

Plataformas de programación web en entorno servidor. Aplicaciones LAMP.

Caso práctico



En la empresa **BK Programación** están a punto de hacerse cargo de un importante proyecto. **Ada**, la directora, va a comprometerse con un cliente y amigo, **Esteban**, que necesita ayuda con un problema concreto.

Esteban fundó hace ahora tres años **una pequeña empresa** dedicada a la venta de material tecnológico: cámaras, televisores, material informático, etc. Con el tiempo, esa pequeña empresa ha crecido. El número de ventas ha aumentado, así como el catálogo de productos que ofrece, e incluso ha abierto dos nuevas tiendas en localidades cercanas.

Pero como sucede muchas veces, **al aumentar el negocio** han ido surgiendo ciertas **necesidades**. El proyecto que Esteban le ha propuesto a Ada consiste en **desarrollar una web**. No una página web que explique dónde está la empresa o qué hace. Quiere una web enfocada a dos temas concretos: mejorar la **comunicación con sus clientes**, y que le aporte **información interna** a la empresa sobre su negocio.

Por ejemplo, quiere que los clientes puedan ver desde su casa los productos que vende, el precio de los mismos, o la disponibilidad en una u otra tienda. O que los empleados de la propia empresa puedan ver de forma sencilla el stock que tienen de los productos en las distintas sucursales, para poder decidir mejor qué productos se piden a los distribuidores y en qué cantidad.

Sin embargo, tal y como Ada ya le ha comentado a Esteban, la experiencia de BK Programación en el desarrollo de aplicaciones web es muy reducida. La mayoría de proyectos que han realizado hasta el momento se han centrado en aplicaciones para plataformas Windows y Linux, o para dispositivos móviles. Sólo uno de sus empleados, **Juan**, tiene cierta experiencia con la **programación web**.

Aun así, Esteban confía en que Ada y su empresa lograrán llevar a cabo el proyecto. No tiene prisa por ponerlo en funcionamiento, y sabe que a BK Programación le servirá para ir formando a sus empleados en temas relacionados en el

desarrollo de aplicaciones web, por lo que finalmente llegan a un acuerdo.

En esta unidad se introducen los conceptos fundamentales en los que se basa el módulo. Se explica el concepto de ejecución de aplicaciones en entorno servidor, los componentes implicados, y la diferencia con las páginas web estáticas y con la ejecución de código en el navegador web.

Se analizan las principales tecnologías disponibles para el desarrollo de aplicaciones web, y los conceptos fundamentales de cada una.

Pues, vayamos a ello.



[Ministerio de Educación y Formación Profesional](#) (Dominio público)

Materiales formativos de FP Online propiedad del Ministerio de Educación y Formación Profesional.

[Aviso Legal](#)

1.- Características de la programación web.

Caso práctico



El nuevo proyecto al que se enfrenta **BK Programación** requiere que sus **empleados actualicen su formación** en temas relacionados con la **programación web**. **Juan**, el único con cierta experiencia en ese ámbito, ha **propuesto** a sus compañeros la realización de unas **jornadas** en las que explique a los demás los **fundamentos del desarrollo** de aplicaciones para la **web**. De esta forma, aquellos que quieran pasarán a formar parte del equipo que se dedique al nuevo proyecto.

Todos se apuntan a la **propuesta** de Juan, **incluyendo a Ada**, la directora.

El **primer objetivo** de las jornadas es ver en qué consiste la programación web. Por ejemplo, ver la **diferencia** existente entre hacer una **página web** y **programar una aplicación web**.

Seguro que **ya sabes** exactamente qué es una **página web**, e incluso conozcas cuáles son los pasos que se suceden para que, cuando visitas una web poniendo su dirección en el navegador, la página se descargue a tu equipo y se pueda mostrar. Sin embargo, este procedimiento que puede parecer sencillo, a veces no lo es tanto. Todo depende de cómo se haya hecho esa página.

Cuando una **página web se descarga** a tu ordenador, su contenido define qué se debe mostrar en pantalla. Este contenido está programado en un **lenguaje de marcado**, formado por etiquetas, que puede ser **HTML** o **XHTML**. Las etiquetas que componen la página indican el objetivo de cada una de las partes que la componen. Así, dentro de estos lenguajes hay etiquetas para indicar que un texto es un encabezado, que forma parte de una tabla, o que simplemente es un párrafo de texto.

Además, si la página está bien estructurada, la información que le indica al navegador el **estilo** con que se debe **mostrar cada parte de la página** estará almacenado en otro fichero, una **hoja de estilos o CSS**. La hoja de estilos se encuentra indicada en la página web y el navegador la descarga junto a ésta. En ella nos podemos encontrar, por ejemplo, estilos que indican que el encabezado debe ir con tipo de letra Arial y en color rojo, o que los párrafos deben ir alineados a la izquierda.

Estos dos ficheros se descargan a tu ordenador desde un servidor web como respuesta a una petición. El proceso es el que se refleja en la siguiente figura.

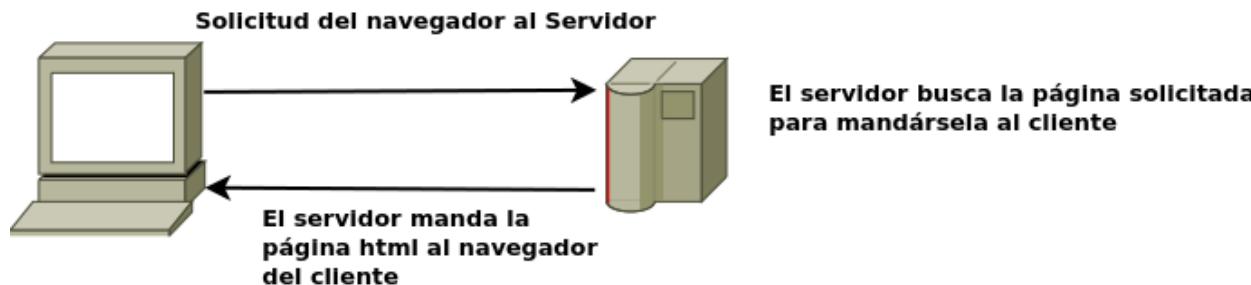


Ilustración realizada con Dia (Elaboración propia)

Los pasos son los siguientes:

- 1.- Un ordenador solicita a un servidor web una página con extensión .htm, .html O .xhtml.
- 2.- El servidor busca esa página en un almacén de páginas (cada una suele ser un fichero).
- 3.- Si el servidor encuentra esa página, la recupera.
- 4.- Y por último se la envía al navegador para que éste pueda mostrar su contenido.

Este es un ejemplo típico de una **comunicación cliente-servidor**. El cliente es el que hace la petición e inicia la comunicación, y el servidor es el que recibe la petición y la atiende. En nuestro caso, el navegador es el cliente web.

Autoevaluación

¿Podemos ver una página web sin que intervenga un servidor web?

- Sí.
- No.

Efectivamente. Podemos ver páginas web con extensión .htm, .html O .xhtml que tengamos almacenadas en nuestro equipo simplemente abriéndolas con el navegador. En este caso la única utilidad del servidor web es enviar la página que

solicitemos a nuestro equipo.

Piénsalo bien... ¿Cuál es realmente la utilidad del servidor web, según lo que acabamos de ver?

Solución

1. Opción correcta
2. Incorrecto

1.1.- Páginas web estáticas y dinámicas (I).

Las páginas que viste en el ejemplo anterior se llaman **páginas web estáticas**. Estas páginas se encuentran almacenadas en su forma definitiva, tal y como se crearon, y su contenido no varía. Son útiles para mostrar una información concreta, y mostrarán esa misma información cada vez que se carguen. La única forma en que pueden cambiar es si un programador la modifica y actualiza su contenido.

En contraposición a las páginas web estáticas, como ya te imaginarás, existen las **páginas web dinámicas**. Estas páginas, como su nombre indica, se caracterizan porque su contenido cambia en función de diversas variables, como puede ser el navegador que estás usando, el usuario con el que te has identificado, o las acciones que has efectuado con anterioridad.

Dentro de las **páginas web dinámicas**, es muy importante distinguir **dos tipos**:

- ✓ Aquellas que **incluyen código que ejecuta el navegador**. En estas páginas el código ejecutable, normalmente en **lenguaje JavaScript**, se incluye dentro del **HTML** (o **XHTML**) y se descarga junto con la página. Cuando el navegador muestra la página en pantalla, ejecuta el código que la acompaña. Este código puede incorporar múltiples funcionalidades que pueden ir desde mostrar animaciones hasta cambiar totalmente la apariencia y el contenido de la página. En este módulo no vamos a ver JavaScript, salvo cuando éste se relaciona con la programación web del lado del servidor.
- ✓ Como ya sabes, hay muchas páginas en Internet que no tienen extensión **.htm**, **.html** o **.xhtml**. Muchas de estas páginas tienen extensiones como **.php**, **.asp**, **.jsp**, **.cgi** o **.aspx**. En éstas, el contenido que se descarga al navegador es similar al de una página web estática: **HTML** (o **XHTML**). Lo que cambia es la forma en que se obtiene ese contenido. Al contrario de lo que vimos hasta ahora, esas páginas no están almacenadas en el servidor; más concretamente, el contenido que se almacena no es el mismo que después se envía al navegador. **El HTML de estas páginas se forma como resultado de la ejecución de un programa**, y esa ejecución tiene lugar en el servidor web (aunque no necesariamente por ese mismo servidor).

El esquema de funcionamiento de una página web dinámica es el siguiente:

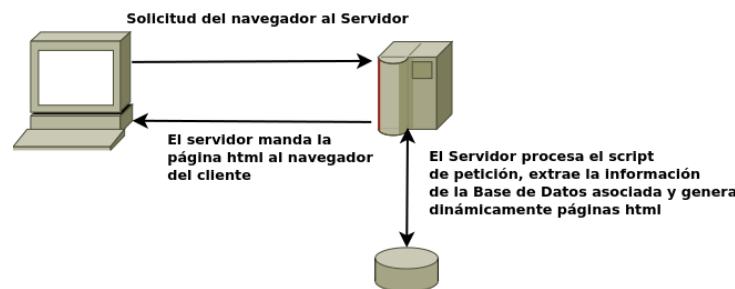


Ilustración realizada con Dia (Elaboración propia)

Pasos:

- 1.- **El cliente web** (navegador) de tu ordenador **solicita** a un servidor web una **página web** con extensión asp, php, jsp....
- 2.- El **servidor busca** esa página y la recupera.
- 3.- En el caso de que se trate de una **página web dinámica**, es decir, que su contenido deba ejecutarse para obtener el HTML que se devolverá, el **servidor web** contacta con el módulo responsable de **ejecutar el código** y se lo envía.
- 4.- Como parte del **proceso de ejecución**, puede ser necesario obtener información de algún **repositorio**, como por ejemplo consultar registros almacenados en una base de datos.
- 5.- **El resultado** de la ejecución será una página en **formato HTML**, similar a cualquier otra página web no dinámica.
- 6.- El **servidor web envía el resultado** obtenido al navegador, que la procesa y muestra en pantalla.

Este procedimiento tiene lugar constantemente mientras consultamos páginas web. Por ejemplo, cuando consultas tu correo en GMail, Outlook, Yahoo o cualquier otro servicio de correo vía web, lo primero que tienes que hacer es introducir tu nombre de usuario y contraseña. A continuación, lo más habitual es que el servidor te muestre una pantalla con la bandeja de entrada, en la que aparecen los mensajes recibidos en tu cuenta. Esta pantalla es un claro ejemplo de una página web dinámica.

Obviamente, el navegador no envía esa misma página a todos los usuarios, sino que la **genera de forma dinámica** en función de quién sea el usuario que se conecte. Para generarla ejecuta un programa que obtiene los datos de tu usuario (tus contactos, la lista de mensajes recibidos) y con ellos compone la página web que recibes desde el servidor web.

Volvamos a lo mismo con una página php dinámica, podemos ver los pasos siguientes:

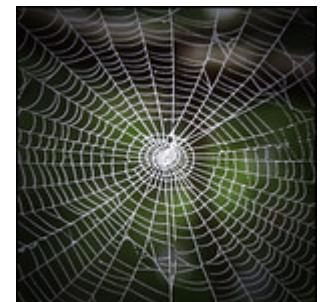
- 1.- El Cliente hace una petición al servidor
- 2.- El servidor tendrá en el servidor web con un "motor" PHP incorporado para procesar esta solicitud
- 3.- El "motor" procesa los scripts de la página en cuestión y genera el código HTML correspondiente
- 4.- El servidor manda la página HTML al cliente



1.1.1.- Páginas web estáticas y dinámicas (II).

Aunque la utilización de páginas web dinámicas te parezca la mejor opción para construir un sitio web, no siempre lo es. Sin lugar a dudas, es la que más potencia y flexibilidad permite, pero las páginas **web estáticas** tienen también **algunas ventajas**:

- ✓ **No es necesario saber programar** para crear un sitio que utilice únicamente páginas web estáticas. Simplemente habría que conocer HTML, XHTML y CSS, e incluso esto no sería indispensable: se podría utilizar algún programa de diseño web para generarlas.
- ✓ La característica diferenciadora de las páginas web estáticas es que **su contenido nunca varía**, y esto en algunos casos también **puede suponer una ventaja**. Sigue, por ejemplo, cuando quieras almacenar un enlace a un contenido concreto del sitio web: si la página es dinámica, al volver a visitarla utilizando el enlace su contenido puede variar con respecto a cómo estaba con anterioridad. O cuando quieras dar de alta un sitio que has creado en un motor de búsqueda como Google.



[Roberto Latxaga \(CC BY\)](#)

Para que Google muestre un sitio web en sus resultados de búsqueda, previamente tiene que indexar su contenido. Es decir, un programa recorre las páginas del sitio consultando su contenido y clasificándolo. Si las páginas se generan de forma dinámica, puede ser que su contenido, en parte o por completo, no sea visible para el buscador y por tanto no quedará indexado. Esto nunca sucedería en un sitio que utilizase páginas web estáticas.

- ✓ Como ya sabes, para que un servidor web pueda procesar una página web dinámica, necesita ejecutar un programa. Esta ejecución la realiza un módulo concreto, que puede estar integrado en el servidor o ser independiente.

Además, puede ser necesario consultar una base de datos como parte de la ejecución del programa. Es decir, la ejecución de una página web dinámica requiere una serie de recursos del lado del servidor.

Estos recursos deben instalarse y mantenerse. **Las páginas web estáticas sólo necesitan un servidor web que se comunique con tu navegador** para enviártela. Y de hecho para ver una página estática almacenada en tu equipo no necesitas siquiera de un servidor web. Son archivos que pueden almacenarse en un soporte de almacenamiento como puede ser un disco óptico o una memoria USB y abrirse desde él directamente con un navegador web.

Pero si decides hacer un sitio web utilizando **páginas estáticas**, ten en cuenta que **tienen limitaciones**. La desventaja más importante ya la comentamos anteriormente: la **actualización** de su contenido debe hacerse de **forma manual** editando la página que almacena el servidor web. Esto implica un mantenimiento que puede ser prohibitivo en sitios web con gran cantidad de contenido.

Reflexiona

¿Qué tipo de tecnología emplearías para crear una página web personal? ¿Sería necesario utilizar páginas dinámicas? ¿Qué tareas de actualización y mantenimiento tendrías que realizar en cada caso?

1.1.2.- Aplicaciones web.

Las **aplicaciones web** emplean **páginas web dinámicas** para crear aplicaciones que se **ejecuten en un servidor web** y se muestren en un navegador. Puedes encontrar aplicaciones web para realizar múltiples tareas. Unas de las primeras en aparecer fueron las que viste antes, los clientes de correo, que te permiten consultar los mensajes de correo recibidos y enviar los tuyos propios utilizando un navegador.

Hoy en día existen aplicaciones web para multitud de tareas como procesadores de texto, gestión de tareas, o edición y almacenamiento de imágenes. Estas aplicaciones tienen ciertas ventajas e inconvenientes si las comparas con las aplicaciones tradicionales que se ejecutan sobre el sistema operativo de la propia máquina.

Ventajas de las aplicaciones web:

- ✓ **No es necesario instalarlas en aquellos equipos en que se vayan a utilizar.** Se instalan y se ejecutan solamente en un equipo, en el servidor, y esto es suficiente para que se puedan utilizar de forma simultánea desde muchos equipos.
- ✓ Como solo se encuentran instaladas en un equipo, **es muy sencillo gestionarlas** (hacer copias de seguridad de sus datos, corregir errores, actualizarlas).
- ✓ **Se pueden utilizar en todos aquellos sistemas que dispongan de un navegador web,** independientemente de sus características (no es necesario un equipo potente) o de su sistema operativo.
- ✓ **Se pueden utilizar desde cualquier lugar en el que dispongamos de conexión con el servidor.** En muchos casos esto hace posible que se pueda acceder a las aplicaciones desde sistemas no convencionales, como por ejemplo teléfonos móviles.



Rob Enslin (CC BY)

Inconvenientes de las aplicaciones web:

- ✓ **El interface de usuario de las aplicaciones web es la página que se muestra en el navegador.** Esto **restringe** las características del interface a aquellas de una página web.
- ✓ **Dependemos de una conexión con el servidor para poder utilizarlas.** Si nos falla la conexión, no podremos acceder a la aplicación web.
- ✓ **La información que se muestra en el navegador debe transmitirse desde el servidor.** Esto hace que cierto tipo de aplicaciones no sean adecuadas para su implementación como aplicación web (por ejemplo, las aplicaciones que manejan contenido multimedia, como las de edición de vídeo).

Hoy en día muchas aplicaciones web utilizan las ventajas que les ofrece la generación de páginas dinámicas. La gran mayoría de su contenido está almacenado en una **base de datos**. Aplicaciones como **Drupal**, **Joomla!** y otras muchas ofrecen dos partes bien diferenciadas:

- ✓ Una parte externa o front-end, que es el conjunto de páginas que ven la gran mayoría de **usuarios** que las usan (usuarios externos).
- ✓ Una parte interna o back-end, que es otro conjunto de páginas dinámicas que utilizan las personas que **producen el contenido** y las que **administran** la aplicación web (usuarios internos) para crear contenido, organizarlo, decidir la apariencia externa, etc.



Ilustración realizada con Dia (Elaboración propia)

Autoevaluación

¿Cuál de las siguientes no es una característica de una aplicación web?

- Sólo es necesario instalarla una vez.
- Se crea a partir de páginas web dinámicas.
- Se puede utilizar en múltiples sistemas.
- Sólo necesita un servidor web para ejecutarse.

No es correcta. Las aplicaciones web sólo se instalan una vez: en el servidor web.

No es correcta. Una aplicación web está compuesta normalmente por muchas páginas web dinámicas relacionadas entre sí.

No es correcta. Debido a que se basa en páginas web, puede ser utilizada en todos aquellos sistemas y dispositivos que dispongan de un navegador web.

Efectivamente ésta opción no es cierta. Las aplicaciones web emplean páginas web dinámicas. Y como ya vimos, para ejecutar páginas web dinámicas se necesitan una serie de recursos adicionales en el servidor, como un módulo que

ejecute el código o un sistema gestor de bases de datos.

Solución

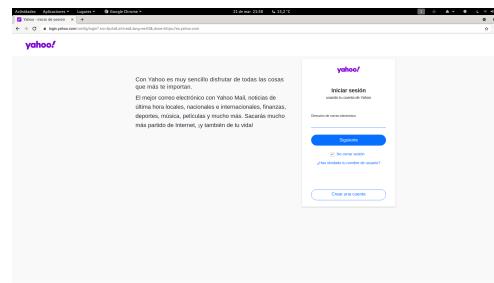
1. Incorrecto
2. Incorrecto
3. Incorrecto
4. Opción correcta

1.2.- Ejecución de código en el servidor y en el cliente.

Como vimos, cuando tu navegador solicita a un servidor web una página, es posible que antes de enviártela haya tenido que ejecutar, por sí mismo o por delegación, algún programa para obtenerla. Ese programa es el que genera, en parte o en su totalidad, la página web que llega a tu equipo. En estos casos, **el código se ejecuta en el entorno del servidor web**.

Además, cuando una página web llega a tu navegador, es también posible que incluya algún programa o fragmentos de código que se deban ejecutar. Ese código, normalmente en lenguaje **JavaScript**, **se ejecutará en tu navegador** y, además de poder modificar el contenido de la página, también puede llevar a cabo acciones como la animación de textos u objetos de la página o la comprobación de los datos que introduces en un formulario.

Estas dos tecnologías se complementan una con otra. Así, volviendo al ejemplo del correo web, el programa que se encarga de obtener tus mensajes y su contenido de una base de datos se ejecuta en el entorno del servidor, mientras que tu navegador ejecuta, por ejemplo, el código encargado de avisarte cuando quieras enviar un mensaje y te has olvidado de poner un texto en el asunto.



[Yahoo](#) (Todos los derechos reservados)

Esta división es así porque el **código que se ejecuta en el cliente** web (en tu navegador) no tiene, o mejor dicho **tradicionalmente no tenía, acceso a los datos** que se almacenan en el **servidor**. Es decir, cuando en tu navegador querías leer un nuevo correo, el código Javascript que se ejecutaba en el mismo no podía obtener de la base de datos el contenido de ese mensaje. La solución era crear una nueva página en el servidor con la información que se pedía y enviarla de nuevo al navegador.

Sin embargo, desde hace unos años existe una **técnica de desarrollo web conocida como AJAX**, que nos posibilita realizar programas en los que el código JavaScript que se ejecuta en el navegador pueda comunicarse con un servidor de Internet para obtener información con la que, por ejemplo, modificar la página web actual.

En nuestro ejemplo, cuando pulsas con el ratón encima de un correo que quieras leer, la página puede contener código Javascript que detecte la acción y, en ese instante, consultar a través de Internet el texto que contiene ese mismo correo y mostrarlo en la misma página, modificando su estructura en caso de que sea necesario. Es decir, sin salir de una página poder modificar su contenido en base a la información que se almacena en un servidor de Internet.

En este módulo vas a aprender a crear aplicaciones web que se ejecuten en el lado del servidor. Otro módulo de este mismo ciclo, Desarrollo Web en Entorno Cliente, enseña las características de la programación de código que se ejecute en el navegador web.

Muchas de las **aplicaciones web actuales utilizan estas dos tecnologías**: la ejecución de **código en el servidor y en el cliente**. Así, el código que se ejecuta en el servidor genera páginas web que ya incluyen código destinado a su ejecución en el navegador. Hacia el final de este módulo verás las técnicas que se usan para programar aplicaciones que incorporen esta funcionalidad.

Autoevaluación

Si quieres verificar que la contraseña introducida en una página web tenga una longitud mínima, ¿dónde sería preferible que se ejecutara el código de comprobación?

- En el navegador web.
- En el servidor web.

Obviamente; no es necesario enviar la contraseña al servidor web para comprobar su longitud, cuando esa tarea se puede hacer perfectamente en el navegador.

Para poder comprobar la contraseña en el servidor, hace falta enviársela. ¿Es realmente necesario?

Solución

1. Opción correcta
2. Incorrecto

2.- Tecnologías para programación web del lado del servidor.

Caso práctico



En BK Programación han formado un **equipo** que se dedicará al nuevo proyecto. **Juan** será el jefe del proyecto, y contará con la **ayuda de María**, que trabaja habitualmente en la instalación y mantenimiento de servidores, y con **Carlos**, un amigo de Juan que pese a **no tener experiencia** en ese tipo de trabajo se siente muy atraído por todo lo relacionado con la programación web.

Juan ha programado aplicaciones en lenguajes C, Java y PHP. **María** por su parte **conoce el lenguaje C** y utiliza Perl en su trabajo habitual. **Carlos es autodidacta** y tiene nociones de programación en lenguaje Pascal. **Todos** conocen bien el lenguaje HTML.

En la primera reunión, los tres ponen en común los objetivos generales del trabajo y deciden estudiar las distintas opciones que tienen para lograr el objetivo.

Cuando programas una **aplicación**, utilizas un **lenguaje de programación**. Por ejemplo, utilizas el lenguaje Java para crear aplicaciones que se ejecuten en distintos sistemas operativos. Al programar cada aplicación utilizas ciertas herramientas como un entorno de desarrollo o librerías de código. Además, una vez acabado su desarrollo, esa aplicación necesitará ciertos componentes para su ejecución, como por ejemplo una máquina virtual de Java.

En este bloque **vas a aprender las distintas tecnologías** que se pueden utilizar para **programar aplicaciones** que se ejecuten en un **servidor web**, y cómo se relacionan unas con otras. Verás las ventajas e inconvenientes de utilizar cada una, y qué lenguajes de programación deberás aprender para utilizarlas.

Los **componentes principales** con los que debes contar para ejecutar aplicaciones web en un servidor son los siguientes:

- ✓ Un **servidor web** para recibir las peticiones de los clientes web (normalmente navegadores) y enviarles la página que solicitan (una vez generada puesto que hablamos de páginas web dinámicas). El servidor web debe conocer el procedimiento a seguir para generar

la página web: qué módulo se encargará de la ejecución del código y cómo se debe comunicar con él. **Servidores web** que tal vez te suenen son: **Apache**, **Nginx** o **IIS** del familia Microsoft.

- ✓ El **módulo encargado de ejecutar el código** o programa y generar la página web resultante. Este módulo debe integrarse de alguna forma con el servidor web, y dependerá del lenguaje y tecnología que utilicemos para programar la aplicación web.
- ✓ Una **aplicación de base de datos**, que normalmente también será un servidor. Este módulo no es estrictamente necesario pero en la práctica se utiliza en todas las aplicaciones web que utilizan grandes cantidades de datos para almacenarlos. Bases de datos muy usadas serán **Mysql**, **MariaDb** y **SQLite**.
- ✓ El **lenguaje de programación** que utilizarás para desarrollar las aplicaciones. A lo largo de este curso veremos **PHP** y **JavaScript** para NodeJs.

Además de los componentes a utilizar, también es importante decidir cómo vas a **organizar el código** de la aplicación. Muchas de las arquitecturas que se usan en la programación de aplicaciones web te ayudan a estructurar el código de las aplicaciones en **capas o niveles**.

El motivo de dividir en capas el diseño de una aplicación es que se puedan **separar las funciones lógicas** de la misma, de tal forma que sea posible ejecutar cada una en un servidor distinto (en caso de que sea necesario).

En una aplicación puedes distinguir, de forma general, **funciones de presentación** (se encarga de dar formato a los datos para presentárselo al usuario final), **lógica** (utiliza los datos para ejecutar un proceso y obtener un resultado), **persistencia** (que mantiene los datos almacenados de forma organizada) y **acceso** (que obtiene e introduce datos en el espacio de almacenamiento).

Cada capa puede ocuparse de una o varias de las funciones anteriores. Por ejemplo, en las aplicaciones de **3 capas** nos podemos encontrar con:

- ✓ Una **capa cliente**, que es donde programarás todo lo relacionado con el interface de usuario, esto es, la parte visible de la aplicación con la que interactuará el usuario.
- ✓ Una **capa intermedia** donde deberás programar la funcionalidad de tu aplicación.
- ✓ Una **capa de acceso a datos**, que se tendrá que encargar de almacenar la información de la aplicación en una base de datos y recuperarla cuando sea necesario.

2.1.- Arquitecturas y plataformas.

La primera elección que harás antes de comenzar a programar una aplicación web es la arquitectura que vas a utilizar. Hoy en día, puedes elegir entre:

- ✓ Java EE (Enterprise Edition), que antes también se conocía como J2EE. Es una plataforma orientada a la programación de aplicaciones en lenguaje Java. Puede funcionar con distintos gestores de bases de datos, e incluye varias librerías y especificaciones para el desarrollo de aplicaciones de forma modular.

Está apoyada por grandes empresas como Sun y Oracle, que mantienen Java, o IBM. Es una buena solución para el desarrollo de aplicaciones de tamaño mediano o grande. Una de sus principales ventajas es la multitud de librerías existentes en ese lenguaje y la gran base de programadores que lo conocen.

Dentro de esta arquitectura existen distintas tecnologías como las páginas JSP y los servlets, ambos orientados a la generación dinámica de páginas web, o los EJB, componentes que normalmente aportan la lógica de la aplicación web.

- ✓ AMP. Son las siglas de Apache, MySQL y PHP/Perl/Python. Las dos primeras siglas hacen referencia al servidor web (Apache) y al servidor de base de datos (MySQL o MariaDB). La última se corresponde con el lenguaje de programación utilizado, que puede ser PHP, Perl o Python, siendo PHP el más empleado de los tres.

Dependiendo del sistema operativo que se utilice para el servidor, se utilizan las siglas LAMP (para Linux), WAMP (para Windows) o MAMP (para Mac). También es posible usar otros componentes, como el gestor de bases de datos PostgreSQL en lugar de MySQL.

Todos los componentes de esta arquitectura son de código abierto (open source). Es una plataforma de programación que permite desarrollar aplicaciones de tamaño pequeño o mediano con un aprendizaje sencillo. Su gran ventaja es la gran comunidad que la soporta y la multitud de aplicaciones de código libre disponibles.



[ApacheFriends](#) (Todos los derechos reservados)

Para saber más

Existen paquetes software que incluyen en una única instalación una plataforma AMP completa. Algunos ni siquiera es necesario instalarlos, son las llamadas portables, e incluso disponen de versiones para distintos sistemas operativos como Linux, Windows o Mac. Uno de los más conocidos es XAMPP.

[XAMPP](#)

- ✓ CGI/Perl. Es la combinación de dos componentes: Perl, un potente lenguaje de código libre creado originalmente para la administración de servidores, y CGI, un estándar para permitir al servidor web ejecutar programas genéricos, escritos en cualquier lenguaje (también se pueden utilizar por ejemplo C o Python), que devuelven páginas web como resultado de su ejecución. Es la más primitiva de las arquitecturas que comparamos aquí.

El principal inconveniente de esta combinación es que CGI es lento, aunque existen métodos para acelerarlo. Por otra parte, Perl es un lenguaje muy potente con una amplia comunidad de usuarios y mucho código libre disponible.

- ✓ ASP.Net es la arquitectura comercial propuesta por Microsoft para el desarrollo de aplicaciones. Es la parte de la plataforma .Net destinada a la generación de páginas web dinámicas. Proviene de la evolución de la anterior tecnología de Microsoft, ASP.

El lenguaje de programación puede ser Visual Basic.Net o C#. La arquitectura utiliza el servidor web de Microsoft, IIS, y puede obtener información de varios gestores de bases de datos entre los que se incluye, como no, Microsoft SQL Server.

Una de las mayores ventajas de la arquitectura .Net es que incluye todo lo necesario para el desarrollo y el despliegue de aplicaciones. Por ejemplo, tiene su propio entorno de desarrollo, Visual Studio, aunque hay otras opciones disponibles. La mayor desventaja es que se trata de una plataforma comercial de código propietario.

- ✓ Node.js. Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación ECMAScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

2.1.1.- Selección de una arquitectura de programación web.

Como has visto, hay muchas decisiones que debes tomar antes de comenzar el desarrollo de una aplicación web. La arquitectura que utilizarás, el lenguaje de programación, el entorno de desarrollo, el gestor de bases de datos, el servidor web, incluso cómo estructurarás tu aplicación.

Para tomar una decisión correcta, deberás considerar entre otros los siguientes puntos:

- ✓ ¿Qué tamaño tiene el proyecto?
- ✓ ¿Qué lenguajes de programación conozco? ¿Vale la pena el esfuerzo de aprender uno nuevo?
- ✓ ¿Voy a usar herramientas de código abierto o herramientas propietarias? ¿Cuál es el coste de utilizar soluciones comerciales?
- ✓ ¿Voy a programar la aplicación yo solo o formaré parte de un grupo de programadores?
- ✓ ¿Cuento con algún servidor web o gestor de base de datos disponible o puedo decidir libremente utilizar el que crea necesario?
- ✓ ¿Qué tipo de licencia voy a aplicar a la aplicación que desarrolle?



[googvt9 \(CC BY\)](#)

Estudiando las respuestas a éstas y otras preguntas, podrás ver qué arquitecturas se adaptan mejor a tu aplicación y cuáles no son viables.

Reflexiona

Cada proyecto tiene sus peculiaridades. Aunque unas arquitecturas son más potentes que otras, no hay una en concreto que podamos considerar mejor que las demás para todos los casos. ¿Crees que vale la pena el esfuerzo de cambiar de arquitectura de desarrollo para cada aplicación? ¿Es necesario en todos los casos?

Autoevaluación

¿Cuál de estas tecnologías permite la ejecución por el servidor web de programas escritos en cualquier lenguaje?

- Java EE.
- PHP.
- AMP.
- CGI.

No es correcta. Si se utiliza Java EE, los programas deben estar escritos en lenguaje Java.

No es correcta porque PHP es un lenguaje de programación.

No es correcta. La P de AMP puede hacer referencia a varios lenguajes: PHP, Perl o Python. Pero en una plataforma AMP no se pueden ejecutar programas escritos en un lenguaje cualquiera.

Efectivamente es correcto, CGI es un estándar que permite al servidor web la ejecución de programas genéricos, escritos en cualquier lenguaje.

Solución

1. Incorrecto
2. Incorrecto
3. Correcto

4. Opción correcta

2.2.- Integración con el servidor web.

Como ya sabes, la comunicación entre un cliente web o navegador y un servidor web se lleva a cabo gracias al protocolo [HTTP](#). En el caso de las aplicaciones web, HTTP es el vínculo de unión entre el usuario y la aplicación en sí. Cualquier introducción de información que realice el usuario se transmite mediante una petición HTTP, y el resultado que obtiene le llega por medio de una respuesta HTTP.

En el lado del servidor, estas peticiones son procesadas por el servidor web (también llamado servidor HTTP). Es por tanto el servidor web el encargado de decidir cómo procesar las peticiones que recibe. Cada una de las arquitecturas que acabamos de ver tiene una forma de integrarse con el servidor web para ejecutar el código de la aplicación.

La tecnología más antigua es [CGI](#). CGI es un protocolo estándar que existe en muchas plataformas. Lo implementan la gran mayoría de servidores web. Define qué debe hacer el servidor web para delegar en un programa externo la generación de una página web. Esos programas externos se conocen como guiones CGI, independientemente del lenguaje en el que estén programados (aunque se suelen programar en lenguajes de guiones como Perl).

El principal problema de CGI es que cada vez que se ejecuta un guión CGI, el sistema operativo debe crear un nuevo proceso. Esto implica un mayor consumo de recursos y menor velocidad de ejecución. Existen algunas soluciones que aceleran la ejecución, como FastCGI, y también otros métodos para ejecutar guiones en el entorno de un servidor web, por ejemplo el módulo [mod_perl](#) para ejecutar en Apache guiones programados en Perl.

Aunque también es posible ejecutar código en lenguaje [PHP](#) utilizando CGI, los intérpretes PHP no se suelen utilizar con este método. Al igual que [mod_perl](#), existen otros módulos que podemos instalar en el servidor web Apache para que ejecute páginas web dinámicas. El módulo [mod_php](#) es la forma habitual que se utiliza para ejecutar guiones en PHP utilizando plataformas [AMP](#), y su equivalente para el lenguaje Python es [mod_python](#).

La arquitectura [Java EE](#) es más compleja. Para poder ejecutar aplicaciones [Java EE](#) en un servidor básicamente tenemos dos opciones: servidores de aplicaciones, que implementan todas las tecnologías disponibles en [Java EE](#), y contenedores de servlets, que soportan solo parte de la especificación. Dependiendo de la magnitud de nuestra aplicación y de las tecnologías que utilice, tendremos que instalar una solución u otra.



[Apache](#) (Apache License, Version 2.0.)

Existen varias implementaciones de servidores de aplicaciones Java EE certificados. Las dos soluciones comerciales más usadas son IBM Websphere y BEA Weblogic. También existen soluciones de código abierto (OpenSource) como JBoss, Geronimo (de la fundación Apache) o Glassfish.

Para saber más

Puedes consultar una lista con los servidores de aplicaciones Java EE certificados por Sun en la Wikipedia.

[Servidores de aplicaciones Java EE certificados por Sun](#)

Sin embargo, en la mayoría de ocasiones no es necesario utilizar un servidor de aplicaciones completo, especialmente si no utilizamos EJB en nuestras aplicaciones, sino que nos será suficiente un contenedor de servlets. En esta área, destaca Tomcat, la implementación por referencia de un contenedor de servlets, que además es de código abierto.

Una vez instalada la solución que hayamos escogido, tenemos que integrarla con el servidor web que utilicemos, de tal forma que reconozca las peticiones destinadas a servlets y páginas JSP y las redirija. Otra opción es utilizar una única solución para páginas estáticas y páginas dinámicas. Por ejemplo, el contenedor de servlets Tomcat incluye un servidor HTTP propio que puede sustituir a Apache.

La arquitectura ASP.Net utiliza el servidor IIS de Microsoft, que ya integra soporte en forma de módulos para manejar peticiones de páginas dinámicas ASP y ASP.Net. La utilidad de administración del servidor web incluye funciones de administración de las aplicaciones web instaladas en el mismo.

2.3.- Lenguajes.

Caso práctico



Tras analizar distintas arquitecturas de programación web, **Juan** y su equipo comprueban que una de las decisiones más importantes antes de ponerse manos a la obra es el lenguaje de programación que utilizarán en el desarrollo.

Además de la sintaxis propia de cada lenguaje, la elección de uno u otro afecta a la complejidad del proyecto, a las herramientas que tendrán que utilizar, e incluso a la forma en que tendrán que programar la aplicación web.

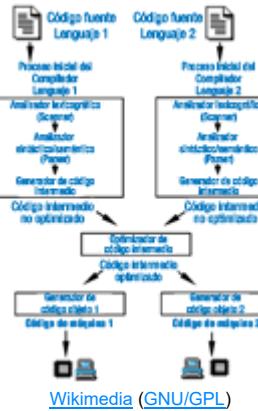
Juan propone utilizar el lenguaje PHP, que ya conoce, pero no sin antes estudiar las ventajas y desventajas que les supondría con respecto a la utilización de otras alternativas.

Una de las diferencias más notables entre un lenguaje de programación web y otro es la manera en que se ejecutan en el servidor web. Debes distinguir tres grandes grupos:

- ✓ **Lenguajes de guiones** (scripting). Son aquellos en los que los programas se ejecutan directamente a partir de su código fuente original. Se almacenan normalmente en un fichero de texto plano. Cuando el servidor web necesita ejecutar código programado en un lenguaje de guiones, le pasa la petición a un intérprete, que procesa las líneas del programa y genera como resultado una página web.

De los lenguajes que estudiaste anteriormente, pertenecen a este grupo Perl, Python, PHP y ASP y su sucesor ASP.net

- ✓ **Lenguajes compilados a código nativo**. Son aquellos en los que el código fuente se traduce a código binario, dependiente del procesador, antes de ser ejecutado. El servidor web almacena los programas en su modo binario, que ejecuta directamente cuando se les invoca.



El método principal para ejecutar programas binarios desde un servidor web es CGI. Utilizándolos podemos hacer que el servidor web ejecute código programado en cualquier lenguaje de propósito general como puede ser C.

- ✓ **Lenguajes compilados a código intermedio.** Son lenguajes en los que el código fuente original se traduce a un código intermedio, independiente del procesador, antes de ser ejecutado. Es la forma en la que se ejecutan por ejemplo las aplicaciones programadas en Java, y lo que hace que puedan ejecutarse en varias plataformas distintas.

En la programación web, operan de esta forma los lenguajes de las arquitecturas Java EE (servlets y páginas JSP) y ASP.Net. En la plataforma ASP.Net y en muchas implementaciones de Java EE, se utiliza un procedimiento de compilación JIT. Este término hace referencia a la forma en que se convierte el código intermedio a código binario para ser ejecutado por el procesador. Para acelerar la ejecución, el compilador puede traducir todo o parte del código intermedio a código nativo cuando se invoca a un programa. El código nativo obtenido suele almacenarse para ser utilizado de nuevo cuando sea necesario.

Cada una de estas formas de ejecución del código por el servidor web tiene sus ventajas e inconvenientes.

- ✓ Los lenguajes de guiones tienen la ventaja de que no es necesario traducir el código fuente original para ser ejecutados, lo que aumenta su portabilidad. Si se necesita realizar alguna modificación a un programa, se puede hacer en el momento. Por el contrario el proceso de interpretación ofrece un peor rendimiento que las otras alternativas.
- ✓ Los lenguajes compilados a código nativo son los de mayor velocidad de ejecución, pero tienen problemas en lo relativo a su integración con el servidor web. Son programas de propósito general que no están pensados para ejecutarse en el entorno de un servidor web. Por ejemplo, no se reutilizan los procesos para atender a varias peticiones: por cada petición que se haga al servidor web, se debe ejecutar un nuevo proceso. Además los programas no son portables entre distintas plataformas.
- ✓ Los lenguajes compilados a código intermedio ofrecen un equilibrio entre las dos opciones anteriores. Su rendimiento es muy bueno y pueden portarse entre distintas plataformas en las que exista una implementación de la arquitectura (como un contenedor de servlets o un servidor de aplicaciones Java EE).

2.3.1.- Código embebido en el lenguaje de marcas.

Cuando la web comenzó a evolucionar desde las páginas web estáticas a las dinámicas, una de las primeras tecnologías que se utilizaron fue la ejecución de código utilizando CGI. Los guiones CGI son programas estándar, que se ejecutan por el sistema operativo, pero que generan como salida el código HTML de una página web. Por tanto, los guiones CGI deben contener, mezcladas dentro de su código, sentencias encargadas de generar la página web.

Por ejemplo, si quieras generar una página web utilizando CGI a partir de un guion de sentencias en Linux, tienes que hacer algo como lo siguiente:

```
#!/bin/sh
# Generamos la cabecera HTTP
echo "Content-Type: text/html"
echo ""
# y el contenido de la página web
echo "<html>"
echo "  <head>"
echo "    <title>Prueba CGI</title>"
echo "  </head>"
echo "  <body>"
echo "    Prueba de guion bash CGI"
echo "  </body>"
echo "</html>"
```

Esta es una de las principales formas de realizar páginas web dinámicas: **integrar las etiquetas HTML en el código de los programas**. Es decir, los programas como el guion anterior incluyen dentro de su código sentencias de salida (`echo` en el caso anterior) que son las que incluyen el código HTML de la página web que se obtendrá cuando se ejecuten. De esta forma se programan, por ejemplo, los guiones CGI y los servlets.

Un enfoque distinto consiste en **integrar el código del programa en medio de las etiquetas HTML de la página web**. De esta forma, el contenido que no varía de la página se puede introducir directamente en HTML, y el lenguaje de programación se utilizará para todo aquello que pueda variar de forma dinámica.

Por ejemplo, puedes incluir dentro de una página HTML un pequeño código en lenguaje PHP que muestre el nombre del servidor:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Prueba PHP</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>
  <body>
    Prueba de página PHP <br>
    El nombre del servidor es: <?php echo $_SERVER['SERVER_NAME']; ?>"
```

Esta metodología de programación es la que se emplea en los lenguajes ASP, PHP y con las páginas JSP de Java EE.

Los servlets de Java EE se diferencian de las páginas JSP en que los primeros son programas Java compilados y almacenados en el contenedor de servlets. Sin embargo, las páginas JSP contienen código Java embebido en lenguaje HTML y se almacenan de forma individual en el servidor web. La primera vez que se necesita una página JSP, se convierte a un servlet y éste se guarda para ser utilizado en posteriores llamadas a la misma página.

En la arquitectura ASP.Net, cada página se divide en dos ficheros: uno contiene las etiquetas HTML y otro el código en el lenguaje de programación utilizado. De esta forma se logra cierta independencia entre el aspecto de la aplicación y la gestión del contenido dinámico. A partir de esos ficheros se obtiene un código intermedio (MSIL en la terminología de la plataforma) que es lo que almacena el servidor.

Autoevaluación

La relación entre la forma de ejecución de un lenguaje, y el método para integrarse con las etiquetas HTML de una página web es:

- Si el lenguaje integra en su código etiquetas HTML, entonces se trata de un lenguaje de guiones.
- Si las instrucciones del lenguaje se integran dentro de las etiquetas HTML de una página web, entonces se trata de un lenguaje de guiones.

- Si las instrucciones del lenguaje se integran dentro de las etiquetas HTML de una página web, entonces se trata de un lenguaje compilado.
- Es indistinto, no hay una relación directa.

No es correcta porque existen lenguajes compilados, como Java en el caso de los servlets, que integran en su código etiquetas HTML.

No es correcta porque existen lenguajes compilados, también como Java en el caso de las páginas JSP, que integra las instrucciones del lenguaje dentro de las etiquetas HTML de una página web.

No es correcta porque existen lenguajes de guiones como PHP, que integra las instrucciones del lenguaje dentro de las etiquetas HTML de una página web.

No existe una relación directa entre la forma en que se ejecuta un lenguaje, y cómo se integra con las etiquetas HTML de una página web.

Solución

1. Incorrecto
2. Incorrecto
3. Incorrecto
4. Opción correcta



2.3.2.- Programación web con Java.

Java es el lenguaje de programación más utilizado hoy en día. Es un lenguaje orientado a objetos, basado en la sintaxis de C y C++ y eliminando algunas características de éstos que daban lugar a errores de programación, como los punteros. Todo el código que escribas en Java debe pertenecer a una clase.

El código fuente se escribe en archivos con extensión `.java`. El compilador genera por cada clase un archivo `.class`. Para ejecutar una aplicación programada en Java necesitamos tener instalado un entorno de ejecución (JRE). Para crear aplicaciones en Java necesitamos el kit de desarrollo de Java (JDK), que incluye el compilador.

Como ya viste, existen básicamente dos tecnologías que te permiten programar páginas web dinámicas utilizando Java EE : servlets y JSP (páginas web que contienen instrucciones para añadir contenido de forma dinámica).

Aunque no es así en todos los casos, la mayoría de implementaciones disponibles para JSP compilan cada página y generan un servlet a partir de la misma la primera vez que se va a ejecutar. Este servlet se almacena para ser usado en futuras peticiones.

Por ejemplo, si quieras calcular la suma de dos números y enviar el resultado al navegador, lo podríamos realizar con una página JSP, incluyendo el código en Java dentro de las etiquetas HTML utilizando los delimitadores `<%` y `%>` de la siguiente manera:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Prueba JSP</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>
  <body>
    Prueba de página JSP <br>
    La suma de 2 y 3 es: <% out.println(2+3) %>
  </body>
</html>
```



[Ordercrazy \(CC0\)](#)

O también utilizando el método `println` dentro de un servlet como el siguiente, que obtiene los valores a sumar de otra página:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class miServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        int num1 = Integer.parseInt(req.getParameter("num1"));
        int num2 = Integer.parseInt(req.getParameter("num2"));

        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("  <head>");
        out.println("    <title>Prueba de servlet</title>");
        out.println("  </head>");
        out.println("  <body>");
        out.println("    Prueba de servlet Java.");
        out.println("    La suma de " + num1 + " + " + num2 + " es: " + (num1 + num2));
        out.println("  </body>");
        out.println("</html>");
    }
}
```

No hay nada que se pueda hacer con JSP que no pueda hacerse también con servlets. De hecho, como ya viste, las primeras se suelen convertir en servlets para ser ejecutadas.

El problema de utilizar servlets directamente es que, aunque son muy eficientes, son muy tediosos de programar puesto que hay que generar la salida en código HTML con gran cantidad de funciones como `println`. Este problema se resuelve fácilmente utilizando JSP, puesto que aprovecha la eficiencia del código Java, para generar el contenido dinámico, y la lógica de presentación se realiza con HTML normal.

De esta forma estas dos tecnologías se suelen combinar para crear aplicaciones web. Los servlets se encargan de procesar la información y obtener resultados, y las páginas JSP se encargan del interface, incluyendo los resultados obtenidos por los servlets dentro de una página web.

3.- Entorno de programación PHP.

Caso práctico



Los miembros del equipo están dispuestos a ponerse manos a la obra pero al inicio de cualquier desarrollo es imprescindible **definir claramente las herramientas a utilizar**. En este caso, hay que decidir los **elementos del entorno de desarrollo PHP** que vamos a utilizar.

Las primeras indagaciones les llevan a la conclusión de que hay distintas formas de **construir un entorno de programación en PHP**, empezando por la plataforma utilizada, Windows, Linux, Mac y la nube, hasta el tipo de IDE más idóneo, Apache Netbeans, Eclipse, Visual Studio, etc.

Descripción de los elementos de un entorno de desarrollo PHP

Los programas necesarios para empezar a desarrollar con PHP son los siguientes:

- ✓ Un programa que interprete el código PHP.
- ✓ Una base de datos, que suele ser MySQL. Lo más habitual es que un desarrollo PHP use datos almacenados en una base de datos para construir el contenido dinámico de la página web.
- ✓ Un servidor web, normalmente Apache o Nginx para recibir la petición de un navegador y lanzar el proceso de ejecución de un script PHP. El proceso de ejecución consiste en interpretar el código PHP incrustado en el fichero con extensión .php que se solicita en la petición HTTP al servidor.
- ✓ Un IDE que apoye al programador en las tareas de desarrollo de los programas tanto en el momento de su escritura como en las pruebas y depuración de los mismos. Más adelante se listan los IDEs más utilizados en el desarrollo de aplicaciones PHP.

Los IDEs proporcionan a los programadores funcionalidades que les ayudan a desarrollar código PHP. Entre estas características se encuentran:

- ✓ **Resaltado de texto.** Muestra con distinto color o tipo de letra los diferentes elementos del lenguaje: sentencias, variables, comentarios, etc. También genera "*indentado*" automático para diferenciar de forma clara los distintos bloques de un programa.
- ✓ **Completado automático.** Detecta qué estás escribiendo y cuando es posible te muestra distintas opciones para completar el texto.
- ✓ **Navegación en el código.** Permite buscar de forma sencilla elementos dentro del texto, por ejemplo, definiciones de variables.
- ✓ **Comprobación de errores al editar.** Reconoce la sintaxis del lenguaje y revisa el código en busca de errores mientras lo escribes.
- ✓ **Generación automática de código.** Ciertas estructuras, como la que se utiliza para las clases, se repiten varias veces en un programa. La generación automática de código puede encargarse de crear la estructura básica, para que sólo tengas que rellenarla.
- ✓ **Ejecución y depuración.** Esta característica es una de las más útiles. El IDE se puede encargar de ejecutar un programa para poder probar su funcionamiento. Además, cuando algo no funciona, te permite depurarlo con herramientas como la ejecución paso a paso, el establecimiento de puntos de ruptura o la inspección del valor que almacenan las variables.
- ✓ **Gestión de versiones.** En conjunción con un sistema de control de versiones, el entorno de desarrollo te puede ayudar a guardar copias del estado del proyecto a lo largo del tiempo, para que si es necesario puedas revertir los cambios realizados.

Los IDE para desarrollar PHP más utilizados en la actualidad son:

- **PHPStorm** de JetBrains (requiere licencia pero estudiantes y profesorado pueden solicitar una gratuita de un año).
- **Visual Studio Code** de Microsoft (muy versátil, se le pueden instalar muchas extensiones para darle una gran funcionalidad en casi cualquier lenguaje de programación).
- **SublimeText** al igual que el anterior se le pueden instalar gran cantidad de extensiones, es software propietario pero se puede usar para enseñanza.
- Clásicos como **NetBeans** y **Eclipse**. Ambos además ofrecen para la descarga versiones personalizadas del IDE que pueden ser usadas directamente para programar en un lenguaje determinado, sin necesidad de cambiar la configuración o instalar módulos.

Netbeans



PHPStorm



[JetBrains](#) (Todos los derechos reservados)

VSCode



[Microsoft Visual Code](#) (Todos los derechos reservados)

SublimeText



[SublimeText](#) (Todos los derechos reservados)

Entorno de desarrollo PHP para el curso

Como hemos visto hasta ahora existen distintas maneras de componer un entorno de desarrollo con el lenguaje de programación PHP en su versión 8.0 o superior. A continuación, se describen los componentes del entorno utilizado en este curso y el proceso de instalación y configuración de cada uno de ellos.

Los programas que se van a instalar en la plataforma Windows son:

XAMPP

XAMPP es una distribución de Apache que contiene MariaDB y PHP. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

Java JDK

Java Development Kit es un software que provee herramientas de desarrollo para la creación de programas en Java, necesario para el funcionamiento del IDE Apache Netbeans

Apache Netbeans

Apache NetBeans IDE es un entorno de desarrollo integrado, gratuito y de código abierto para el desarrollo de aplicaciones.

Las secciones de este manual cubren el proceso de instalación y configuración de estos paquetes de SW necesarios para crear el entorno adecuado para el desarrollo de aplicaciones PHP

3.1.- XAMPP.



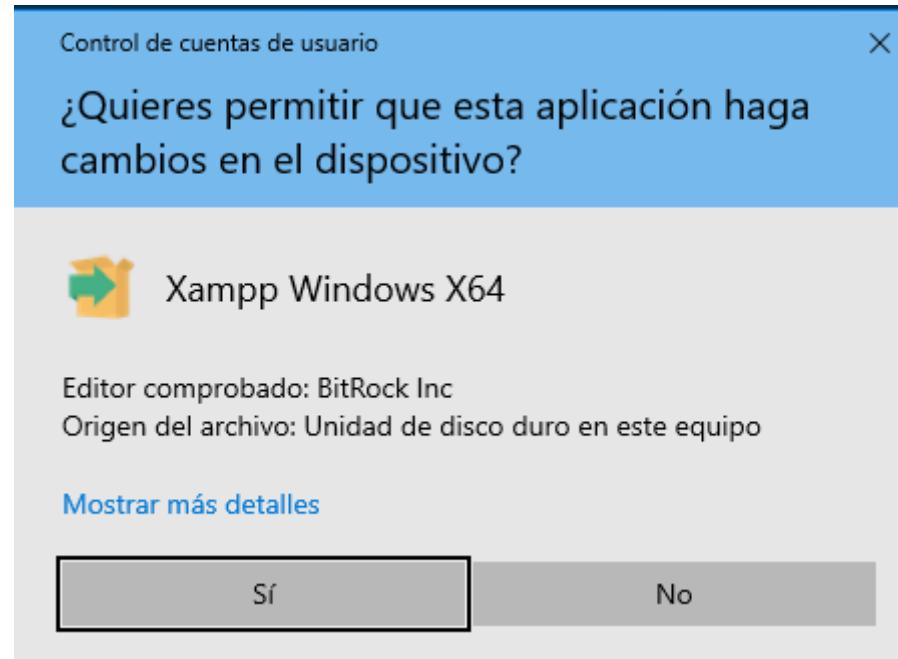
XAMPP es una distribución de Apache que incluye varios software libres. El nombre es un acrónimo compuesto por las iniciales de los programas que lo constituyen: el servidor web Apache, los sistemas relacionales de administración de bases de datos MariaDB (MySQL) y phpMyAdmin, así como los lenguajes de programación Perl y PHP. La "X" inicial indica que es portable para las plataformas de Windows y Linux.

Tomik2001. Logo XAMPP ([CC BY-SA](#))

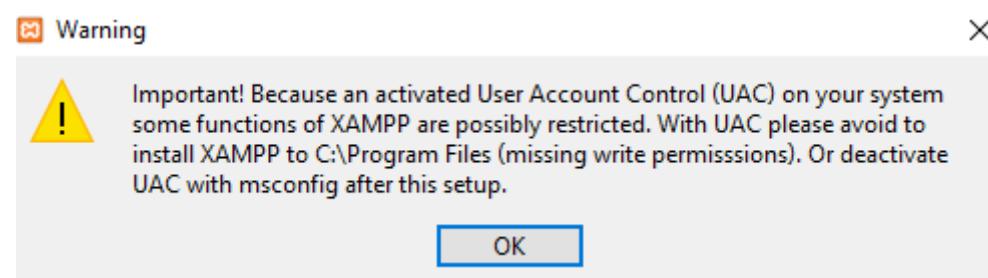
Instalación de XAMPP

Nuestro tutorial muestra la instalación de XAMPP 8.0.19 o superior SO Windows 10. La descarga del paquete XAMPP se puede hacer gratuitamente desde la página del proyecto [Apache Friends](#).

Una vez descargado el paquete, ejecutaremos el instalador y Windows pedirá permiso para realizar cambios en nuestra máquina.

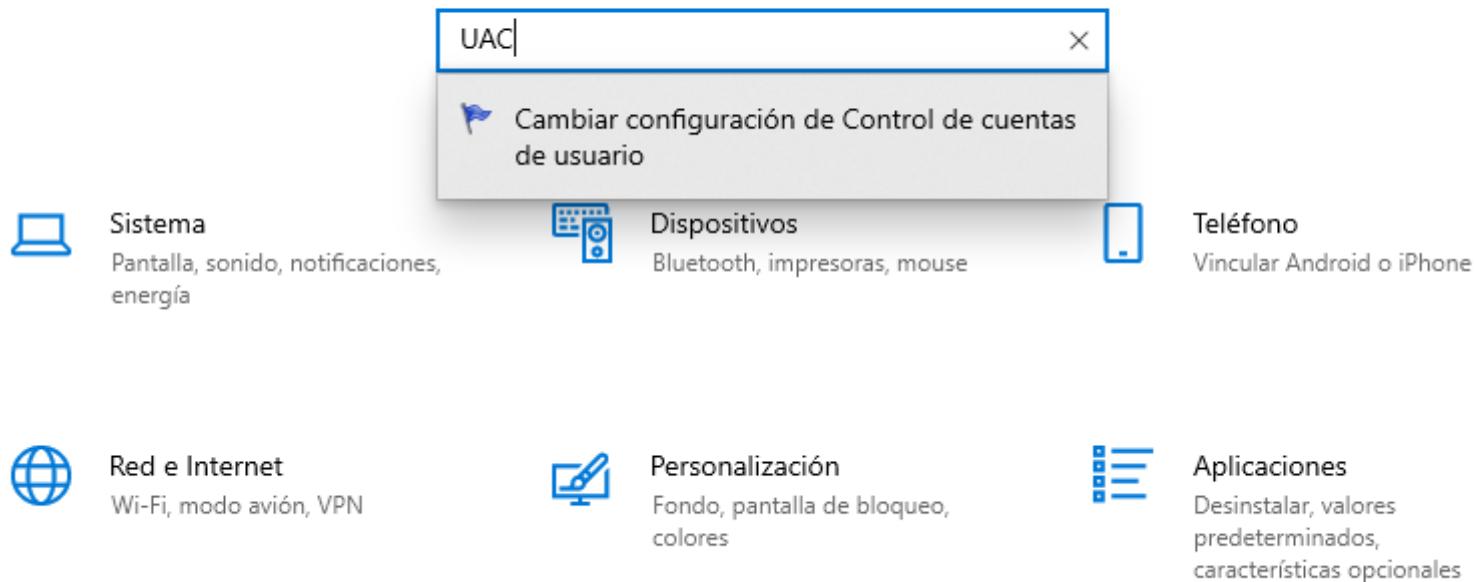


En el caso en que hayamos configurado UAC (User Account Control) en nuestro sistema operativo habrá que desactivarlo para que todos los programas incluidos en el paquete se instalen correctamente.



Para deshabilitarlo durante la instalación debemos ir a «Configuración de Windows» y allí teclearemos UAC.

Configuración de Windows



Seleccionamos la opción "Cambiar configuración de Control de Cuentas de usuario" y se mostrará la ventana siguiente. Para desactivar UAC bajamos los permisos al nivel mínimo.

Elija cuándo desea recibir notificaciones acerca de cambios en el equipo

Control de cuentas de usuario ayuda a impedir que programas perjudiciales realicen cambios en el equipo.

[Más información acerca de la configuración de Control de cuentas de usuario](#)

Notificarme siempre



No notificarme nunca cuando:

- Las aplicaciones intentan instalar software o hacer cambios en el equipo
- Realice cambios en la configuración de Windows

 No se recomienda.

No notificarme nunca

 Aceptar

Cancelar

El sistema operativo pedirá confirmación para realizar la operación.

Control de cuentas de usuario

X

¿Quieres permitir que esta aplicación haga cambios en el dispositivo?



UserAccountControlSettings

Editor comprobado: Microsoft Windows

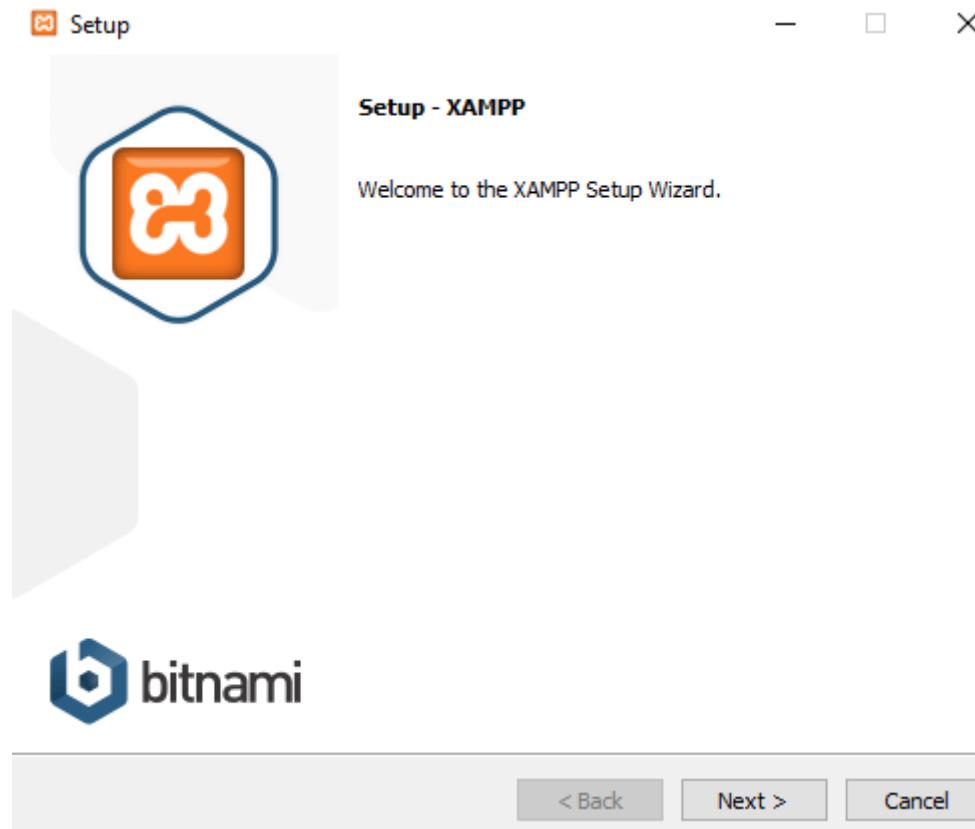
[Mostrar más detalles](#)

Sí

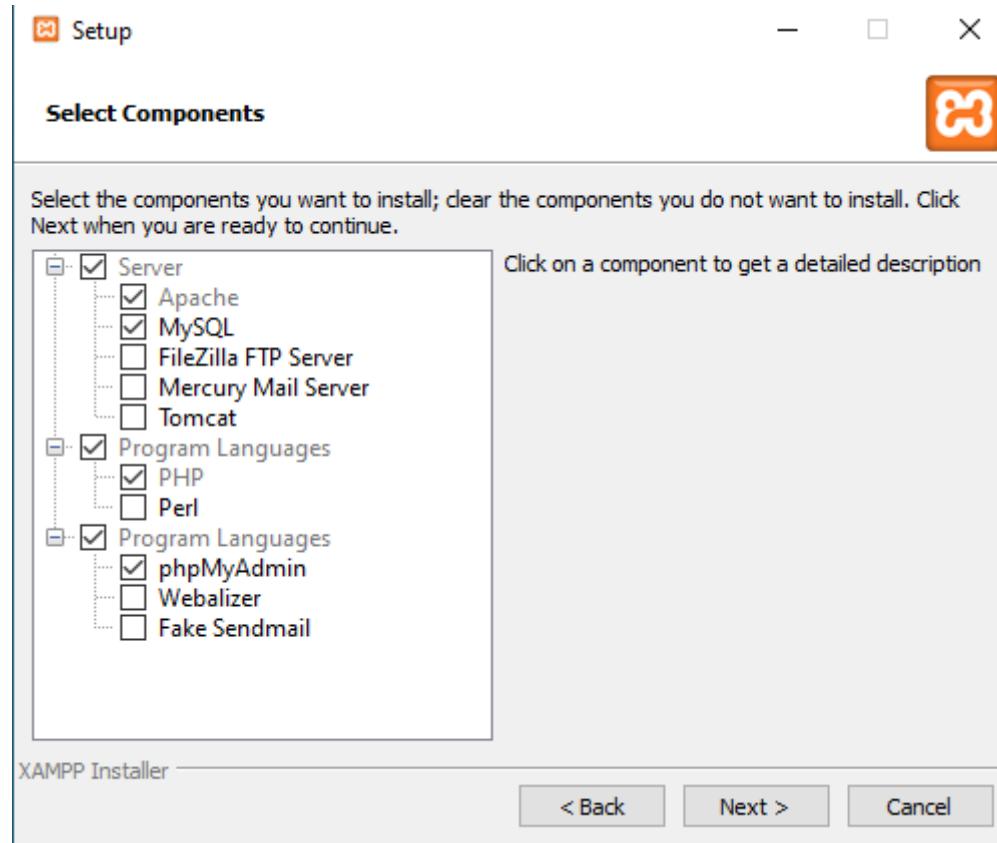
No

Es necesario reiniciar el sistema operativo para proceder de nuevo con la instalación de XAMPP.

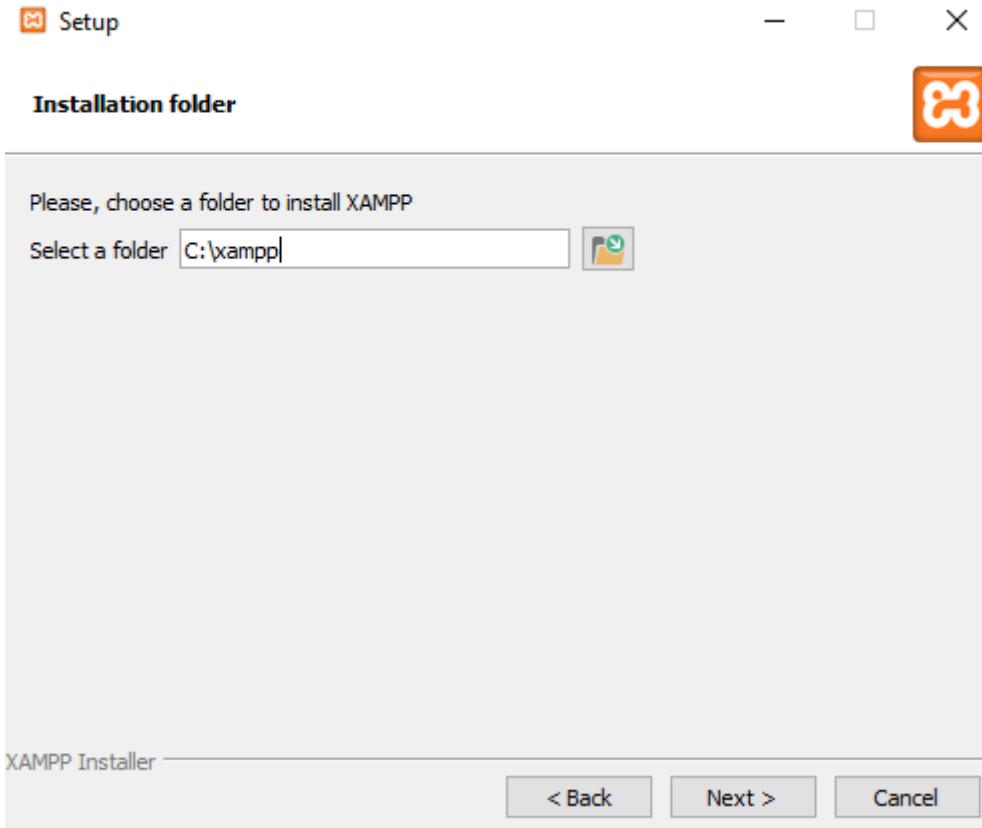
Una iniciada al instalación de nuevo.



Pinchamos en Next y nos aparecerán los diferentes componentes que podemos instalar. En nuestro caso y con el fin de crear el entorno de programación PHP que nos permita realizar los programas del módulo necesitaremos únicamente Apache, MySQL, PHP y phpMyAdmin. Seleccionamos esos componentes y pinchamos sobre Next.



A continuación se solicita la ruta donde se desean instalar los programas. Por defecto lo instala en "C:\xampp".



A continuación nos mostrará otra ventana que nos ofrece acceder a la página web de XAMPP, desmarcamos y pinchamos sobre Next para que empiece la instalación.

Bitnami for XAMPP



Bitnami for XAMPP provides free installers that can install
Drupal, Joomla!, WordPress and many other popular open
source apps on top of your existing XAMPP installation.

<https://bitnami.com/xampp>

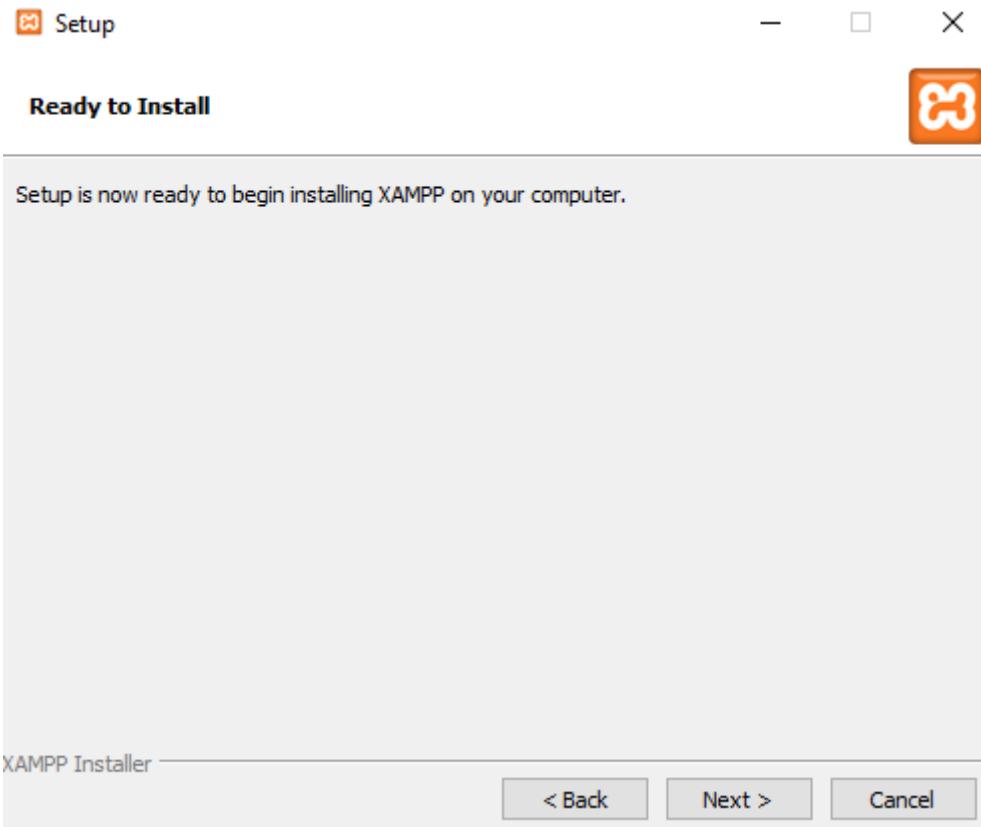
Learn more about Bitnami for XAMPP

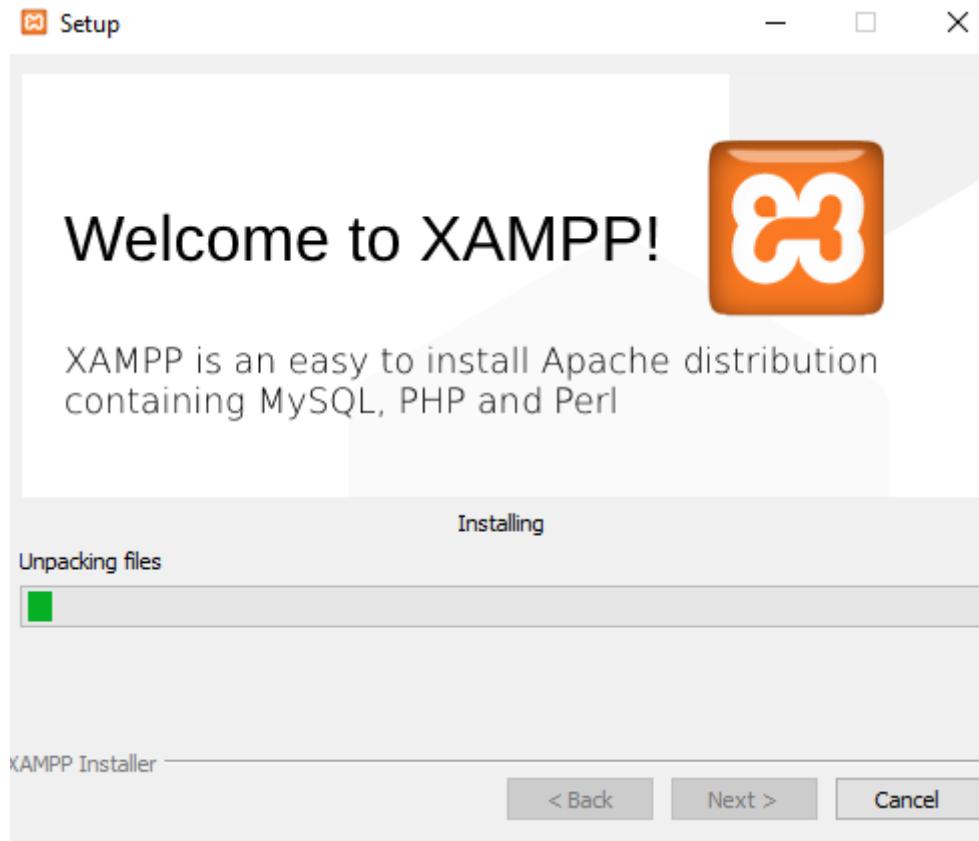
XAMPP Installer

< Back

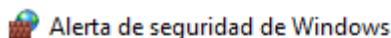
Next >

Cancel





Es posible que el cortafuegos del sistema operativo nos solicite permisos para utilizar los puertos asociados a las aplicaciones instaladas.



Alerta de seguridad de Windows



Firewall de Windows Defender bloqueó algunas características de esta aplicación

Firewall de Windows Defender bloqueó algunas características de Apache HTTP Server en todas las redes públicas y privadas.



Nombre: [Apache HTTP Server](#)

Editor: Apache Software Foundation

Ruta de acceso: C:\xampp\apache\bin\httpd.exe

Permitir que Apache HTTP Server se comunique en estas redes:

Redes privadas, como las domésticas o del trabajo

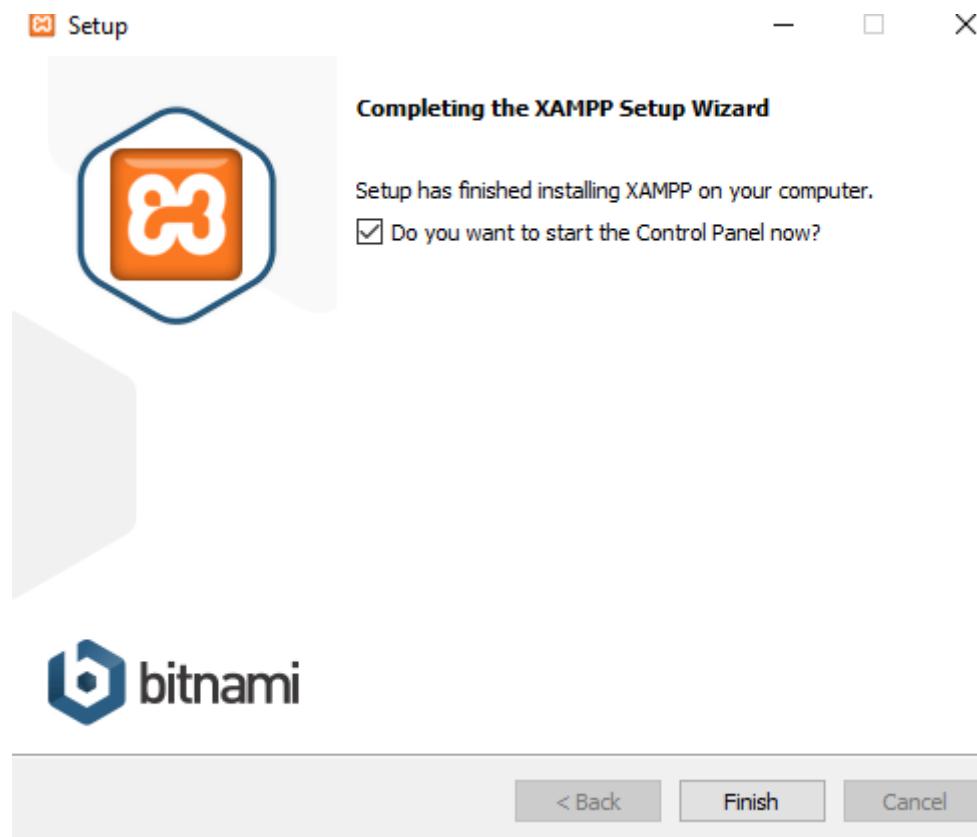
Redes públicas, como las de aeropuertos y cafeterías (no se recomienda porque estas redes públicas suelen tener poca seguridad o carecer de ella)

[¿Cuál es el riesgo de permitir que una aplicación pase a través de un firewall?](#)

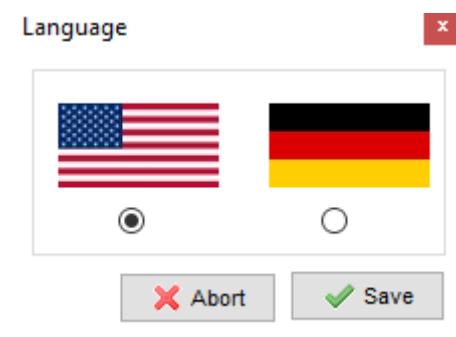
[Permitir acceso](#)

[Cancelar](#)

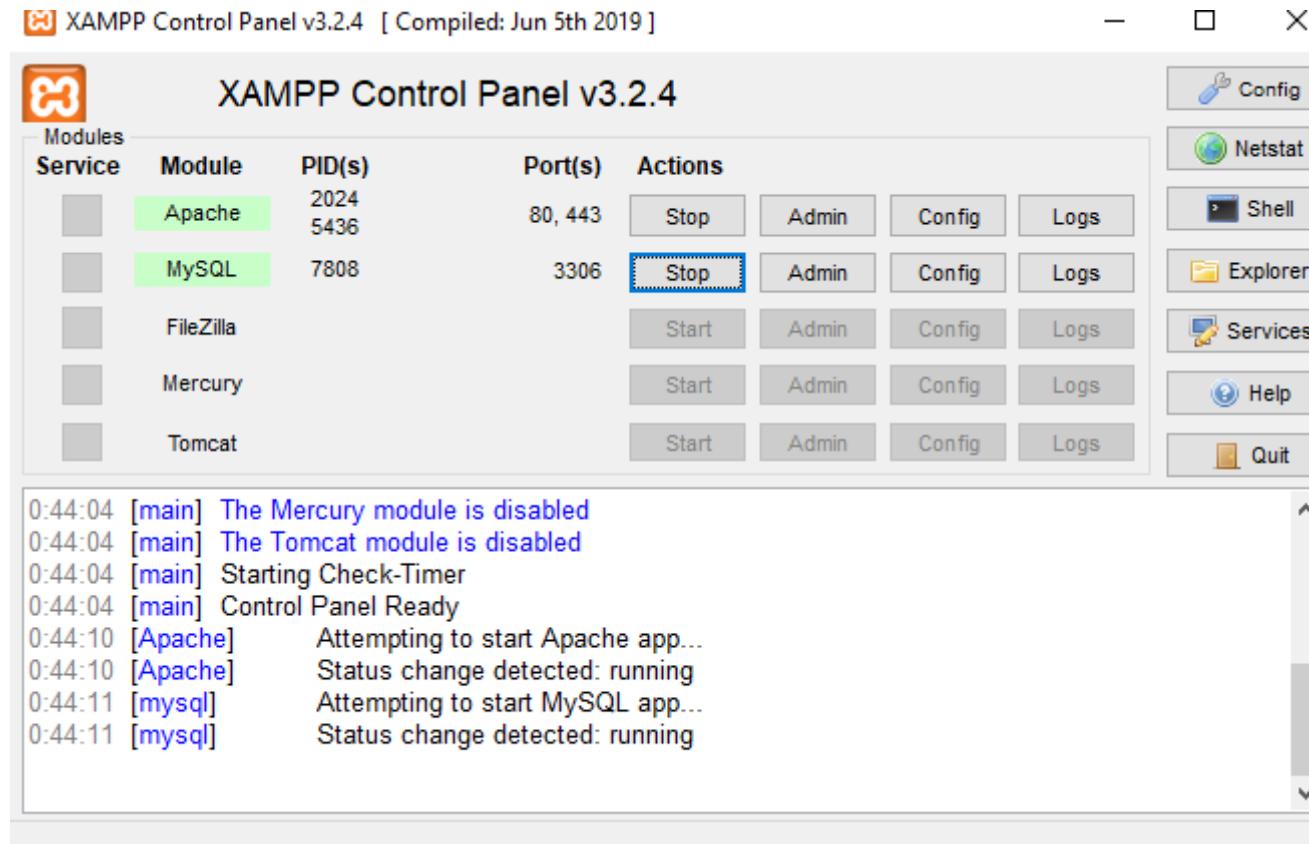
La distribución de XAMPP incluye un panel de control que nos facilitará las labores de arranque y parada de los servidores instalados. Al final de la instalación tenemos la opción de iniciar lo.



También se nos permite configurar el idioma del panel de control.



Una vez iniciado observaremos una pantalla como la siguiente. Desde esta ventana podemos iniciar los servicios pinchando en el botón de "Start" una vez arrancados se pueden parar con el botón de "Stop".



El servidor web instalado está configurado por defecto para buscar los recursos solicitados por el cliente en la ruta "C:\xampp\htdocs\".

Por defecto los servicios no se arrancan cuando se inicia el sistema operativo y por lo tanto se deben iniciar manualmente utilizando el panel de control. Esta es la configuración recomendada para que los servidores no consuman recursos del PC de manera innecesaria.

Es posible cambiar esta configuración si se pincha en el botón "Config" del panel de control.



Configuration of Control Panel



Editor:

notepad.exe



Browser (empty = system default)



Autostart of modules

Apache

FileZilla

Tomcat

MySQL

Mercury

Selected modules will be started on next launch of the Control Panel.

Start Control Panel Minimized

Enable Tomcat output window

Check default ports on startup

Show debug information

[Change Language](#)

[Service and Port Settings](#)

[User Defined Files](#)

[Log Options](#)

Abort

Save

3.2.- Apache Netbeans IDE.



Apache NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. Apache NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

Apache NetBeans IDE 12.5 está desarrollado en Java y requiere un entorno de ejecución adecuado. Apache Netbeans 12.5 es soportado oficialmente si corre sobre Java JDK LTS versiones 11 y 17.

A continuación se presenta un tutorial para instalar Java JDK LTS 17, si se desea instalar una versión más antigua se puede usar el mismo tutorial descargando las distribuciones de Java JDK LTS 11.

Después instalaremos y configuraremos Apache Netbeans para ser utilizado como IDE de desarrollo de aplicaciones PHP.

Instalación Java JDK 17

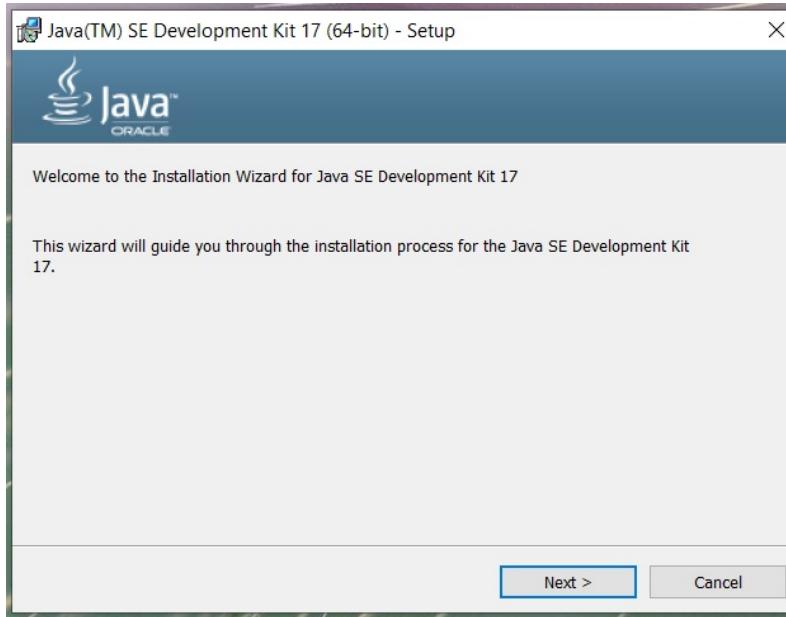
Java™ Development Kit (JDK) es un software para los desarrolladores de Java. Incluye el intérprete Java, clases Java y herramientas de desarrollo Java (JDT): compilador, depurador, desensamblador, visor de applets, generador de archivos de apéndice y generador de documentación.

Este tutorial muestra como instalar el paquete Java JDK 17 en la plataforma Windows 10. Java JDK se puede descargar gratuitamente de la página de descargas de Java de [Oracle](#). La distribución a descargar es la versión x64 Installer.

The screenshot shows a web browser window with the URL <https://www.oracle.com/java/technologies/downloads/#jdk17-windows>. The page header includes the Oracle logo, a search bar, and navigation links for Products, Industries, Resources, Support, Events, and Developer. A note at the top states: "JDK 17 will receive updates under these terms, until at least September 2024." Below this, a section titled "Java SE Development Kit 17 downloads" provides information about the JDK and its tools. A "Documentation Download" button is available. The download section is titled "Windows" and lists three options: "x64 Compressed Archive" (170.64 MB), "x64 Installer" (151.99 MB), and "x64 MSI Installer" (150.88 MB). Each option includes a download link and a SHA256 hash.

Product/file description	File size	Download
x64 Compressed Archive	170.64 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256 🔗)
x64 Installer	151.99 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256 🔗)
x64 MSI Installer	150.88 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256 🔗)

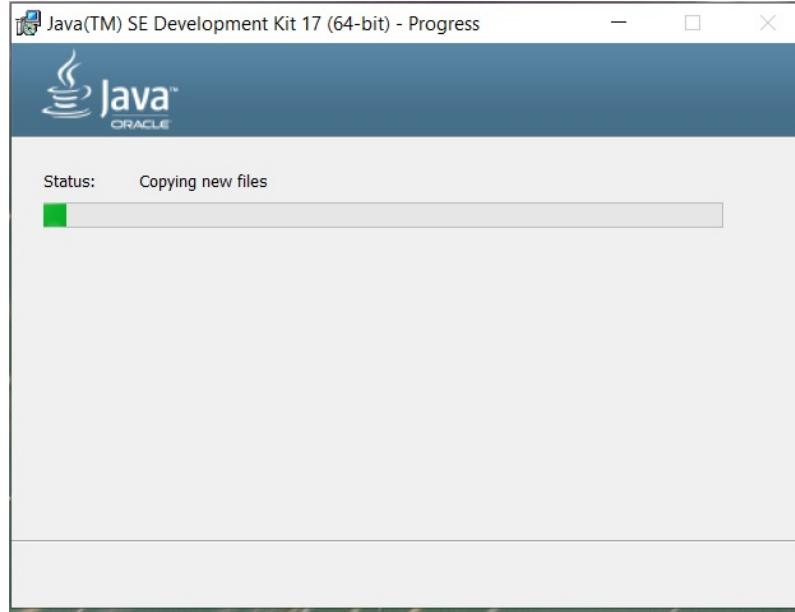
Una vez descargado el paquete de SW iniciamos la instalación ejecutando el instalador descargado.



Se pincha sobre el botón "Next" para iniciar la instalación. La ventana muestra la ruta por defecto donde se instalará esta distribución de Java. Puedes cambiar dicha ruta pinchando en el botón "Change ..".



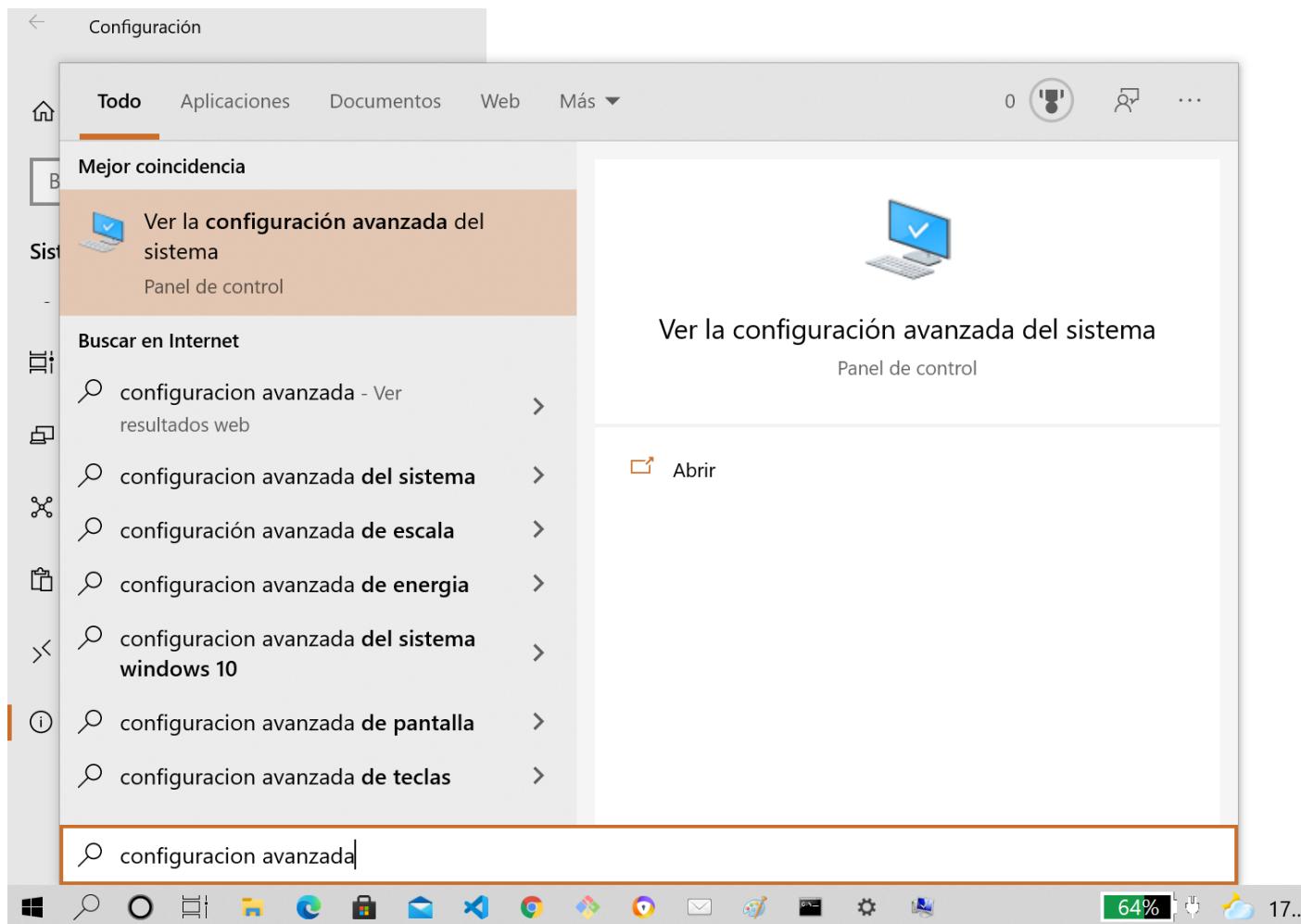
Pincha en el botón "Next" para continuar con la instalación.



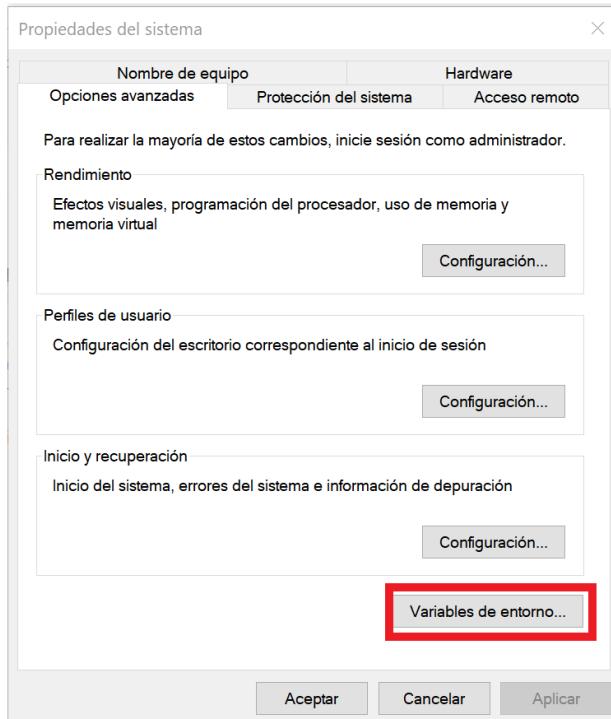
Una vez completada la instalación se muestra una ventana que informa de su finalización. Se puede pinchar en "Close"



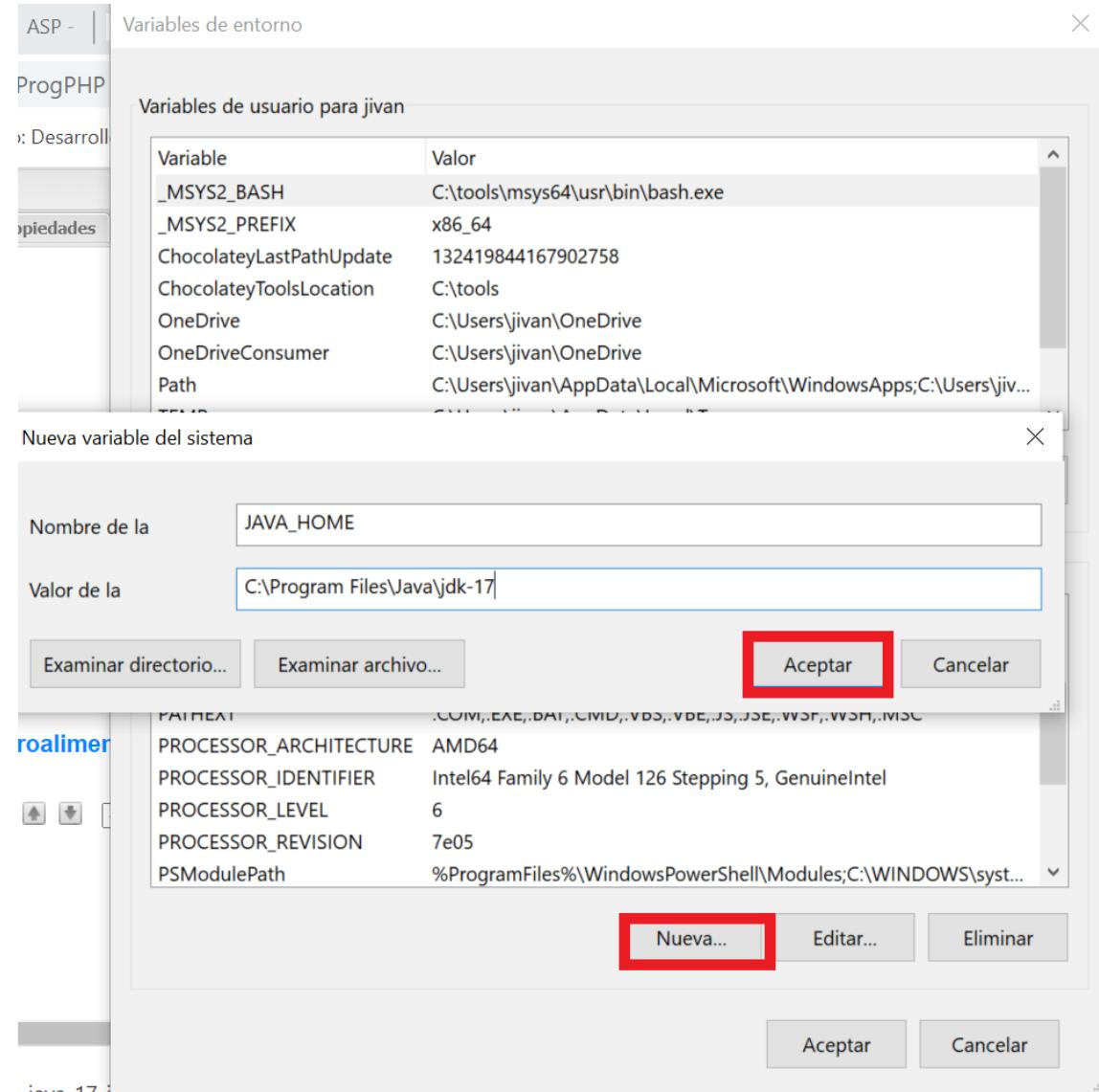
Una vez terminada la instalación es recomendable definir la variable de entorno del sistema JAVA_HOME con el valor de la ruta donde está instalado y también actualizar la variable del sistema path, añadiendo la ruta a la carpeta donde se encuentran los ejecutables de la máquina virtual. Para ello, el menú de inicio introduzca la cadena de búsqueda "configuración avanzada" y acceda a la opción "Ver la configuración avanzada del sistema"



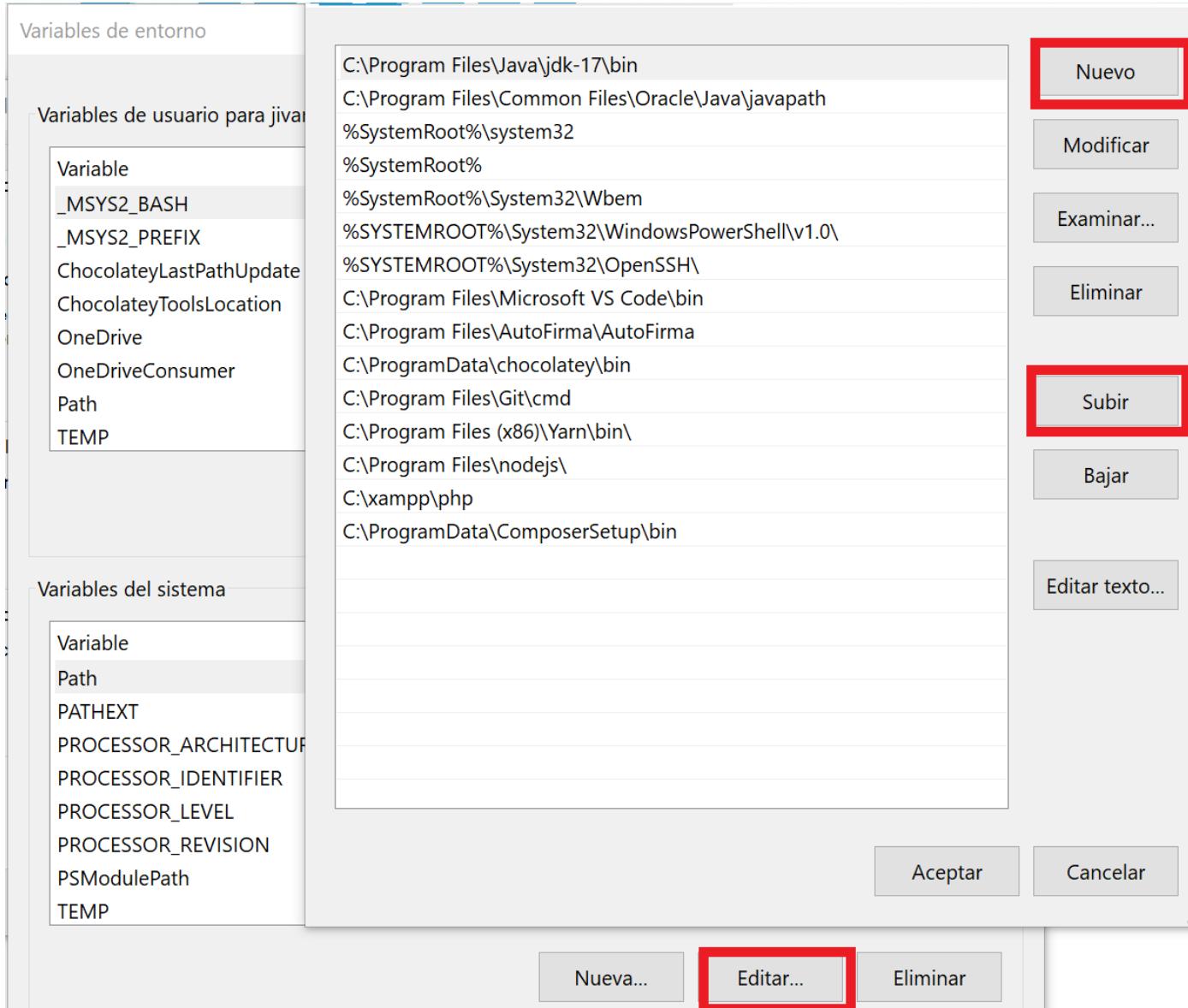
En la ventana de configuración avanzada pinchamos en la opción de "Variables de entorno" en la parte inferior de la ventana.



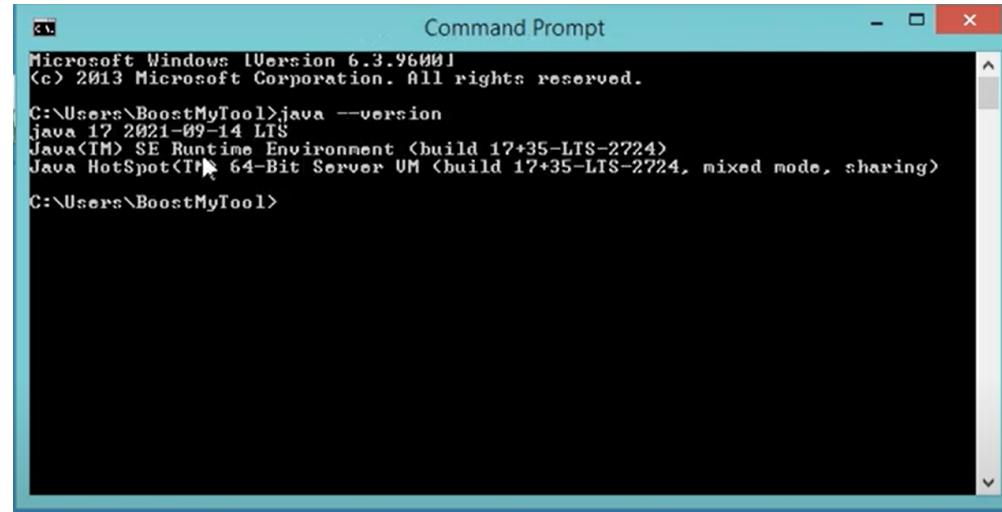
Para crear la variable de entorno de sistema JAVA_HOME pincho en el botón "Nueva" de la sección de variables del sistema. Relleno el nombre de la variable (JAVA_HOME) y el valor (ruta de instalación del paquete de Java JDK) y pincho en el botón "Aceptar".



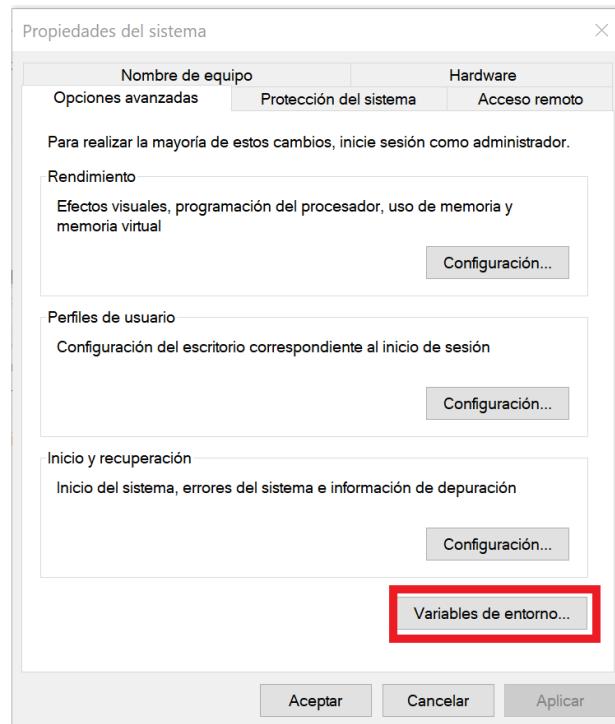
Además, busco y selecciono la variable de sistema "Path" y pincho en el botón "Editar". Añado una entrada nueva pinchando en el botón de "Nuevo". Edito la entrada introduciendo la ruta a la carpeta bin de la instalación de Java JDK (C:\Program Files\Java\jdk-17\bin). Luego subo la entrada a la primera posición pinchando en el botón de "Subir". Finalmente pincho en el botón de "Aceptar".



Para comprobar que la instalación se ha realizado correctamente se puede abrir la terminal de comandos de windows (cmd) y ejecutar el comando `java --version`. El comando devuelve la versión que hemos instalado.



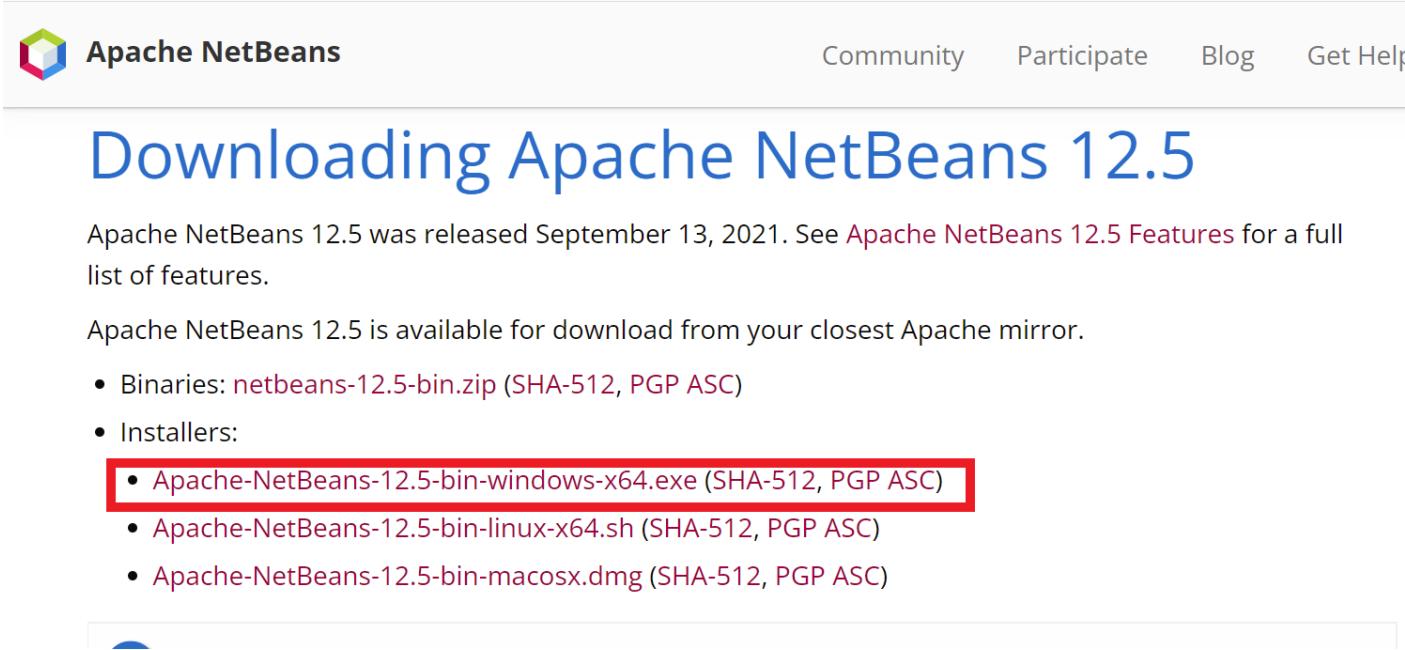
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\BoostMyTool>java --version
java 17 2021-09-14 LTS
Java(TM) SE Runtime Environment (build 17+35-LTS-2724)
Java HotSpot(TM) 64-Bit Server VM (build 17+35-LTS-2724, mixed mode, sharing)
C:\Users\BoostMyTool>



Para crear la variable de entorno de sistema JAVA_HOME pincho en el botón "Nueva" de la sección de variables del sistema. Relleno el nombre de la variable (JAVA_HOME) y el valor (ruta de instalación del paquete de Java JDK) y pincho en el botón "Aceptar".

Instalación Apache Netbeans 12.5 o superior

Este tutorial muestra como instalar el paquete Apache Netbeans 12.5 o superior en la plataforma Windows 10. Apache Netbeans 12.5 se puede descargar gratuitamente de la página de descargas de [Netbeans](#).



The screenshot shows the Apache NetBeans download page. At the top, there's a navigation bar with the Apache NetBeans logo, Community, Participate, Blog, and Get Help links. Below the navigation bar, the title "Downloading Apache NetBeans 12.5" is displayed in large blue text. A paragraph below the title states: "Apache NetBeans 12.5 was released September 13, 2021. See [Apache NetBeans 12.5 Features](#) for a full list of features." Another paragraph says: "Apache NetBeans 12.5 is available for download from your closest Apache mirror." A bulleted list follows, with the first item "Binaries: [netbeans-12.5-bin.zip \(SHA-512, PGP ASC\)](#)" and the second item "Installers:" followed by three links: "Apache-NetBeans-12.5-bin-windows-x64.exe (SHA-512, PGP ASC)" (which is highlighted with a red box), "Apache-NetBeans-12.5-bin-linux-x64.sh (SHA-512, PGP ASC)", and "Apache-NetBeans-12.5-bin-macosx.dmg (SHA-512, PGP ASC)".

Una vez descargado el SW ejecutamos el instalador de Windows. La instalación comienza permitiendo que se ajuste la instalación a nuestras necesidades.

Welcome to the Apache NetBeans IDE 12.5 Installer

The installer will install the NetBeans IDE with the following packs and runtimes.
Click Customize to select the packs and runtimes to install.

Base IDE
Java SE
Java EE
HTML5/JavaScript
PHP



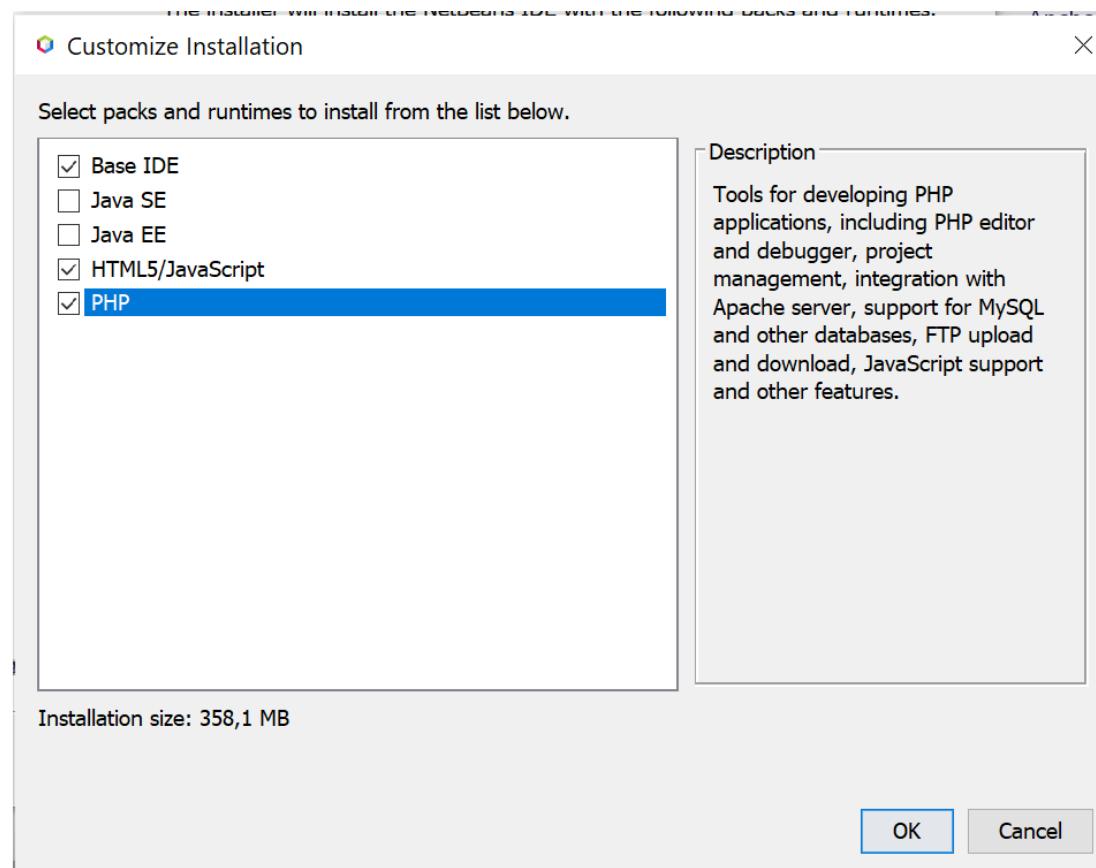
Customize...

Installation Size: 761,1 MB

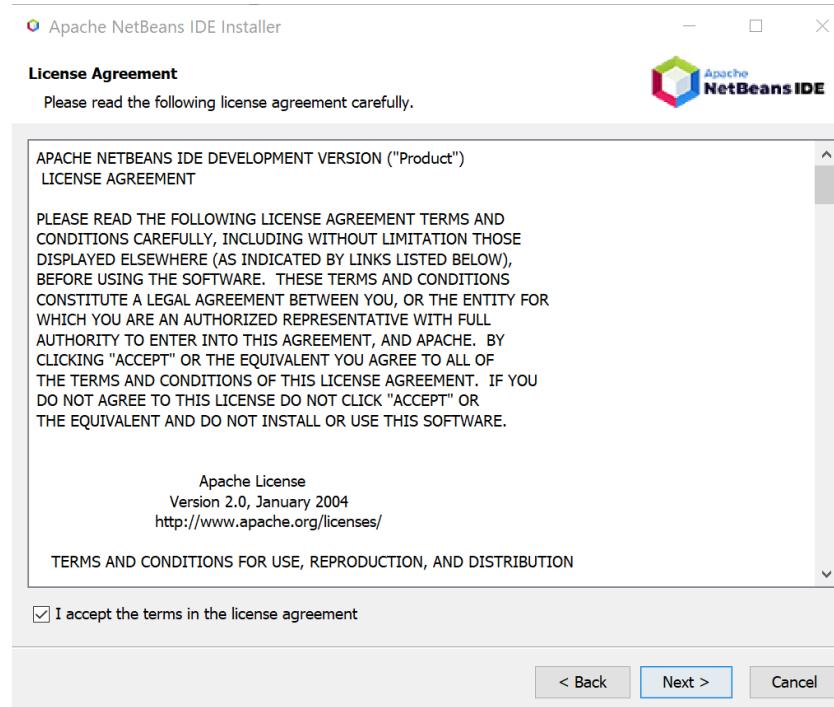
Next >

Cancel

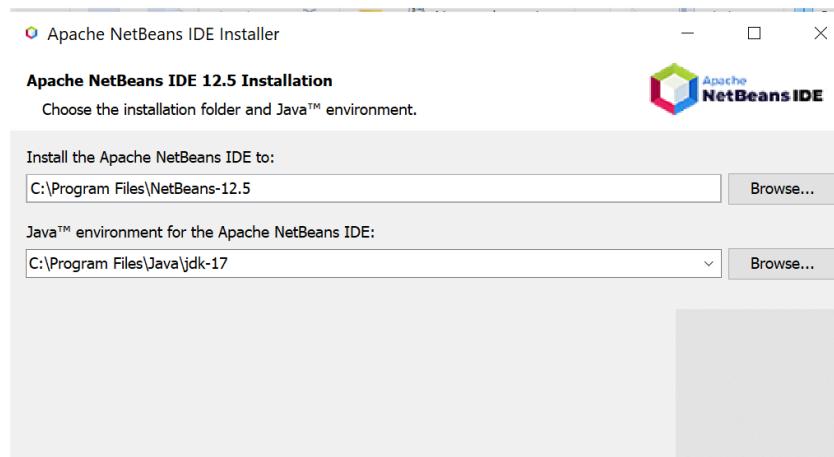
Para éste módulo necesitaremos los componentes de Base IDE, HTML5/JavaScript y PHP. Si se va a usar Netbeans como IDE para otros módulos del ciclo se pueden instalar el resto de componentes.



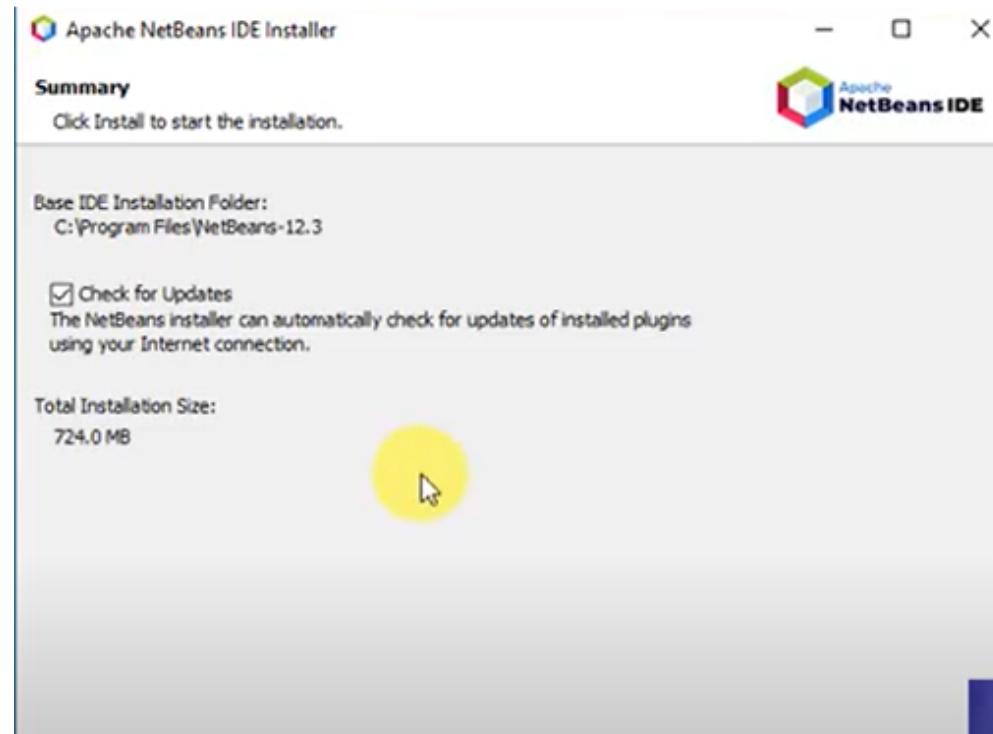
A continuación aceptamos el acuerdo de licencia



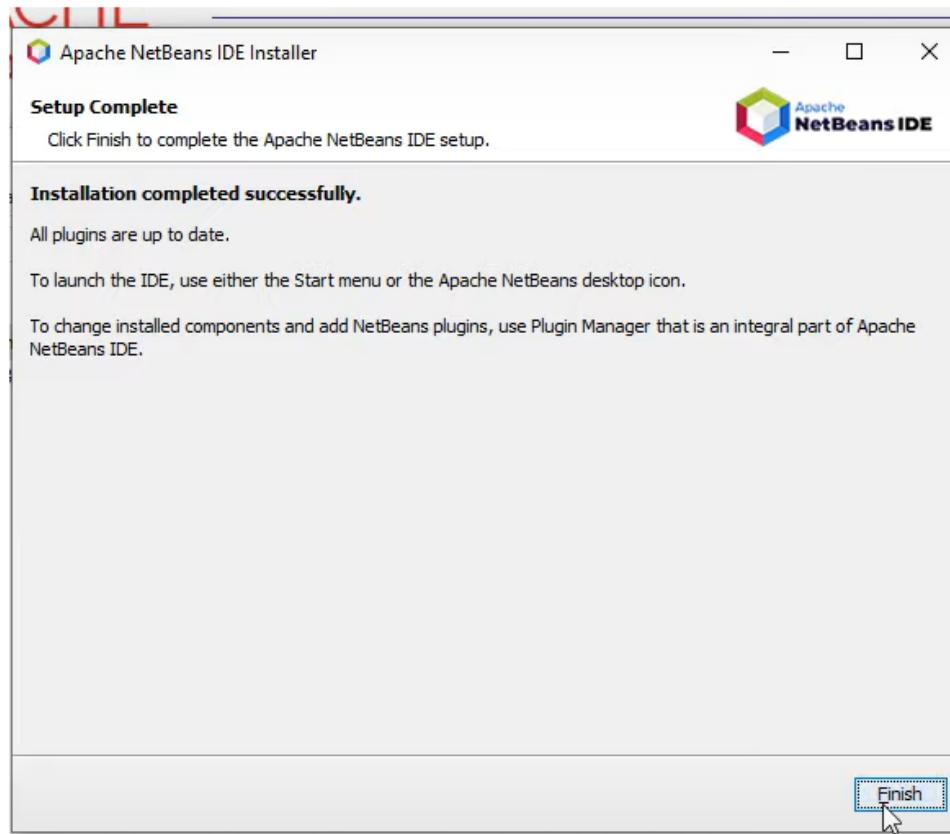
A continuación, aparece una ventana con las rutas donde se va a instalar NetBeans y la ruta a la distribución de java que tienes instalada en tu ordenador y a la que apunta la variable JAVA_HOME. Si todo es correcto puedes continuar, si tienes que hacer alguna modificación a las rutas cambialas pinchando en el botón "Browse".



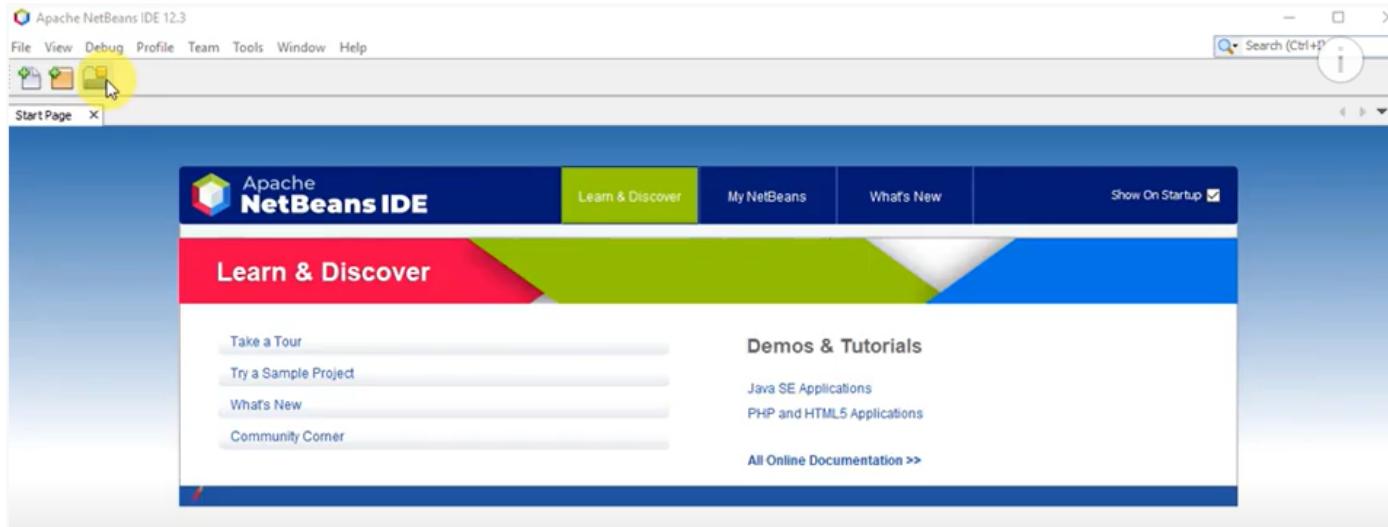
La siguiente ventana nos pregunta si queremos que se descarguen actualizaciones de manera automática. Podemos activarlo si queremos mantener el IDE actualizado a la última versión.



Una vez instalado se mostrará la ventana que informa del final de la instalación. La instalación crea su ícono en el escritorio para poder arrancar el IDE con mayor facilidad



Para comprobar que la instalación ha sido correcta prueba a iniciar el IDE. Te aparecerá una ventana como la siguiente.

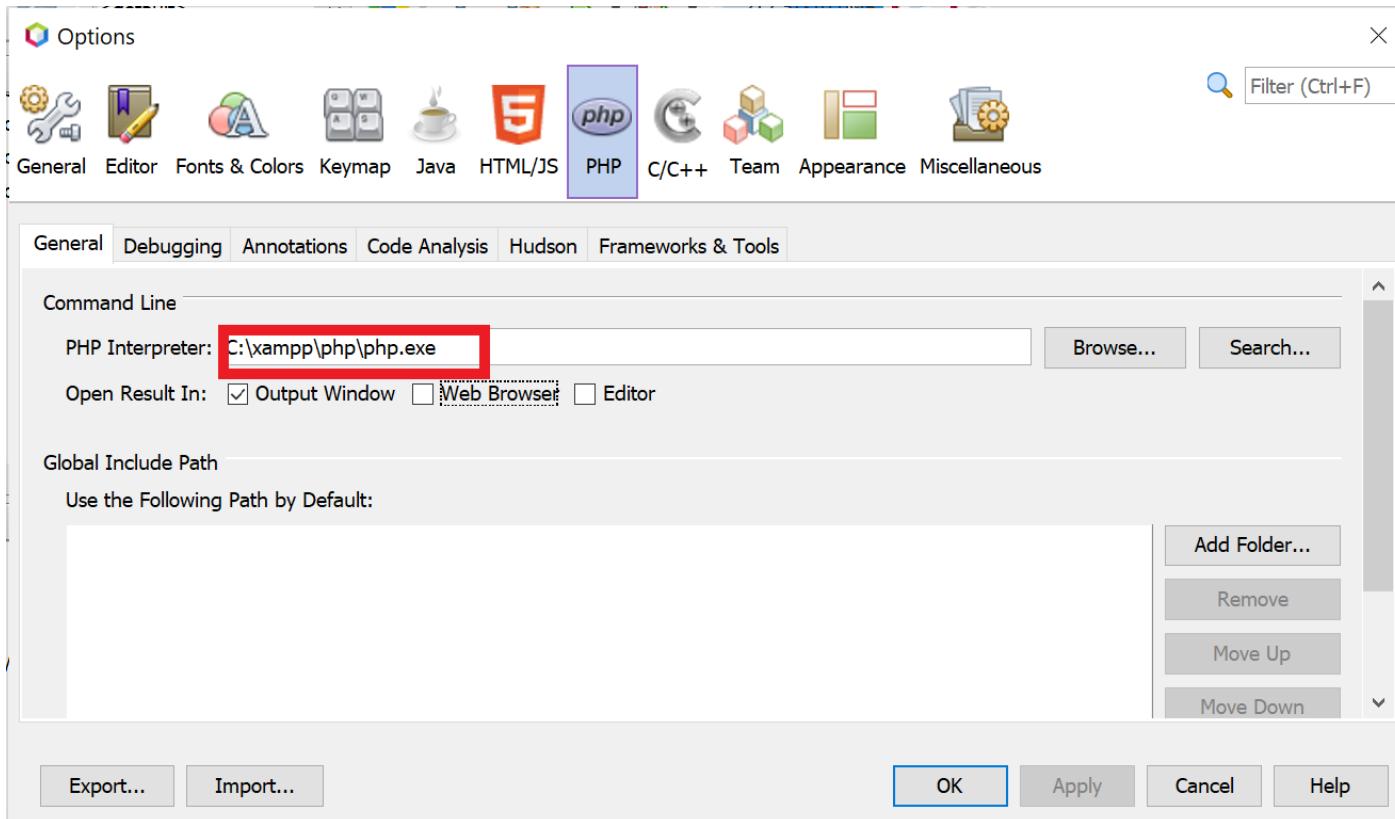


Configuración Apache NetBeans para PHP

Apache Netbeans debe ser configurado para que se convierta en un IDE de desarrollo de PHP. Por un lado se debe configurar la ruta al ejecutable que implementa la funcionalidad de intérprete PHP y por otra se debe instalar y configurar la librería XDebug utilizada para proporcionar la funcionalidad de depuración.

Configuración ruta de intérprete PHP

Desde la opción del menú principal de Tools/Herramientas seleccionamos la opción de Options/Opciones. La ventana de dialogo ofrece varias pestañas en su parte superior y nos dirigimos a la pestaña de PHP. En la ventana de configuración de PHP configuramos la entrada de PHP Interpreter con el ejecutable de PHP incluido en la distribución de XAMPP. El interprete PHP está localizado dentro de la carpeta "php" de la distribución de xampp, C:\xampp\php. (Asegúrate que seleccionas el fichero php.exe y no php.ini)



Configuración XDebug

La función de depuración con NetBeans IDE en PHP se hace por medio de la librería XDebug. La instalación de NetBeans no incluye XDebug por defecto así que hay que instalarla aparte.

Para descargar la versión correcta de la librería de depuración vamos a utilizar el "[wizard](#)" facilitado por la web de XDebug.



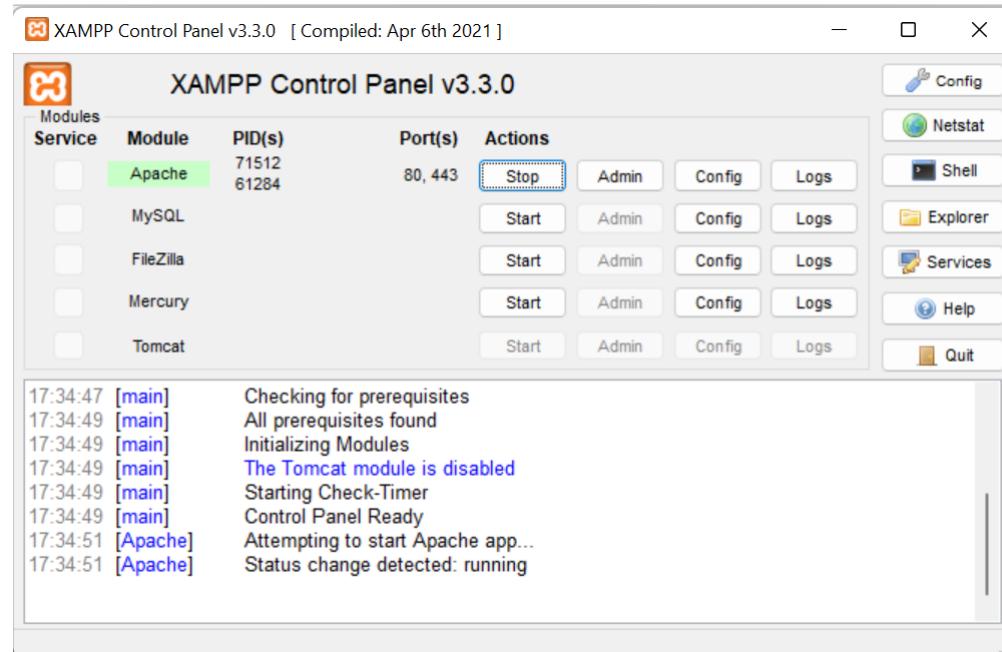
If you find Xdebug useful, please consider [supporting the project](#).

Installation Wizard

This page helps you finding which file to download, and how to configure PHP to get Xdebug running. Please paste the **full** output of `phpinfo()` (either a copy & paste of the HTML version, the HTML source or `php -i` output) and submit the form to receive tailored download and installation instructions.

Para utilizar esta herramienta tenemos que obtener la información sobre el entorno PHP que acabamos de instalar mediante la función `phpinfo`. Dicha información se puede conseguir de varias maneras. A continuación, se presenta un método sencillo de hacerlo.

Empezamos arrancando el servidor Apache utilizando el panel de control de Xampp



Despues visitamos la URL <http://localhost> utilizando nuestro navegador favorito.

A screenshot of the XAMPP welcome page. The header has links for Apache Friends, Applications, FAQs, HOW-TO Guides, PHPInfo (which is highlighted with a red box), and phpMyAdmin. The main content features the XAMPP logo and the text "XAMPP Apache + MariaDB + PHP + Perl".

Welcome to XAMPP for Windows 8.0.10

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

Pinchamos sobre el enlace `phpinfo()` situado en la parte superior derecha y obtenemos una página con toda la información de configuración de nuestro entorno PHP.



System	Windows NT IZA 10.0 build 22000 (Windows 10) AMD64
Build Date	Aug 25 2021 08:42:19
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cscript /nologo /e:jscript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=..\..\..\instantclient\sdk\shared" "--with-oci8-19=..\..\..\instantclient\sdk\shared" "--enable-object-out-dir=..\.obj" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	<i>no value</i>
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20200930
PHP Extension	20200930
Zend Extension	420200930
Zend Extension Build	API420200930,TS,VS16
PHP Extension Build	API20200930,TS,VS16
Debug Build	no
Thread Safety	enabled

Para utilizar el "wizard" de Xdebug debemos copiar toda la información de la página anterior y copiarla en la ventana de la herramienta mostrada al principio.

[Install](#)[Documentation](#)[Report Issues](#)[GitHub](#)

If you find Xdebug useful, please consider [supporting the project](#).

Installation Wizard

This page helps you finding which file to download, and how to configure PHP to get Xdebug running. Please paste the **full** output of `phpinfo()` (either a copy & paste of the HTML version, the HTML source or `php -i` output) and submit the form to receive tailored download and installation instructions.

```
PHP Version 8.0.10
System Windows NT IZA 10.0 build 22000 (Windows 10) AMD64
Build Date Aug 25 2021 08:42:19
Build System Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler Visual C++ 2019
Architecture x64
Configure Command cscript /nologo /e:jscript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=..\..\..\instantclient\sdk,shared" "--with-oci8-19=..\..\..\instantclient\sdk,shared" "--enable-object-out-dir=../obj/" "--
```

Para que el "wizard" nos ayude a escoger la versión de la librería correcta para nuestro entorno pinchamos en el botón [Analyze my `phpinfo\(\)` output](#) y recibimos información que incluye un enlace de descarga de la librería correcta y las instrucciones para instalarla.

[Install](#)[Documentation](#)[Report Issues](#)[GitHub](#)

If you find Xdebug useful, please consider [supporting the project](#).

Installation Wizard

Summary

- **Windows:** yes

Instructions

1. Download [php_xdebug-3.1.5-8.0-vs16-x86_64.dll](#)
2. Move the downloaded file to C:\xampp\php\ext, and rename it to
`php_xdebug.dll`
3. Update `C:\xampp\php\php.ini` to have the line:
`zend_extension = xdebug`
4. Restart the Apache Webserver

Una vez descargada la librería dinámica se renombra a `php_xdebug.dll` y se copia en la carpeta `C:\xampp\php\ext` junto con el resto de librerías de PHP instaladas con Xampp.

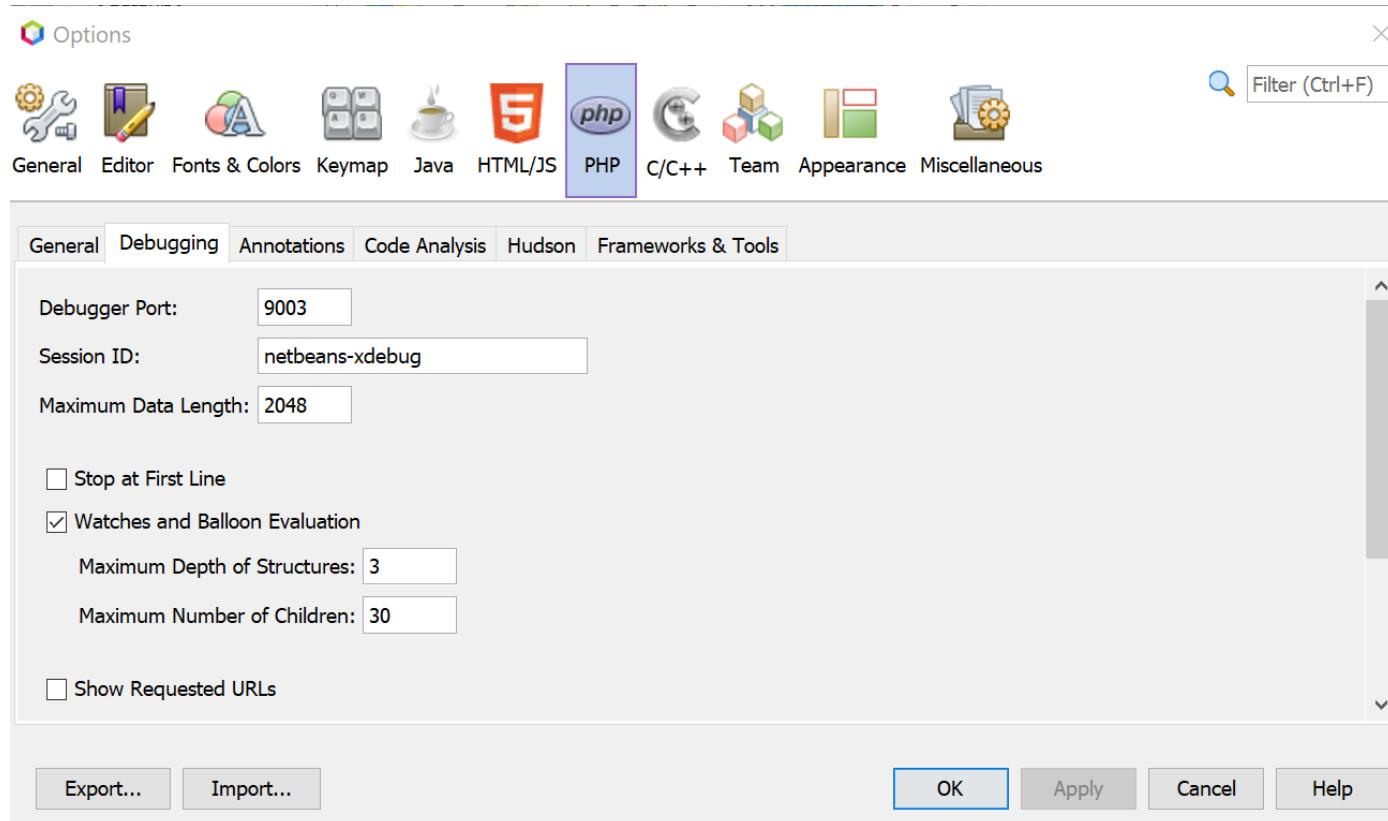
Ya solamente queda configurar el fichero `php.ini` situado en la carpeta `php` de la distribución de Xampp, `C:\xampp\php`. Se deben añadir los siguientes datos de configuración al final del fichero.

```
[xdebug]
zend_extension = "C:\xampp\php\ext\php_xdebug.dll"
xdebug.mode=debug
xdebug.client_host=localhost
xdebug.client_port=9003
xdebug.idekey="netbeans-xdebug"
```

La ruta en el parámetro `zend_extension` debe apuntar a la ubicación donde se copió la librería en el paso anterior.

Una vez realizados estos pasos podemos terminar de configurar la función de depuración en Apache NetBeans IDE accediendo a la ventana de configuración de PHP a la que accedemos desde el menú de Tools (Herramientas), seleccionando Options (Opciones).

Se recomienda la siguiente configuración de la sección de depuración (Debugging)



4.- Conceptos básicos de programación web con PHP.

Caso práctico



El equipo ya está ansioso de **empezar a programar en PHP** aunque sean fragmentos de código muy sencillos. La idea es empezar a familiarizarse con los **elementos más básicos** del lenguaje.

También hay ganas de poner a prueba el entorno de programación que acaban de instalar creando un **primer proyecto PHP en el IDE**.

PHP es un lenguaje de guiones de propósito general, pero diseñado para el desarrollo de páginas web dinámicas utilizando código embebido dentro del lenguaje de marcas. Su sintaxis está basada en la de C / C++, y por lo tanto es muy similar a la de Java. Aunque lo puedes hacer de otras formas, los delimitadores recomendados para incluir código PHP dentro de una página web son <?php y ?>.

El código se ejecuta por un entorno de ejecución con el que se integra el servidor web (normalmente utilizando Apache con el módulo `mod_php`). La configuración tanto del servidor web Apache, como de PHP, se realiza por medio de ficheros de configuración. El de Apache es `httpd.conf`, y el de PHP es `php.ini`. Este fichero, `php.ini`, puede encontrarse en distintas ubicaciones. La función `phpinfo()` que ejecutaste antes te informa, entre otras muchas cosas, del lugar en que se encuentra almacenado el fichero `php.ini` en tu ordenador. En nuestro entorno se encuentra en `C:/xampp/php/php.ini`.



[Colin Viebrock \(CC BY-SA\)](#)

Dependiendo de cómo se integre PHP con Apache, los cambios que realices en su fichero de configuración se aplicarán en un momento o en otro. Si como es nuestro caso, utilizamos `mod_php` para ejecutar PHP como un módulo de Apache, las opciones de configuración de PHP se aplicarán cada vez que se reinicie Apache. Por tanto, no te olvides de hacerlo cada vez que hagas cambios en `php.ini`. Por ejemplo: `sudo systemctl restart apache2`

Algunas de las directivas más utilizadas que figuran en el fichero `php.ini` son:

- ✓ `short_open_tags`. Indica si se pueden utilizar en PHP los delimitadores cortos `<? y ?>`. Es preferible no usarlos, pues puede causarnos problemas si utilizamos páginas con XML. Para prohibir la utilización de estos delimitadores con PHP le asignamos a esta directiva el valor `Off`. Por defecto suelen estar a `On`.
- ✓ `max_execution_time`. Permite que puedas ajustar el número máximo de segundos que podrá durar la ejecución de un script PHP. Evita que el servidor se bloquee si se produce algún error en un script.
- ✓ `display_errors`. Permite visualizar los errores que se produzcan en el código PHP, Para ver el nivel de detalles de los errores mostrados se complementa con la directiva siguiente, los valores recomendados por defecto son: Para un entorno en producción a `Off`, para un entorno en desarrollo a `On`. Por defecto suele estar en `On`.
- ✓ `error_reporting`. Indica qué tipo de errores se mostrarán en el caso de que se produzcan. Por ejemplo, si haces `error_reporting = E_ALL`, te mostrará todos los tipos de errores. Si no quieres que te muestre los avisos pero sí otros tipos de errores, puedes hacer `error_reporting = E_ALL & ~E_NOTICE`.
- ✓ `file_uploads`. Indica si se pueden o no subir ficheros al servidor por HTTP.
- ✓ `upload_max_filesize`. En caso de que se puedan subir ficheros por HTTP, puedes indicar el límite máximo permitido para el tamaño de cada archivo. Por ejemplo, `upload_max_filesize = 1M`.
- ✓ `post_max_size`. Complementa la directiva anterior, establece el tamaño máximo de un archivo subido por POST, Por ejemplo, `post_max_size = 1M`.

Iremos viendo otras directivas a lo largo del módulo cuando las vayamos necesitando. Realiza la siguiente actividad para comprobar si recuerdas las que acabamos de ver.

Actividad de repaso

Responde verdadero o falso a los siguientes supuestos.

Caso 1: Para mi proyecto necesito subir pdf de hasta 5M de tamaño. Las subidas serán todas por formularios (POST), he revisado la directiva siguiente y su valor, siendo esta: `upload_max_filesize = 15M`.

¿Podré subir mis archivos?

Sugerencia

- Verdadero Falso

Falso

No es suficiente con la directiva anterior, necesitamos establecer el valor de la directiva `post_max_size` a 5M como mínimo y `file_uploads` a **On**.

Caso 2: En un archivo php que debo modificar he visto el código siguiente:

```
<?
$var=45;
echo $var;
?>
```

El valor de la directiva `short_open_tags` es **Off**.

¿Funcionará dicho código?

- Verdadero Falso

Falso

Para que el código funcione el valor de la directiva `short_open_tags`, debe ser **On**. Aunque no es recomendable.

Caso 3: Acabo de cambiar mi proyecto PHP, de producción a desarrollo, el valor de la directiva `display_errors` es el que viene por defecto.

¿Lo he hecho bien?

- Verdadero Falso

Falso

Por defecto el valor de la directiva `display_errors` es **On** que no se recomienda para un entorno en producción, solo para entornos en desarrollo. Lo correcto hubiese sido ponerlo a **Off**.

Para saber más

En la documentación de PHP se incluye una lista completa de las directivas que se pueden utilizar en `php.ini`.

[Lista completa de las directivas](#)

Autoevaluación

Relaciona cada parámetro de la configuración de PHP con su finalidad:

Ejercicio de relacionar

Parámetro	Relación	Finalidad
<code>file_uploads</code>	<input type="checkbox"/>	1. Indica si se pueden subir ficheros al servidor web.
<code>max_execution_time</code>	<input type="checkbox"/>	2. Indica qué tipo de delimitadores se pueden usar para el código PHP.
<code>upload_max_filesize</code>	<input type="checkbox"/>	3. Limita el tamaño máximo de los ficheros que se suben al servidor.
<code>short_open_tags</code>	<input type="checkbox"/>	4. Limita el tiempo máximo de ejecución de un guión PHP.

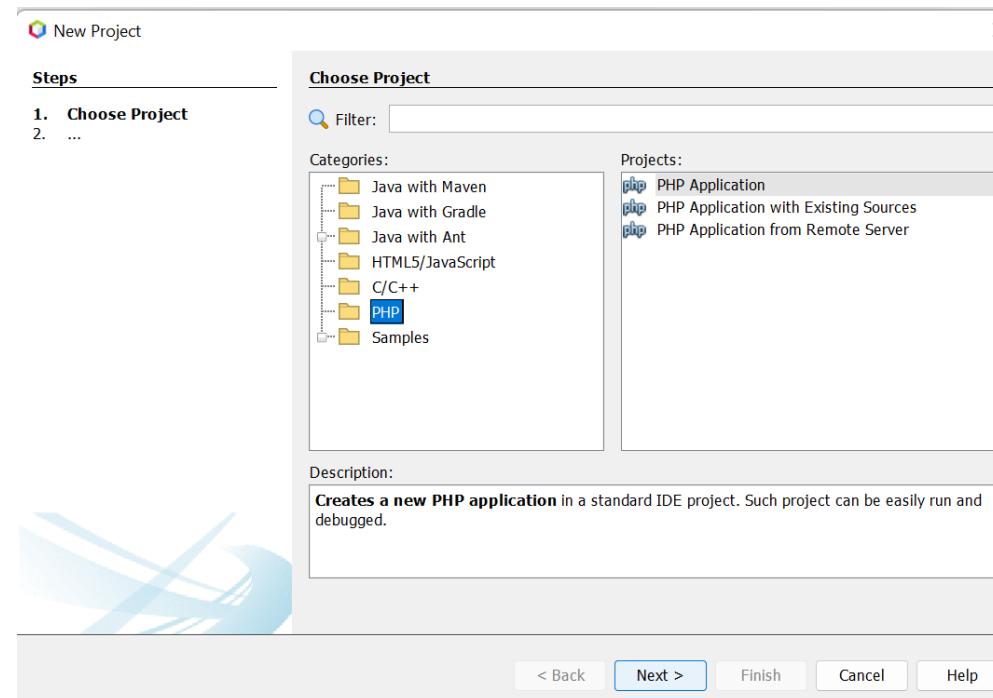
Enviar

Hay muchos parámetros de configuración en `php.ini`. Éstos son solo algunos de los más importantes.

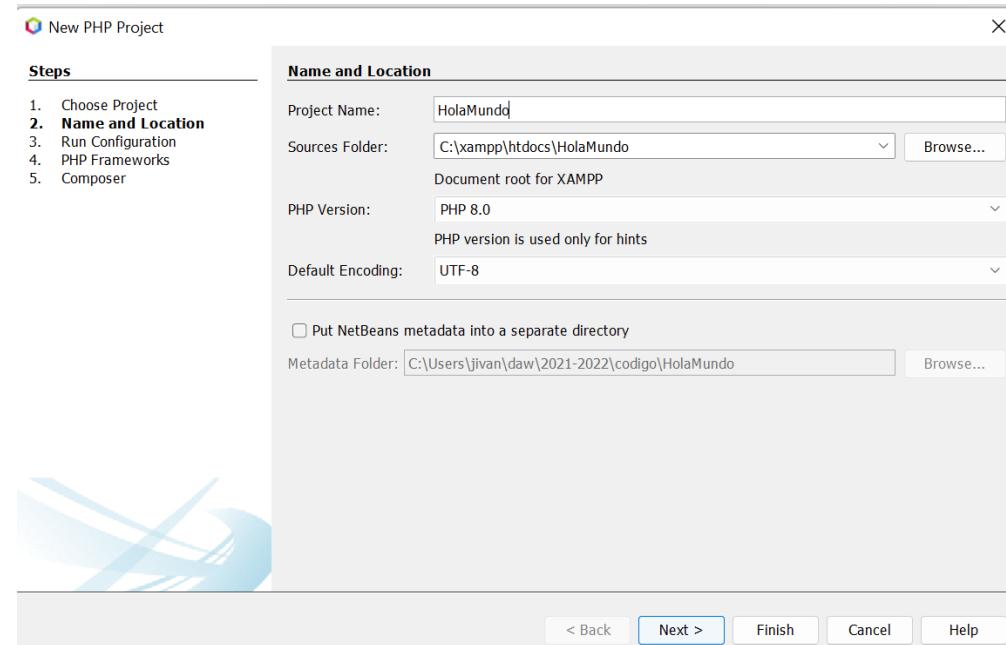
4.1.- Crear proyecto PHP "Hola mundo".

En esta sección vamos a mostrar cómo se crea un proyecto PHP utilizando Apache Netbeans. El proyecto consiste en programar una aplicación PHP que imprima el mensaje "Hola mundo" en el navegador.

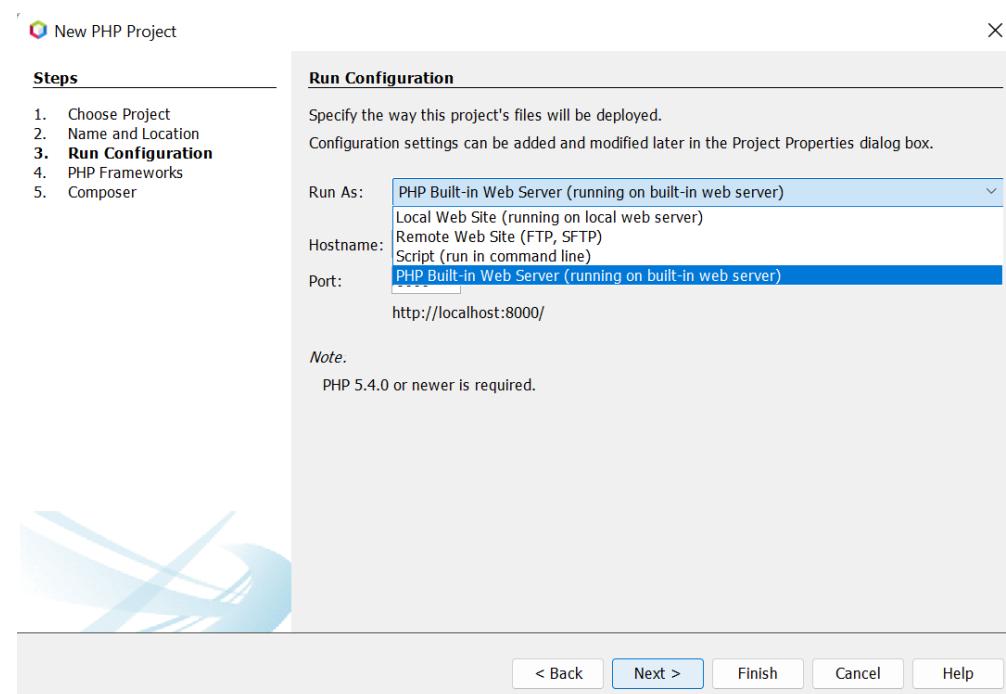
Una vez arrancado Apache Netbeans se solicita la creación de un nuevo proyecto desde el menu **Archivo (File)** seleccionando la opción **Nuevo Proyecto (New Project)**. Para crear un nuevo proyecto pinchamos en **Siguiente (Next)**.



A continuación, podemos darle nombre al nuevo proyecto y decidir donde queremos ubicar nuestro proyecto en el disco duro de nuestro PC. También podemos especificar la versión del intérprete PHP que queremos que ejecute nuestro código. En este curso vamos a optar por utilizar la versión 8.0 siempre que sea posible. Por último podemos definir el modo de codificación de los caracteres que dejaremos con el valor por defecto.



El siguiente aspecto a configurar es el método de ejecución de nuestro proyecto PHP. Existen las siguientes opciones:

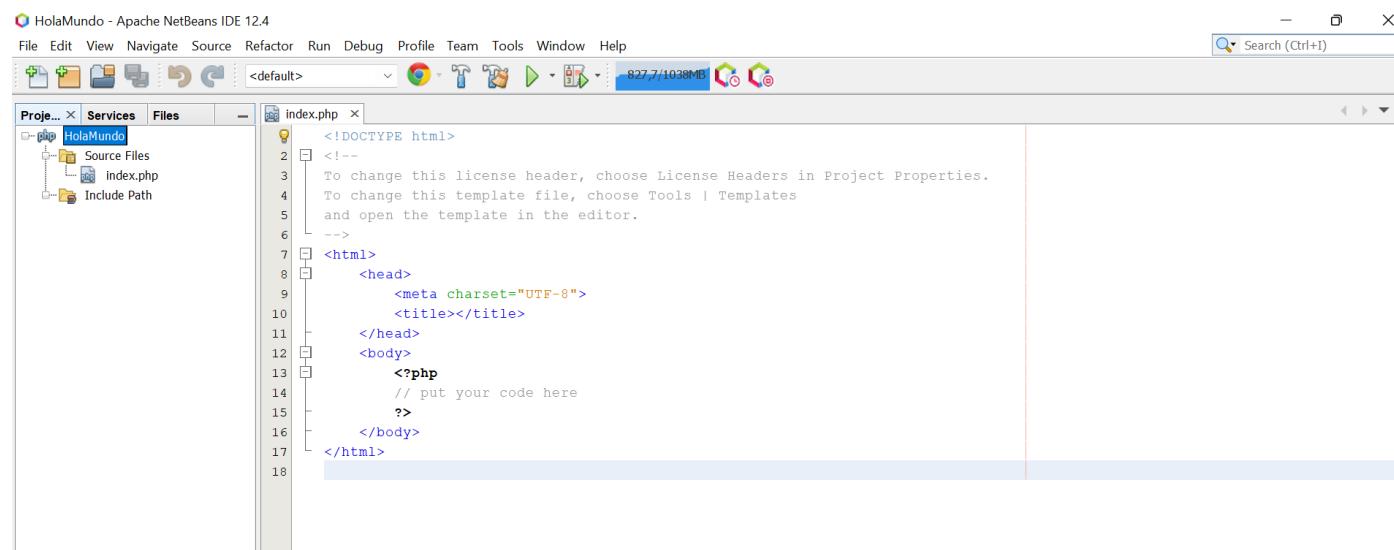


- ✓ **Sitio Local (Local Web Site):** El proyecto se ejecuta utilizando un servidor web local, por ejemplo el servidor Apache instalado con XAMPP. En este caso hay que asegurarse que el servidor web está arrancado y que el proyecto se guarda en la carpeta C:\xampp\htdocs. Se puede especificar la ubicación del archivo raíz del proyecto usando el parámetro URL de proyecto (Project URL).
- ✓ **Servidor Web integrado PHP (PHP Built-in Web Server):** El proyecto utiliza un servidor web integrado en el IDE y desplegado en el puerto 8000, a diferencia de Apache que lo hace en el puerto 80 por defecto. Esta es la manera más sencilla de ejecutar nuestras aplicaciones mientras las desarrollamos.
- ✓ **Sitio web remoto (Remote Web Site):** El proyecto se ejecuta después de ser desplegado vía FTP en un servidor de hosting remoto. Es necesario aportar datos de acceso al servidor vía FTP.
- ✓ **Script (Script run in command line):** La aplicación PHP se ejecuta desde la línea de comandos y no mediante peticiones realizadas desde un navegador web.

Durante este curso se recomienda que se use la configuración que usa el servidor web integrado en Apache Netbeans, de manera opcional también se puede usar un servidor web local, en nuestro caso Apache.

Posteriormente, el proceso de creación nos ofrece la posibilidad de utilizar un framework PHP tal como Symfony, Zend, etc, aunque en nuestro caso no haremos uso de ninguno de ellos. Finalmente tampoco utilizaremos la herramienta composer en las primeras unidades del curso.

Una vez llegados a este punto podemos pinchar en el botón Fin (Finish) y se creará el proyecto php en el panel de proyectos ubicado en la izquierda del interfaz.

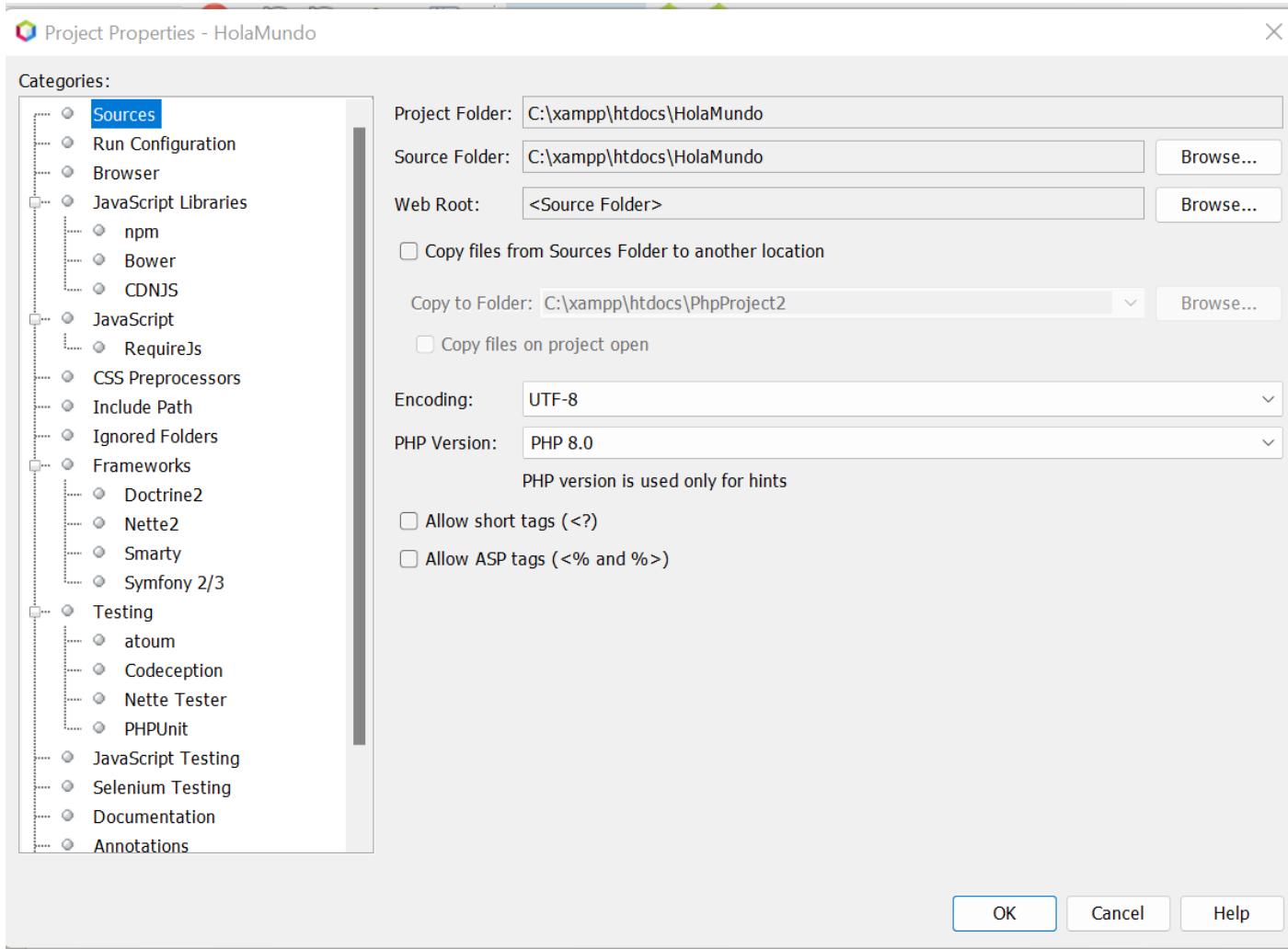


Para completar el proyecto de manera que imprima el mensaje de "Hola mundo" añadimos el siguiente código entre las etiquetas <?php y ?>. Estas etiquetas indican al interprete PHP que debe ejecutar el código que hay dentro y colocar en su lugar la salida generada por dicho

código. En este caso la sentencia echo escribe la cadena "Hola mundo" a la página dinámica que se muestra cuando la página se carga en el navegador.

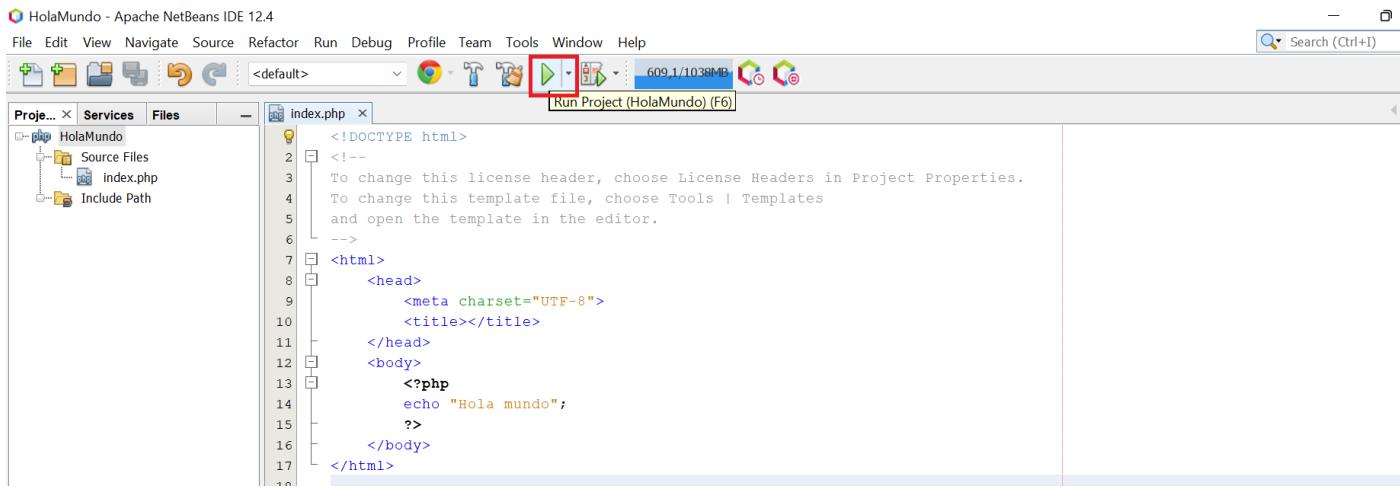
```
<?php  
    echo "Hola mundo";  
?>
```

Una vez creado el proyecto es posible modificar sus propiedades pinchando con el botón derecho del ratón en el nombre del proyecto y seleccionando la opción de **Propiedades (Properties)**.



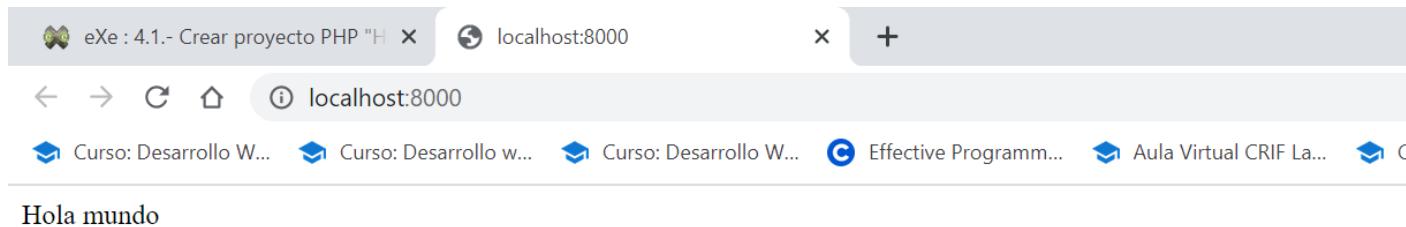
Desde esta ventana se pueden modificar aspectos importantes del proyecto tales como ubicaciones de los fuentes del proyecto, configuración de ejecución del proyecto y navegador utilizado para ejecutar el proyecto. Abre la sección **Navegador (Browser)** y configura Chrome como navegador utilizado en la ejecución.

Podemos ejecutar el proyecto de varias maneras, se puede pinchar con el botón derecho sobre el nombre del proyecto y seleccionar la opción **Ejecutar (Run)**. Alternativamente se puede pinchar en el ícono de ejecución que se muestra a continuación. También se puede pulsar el atajo F6.



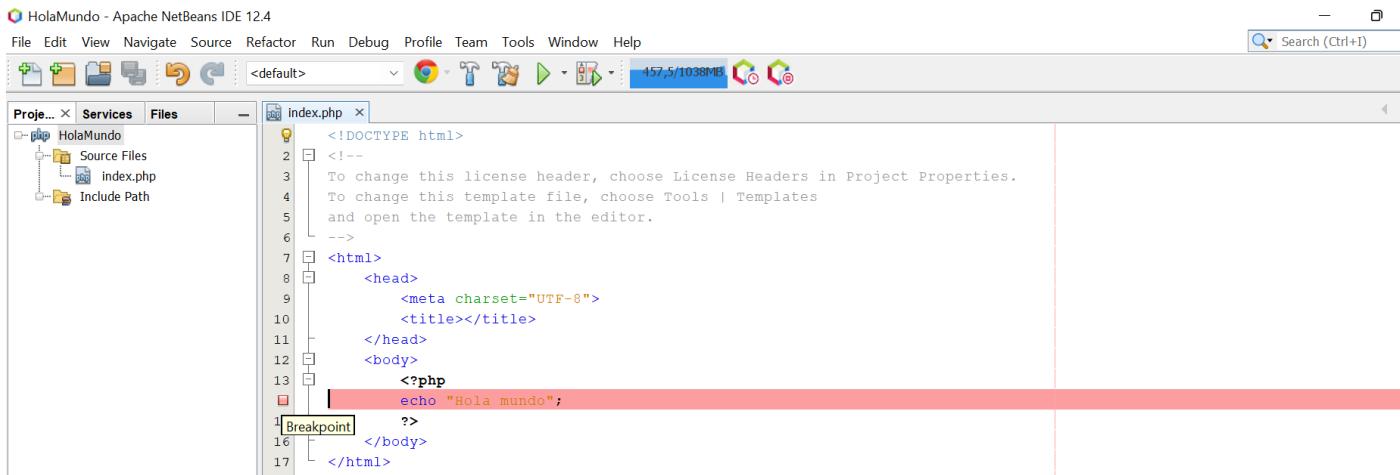
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>
<?php
echo "Hola mundo";
?>
</body>
</html>
```

La ejecución del script genera la siguiente página web mostrada por el navegador.



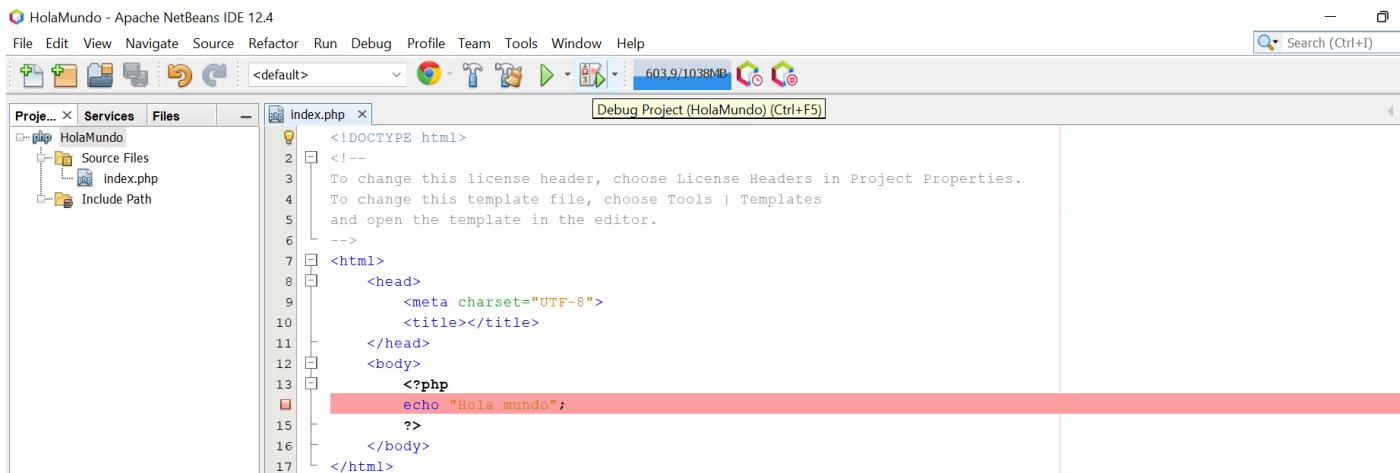
Para terminar esta sección vamos a realizar las comprobaciones necesarias para asegurarnos que hemos instalado y configurado la funcionalidad de depuración correctamente. Para ello vamos a crear un punto de interrupción en la línea de la instrucción `echo` y vamos a lanzar la aplicación con la opción de depuración para comprobar que la ejecución de la aplicación se detiene en esa línea.

Para crear el punto de interrupción pinchamos sobre el número de línea del script donde figura la instrucción `echo`.



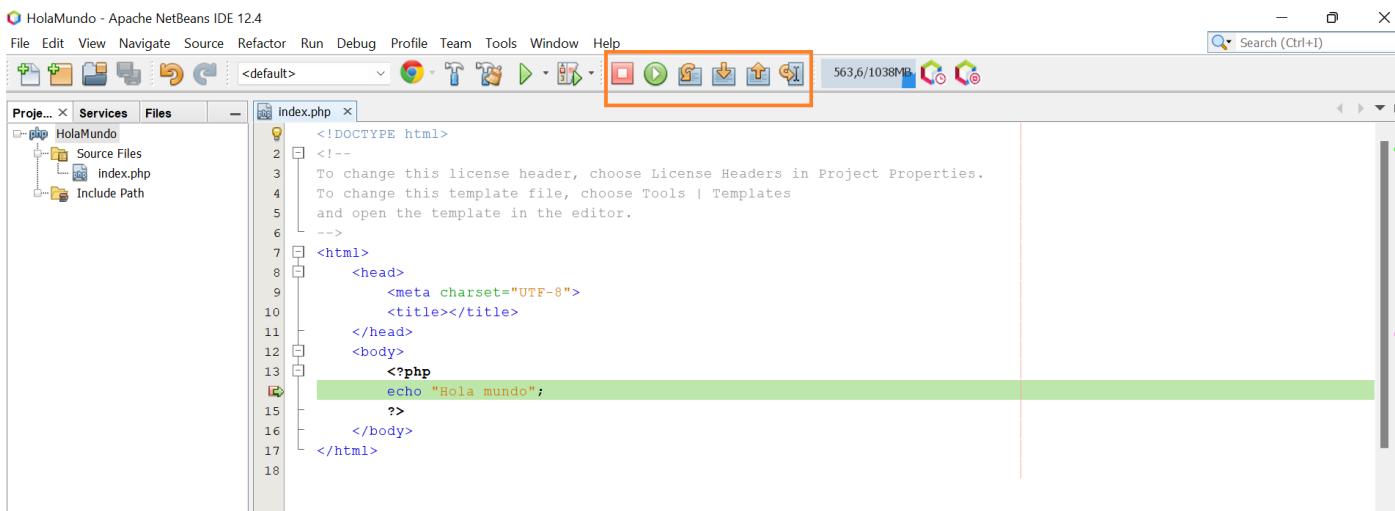
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>
<?php
echo "Hola mundo";
?>
</body>
</html>
```

Después ejecutamos la aplicación en modo depuración, del mismo modo que la ejecución normal se puede conseguir de varias maneras, se puede pinchar con el botón derecho sobre el nombre del proyecto y seleccionar la opción Depurar (Debug). Alternativamente se puede pinchar en el ícono de ejecución con depuración que se muestra a continuación. También se puede pulsar el atajo Ctrl + F5.



```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>
<?php
echo "Hola mundo";
?>
</body>
</html>
```

Cuando se ejecuta el proyecto en modo depuración deberíamos ver una línea verde en el punto del script donde hemos solicitado que se detenga. A partir de ese momento tenemos el control para ir ejecutando el script tal y como nos interese para encontrar los errores que observemos en el comportamiento del programa en ejecución. Para ello utilizaremos la botonera que aparece resaltada en la imagen.



Si la línea verde no aparece en el editor de Apache Netbeans deberíamos repasar los pasos de instalación y configuración del depurador Xdebug de la sección anterior.

Hablaremos más extensamente del depurador en la siguiente unidad.

4.2.- Literales, constantes, variables y tipos de datos.

Los lenguajes de programación imperativos permiten escribir algoritmos que transforman unos datos o valores de entrada en unos datos de salida o resultado. Los datos son representados y almacenados en un programa PHP por medio de símbolos tales como literales, constantes y variables. Los valores manipulados por un programa pueden ser de distintos tipos, tales como enteros, decimales, cadenas de caracteres y booleanos.

En las siguientes secciones profundizaremos un poco más en estos conceptos.



Debes conocer

La función `var_dump()` devuelve una cadena con el valor y el tipo del valor de lo que pasamos a la función como parámetro de entrada. Esta función nos va a permitir visualizar los valores manipulados en nuestro programa durante su ejecución. A continuación se muestran ejemplos del uso de `var_dump` con literales.

- `var_dump(45)`: devuelve la cadena `int(45)` indicando que se trata del entero 45.
- `var_dump("HOLA")`: devuelve la cadena `string(4) "HOLA"` indicando que se trata de la cadena o *string* "HOLA" de 4 caracteres.

Se puede investigar mas sobre `var_dump` en el [manual de PHP](#).

Literales

Los literales en PHP se utilizan para representar valores concretos de tipo entero, decimal, cadena de caracteres y booleanos. Por ejemplo el literal `32` representa al entero 32, el literal `0.45` representa al decimal 0,45, el literal `false` representa el valor booleano `false` y el literal `"HOLA"`

representa a la cadena de caracteres "HOLA".

Los literales enteros pueden expresarse en mediante notación decimal (base 10), hexadecimal (base 16), octal (base 8) o binaria (base 2), opcionalmente precedidos por un signo (- o +).

```
<?php
    var_dump(1234); // número decimal int(1234)
    var_dump(-123); // un número negativo int(-123)
    var_dump(0123); // número octal (equivale a 83 decimal) int(83)
    var_dump(0x1A); // número hexadecimal (equivale a 26 decimal) int(26)
    var_dump(0b11111111); // número binario (equivale al 255 decimal) int(255)
?>
```

Los literales en "coma flotante" o float representan los valores decimales. Estos valores pueden expresarse mediante números decimales o notación científica.

```
<?php
    var_dump(1.234); // float(1.234)
    var_dump(1.2e3); // float(1200)
    var_dump(7E-10); // float(7.0E-10)
?>
```

Los literales cadena de caracteres pueden crearse utilizando comilla simple, comilla doble y también con sintaxis HEREDOC y NOWDOC. La diferencia entre la comilla simple y la comilla doble radica en cómo se procesan las caracteres de la cadena antes de devolver su valor.

Cuando se usan comillas simples, sólo se realizan dos sustituciones dentro de la cadena:

1. Cuando se encuentra la secuencia de caracteres '\', se muestra en la salida una comilla simple. Si no se hace así la comilla cerraría la comilla inicial y se produciría un error sintáctico.
2. Cuando se encuentra la secuencia \\, se muestra en la salida una barra invertida.

```
<?php
    var_dump('Le dió p\'al pelo'); // string(17) "Le dió p'al pelo"
?>
```

Sin embargo, las cadenas creadas con comillas dobles pueden incorporar los siguientes caracteres especiales:

Secuencias de escape.

Secuencia	Resultado
\\	se muestra una barra invertida.
\'	se muestra una comilla simple.
\"	se muestra una comilla doble.
\n	se muestra un avance de línea (LF o 0x0A (10) en ASCII).
\r	se muestra un retorno de carro (CR o 0x0D (13) en ASCII).
\t	se muestra un tabulador horizontal (HT o 0x09 (9) en ASCII).
\v	se muestra un tabulador vertical (VT o 0x0B (11) en ASCII).
\f	se muestra un avance de página (FF o 0x0C (12) en ASCII).
\\$	se muestra un signo del dólar.

```
<pre>
<?php
var_dump("incluyo algún \t tabulador y algún \n retorno de carro");
?>
</pre>
```

Se añade la etiqueta html `<pre>` para que el navegador muestre los caracteres de tabulador y retorno de carro ya que dichos caracteres son filtrados por el navegador si son parte de un fichero HTML.

Otra particularidad muy interesante de los literales cadena creados con comillas dobles es que pueden contener variables que son sustituidas por su valor antes de devolver la cadena. Lo veremos con más detalle cuando introduzcamos las variables en PHP.

Los literales booleanos son `true` y `false` y representan los valores booleanos *Verdadero* o *true* y *Falso* o *false*.

```
<?php
var_dump(true); // bool(true)
var_dump(false); // bool(false)
?>
```

Constantes

Una constante es un identificador o un nombre que asociamos a un valor simple. Una vez definida la asociación entre el identificador y el valor no puede ser modificada. Una vez definida, la constante puede ser usada en el programa para referirse al valor concreto de manera que si dicho valor varía solo habría que redefinir la constante al inicio del programa.

```
<?php
define('MIN_VALUE', 0.5);
define('CONTRASEÑA', 'PASSWORD');
var_dump(MIN_VALUE); // float(0.5)
```

```
var_dump(CONTRASEÑA); // string(8) "PASSWORD"
?>
```

Si la función var_dump() recibe una constante como parámetro de entrada devuelve el valor y su tipo asociado a dicha constante.

Variables

Como en todos los lenguajes de programación, en PHP puedes crear variables para almacenar valores. Las variables en PHP siempre deben comenzar por el signo \$. Los nombres de las variables deben comenzar por letras o por el carácter _, y pueden contener también números. Sin embargo, al contrario que en muchos otros lenguajes, en PHP no es necesario declarar una variable ni especificarle un tipo (entero, cadena,...) concreto. Para empezar a usar una variable, simplemente asignarle un valor utilizando el operador =.

En esta sección vamos a usar la función var_dump que recibe como parámetro de entrada una o más variables y devuelve una cadena con sus valores y sus tipos. Por ejemplo si ejecutamos el siguiente código:

```
<?php
$mi_variable = 7;
var_dump($mi_variable);
?>
```

Obtendremos el resultado:

```
int(7)
```

Dependiendo del valor que se le asigne, a la variable se le aplica un tipo de datos, que puede cambiar si cambia su contenido. Esto es, el tipo de la variable se decide en función del contexto en que se emplee.

```
<?php  
// Al asignarle el valor 7, la variable es de tipo "entero"  
$mi_variable = 7;  
var_dump($mi_variable); // int(7)  
// Si le cambiamos el contenido  
$mi_variable = "siete";  
var_dump($mi_variable); // string(5) "siete"  
// La variable puede cambiar de tipo  
// En este caso pasa a ser de tipo "cadena"  
?>
```

Tipos de datos

Los tipos de datos simples en PHP son:

- ✓ **booleano** (boolean). Sus posibles valores son `true` y `false`. Además, cualquier número entero se considera como `true`, salvo el `0` que es `false`.
- ✓ **entero** (integer). Cualquier número sin decimales. Se pueden representar en formato decimal, octal (comenzando por un `0`), o hexadecimal (comenzando por `0x`).
- ✓ **real** (float). Cualquier número con decimales. Se pueden representar también en notación científica.
- ✓ **cadena** (string). Conjuntos de caracteres delimitados por comillas simples o dobles.
- ✓ **null**. Es un tipo de datos especial, que se usa para indicar que la variable no tiene valor.

Por ejemplo:

```
<?php  
$mi_booleano = false;  
var_dump($mi_booleano); // bool(false)  
$mi_entero = 0x2A;  
var_dump($mi_entero); // int(42)  
$mi_real = 7.3e-1;  
var_dump($mi_real); // float(0.73)  
$mi_cadena = "texto";
```

```
var_dump($mi_cadena); // string(5)"texto"  
$mi_variable = Null;  
var_dump($mi_variable); // NULL  
?>
```

Si realizas una operación con variables de distintos tipos, ambas se convierten primero a un tipo común. Por ejemplo, si sumas un entero con un real, el entero se convierte a real antes de realizar la suma:

```
<?php  
$mi_entero = 3;  
$mi_real = 2.3;  
$resultado = $mi_entero + $mi_real;  
var_dump($resultado); // float (5.3)  
// La variable $resultado es de tipo real  
?>
```

Estas conversiones de tipo, que en el ejemplo anterior se lleva a cabo de forma automática, también se pueden realizar de forma forzada:

```
<?php  
$mi_entero = 3;$mi_real = 2.3;  
$resultado = $mi_entero + (int) $mi_real;  
var_dump($resultado); // int(5)  
// La variable $mi_real se convierte a entero (valor 2) antes de sumarse.  
// La variable $resultado es de tipo entero (valor 5)  
?>
```

Por otra parte tenemos unas funciones para comprobar el tipo de dato como:

- ✓ `is_bool()`: Comprueba si una variable es de tipo `booleano`.
- ✓ `is_float()`: Comprueba si el tipo de una variable es `float`.
- ✓ `is_numeric()`: Comprueba si una variable es un número o un `string` numérico.
- ✓ `is_string()`: Comprueba si una variable es de tipo `string`.

- ✓ `is_array()`: Comprueba si una variable es un array.
- ✓ `is_object()`: Comprueba si una variable es un objeto.

Todas devuelven `true` o `false`, por ejemplo:

```
<?php  
$var=23;  
var_dump(is_int($var)); // bool(true)  
var_dump(is_int(23)); // bool(true)  
var_dump(is_int('23'));// bool(false)  
?>
```

Debes conocer

En la documentación de PHP se especifican las conversiones de tipo posibles y los resultados obtenidos con cada una:

[Conversiones de tipos posibles y los resultados obtenidos](#)

4.3.- Operadores y funciones de librería.

Como en muchos otros lenguajes, en PHP se utilizan expresiones para escribir fórmulas que se evalúan cuando se ejecuta un programa. Las expresiones se suelen utilizar para obtener y almacenar valores calculados a partir de unos datos de entrada y para escribir condiciones lógicas en sentencias de control condicionales e iterativas.



[Edson Suares \(CC BY-NC\)](#)

```
<?php  
$mi_variable = 7; // Operador de asignación =  
$a = $b + $c; // Operador de suma + y asignación =  
$res = $uno && $otro; // Operador and lógico y asignación =  
$valor++; // Operador de incremento ++ y asignación =  
$compara = 5 < 6; // Operador de comparación < y asignación =  
?>
```

Las expresiones se componen normalmente de uno o más operadores que se aplican a otros valores (o expresiones) llamados operandos.

Operadores

En PHP existe un amplio repertorio de operadores. A continuación se presentan algunos de los más utilizados por los programadores:

- ✓ **Operadores aritméticos:** cambio de signo -, suma +, resta -, multiplicación *, división / y módulo %.
- ✓ **Operadores de asignación.** Además del operador de asignación =, existen operadores con los que realizar operaciones y asignaciones en un único paso (+=, -=,...). Entre éstos se incluyen operadores de pre y post incremento y decremento, ++ y --. Estos operadores incrementan o decrementan el valor del operando al que se aplican. Si se utilizan junto a una expresión de asignación, modifican el operando antes o después de la asignación en función de su posición (antes o después) con respecto al operando. Por ejemplo:

```
<?php  
$a = 5;  
$b = ++$a;
```

```

// Antes se le suma uno a $a (pasa a tener valor 6)
// y después se asigna a $b (que también acaba con valor 6).

$a = 5;
$b = $a++;
// Antes se asigna a $b el valor de $a (5).
// y después se le suma uno a $a (pasa a tener valor 6)
?>

```

Es importante resaltar que los operadores de asignación producen un efecto secundario que permite almacenar el valor obtenido en la parte derecha del operador en la variable que aparece en la parte izquierda. Como resultado de la operación se devuelve dicho valor.

- ✓ **Operadores de comparación.** Además de los que nos podemos encontrar en otros lenguajes (`>`, `>=`, ...), en PHP tenemos dos operadores para comprobar igualdad (`==`, `===`) y tres para comprobar diferencia (`<>`, `!=` y `!==`).

Los operadores `<>` y `!=` son equivalentes. Comparan los valores de los operandos.

El operador `==` devuelve verdadero (true) sólo si los operandos son del mismo tipo y además tienen el mismo valor. El operador `!=` devuelve verdadero (true) si los valores de los operandos son distintos o bien si éstos no son del mismo tipo. Por ejemplo:

```

<?php
$x = 0;
var_dump($x == false); // bool(true) ya que false se convierte a 0
var_dump($x === false); // bool (false) debe coincidir el valor y el tipo
?>

```

- ✓ **Operadores lógicos o booleanos.** Tratan a los operandos como variables booleanas (true O false). Existen operadores para realizar un **y** lógico (operadores `and` o `&&`), **o** lógico (operadores `or` o `||`), **No** lógico (operador `!`) y **O lógico exclusivo** (operador `xor`).
- ✓ **Operadores bit a bit:** Tratan a los operandos como secuencias de bits. Pueden buscar bits que coinciden o no en ambos datos (`&`, `|`, `^` y `~`), desplazar los bits del operando hacia la izquierda (`<<`) o hacia la derecha (`>>`).
- ✓ **Operador de concatenación de cadenas.** Este operador `(.)` recibe dos cadenas y devuelve la cadena resultado de su concatenación.
- ✓ **Operador ternario:** Recibe como operandos tres expresiones, una expresión booleana que se evalúa y si el resultado es verdadero devuelve el resultado de evaluar la segunda expresión, si el resultado es falso devuelve el resultado de evaluar la tercera expresión.

```
<?php
$x = 5;
$y = 8;
$resultado = ($x > $y) ? "x mayor" : "y mayor o igual";
var_dump($resultado);
?>
```

- ✓ **Fusión de null:** Este operador recibe un número variable de parámetros y devuelve el primer operando de izquierda a derecha que exista y no sea null. Devuelve null si no hay valores distintos de null. Es muy útil para establecer valores por defecto.

```
<?php
$parametro = $valor ?? 6; // Si $valor no está establecido se usa el valor 6 ya que su valor es null
var_dump($parametro);
?>
```

- ✓ **Expresión match:** La expresión match ramifica la evaluación basada en una comprobación de identidad de un valor. Una expresión match tiene una expresión de sujeto que se compara con múltiples alternativas. Se evaluará a un valor muy parecido al de las expresiones ternarias. La comparación es una comprobación de identidad (==) en lugar de una comprobación de igualdad débil (==).

```
<?php
$a=8;

$x=10;
$y=8;
$z=5;

$resultado = match ($a) {
    $x => "Valor igual a X",
    $y => "Valor igual a Y",
    '8.0' => "Esta no es una coincidencia exacta",
    8.0 => "Valor devuelto 8",
    default => "No lo encontre",
```

```
};  
var_dump($resultado);  
>
```

Los operadores siempre quieren poder aplicarse sin generar errores que puedan entorpecer la generación de las páginas dinámicas. Hay casos en los que nuestro programa intenta aplicar operadores a valores que no concuerdan en tipo a lo que espera el operador. Por ejemplo, aplicar un operador booleano a un entero. En su afán de no generar errores los operadores fuerzan ciertos cambios de tipo (*casting*) implícitos de valores de un tipo a valores de otro tipo. Es importante que conozcamos dicho comportamiento oculto de los operadores para no llevarnos sorpresas con el comportamiento de nuestros programas.

A continuación se presentan enlaces que explican las conversiones más útiles de conocer:

- ✓ [Conversión a booleano](#)
- ✓ [Conversión a entero](#)
- ✓ [Conversión a número de punto flotante](#)
- ✓ [Conversión a cadena](#)

Debes conocer

En la documentación de [PHP](#) se explican en detalle todos los operadores disponibles en el lenguaje y la forma en que se utilizan:

[Operadores disponibles en el lenguaje](#)

Debes conocer

Es importante que te familiarices con el orden de precedencia establecido en la evaluación de expresiones con varios operadores. Si los operadores de la expresión tienen el mismo orden de precedencia se evalúan siguiendo su criterio de asociatividad.

[Orden de precedencia de los operadores en PHP](#)

Ejercicio Resuelto

Del mismo modo que se evalúan expresiones aritméticas en Matemáticas aplicando operadores de manera progresiva siguiendo su orden de precedencia también podemos reducir una expresión PHP a un valor final. Por ejemplo la expresión matemática:

$$4 + (3 \times 2)/2^2$$

$$4 + 6/2^2$$

$$4 + 6/4$$

$$4 + 1.5$$

$$5.5$$

Estudia el proceso de evaluación de las siguientes expresiones en PHP:

- ✓ $5 + 3 * 4 / (2 + 2)$
- ✓ $\$x || (\$y \&& !\$z) > \text{false}$ con $\$x=\text{false}$, $\$y=\text{true}$ y $\$z=0$

[Mostrar retroalimentación](#)

✓ $5 + 3 * 4 / (2 + 2)$

$5 + 3 * 4 / 4$ // Aplico el operador entre paréntesis +

$5 + 12 / 4$ // Los operadores * y / tienen la misma precedencia pero su asociatividad es de izquierda a derecha así que aplico el operador *

$5 + 3$ // Aplico el operador / porque tiene mayor precedencia

8 // Aplico el operador +

```
<?php
    var_dump(5+3*4/(2+2)) // int(8)
?>
```

La función `var_dump()` también puede recibir expresiones como parámetro de entrada y devuelve el valor y su tipo resultante de la evaluación de dicha expresión.

✓ $\$x || (\$y \&& !\$z) > \text{false}$ con $\$x=\text{false}$, $\$y=\text{true}$ y $\$z=0$

$\$x || (\text{true} \&& \text{!}0) > \text{false}$ // sustituyo las variables por sus valores y ejecuto lo que hay dentro del paréntesis

$\$x || (\text{true} \&& \text{true}) > \text{false}$ // Aplico el operador ! sobre el 0, pero antes realizo una conversión del 0 a booleano y obtengo false, por lo tanto !0 es true

$\$x || \text{true} > \text{false}$ // Aplico el operador && por estar dentro de un paréntesis

$\$x || \text{true}$ // Aplico el operador de comparación > porque tiene mayor precedencia, pero antes realizo la conversión de true y false a entero y obtengo 1 y 0 respectivamente.

$\text{false} || \text{true}$ // Sustituyo el valor del operando $\$x$ para aplicar el operador ||

true // Aplico el operador ||

true

```
<?php
    \$x = false;
    \$y = true;
    \$z = 0;
```

```
?> var_dump($x || ($y && !$z) > false); // bool(true)
```

Funciones de librería

El lenguaje PHP pone a disposición del programador una serie de librerías que facilitan la tarea de programar. Algunas de estas librerías contienen un repertorio de funciones que transforman datos en otros valores resultado. En esta sección vamos a presentar algunas de las funciones más interesantes de tipo matemático y de tipo cadena de caracteres.

Funciones matemáticas

A continuación, se presenta un conjunto de funciones numéricas de uso frecuente:

- ✓ `round(int|float $num, int $precision = 0, int $mode = PHP_ROUND_HALF_UP): float` : redondea un número al entero más próximo.
- ✓ `floor(int|float $num): float` : redondea un número al entero inferior (es decir, devuelve la parte entera).
- ✓ `ceil(int|float $num): float` : redondea un número al entero superior.
- ✓ `pow(mixed $num, mixed $exponent): int|float|object` : calcula la potencia de un número.
- ✓ `sqrt(float $num): float` : calcula la raíz cuadrada de un número.
- ✓ `max(mixed $value, mixed ...$values): mixed` : devuelve el máximo de los valores de entrada.
- ✓ `min(mixed $value, mixed ...$values): mixed` : devuelve el mínimo de los valores de entrada.
- ✓ `mt_rand(int $min, int $max): int` : genera un numero aleatorio entre los valores min y max.

También es útil conocer cómo formatear un número para que se imprima de la manera adecuada. Para ello se utiliza la siguiente función:

```
number_format(float $num, int $decimals = 0, ?string $decimal_separator = ".", ?string $thousands_separator = "," ): string
```

La función requiere dos o cuatro argumentos:

- el primer argumento es el número a formatear.
- el segundo argumento es el número de decimales a mostrar (el número se redondea o trunca dependiendo de la longitud del número, compárese el segundo y el cuarto ejemplo de los ejemplos anteriores).

- el tercer argumento es el carácter a utilizar como separador de la parte entera de la decimal.
- el cuarto argumento es el carácter a utilizar como separador de miles.

La función devuelve el número formateado. Si sólo se utilizan dos argumentos, se utiliza el punto como separador de parte entera y decimal y la coma como separador de miles (notación inglesa).

```
<?php
    var_dump(round(12.6)); // float(13)
    var_dump(round(12.3)); // float(12)
    var_dump(round(-12.6)); // float(-13)
    var_dump(round(-12.3)); // float(-12)
    var_dump(round(12.6574, 2)); // float(12.66)
    var_dump(floor(12.6)); // float(12)
    var_dump(floor(12.3)); // float(12)
    var_dump(ceil(12.6)); // float(13)
    var_dump(ceil(-12.6)); // float(-12)
    var_dump(pow(2, 3)); // int(8)
    var_dump(sqrt(25)); // float(5)
    var_dump(max(22, 40, 28.1, 17.7)); // int(40)
    var_dump(min(12, 4, 26.1, 18.7)); // int(4)
    var_dump(mt_rand(1, 6)); // int(4)
    var_dump(number_format(34534556.34544, 2)); //string(13) "34,534,556.35"
    var_dump(number_format(123222343, 0, ",", ".")); // string(11) "123.222.343"
    var_dump(number_format(345345345347.789, 2, ",", ".")); // string(18) "345.345.345.347,79"
?>
```

Funciones de cadenas de caracteres

Del mismo modo, PHP proporciona una gran número de funciones de cadenas de caracteres muy útiles para el programador. A continuación se presentan las funciones de uso más frecuente:

- ✓ **ltrim(string \$string, string \$characters = "\n\r\t\x00")**: string : Elimina los caracteres en blanco que se encuentren al principio de la cadena. Lo mismo hace **rtrim** por la derecha.
- ✓ **str_contains(string \$haystack, string \$needle)**: bool : Determina si una cadena contiene otra cadena dada.
- ✓ **str_repeat(string \$string, int \$times)**: string : Repite una cadena dada.

- ✓ `str_replace(array|string $search, array|string $replace, string|array $subject, int &$count = null): string|array` : Reemplaza una cadena por otra dada.
- ✓ `strcmp(string $string1, string $string2): int` : Compara dos cadenas de entrada.
- ✓ `strpos(string $haystack, string $needle, int $offset = 0): int|false` : Encuentra la posición de la primera ocurrencia de una cadena en otra dada.
- ✓ `strlen(string $string): int` : Devuelve la longitud de una cadena.
- ✓ `strtolower(string $string): string` : Convierte una cadena a minúsculas. `strupper` convierte la cadena a mayúsculas.
- ✓ `substr(string $string, int $offset, ?int $length = null): string` : Devuelve una parte de la cadena

A continuación se muestran ejemplos de uso de estas funciones.

```
<?php  
    var_dump(ltrim("    HOLA    ")); // string(7) "HOLA "  
    var_dump(str_contains("HOLA amigo", "HOLA")); // bool(true)  
    var_dump(str_repeat("HOLA", 3)); // string(12) "HOLAHOLAHOLA"  
    var_dump(str_replace("HOLA", "ADIOS", "HOLA amigo")); // string(11) "ADIOS amigo"  
    var_dump(strcmp("HOLA", "HOLA")); // int(0)  
    var_dump(strpos("HOLA amigo", "ami")); // int(5)  
    var_dump(strlen("HOLA amigo")); // int(10)  
    var_dump(strtolower("HOLA")); // string(4) "hola"  
    var_dump(substr("HOLA amigo", 1, 3)); // string(4) "OLA"  
?  
>
```

Debes conocer

Es importante que vayas adquiriendo familiaridad con el máximo de funciones de librería numéricas y de cadena de caracteres para conseguir programas mas robustos en menos tiempo.

No dejes de visitar las páginas del manual PHP donde se listan las [funciones matemáticas](#) y las [funciones de cadenas de caracteres](#), no hace falta que las aprendas de memoria pero sí deberías ser capaz de reconocerlas y saber que existe una función en la librería que puede ser útil para el problema que estás resolviendo.

4.4.- Evaluación de expresiones.

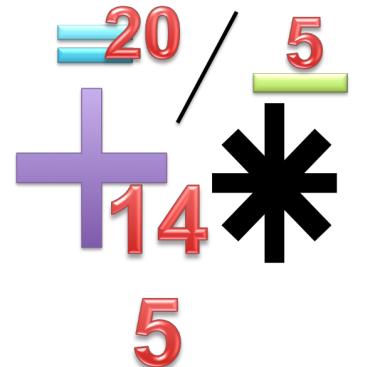
Las expresiones con los elementos de programación más importantes del lenguaje PHP. Las expresiones permiten crear fórmulas que, cuando se evalúan, transforman los datos de entrada en un resultado que puede ser mostrado en la página web dinámica o utilizado por las sentencias de control de nuestro algoritmo.

Se pueden crear expresiones con distintos tipos de elementos pero lo más habitual es que contengan literales, variables y operadores de distintos tipos.

Cuando el interprete PHP se encuentra con una expresión su misión consistirá en evaluarla y sustituir dicha expresión por el valor obtenido. Es importante que entendamos las reglas que se aplican en el proceso de evaluación de las expresiones en PHP para evitar errores al escribirlas y mejorar nuestra comprensión del código cuando lo vemos escrito.

Las expresiones pueden ser muy sencillas, formadas por un único literal, una constante o una variable, en ese caso el valor resultante sería el valor representado por el literal, el valor asociado a la constante o el valor almacenado en la variable en ese momento respectivamente.

Si se trata de expresiones con operadores habrá que establecer un orden de evaluación de dichos operadores basado en la tabla de precedencia y asociatividad de los mismos tal como se muestra en el [manual de PHP](#). El concepto de asociatividad entra en juego cuando estamos evaluando una expresión con operandos de la misma precedencia y tenemos que decidir en qué dirección se evalúan, es decir, evaluamos de izquierda a derecha o de derecha a izquierda.



```
<?php  
var_dump(!(5 - 3 * 2 / 2 > 7 % 3)) // bool(false) La multiplicación se evalúa antes de la división por asociatividad ?>
```

La asignación en PHP se realiza por medio del operador asignación que conlleva el efecto secundario de almacenar en la variable de la izquierda el valor obtenido en la derecha. En este caso el operador tiene asociatividad *derecha*, lo que significa que si tengo varias asignaciones en la expresión se ejecutan de derecha a izquierda.

```
<?php  
var_dump($a = $b = $c = 10); // int(10)  
var_dump($a); // int(10)
```

```
<?php  
    var_dump($b); // int(10)  
    var_dump($c); // int(10)  
?>
```

El proceso de evaluación también puede hacer uso de un recurso llamado cortocircuito (short-circuit) por el que un operador no necesite conocer el valor de todos sus operandos para devolver un resultado. Por ejemplo, los operadores lógicos `&&` y `||` ya pueden devolver un resultado sin conocer el valor de los dos operandos. En el caso de `&&` devuelve falso si el primer operando es falso y en el caso de `||` devuelve true si el primer operando es true. En el ejemplo vemos que el segundo operando (1/0) no se evalúa

```
<?php  
    var_dump(false and 1/0); // bool(false)  
    var_dump(true || 1/0); // bool(true)  
    var_dump(1/0 || true); // error division por cero  
?>
```

El operador ternario (`:=`) funciona de manera parecida ya que dependiendo del resultado de la evaluación de la expresión booleana se evaluará solamente la segunda o la tercera expresión.

Este operador tiene una variante que consiste en omitir el tercer operando de tal manera que si el resultado de evaluar la expresión resulta un valor que corresponde a verdadero el operador devuelve dicho valor.

```
<?php  
    $x = 10;  
    $y = 5;  
    $path = false;  
    var_dump($x > $y ? "X es mayor que Y" : 1/0); // string(16) "X es mayor que Y"  
    var_dump($path ?: '/'); // string(1) "/" Si $path tuviera un valor distinto de false devolvería dicho valor. Muy útil para asignar valor  
?>
```



El operador de fusión de nulo (`??`) el comportamiento es parecido al operador ternario y solo se evalúan operandos hasta llegar al primero no nulo empezando por la izquierda.

```
<?php  
    $path = null;  
    var_dump($path ?? '/'); // string(1) "/" En este caso se comprueba si el valor es nulo en vez de falso  
?>
```

La expresión `match` es otro tipo de operador que puede sustituir a la sentencia de control `switch` ya que funciona de manera similar. Dado un valor de entrada se proporcionan un número de expresiones asociadas a valores que se comparan con el primer valor. Si alguna de las comparaciones tiene éxito se devuelve el valor resultante de la evaluación de la expresión asociada. Es importante resaltar que la comparación de valores se realiza con el operador `(==)`.

```
<?php  
    $nota = 15;  
    var_dump(  
        match ($nota) {  
            1, 2 => "Muy deficiente",  
            3, 4 => "Insuficiente",  
            5 => "Suficiente",  
            6 => "Bien",  
            7, 8 => "Notable",  
            9, 10 => "Sobresaliente",  
            default => "Nota incorrecta",  
        }  
    ); // string(10) "Suficiente". Se puede añadir una cláusula default por si el valor no encuentra dentro de los valores listados  
?>
```

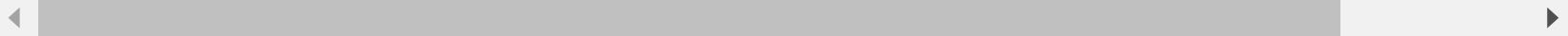
Ejercicio Resuelto

Se trata de escribir una expresión que devuelva un valor booleano indicando si un año es bisiesto o no. Recuerda que la condición para que un año sea bisiesto es que sea divisible por 4 pero no por 100 aunque si por 400.

Ocultar retroalimentación

La solución puede ser:

```
<?php  
    $anyo=1345;  
    var_dump((($anyo%4==0 && $anyo%100!=0) || $anyo%400==0); // Puedes modificar el valor de la variable $anyo para compro  
?>
```



Se trata de escribir una expresión que dado un valor de \$dia, \$mes y \$anyo se compruebe que la fecha es válida. No se pueden usar funciones de fecha para realizar la comprobación. La idea es que vayas analizando el valor de la variable \$mes y en base a ese valor establezcas una expresión booleana que compruebe si el valor de la variable \$dia está en el rango correcto. No dejes de usar la expresión booleana de año bisiesto para tratar ese caso.

Ejercicio Resuelto

Se trata de programar una expresión que convierta un valor numérico de una nota en su nota textual siguiendo la siguiente correspondencia:

1, 2 => Muy deficiente; 3, 4 => Insuficiente; 5 => Suficiente; 6 => Bien; 7, 8 => Notable; 9, 10 => Sobresaliente

Recuerda que se pide que programes una expresión por lo que no puedes usar sentencias de control estructurado.

Te recomendamos que uses la el operador de expresión `match`.

Ocultar retroalimentación

La solución propuesta al ejercicio es:

```
<?php
$nota = 5;
var_dump(
    match ($nota) {
        1, 2 => "Muy deficiente",
        3, 4 => "Insuficiente",
        5 => "Suficiente",
        6 => "Bien",
        7, 8 => "Notable",
        9, 10 => "Sobresaliente",
        default => "Nota incorrecta",
    }
); // string(10) "Suficiente". Se puede añadir una clausula default por si el valor no encuentra dentro de los val
?>
```

