**Apuntes tema 1**

**PHP**

Puede generar contenido HTML, el servidor procesa el código PHP y envía el resultado HTML al cliente. Primero se procesa el PHP y después el HTML. Se puede crear contenido dinámico basado en datos del servidor.

Para construir largas cadenas con muchas variables se recomienda usar comillas dobles.

Para escribir código PHP se debe envolver en “<?php ‘código’ ?>”. También se puede usar <? ‘codigo’ ?> pero requiere configuración y no se recomienda.

Es un lenguaje débilmente tipado, y las variables son dinámicas(Pueden cambiar en plena ejecución)

**Arrays**

* Arrays indexados ( $colores = [datos] ): Se pueden indexar por numero de posición empiezan desde 0.
* Arrays asociativos ($persona = (‘nombre’ = Carlos, ‘apellido’ = ‘Salazar’)): Pares clave valor.
* Arrays multidimensionales: Son arrays dentro de arrays.

Algunas de los métodos de los arrays son:

* count() - Cuenta elementos
* array\_push() – Añade elementos al final.
* array\_merge() – Combina arrays.
* in\_array() – Busca un valor.
* array\_keys – Obtiene las claves de un array asociativo

**Bucles y funciones**

* For: Cuando sabemos exactamente cuantas veces queremos que se repita.
* While: Se repite mientras la condición sea verdadera.
* Foreach: Recorrer arrays, elegante y eficiente.

Usando ‘break’ salimos del bucle y con ‘continue’ pasamos a la siguiente iteración.

Las funciones son útiles si queremos reutilizar bloques de código se le puede añadir parámetros. Para usar variables de ámbito global dentro de una función podemos usar ‘global $var’ para que podamos verla dentro.

**Arquitectura MVC**

Este patrón de software tiene soluciones comprobadas, es reutilizable y tiene buena documentación.

Patrón Singleton: Asegura que en una aplicación solo puede generar una instancia de objeto.

**Arquitectura Física (Multitier)**

Maquinas conectadas en red ejecutando conjuntamente una aplicación.

**Arquitectura Lógica (Multilayer)**

Capas de software que colaboran mediante interfaces definidas.

* Capa ‘Usuario’.
* Capa ‘Intermedias’.
* Capa ‘Hardware’.

Ventajas de la multicapa.

* Desarrollo en paralelo, velocidad de desarrollo y robustez.
* Mantenimiento sencillo, flexibilidad y seguridad.
* Tiene mejor escalabilidad y rendimiento ya que podemos trabajar en cada capa.
* Mas fácil hacer el control de calidad y las pruebas.

**3 capas**

Modelo: Logica de negocio y acceso a datos

Vista: Interfaz de usuario y presentación.

Controlador: Control del flujo y coordinación.

**Evolución del código:**

1. Codigo Monolitico: Un solo bloque, inmanejable.
2. Controlador + Vista: Separación lógica de datos y presentación
3. Modelo+Vista+Controlador: 3 Capas especializadas
4. Con abstracción de datos: Gestor de BD independiente.

**Implementación practica:**

* index.php = Punto de entrada al controlador
* controller.php = Invoca los modelos y vistas en orden, captura peticiones y dirige.
* clase.php = Acceso a datos y lógica de negocio, algoritmos, restricciones…
* vista.php = El HTML, CSS y Javascript

