

**DESARROLLO DE APLICACIONES WEB AVANZADO**

**LABORATORIO N° 01**

**Aplicación web básica y Configuración del Entorno de node.js**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Alumno(s):*** | ***Juan Manuel Pino Cáceres*** | | | | | | ***Nota*** |  | |
| ***Grupo:*** | ***A*** | | ***Ciclo:VI*** | | | | | | |
| ***Criterio de Evaluación*** | | ***Excelente (4pts)*** | | ***Bueno (3pts)*** | ***Requiere mejora (2pts)*** | ***No acept. (0pts)*** | | | ***Puntaje Logrado*** |
| Instala el servicio de node.js en Windows | |  | |  |  |  | | |  |
| Instala el servicio de Apache en Windows | |  | |  |  |  | | |  |
| Ejecuta los formularios propuestos de BackEnd y FrontEnd | |  | |  |  |  | | |  |
| Logra solucionar lo propuesto en los formularios JavaScript | |  | |  |  |  | | |  |
| Es puntual y redacta el informe adecuadamente | |  | |  |  |  | | |  |

**Laboratorio 1: Aplicación web desde el cliente**

**y desde el servidor**

**Objetivos:**

Al finalizar el laboratorio el estudiante será capaz de:

* Identificar las principales características del XAMPP
* Instalar Apache
* Identificar las diferencias de trabajar formularios en el lado del servidor y del lado del cliente.

**Seguridad:**

* Ubicar maletines y/o mochilas en el gabinete del aula de Laboratorio.
* No ingresar con líquidos, ni comida al aula de Laboratorio.
* Al culminar la sesión de laboratorio apagar correctamente la computadora y la pantalla, y ordenar las sillas utilizadas.

**Equipos y Materiales:**

* Una computadora con:
* Windows 7 o superior
* VMware Workstation 10+ o VMware Player 7+
* Conexión a la red del laboratorio
* Máquinas virtuales:
* Windows 7 Pro 64bits Español - Plantilla
* Instalador de XAMPP

**Procedimiento:**

**Lab Setup**

1. **Creación del equipo virtual**
   1. Encender el equipo
   2. Acceder empleando la cuenta de **usuario**: Tecsup, **contraseña**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. Iniciar el Software VMWare.
   4. Abrir la plantilla ubicada en:

**E:\Equipos virtuales\Windows 7 Pro 64bits Español - Plantilla**

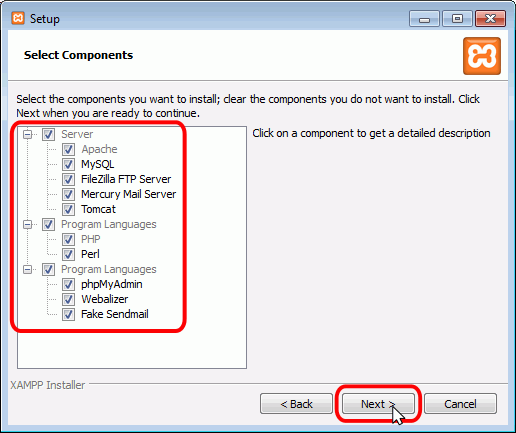
* 1. Crear un clon de la maquina anterior con los siguientes datos:
  2. Nombre del clon: **C15-BDAV**
  3. Ubicación del clon: **D:\C15-BDAV**
  4. Cerrar la plantilla
  5. Iniciar el equipo virtual **C15-BDAV**
  6. Identifíquese con la cuenta de usuario: **Redes**. Contraseña: **RCDTecsup2**

**Instalación de XAMPP**

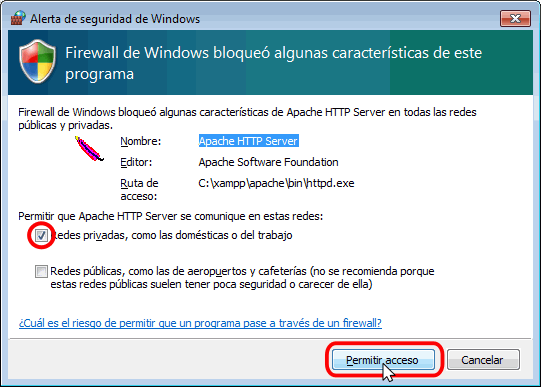
1. **Proceso de instalación de XAMPP**
   1. Solicite al instructor, el archivo de instalación de XAMPP

**xampp-win32-1.8.1-VC9-installer.exe**

* 1. Copie el archivo instalador al escritorio del equipo virtual
  2. Doble clic en el instalador para iniciar el proceso de instalación.
  3. Al llegar a la ventana “**Select Components**”, veremos los distintos componentes instalados durante el proceso. **Anótelos y defínalos en su informe sobre este laboratorio.**



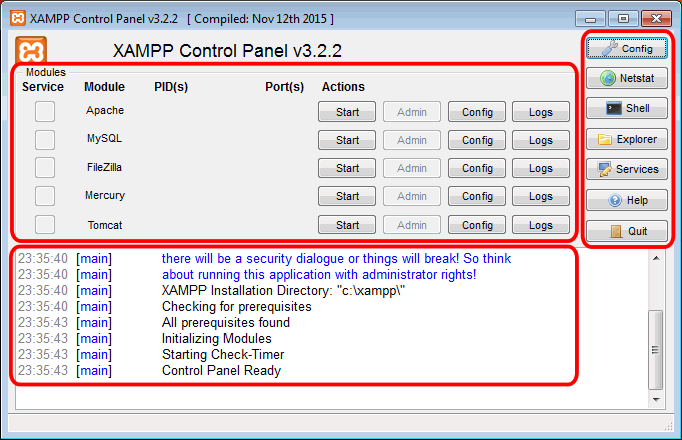
* 1. Durante la instalación, si en el ordenador no se había instalado Apache anteriormente, se mostrará un aviso del cortafuego de Windows para autorizar a Apache para comunicarse en las redes domésticas o de trabajo, lo que debemos permitir haciendo clic en el botón **"Permitir acceso".**



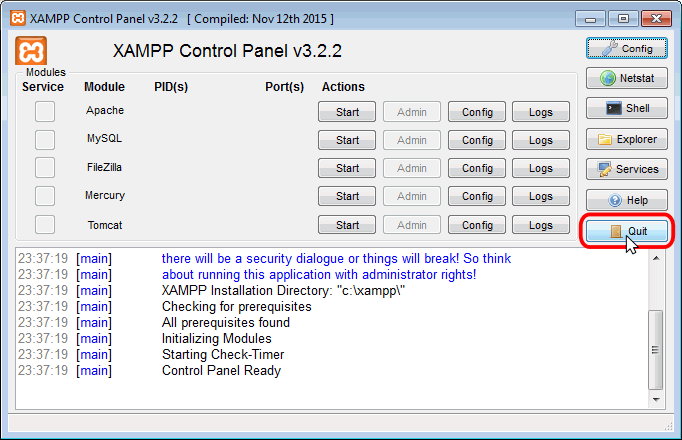
* 1. Proceda con la instalación del programa
  2. Espere que finalice el proceso de **instalación**. Clic en el botón **Finish**

**XAMPP**

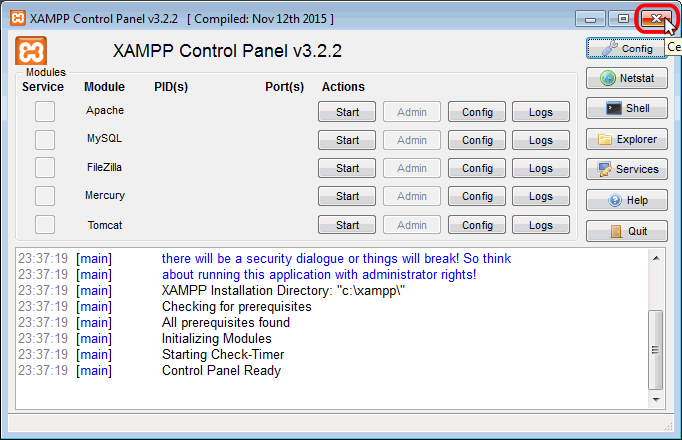
1. Identificación del Panel de Control de XAMPP
   1. El panel de control de XAMPP se divide en tres zonas:
   2. la zona de módulos, que indica para cada uno de los módulos de XAMPP: si está instalado como servicio, su nombre, el identificador de proceso, el puerto utilizado e incluye unos botones para iniciar y detener los procesos, administrarlos, editar los archivos de configuración y abrir los archivos de registro de actividad.
   3. la zona de notificación, en la que XAMPP informa del éxito o fracaso de las acciones realizadas
   4. la zona de utilidades, para acceder rápidamente



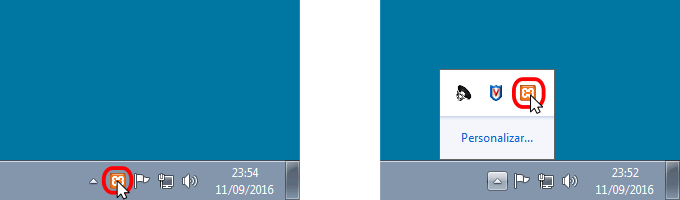
* 1. Para cerrar el panel de control de XAMPP hay que hacer clic en el botón Quit (al cerrar el panel de control no se detienen los servidores):



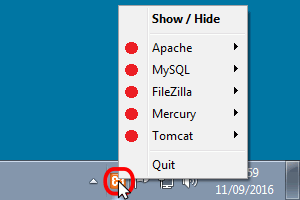
* 1. El botón Cerrar en forma de aspa no cierra realmente el panel de control, sólo lo minimiza:



* 1. Si se ha minimizado el panel de control de XAMPP, se puede volver a mostrar haciendo doble clic en el icono de XAMPP del área de notificación.



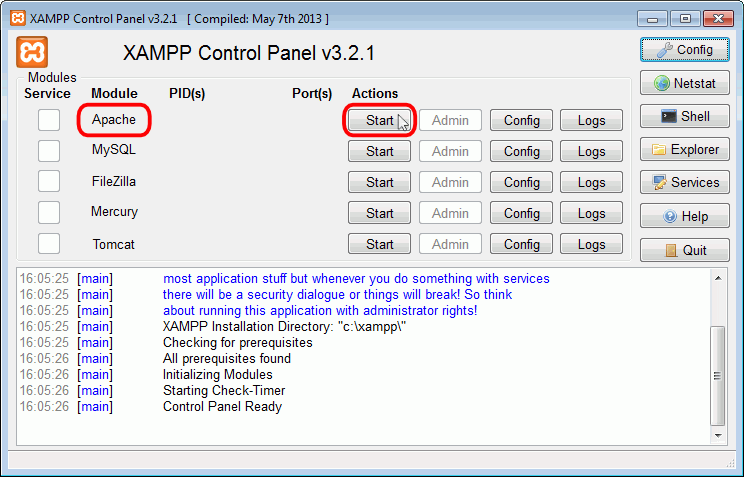
* 1. Haciendo clic derecho en el icono de XAMPP del área de notificación se muestra un menú que permite mostrar u ocultar el panel de control, arrancar o detener servidores o cerrar el panel de control.



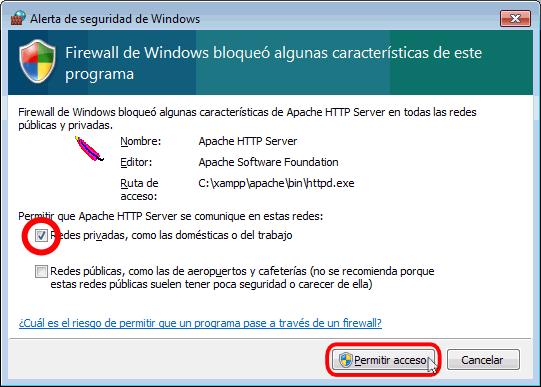
* 1. Se pueden abrir varios paneles de control simultáneamente y cualquiera de ellos puede iniciar o detener los servidores, pero no es aconsejable hacerlo ya que puede dar lugar a confusiones (por ejemplo, al detener un servidor desde un panel de control los otros paneles de control interpretan la detención como un fallo inesperado y muestran un mensaje de error).

Cuando se pone en marcha por primera vez cualquiera de los servidores que instala XAMPP, el cortafuego de Windows pide al usuario confirmación de la autorización.

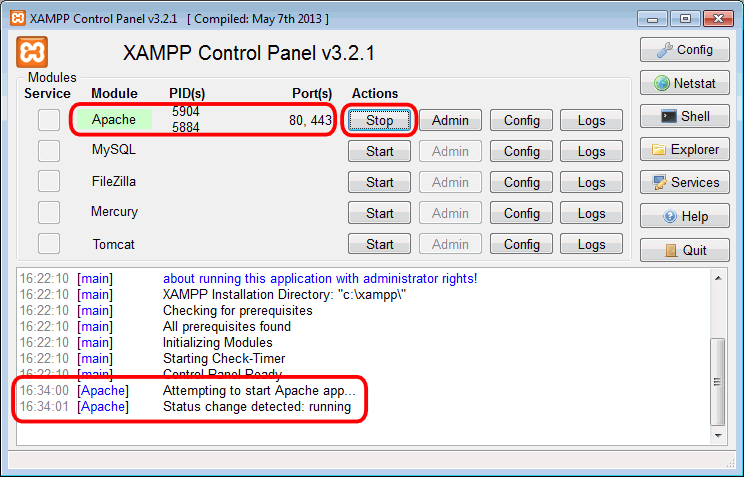
Por ejemplo, la primera vez que se pone en marcha Apache mediante el botón Start correspondiente...



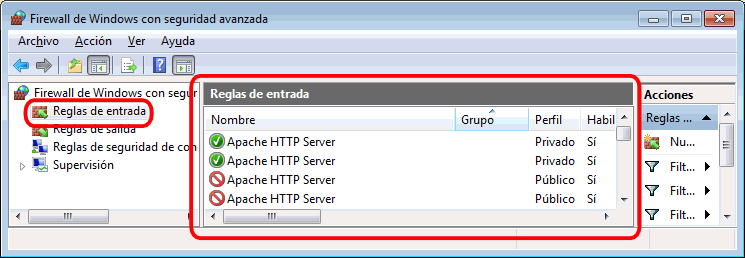
* 1. ... como Apache abre puertos en el ordenador (por primera vez), el cortafuegos de Windows pide al usuario confirmación. Para poder utilizarlo hace falta al menos autorizar el acceso en redes privadas:



* 1. Si el arranque de Apache tiene éxito, el panel de control mostrará el nombre del módulo con fondo verde, su identificador de proceso, los puertos abiertos (http y https), el botón "Start" se convertirá en el botón "Stop" y en la zona de notificación se verá el resultado de las operaciones realizadas.

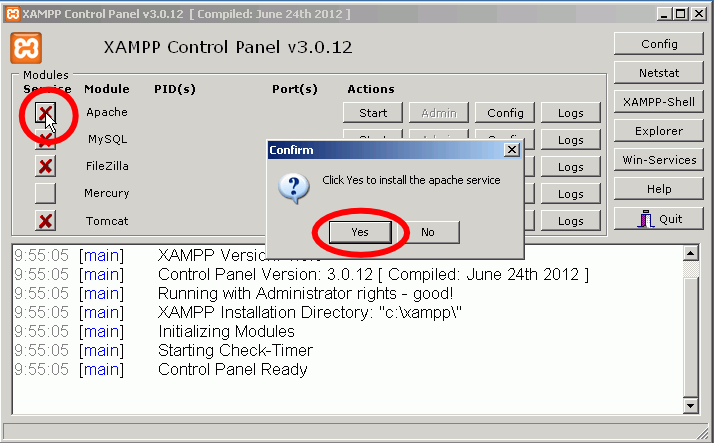


* 1. Si se abre el programa "Firewall de Windows con seguridad avanzada", en el apartado de Reglas de entrada pueden verse las nuevas reglas añadidas.

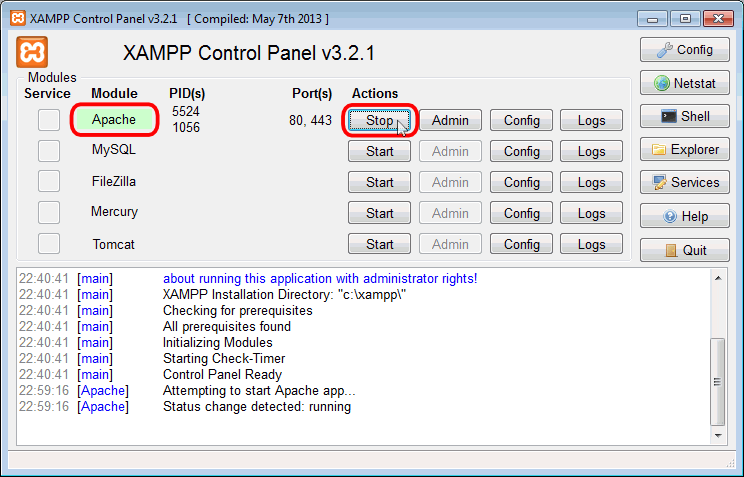


* 1. Instalar Apache como servicio de Windows

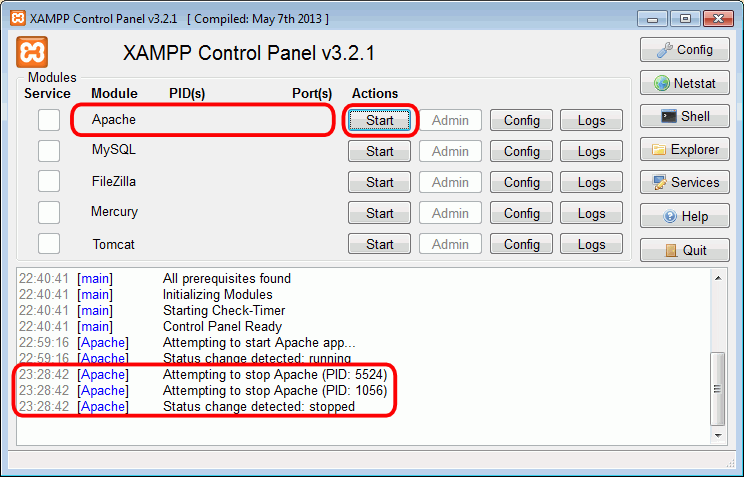
Si queremos que un servidor arranque como servicio, es decir, que se ponga en marcha cada vez que arrancamos el ordenador, hay que marcar la casilla Service correspondiente.



* 1. Si el servicio se instala correctamente, se indica en el panel inferior. Los servicios instalados se indican con una marca verde en la columna Service

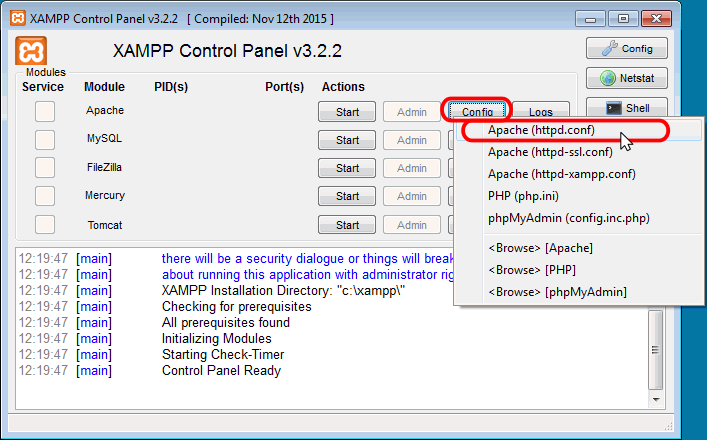


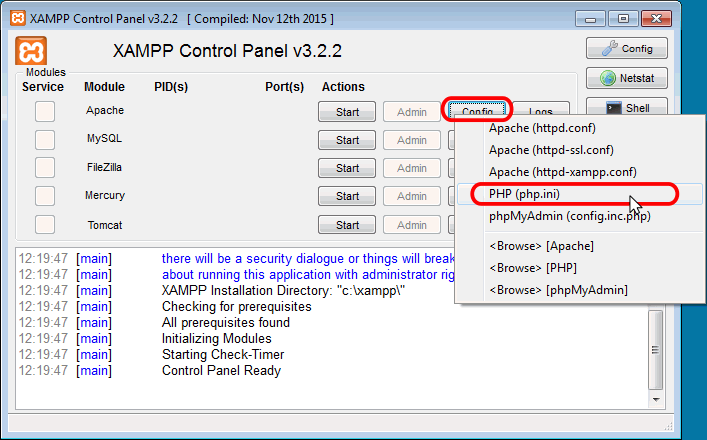
* 1. Al reiniciar el ordenador, el panel de control de XAMPP indica los servicios arrancados:



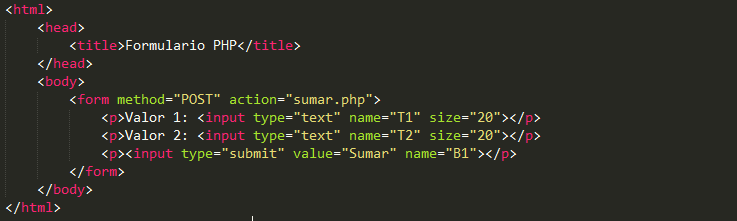
* 1. Editar archivos de configuración de Apache o PHP

Los dos archivos principales de configuración son los archivos httpd.conf (Apache) y php.ini (PHP). Para editarlos se puede utilizar el panel de control de XAMPP, que los abre directamente en el bloc de notas. Para ello hay que hacer clic en el botón "Config" correspondiente a Apache y hacer clic en el archivo que se quiere editar.

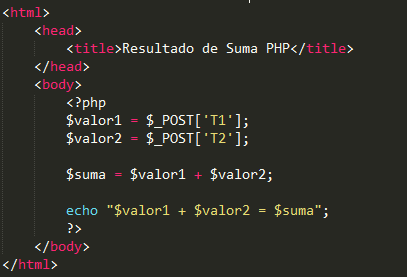




1. **Formulario desde BackEnd**
   1. Cree el archivo **formulario.php** en la carpeta C:\xampp\htdocs con el siguiente contenido:



* 1. Cree el archivo **sumar.php** en la misma carpeta con el siguiente contenido:



* 1. Visualice lo redactado accediendo a la dirección [**http://localhost/formulario.php**](http://localhost/formulario.php). Adjunte capturas del proceso.

Después de haber visto lo que se puede obtener, intentemos reducir la cantidad de archivos.

Cree el archivo **formulario2.php** con el siguiente contenido:

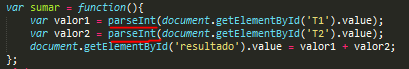


* 1. Visualice lo redactado accediendo a la dirección [**http://localhost/formulario2.php**](http://localhost/formulario2.php)**.**

1. **Formulario desde FrontEnd**
   1. Crearemos el archivo **formulario3.html** con el siguiente contenido:

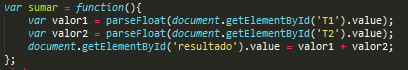


* 1. Visualice lo redactado accediendo a la dirección [**http://localhost/formulario3.**](http://localhost/formulario.php)**html**. Adjunte capturas del proceso.
  2. Intente sumar los siguientes valores, anote el resultado y explique:
  + 1 + 1 = 2 => Al sumar dos números enteros no existe problema.
  + 1000 + 152 = 1152 => Al igual que el caso anterior no se presenta problema.
  + 5 + prueba = NaN =>No puede sumar un número con un cadena de caracteres.
  + 2.5 + 6.8 =8 => Toma la parte entera de los números y la suma, la parte decimal es ignorada.
  1. Vemos que la última suma no es como deseamos. Explique por qué:



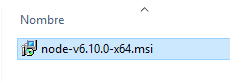
|  |
| --- |
| Los datos están parseados a enteros de manera que para obtener la suma de decimales debemos parsear a variables flotantes. |

* 1. Adjunte una captura del código con la solución a nuestro problema actual.

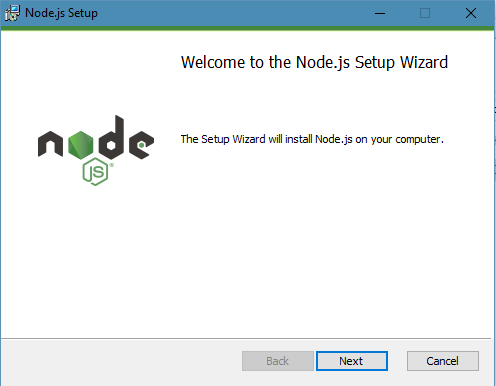


**Instalación de node.js**

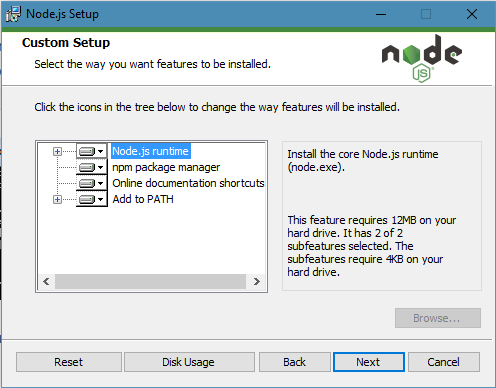
1. Obtención del instalador
   1. El instalador de node.js se puede obtener de la página oficial: <https://nodejs.org/en/download/>
   2. Veremos varias opciones, aparte de las plataformas de sistema operativo como son:
      * **LTS** (recomendada para la mayoría de usuarios): es la versión de Node.js con Long Term Support (LTS), es decir a la que se le da soporte a largo plazo. Esta versión puede no tener disponibles las últimas tecnologías que todavía no se consideran estables.
      * **Current**: esta es la versión más reciente de Node.js e incluye todas las funcionalidades, incluso aquellas más novedosas y que no se consideran estables.
   3. Por norma general, **seleccionar la versión LTS es la opción recomendada y estable.**
   4. Esta sección es **informativa**, para ahorrar tiempo, el instructor le facilitará el archivo de instalación del node.js, aunque como podrá usted comprobar, el instalador no es tan pesado (aproximadamente unos 12MB)
2. **Proceso de instalación de node.js**
   1. Solicite al instructor, el archivo de instalación de node.js



* 1. Copie el archivo instalador al escritorio del equipo virtual
  2. Doble clic en el instalador para iniciar el proceso de instalación. Haga click en Next



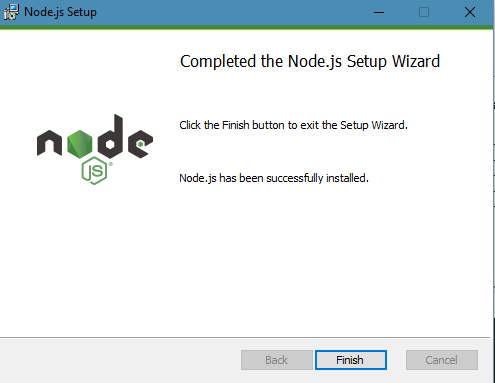
* 1. No deberá tener problemas en el instalador ya que se seleccionarán las opciones por defecto. Llegará usted a la siguiente pantalla:



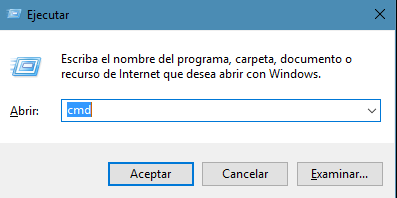
Haga click en cada una de las características a instalar e indique que es lo que realizarán al ser instaladas:

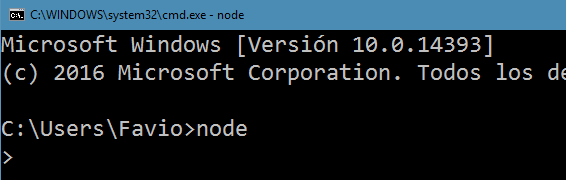
|  |
| --- |
| * Node.js: Es el núcleo del servidor Node.js. * Performance counters: Soporte para los contadores de desempeño específicos de Node.js. * Event tracing: Soporte para el seguimiento de eventos generados por Node.js. * Npm package manager: El manejador de paquetes para Node.js. * Online documentation shortcuts: Agrega enlaces al menú vinculadas a la documentación de Node.js. * Add to path: Agrega Node.js, npm, y modules instalados globalmente por npm a la ruta variable del entorno. * Node.js and npm: Agrega Node y npm agregados la ruta variable del entorno. * Npm modules: Agrega módulos instalados a la ruta variable del entorno. |

* 1. Siga con el proceso de instalación con las opciones por defecto.



1. **Verificación de la instalación**
   1. Abra una consola de comandos y ejecutemos el comando **node**





* 1. Uso de JavaScript en la consola del Shell de MongoDB
     1. Definir variables en mongoDB

**> var a = 20, b = 30, c = 40;**

* + 1. Realizar operaciones aritméticas

**> var suma = a + b;**

**> var resta = c - b;**

**> var producto = a \* b;**

**> var cociente = a / b;**

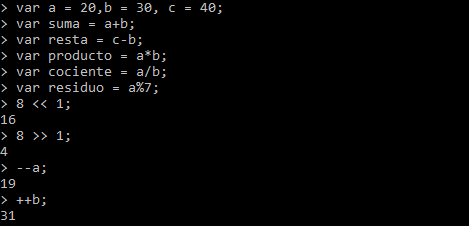
**> var residuo = a % 7;**

**> 8 << 1;**

**> 8 >> 1;**

**> --a;**

**> ++b;**



* + 1. Visualizar valor de variables

**> console.log(a, b, c, suma, resta, producto, cociente, residuo);**



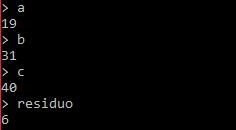
* + 1. Visualizar el valor de una única variable en forma simple

**> a**

**> b**

**> c**

**> residuo**



* + 1. Funciones de la librería javascript Math

> Math.min(0, 150, 30, 20, -8, -200);

> Math.max(0, 150, 30, 20, -8, -200);

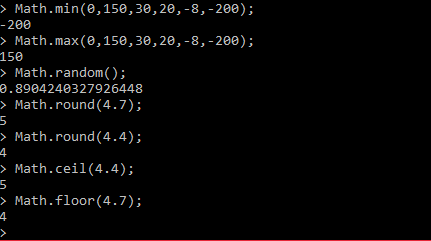
> Math.random();

> Math.round(4.7);

> Math.round(4.4);

> Math.ceil(4.4);

> Math.floor(4.7);



* + 1. Implemente algunos ejemplos donde haga uso de las siguientes funciones o constantes:

**Math Object Methods**

**Method** **Description**

abs(x) Returns the absolute value of x

acos(x) Returns the arccosine of x, in radians

asin(x) Returns the arcsine of x, in radians

atan(x) Returns the arctangent of x as a numeric value between -PI/2 and

PI/2 radians

atan2(y,x) Returns the arctangent of the quotient of its arguments

ceil(x) Returns x, rounded upwards to the nearest integer

cos(x) Returns the cosine of x (x is in radians)

exp(x) Returns the value of Ex

floor(x) Returns x, rounded downwards to the nearest integer

log(x) Returns the natural logarithm (base E) of x

max(x,y,z,...,n) Returns the number with the highest value

min(x,y,z,...,n) Returns the number with the lowest value

pow(x,y) Returns the value of x to the power of y

random() Returns a random number between 0 and 1

round(x) Rounds x to the nearest integer

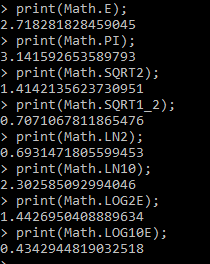
sin(x) Returns the sine of x (x is in radians)

sqrt(x) Returns the square root of x

tan(x) Returns the tangent of an angle

**Math Constants**

Math.E          // returns Euler's number  
Math.PI         // returns PI  
Math.SQRT2      // returns the square root of 2  
Math.SQRT1\_2    // returns the square root of 1/2  
Math.LN2        // returns the natural logarithm of 2  
Math.LN10       // returns the natural logarithm of 10  
Math.LOG2E      // returns base 2 logarithm of E  
Math.LOG10E     // returns base 10 logarithm of E



* + 1. Funciones de cadena JavaScript en MongoDB

**String Methods**

**Method** **Description**

charAt() Returns the character at the specified index (position)

charCodeAt() Returns the Unicode of the character at the specified index

endsWith() Checks whether a string ends with specified string/characters

fromCharCode() Converts Unicode values to characters

includes() Checks whether a string contains the specified string/characters

indexOf() Returns the position of the first found occurrence of a specified value in a string

lastIndexOf() Returns the position of the last found occurrence of a specified value in a string

match() Searches a string for a match against a regular expression, and returns the matches

repeat() Returns a new string with a specified number of copies of an existing string

replace() Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced

search() Searches a string for a specified value, or regular expression, and returns the position of the match

slice() Extracts a part of a string and returns a new string

split() Splits a string into an array of substrings

startsWith() Checks whether a string begins with specified characters

substr() Extracts the characters from a string, beginning at a specified start position, and through the specified number of character

substring() Extracts the characters from a string, between two specified indices

toLowerCase() Converts a string to lowercase letters

toUpperCase() Converts a string to uppercase letters

trim() Removes whitespace from both ends of a string

**Ejemplos de uso:**

**> var cad = ‘Tecsup Arequipa’;**

**> cad.length**

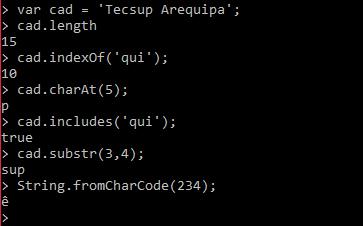
**> cad.indexOf( ‘qui’ );**

**> cad.charAt(5);**

**> cad.includes( ‘qui’ );**

**> cad.substr( 3, 4 );**

**> String.fromCharCode( 234 );**



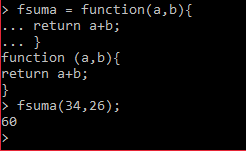
* + 1. Creación de funciones JavaScript en MongoDB

**> fsuma = function(a,b) {**

**... return a+b;**

**... }**

**> fsuma(34,26);**

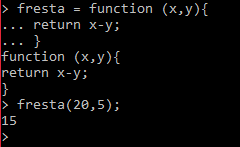


**> fresta = function (x,y) {**

**... return x-y;**

**... }**

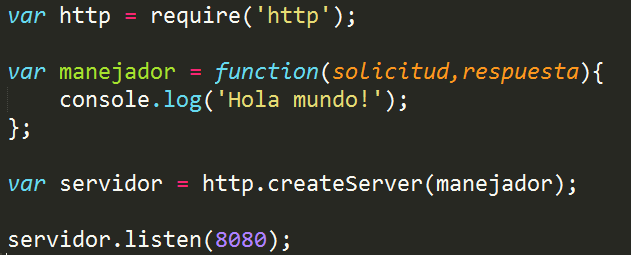
**> fresta(20,5);**



* + 1. Para salir de la consola de node, utilizamos el comando:



1. **Hola mundo con node.js**
   1. Cree el archivo **hola.js** en la unidad **C:** con el siguiente contenido:



* 1. En una ventana de comandos sitúese en la unidad **C:\** y escriba el siguiente comando:



* 1. Abra un navegador de Internet, y escriba la siguiente URL:



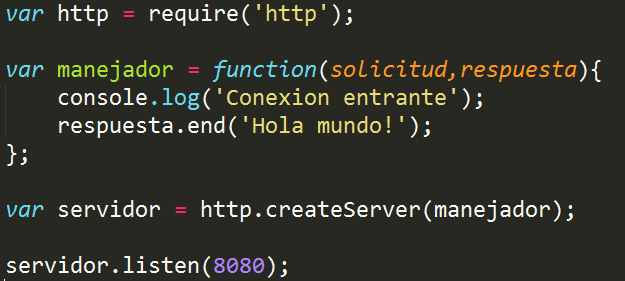
* 1. Verificamos en la consola el resultado.



* 1. Anote sus conclusiones de lo que se acaba de realizar. Para finalizar la actividad del servidor, debe presionar **Ctrl + C**

|  |
| --- |
| En Node.js utilizamos console.log para mostrar un mensaje en la ventana de comandos y especificamos el puerto con la función listen del servidor. |

* 1. A continuación, cree el archivo **hola.js**



* 1. Una vez más, ejecute el comando:



* 1. Anote sus observaciones y mencione en que se diferencia de la versión anterior del archivo.

|  |
| --- |
| Para mostrar un mensaje en el navegador web, se puede utilizar la función end del response. |

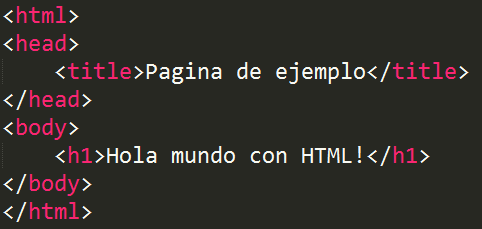
* 1. Ahora creemos el archivo **bucle.js** con el siguiente contenido:



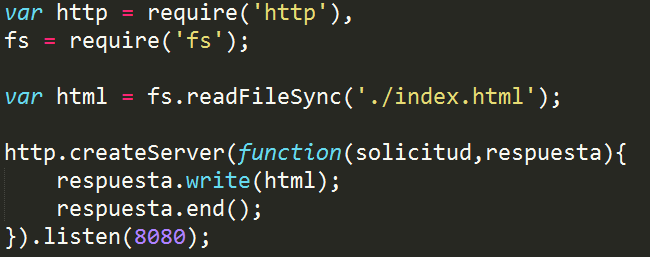
* 1. Ejecute node.js apuntando al nuevo archivo. Anote sus observaciones de acuerdo a lo que sucede en este caso.

|  |
| --- |
| Ser puede crear un bucle infinito en un while true, enviando mensajes al navegador con la función write del response. Así mismo observamos que node no deja de funcionar. |

1. **Leer archivo de servidor**
   1. Creamos el archivo **index.html** con el siguiente contenido:



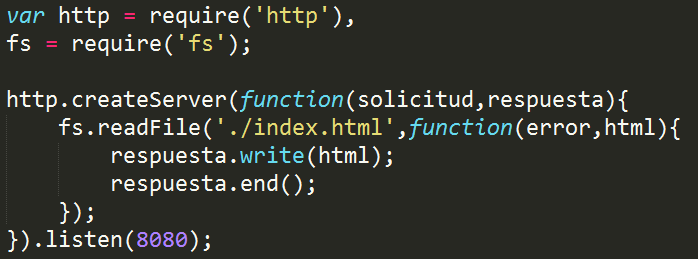
* 1. Creamos el archivo **web.js** con el siguiente contenido



* 1. Ejecútelo y anote sus observaciones.

|  |
| --- |
| Utilizando readFileSync, obtenemos un buffer del archivo en la ubicación especificada y este se envia en la función write para mostrarlo en el navegador. |

* 1. A continuación, modificaremos e archivo para que luzca de la siguiente manera.



* 1. Busque en Internet la diferencia entre los comandos **readFileSync** y **readFile** y explique cuál es la diferencia entre ambos.

|  |
| --- |
| ReadFile lee el archivo completo de forma asíncrona y funciona con 2 argumentos, un error y la data del archivo, por otro lado ReadFileSync lee el archivo de forma síncrona, retorna el contenido del a dirección especificada. |

1. **Finalizar la sesión**
   1. Apagar el equipo virtual
   2. Apagar el equipo

**Tarea:**

1. Desarrolle una página web para la empresa ACME (solicite el rubro de la empresa a su docente), con sus páginas básicas: Inicio, Nosotros, Nuestros Servicios, Catálogo de Clientes y Contáctenos. Esta última página deberá tener un formulario de contacto. Toda la web debe utilizar **node.js**

**Conclusiones:**

Indicar las conclusiones que llegó después de los temas tratados de manera práctica en este laboratorio.

|  |
| --- |
| * En este laboratorio identificamos las partes del módulo de control de XAMPP para administrar los servicios de apache, mySQL, etc. * Desarrollamos formularios gestionados con PHP, Javascript y Node.js e identificamos que, en primer lugar PHP requiere de cambiar o recargar la página, a diferencia de Javacript que puede modificar el contenido a partir de un evento como un click. * Un script de Javascript puede ser incluido como <script> en <head> y utilizar la función en un elemento del <body> como atributo, por ejemplo en onClick. * Node.js por si mismo a diferencia de Tomcat u otro entorno de servidor, posee muy pocas cosas, las mayoría de funcionalidades requieren de paquetes que deben ser instalados, esto con el objetivo de solo tener los elementos necesarios en el servidor. * Los paquetes de Node.js se obtienen por medio del Npm manager que está incluido en la instalación. * En el servidor en Node, importamos http, y el manejador funciona con los parámetros request y response, o solicitud y respuesta, cada una con sus propiedades. En la variable de servidor, especificamos el puerto al cual escuchar. * Probamos que en Node se puede crear un bucle infinito sin que el servidor se detenga o colapse. * Finalmente observamos las formas de leer un archivo desde Node, utilizando readFileSync y ReadFile de forma síncrona o asíncrona. |