Clase 01 - 27/03/2023

Profesores: Franco Lippi y Pablo Musella.

Esta materia va estar enfocada al aspecto del backend de la programación y se va a enfocar en las APIs (*del inglés, application programming interface, en español, interfaz de programación de aplicaciones*) y en la creación de Microservicios, dígase, las operaciones de lado del servidor.

## PHP.

PHP : Lanzado en 1995 por Danes-Canadiense Rasmus Lerdorf. Originalmente sus siglas significaban Personal Home Page, hoy significa el inicialismo​ recursivo PHP: Hypertext Preprocessor.

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo Web y que puede ser incrustado en HTML.

**Características**:

* Popular: porque un gran número de páginas y portales Web están creadas con PHP (\*).
* Código abierto: significa que es de uso libre y gratuito para todos los programadores que quieran usarlo.
* Incrustado en HTML: significa que en un mismo archivo vamos a poder combinar código PHP con código HTML.

**Características técnicas:**

* Es multiplataforma Windows, Mac, Linux
* El código PHP es invisible al navegador.
* Posee capacidad de conexión a BBDD.
* Puede expandir su capacidad por medio de plug-ins.

## Cliente - Servidor

### Servidor:

En el contexto de las aplicaciones web, un servidor se refiere a un programa informático que proporciona servicios de red a través de internet a través del protocolo **HTTP (*Hypertext Transfer Protocol*).** Un servidor web es esencialmente un software que se ejecuta en una computadora y está diseñado para gestionar la solicitud y respuesta de las páginas web a los clientes que acceden a través de internet.

El servidor web es responsable de almacenar los archivos necesarios para servir las páginas web, como imágenes, archivos HTML, scripts y hojas de estilo. Cuando un usuario solicita una página web a través de un navegador web, el servidor web recibe la solicitud y busca los archivos necesarios para servir la página solicitada. Luego, el servidor web construye la página y la envía al navegador del usuario en forma de respuesta HTTP.

Existen varios servidores web populares, como Apache, Nginx, IIS, entre otros, que son utilizados para alojar sitios web y proporcionar servicios web a los usuarios. Estos servidores web pueden ser configurados para cumplir con los requerimientos específicos de los sitios web, tales como la gestión de múltiples sitios web en una misma máquina, el control de acceso y la autenticación, la protección contra ataques cibernéticos, la generación de estadísticas y registros de acceso, entre otros.

En resumen, un servidor web es esencial para el funcionamiento de las aplicaciones web, ya que es el encargado de proveer los recursos necesarios para servir las páginas web y entregar la información al usuario final a través de la red.

### Cliente:

Un cliente se refiere a un programa o software que se ejecuta en la computadora del usuario final y está diseñado para solicitar y recibir recursos y servicios desde un servidor web a través de internet.

El cliente web, también conocido como navegador web, es una aplicación que permite a los usuarios acceder y visualizar páginas web a través de la red. El navegador web actúa como una interfaz de usuario para permitir que los usuarios interactúen con los sitios web y los servicios en línea.

Cuando un usuario introduce una URL en el navegador web, este envía una solicitud al servidor web correspondiente, solicitando los archivos necesarios para construir y presentar la página web solicitada. Estos archivos pueden incluir archivos HTML, imágenes, scripts y hojas de estilo.

Una vez que el servidor web recibe la solicitud del cliente, procesa la solicitud y envía la respuesta de vuelta al cliente. El navegador web entonces interpreta la respuesta y presenta la página web solicitada al usuario final.

Además de la visualización de páginas web, los clientes web también pueden realizar otras funciones como enviar formularios, cargar y descargar archivos, interactuar con aplicaciones web, y mucho más.

Existen varios clientes web populares, tales como Google Chrome, Mozilla Firefox, Microsoft Edge, entre otros, que son utilizados por los usuarios finales para acceder a los servicios y recursos en línea.

En resumen, un cliente web es una aplicación que se ejecuta en la computadora del usuario final y que permite la interacción con los servicios y recursos en línea a través de internet, solicitando y recibiendo información y recursos desde el servidor web correspondiente.

### Interacción Cliente Servidor.

La interacción cliente-servidor se refiere al proceso de comunicación entre el cliente (navegador web) y el servidor web, que permite a los usuarios acceder y consumir servicios y recursos en línea.

El proceso de interacción cliente-servidor en una aplicación web sigue el modelo de solicitud-respuesta. El cliente envía una solicitud al servidor web, y el servidor responde a la solicitud proporcionando los recursos o servicios solicitados.

La interacción cliente-servidor comienza cuando el usuario ingresa una URL en el navegador web y presiona "Enter". El navegador web entonces envía una solicitud al servidor web correspondiente, solicitando los archivos necesarios para construir y presentar la página web solicitada. Estos archivos pueden incluir archivos HTML, imágenes, scripts y hojas de estilo.

Una vez que el servidor web recibe la solicitud del cliente, procesa la solicitud y envía la respuesta de vuelta al cliente. La respuesta puede incluir la página web solicitada, junto con otros recursos adicionales, como imágenes y scripts.

El cliente web interpreta la respuesta del servidor y presenta la página web solicitada al usuario final. El usuario puede interactuar con la página web y enviar nuevas solicitudes al servidor web, solicitando recursos adicionales o servicios en línea.

La interacción cliente-servidor también puede incluir la autenticación del usuario y la gestión de sesiones. Cuando un usuario se autentica en un sitio web, el servidor web puede crear una sesión para el usuario, lo que le permite acceder a recursos y servicios adicionales durante un período de tiempo limitado.

En resumen, la interacción cliente-servidor en una aplicación web implica la comunicación entre el cliente (navegador web) y el servidor web, donde el cliente solicita recursos y servicios en línea y el servidor proporciona la respuesta correspondiente. Este proceso se basa en el modelo de solicitud-respuesta y permite a los usuarios acceder y consumir servicios y recursos en línea de manera eficiente y efectiva.

## Estructura Básica de PHP

• Todo código va entre <?php y ?>.

• Con ʻechoʼ (\*) se muestra el mensaje en el navegador.

• Al igual que en C o C#, toda instrucción se debe finalizar con punto y coma (;)

• La extensión del archivo fuente debe ser .php

Ejemplo:

<?php

echo “HOLA MUNDO””;

?>

Siempre hay que verificar de terminar las instrucciones con punto y coma, ya que en el caso que hagamos la prueba dentro del servidor, esto nos puede generar errores.

### Sintaxis en PHP

### Las variables en PHP.

PHP no es un lenguaje de programación tipado. En PHP, las variables no están obligadas a tener un tipo específico, lo que significa que una variable puede contener un valor de cualquier tipo y se puede cambiar su tipo en tiempo de ejecución.

Por ejemplo, en PHP se puede asignar una cadena de texto a una variable y luego cambiar su tipo a un número o un objeto en tiempo de ejecución sin generar errores. Esta característica se conoce como tipado dinámico.

A diferencia de los lenguajes de programación tipados, como Java o C++, donde las variables deben declararse con un tipo específico y no pueden cambiar su tipo durante la ejecución del programa, el tipado dinámico en PHP ofrece más flexibilidad al programador. Sin embargo, también puede resultar en errores difíciles de detectar si no se maneja adecuadamente.

Es importante tener en cuenta que aunque PHP no es un lenguaje de programación tipado, aún se deben seguir ciertas convenciones de nombres y tipos de datos para mantener la claridad y la legibilidad del código. Además, se pueden usar herramientas adicionales, como PHPDoc, para documentar explícitamente el tipo de datos que se espera que tenga una variable.

Comentarios en PHP:

// De una sola línea.

# De una sola línea alternativa.

/\* Comentarios de más

de una línea \*/

### Los tipos en PHP.

* PHP soporta ocho tipos primitivos.
  + **Cuatro tipos escalares:**
    - Boolean
    - Integer
    - Float
    - String
  + **Dos tipos compuestos**
    - Array
    - Object

Nota: En php todos son arrays.

* + **Dos especiales**
    - Resource //Dato especial para el manejo de archivos y peticiones http.
    - NULL

### Las variables en PHP.

La definición de variables en php se empieza con $ antes del nombre de la variable. Si no se le asigna el símbolo adelante, el interprete no va a interpretarlo como una variable.

Por otro lado al igual que en JavaScript.

### Conversión de tipos en PHP.

Las conversiones las realiza automáticamente PHP dependiendo del contenido de las variables.

• Sin embargo, si se desea explícitamente realizar una

conversión de tipos:

* (int), (integer) -> convierte a entero
* (bool), (boolean) -> convierte a booleano
* (float), (double), (real) -> convierte a decimal
* (string) -> convierte a cadena de caracteres
* (array) -> convierte a array
* (object) -> convierte a objeto
* ●(unset) -> convierte a nulo