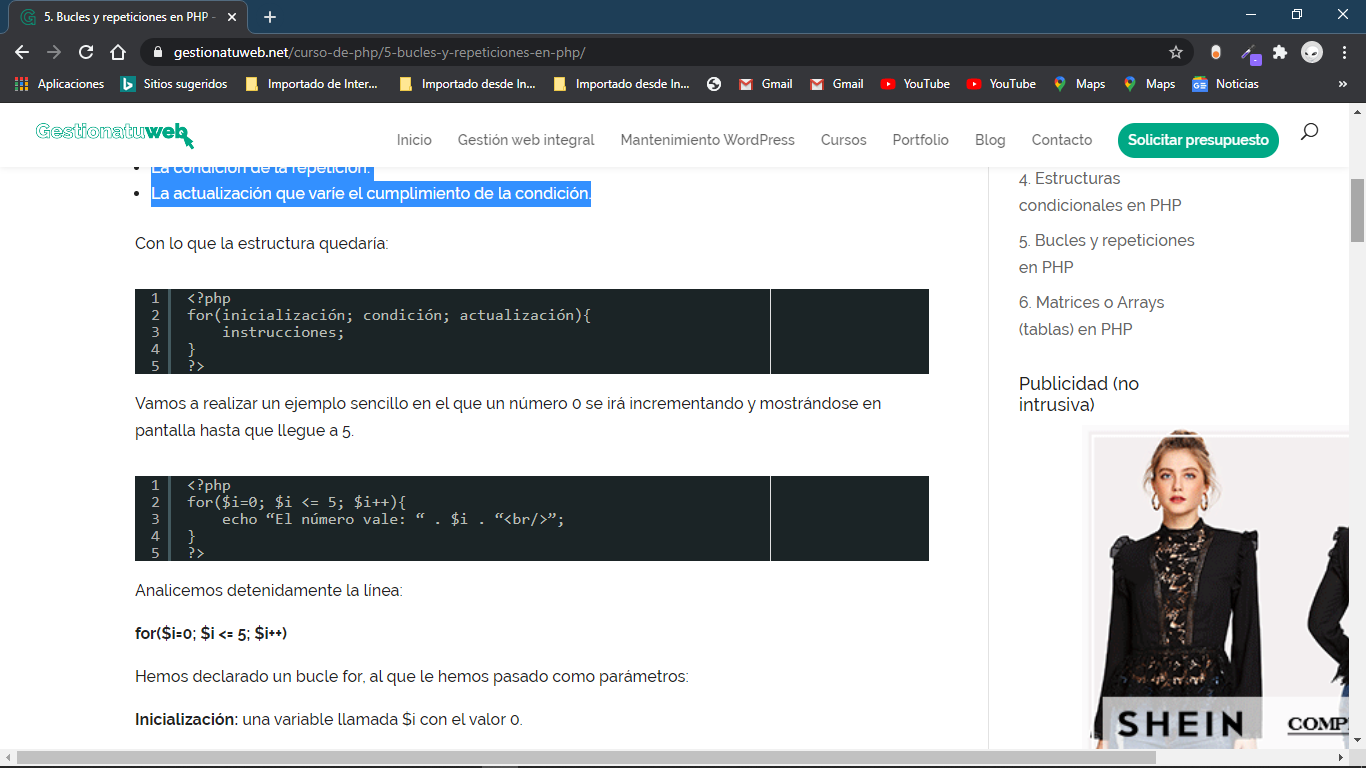
# Bucles

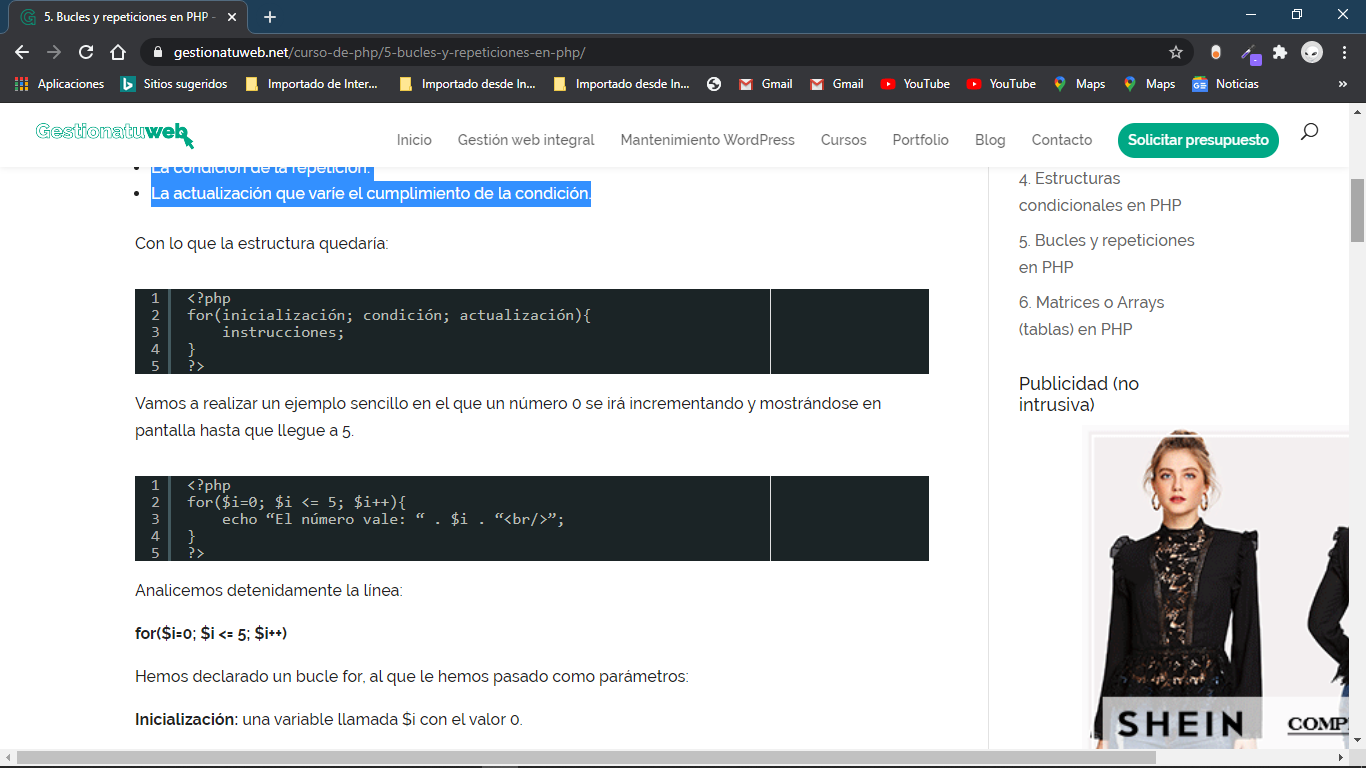
Los bucles son una única instrucción podemos hacer que se lean todos los registros de una base de datos y que para cada registro se monte una fila propia en HTML.

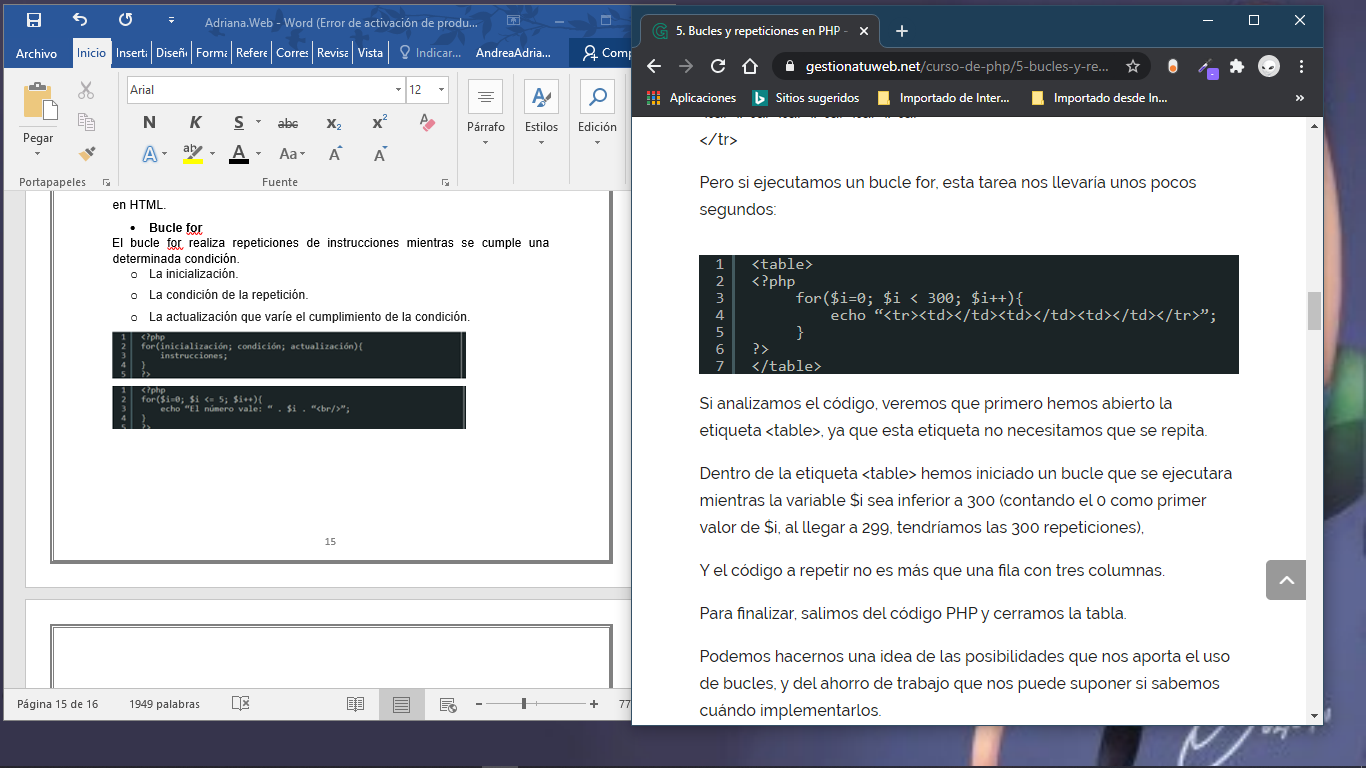
* **Bucle for**

El bucle for realiza repeticiones de instrucciones mientras se cumple una determinada condición.

* La inicialización.
* La condición de la repetición.
* La actualización que varíe el cumplimiento de la condición.





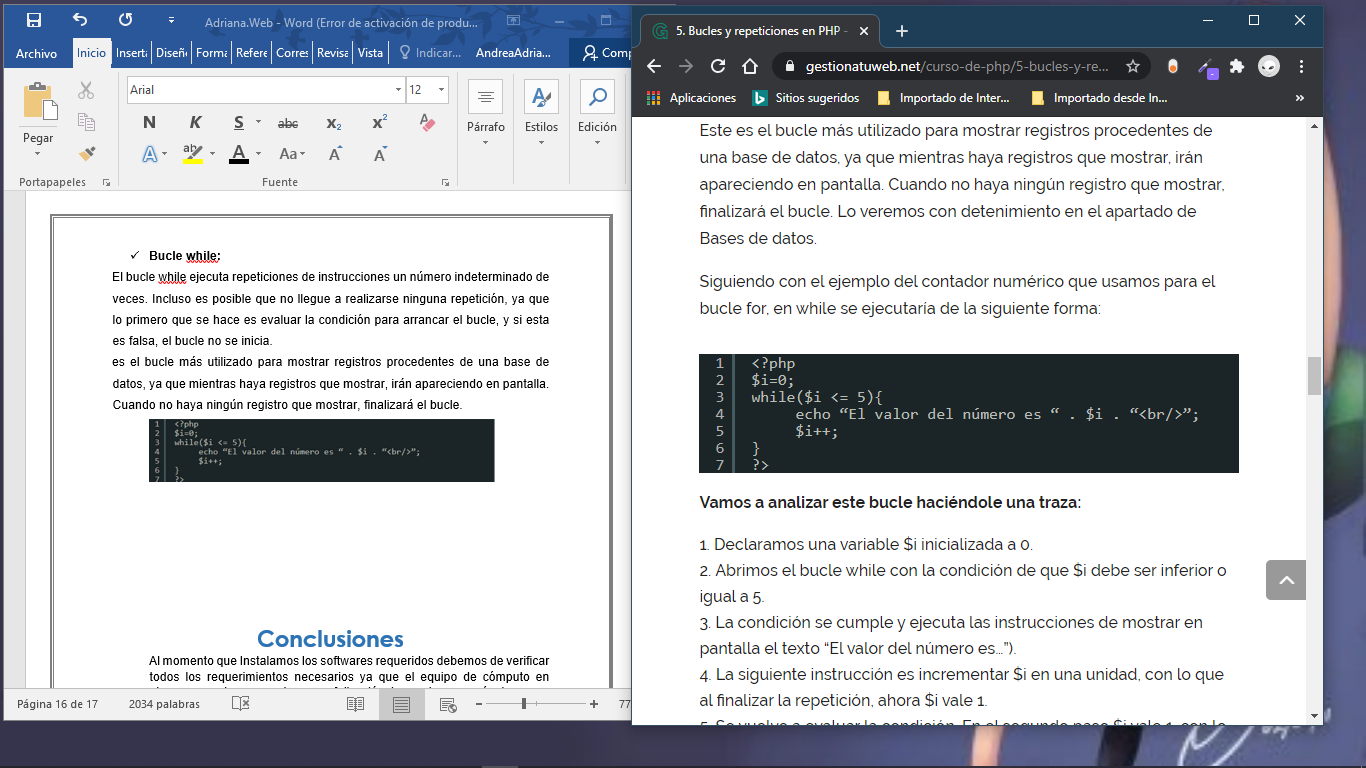


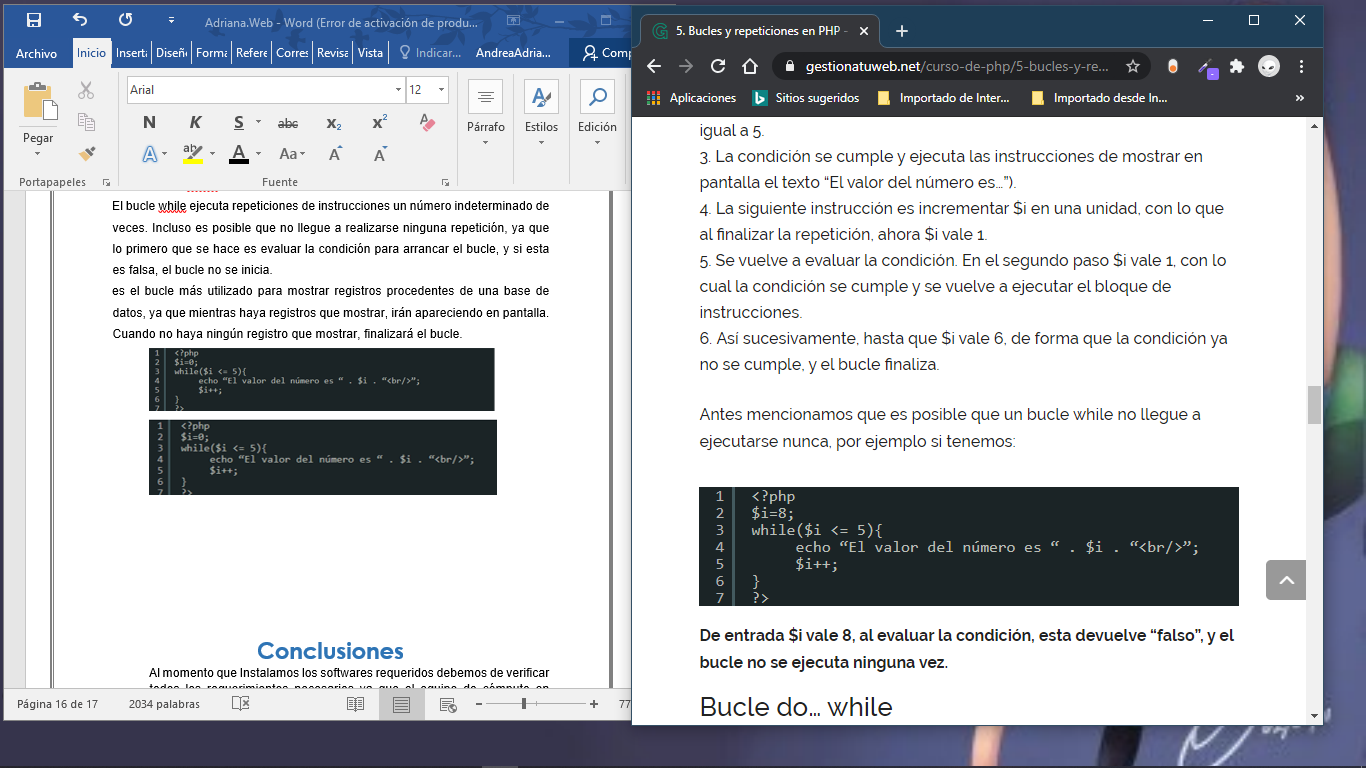
* **Bucle while:**

El bucle while ejecuta repeticiones de instrucciones un número indeterminado de veces. Incluso es posible que no llegue a realizarse ninguna repetición, ya que lo primero que se hace es evaluar la condición para arrancar el bucle, y si esta es falsa, el bucle no se inicia.

es el bucle más utilizado para mostrar registros procedentes de una base de datos, ya que mientras haya registros que mostrar, irán apareciendo en pantalla. Cuando no haya ningún registro que mostrar, finalizará el bucle.

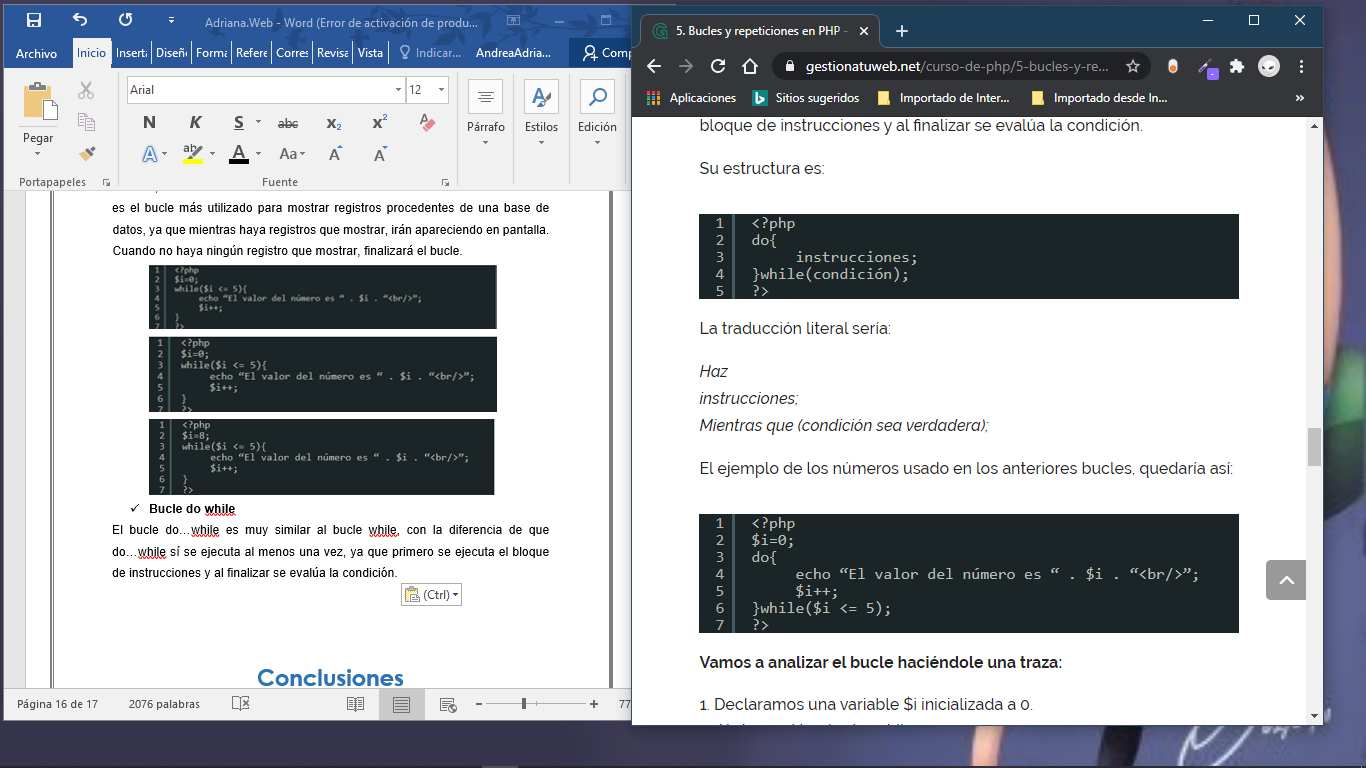


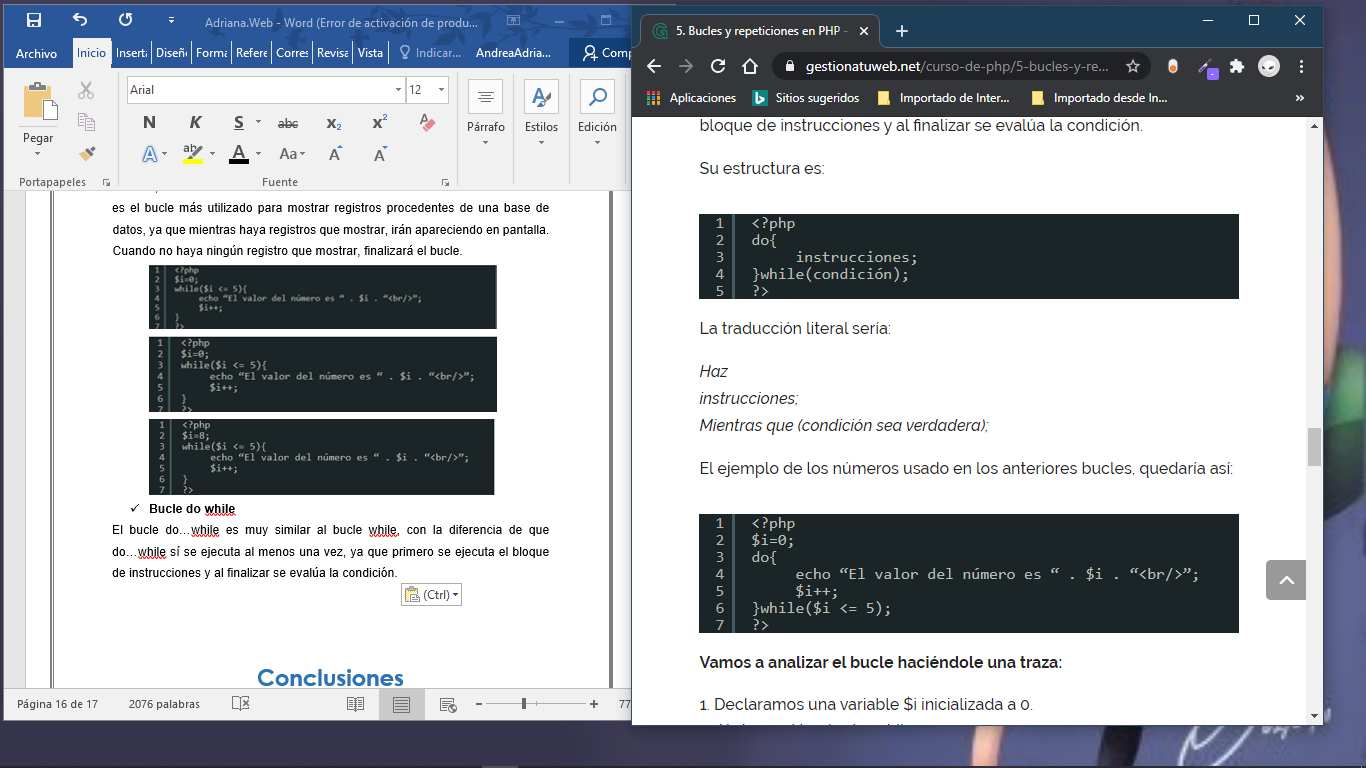




* **Bucle do while**

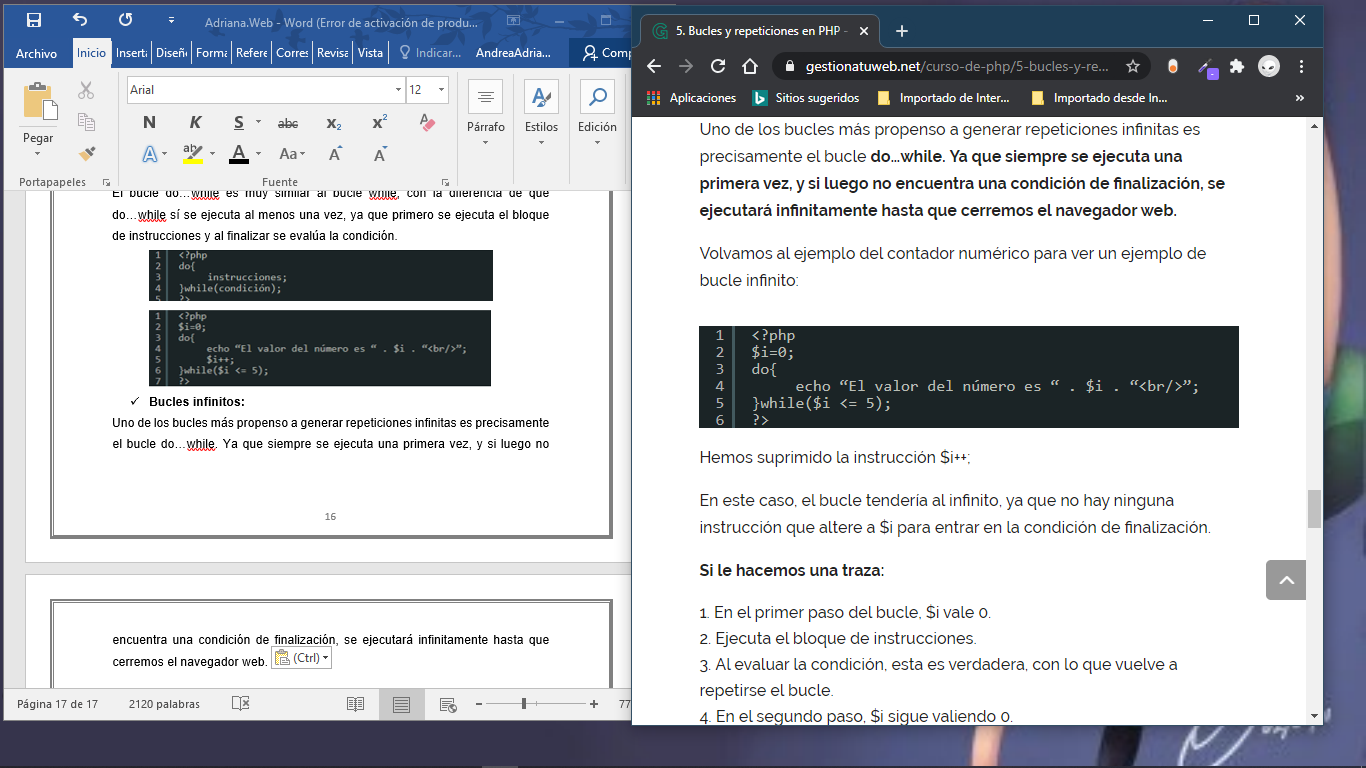
El bucle do…while es muy similar al bucle while, con la diferencia de que do…while sí se ejecuta al menos una vez, ya que primero se ejecuta el bloque de instrucciones y al finalizar se evalúa la condición.





* **Bucles infinitos:**

Uno de los bucles más propenso a generar repeticiones infinitas es precisamente el bucle do…while. Ya que siempre se ejecuta una primera vez, y si luego no encuentra una condición de finalización, se ejecutará infinitamente hasta que cerremos el navegador web.



# Patrón MVC para aplicaciones web con PHP

**Que es:**

El paradigma modelo vista controlador (MVC) es un patrón de diseño que separa el código en tres capas. Utilizar un patrón de diseño a menudo es muy recomendable, ya que es una forma estandarizar nuestro código, optimizarlo y hacer que sea más legible.

* **Modelo**

Gestiona todo lo relacionado con la información y la iteración con los datos de nuestra aplicación. Toda la petición de acceso a los datos pasará por esta capa.

* **Controlador**

Une la vista y el modelo. El usuario solicitará información por medio de la vista y esta hará la petición al controlador. Posteriormente, este, realizará la petición al modelo.

* **Vista**

Es la capa que nos mostrará la información formateada. También desde dónde el usuario puede solicitar más información.

* **Aplicación del patrón MVC en PHP**

En este caso no realiza otra tarea, sólo incluir e iniciar el FrontController**.**

**Ejemplo :**

< ?php  
//Incluimos el FrontController  
require 'libs/FrontController.php';  
//Lo iniciamos con su método estático main.  
FrontController::main();  
?>

< ?php

class FrontController

{

static function main()

{

//Incluimos algunas clases:

require 'libs/Config.php'; //de configuracion

require 'libs/SPDO.php'; //PDO con singleton

require 'libs/View.php'; //Mini motor de plantillas

require 'config.php'; //Archivo con configuraciones.

//Con el objetivo de no repetir nombre de clases, nuestros controladores

//terminarán todos en Controller. Por ej, la clase controladora Items, será ItemsController

//Formamos el nombre del Controlador o en su defecto, tomamos que es el IndexController

if(! empty($\_GET['controlador']))

$controllerName = $\_GET['controlador'] . 'Controller';

else

$controllerName = "IndexController";

//Lo mismo sucede con las acciones, si no hay acción, tomamos index como acción

if(! empty($\_GET['accion']))

$actionName = $\_GET['accion'];

else

$actionName = "index";

$controllerPath = $config->get('controllersFolder') . $controllerName . '.php';

//Incluimos el fichero que contiene nuestra clase controladora solicitada

if(is\_file($controllerPath))

require $controllerPath;

else

die('El controlador no existe - 404 not found');

//Si no existe la clase que buscamos y su acción, mostramos un error 404

if (is\_callable(array($controllerName, $actionName)) == false)

{

trigger\_error ($controllerName . '->' . $actionName . '` no existe', E\_USER\_NOTICE);

return false;

}

//Si todo esta bien, creamos una instancia del controlador y llamamos a la acción

$controller = new $controllerName();

$controller->$actionName();

}

}

?>

# Reflexión

En la investigación que se realizo fue para analizar para que sirve cada uno de los de las descripciones como puede darse las variables, operadores, superglobal, condicionales, bucles y al igual como como se aplica el patrón de MVC en PHP, dado que cada una de las explicaciones de ellas se dan a notar por cada una de ellas las explicaciones y ejemplos

# Bibliografía

Andalucia, J. d. (s.f.). *Marco de Desarrollo de la Junta Andalucia*. Obtenido de Marco de Desarrollo de la Junta Andalucia: http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/257

Baulcode. (2019 de Febrero de 2019). *BaulCode*. Obtenido de BaulCode: https://baulcode.com/codigos/tipos-de-condicionales-php-ejemplos-completos

Marco, B. S. (5 de Octubre de 2017). *Progrmacion web en PHP*. Obtenido de PHP: https://www.mclibre.org/consultar/php/lecciones/php-variables-2.html

PHP. (14 de Mayo de 2013). *PHP*. Obtenido de PHP: https://www.php.net/manual/es/language.variables.superglobals.php

Redactor. (10 de Enero de 2017). *Ada web*. Obtenido de Ada web: https://www.adaweb.es/modelo-vista-controlador-mvc-php/#:~:text=Ejemplo%20del%20paradigma%20modelo%20vista%20controlador%20en%20PHP&text=El%20paradigma%20modelo%20vista%20controlador,hacer%20que%20sea%20m%C3%A1s%20legible.

Sánchez, D. (25 de junio de 2016). *Gestionatuweb*. Obtenido de Gestionatuweb: https://www.gestionatuweb.net/curso-de-php/5-bucles-y-repeticiones-en-php/

WIKIPEDIA. (1995). *WIKIPEDIA*. Obtenido de WIKIPEDIA: https://es.wikipedia.org/wiki/PHP