



## SESIÓN 1

Vamos a comenzar a desarrollar nuestro conocimiento en otra manera de programar, manteniendo el sentido de la POO.

Durante esta primera sesión el desarrollo del contenido será teórica, este aprendizaje es la base para 6to año.

Cada duda que surja deberá ser mencionada en el espacio establecido por el docente, es importante hacer la aclaratoria del mismo para su mayor entendimiento.

En esta sesión se presentarán los siguientes puntos:

### Framework:

- ¿Qué es un Framework?
- Modelo Vista Controlador (MVC)
- ¿Para qué sirven los Framework?
- Características
- Ventajas
- Tipos de Framework
- **Reto de la semana.**

## FRAMEWORK

Es la unión o combinación de 2 palabras, las cuales le dan vida a esta forma de trabajo, **Marco (FRAME)**, y **Trabajo (WORK)**, donde vendría



definido por ser un “*Marco de Trabajo*”, cuando se habla de este término a nivel de programación, es que tiene una estructura definida de código, es decir, es una plantilla que ya viene definida pero incompleta para integrar lo necesario para el proyecto.

### Ejemplo:

Una plantilla de HTML que ya está construida y lo único que debe hacer el desarrollador es trabajar bajo esos parámetros y elementos, usando todos los beneficios y herramientas que posee para finalizar su proyecto.

Esta herramienta es construida para *desarrolladores* hechas por otros **desarrolladores**. Facilita el trabajo, construcción, y elaboración de un proyecto, evitando el desarrollo de soluciones desde cero, sino que se usan las herramientas que el *Framework* brinda, tomando en cuenta que al usar esto, la cantidad de líneas de código se reducen y el mismo se vuelve más eficiente y recursivo.

En el mercado nacional e internacional, en las grandes y pequeñas empresas en su mayoría trabajan con el uso de *FRAMEWORK*, además están diseñados en su mayoría por la arquitectura conocida como MVC (Modelo Vista Controlador), este tipo de estructura divide el trabajo de una forma considerable, permitiendo la unión de diversas personas y áreas al equipo de proyecto.



Antes de seguir avanzando debemos conocer cada una de estas partes, para poder entender con claridad el manejo de los *Framework*.



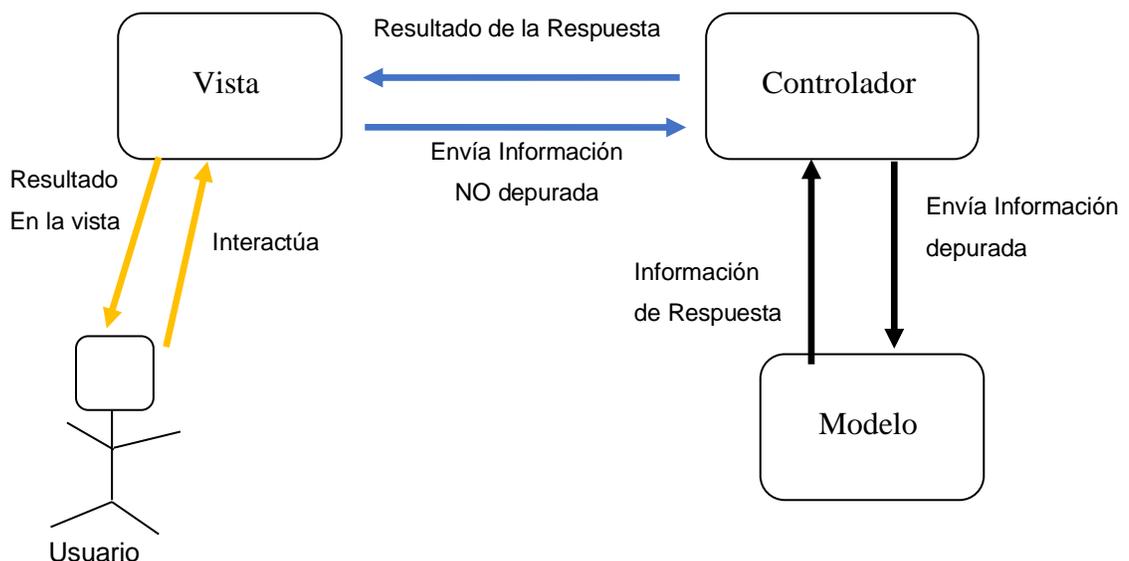
**Modelo:** Es donde se desarrollan los procesos que conectan con la base de datos, aquí es donde van desarrolladas las llamadas **API's** (más adelante veremos este concepto). El proceso interno de estos mayormente es desarrollado en PHP y lenguajes de consultas de Base de datos.

En esta permite la interacción con los registros de la Base de Datos, que incluyen la inserción, consultas, actualización y eliminación dentro de ella.

**Vista:** Aquí es donde los diseñadores gráficos conocidos como desarrolladores de **Frontend**, es toda la parte estética y de maquetación para la visualización del usuario. Es el producto que ve el consumidor, esto es desarrollado en HTML, CSS y JS, este último es actualmente uno de los más usado para este desarrollo.

**Controlador:** Elaborada por desarrolladores de códigos llamados desarrolladores de **Backend**, es decir, esta parte es la encargada de recibir la información de la *Vista*, depurarla, validarla y realizar cualquier proceso interno que permita su envío satisfactorio hacia el *Modelo*. El *Controlador* es el puente de conexión entre la *Vista* y el *Modelo*. Toda la información es enviada y recibida por el controlador, para que se encargue de redireccionarla a su lugar de destino. Este es desarrollado en su mayoría por JS y PHP.

### Funcionamiento del MVC.





Para entender bien este diagrama podemos reflejar el siguiente ejemplo:

Un programa de televisión donde el televisor es la Vista, el canal el Controlador y el Modelo los diversos programas.

El canal compra diversos programas de televisión donde deben definir y aprobar el contenido para su transmisión, lo que hace que el televisor obtenga la señal televisiva para ver el programa. ***Aquí se apreció la interacción Modelo, Controlador, Vista.***

Cuando el usuario programa un canal que desea ver. ***Aquí la interacción sería Vista, Controlador, Modelo.***

### ¿Para qué sirve?

Permite organizar el trabajo y poder minimizar el riesgo de errores. Gracias a la estructura del *Framework* se puede aumentar la productividad del trabajo, así como:

- Permite ahorrar el tiempo en desarrollar el espacio de trabajo, ya que el framework viene prediseñado.
- Facilita la colaboración de varios programadores, haciendo estandarizar los códigos.
- Tienen herramientas que permite trabajar con él, estas son bibliotecas, plugin o utensilios para su desarrollo.
- Permite crear funciones para resolver diversos problemas o para olvidar la repetición de código innecesario.
- La estructura de los directorios de trabajo es sencilla, permite que el desarrollador se intuitivo al momento de revisar cada uno.



- Permite ayudar al programador a trabajar eficientemente sin tener que desarrollar toda la comunicación del sistema desde cero.

### Características.

La mayoría de los *Framework* poseen características similares en su entorno de desarrollo, entre las cuales podemos destacar:

- **Autenticación:** Permite el acceso controlado de los usuarios al sistema, donde con un nombre de usuario y contraseña pueda ser manipulado y verificado.
- **Acceso a datos:** La integración de las herramientas e interfaces necesarias para poder integrar las herramientas de BD, XML, entre otras.
- **Abstracción de URL y Sesiones:** El manejo directo de las URL ya no se hace, porque el *Framework* se encarga de este proceso, es decir, mayor seguridad.
- **Internacionalización:** Posee herramientas que permite integrar el uso de diversos idiomas.
- **Controladores:** Son archivos que permite controlar o gestionar eventos y peticiones para la introducción y extracción de datos, también para el acceso a las páginas. Son fácilmente adaptables para cada uno de los proyectos.



## Ventajas

- Compatibilidad de Lenguajes.
- Integración de múltiples dispositivos móviles y de escritorios.
- Desarrollo mas eficiente, ya que cuenta con componentes necesarios incluidos.
- Integrar diversas áreas de la empresa en un proyecto.
- Reutilización de código.
- Similitud entre los Framework
- Estructura de desarrollo parecidas
- Capacidad de trabajar y tener conocimientos con diversos Framework
- Conexión con diversos gestores de BD SQL y NO SQL.

## Tipos de Framework

Para el desarrollo con *Framework* hay que tener en cuenta el tipo, ya que cada uno de ellos tienen características distintas donde escoger el indicado es sumamente necesario, aunque el 60% de los programadores escogen uno o dos para toda su vida de desarrollo, no todos son eficientes en cuanto al trabajo del proyecto.

Es importante mencionar cuales son los tipos de *Framework*: están los de **Frontend** y **Backend**. Estos son los dos (2) grandes hitos de la programación.



**Reto de la semana.**

Investigar los siguientes, donde pueda realizar una infografía donde presente toda esta información. Tiempo de entrega máxima **sábado 25/4**.

- Que es un framework de Frontend y Backend
- Cuales son los framework incluidos en cada uno
- Cuales son compatibles entre ellos.
- Por qué la comunidad de programadores los prefiere.