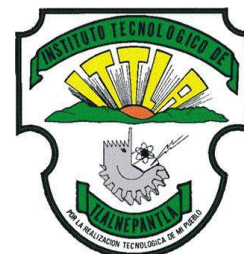




Tecnológico nacional de México
Instituto tecnológico de Tlalnepantla



Ingeniería en tecnologías de la información y comunicaciones

Manejo de Frameworks

Materia: Programación Web

Profesor(a): JOSÉ ANTONIO GALLARDO GODINEZ

Grupo: T63

Alumno:

EDGAR ADALID CISNEROS VILLA

Numero de control: 18251057



Que es un frameworks

Un framework es un marco o esquema de trabajo generalmente utilizado por programadores para realizar el desarrollo de software. Utilizar un framework permite agilizar los procesos de desarrollo ya que evita tener que escribir código de forma repetitiva, asegura unas buenas prácticas y la consistencia del código.

Un framework es por tanto un conjunto de herramientas y módulos que pueden ser reutilizados para varios proyectos. Uno de los frameworks más conocidos y utilizados es el .NET Framework de Microsoft para webs.

Entre las ventajas de utilizar un framework para el desarrollo de software distinguimos:

- El programador ahorra tiempo ya que dispone ya del esqueleto sobre el que desarrollar una aplicación.
- Facilita los desarrollos colaborativos, al dejar definidos unos estándares de programación.
- Al estar ampliamente extendido, es más fácil encontrar herramientas, módulos e información para utilizarlo.
- Proporciona mayor seguridad, al tener gran parte de las potenciales vulnerabilidades resueltas.
- Normalmente existe una comunidad detrás, un conjunto de desarrolladores que pueden ayudar a responder consultas.

Ejemplos de Frameworks

- .NET

El ecosistema Microsoft .NET tiene el concepto de frameworks. Dichos frameworks definen el API que puedes usar para enfocarte en una plataforma en concreto.

- 1- Tenemos el framework .NET Standard que busca solucionar el problema de compartición del código entre plataformas para los desarrolladores de .NET al aportar todas las APIs que necesitas en todos los entornos: aplicaciones de escritorio, apps móviles, juegos y servicios cloud.

- 2- Por otro lado, está .NET Core que es un framework más pequeño para aplicaciones de servidor, de consola, aplicaciones web y como entorno en tiempo de ejecución principal del cual crear otros sistemas. Está concebido para crear aplicaciones multiplataforma con un buen rendimiento en cloud.
- 3- Esta el “clásico” .NET Framework para aplicaciones y sistemas Windows.

.NET Framework 4 es compatible con las aplicaciones que se crearon con .NET Framework versiones 1.1, 2.0, 3.0 y 3.5. En otras palabras, las aplicaciones y componentes creados con versiones anteriores de .NET Framework funcionarán en .NET Framework 4.

Descarga:

[Download Microsoft .NET Framework 4.5 from Official Microsoft Download Center](#)

- Symphony

Symfony es un entorno de trabajo estandarizado (framework PHP) que se utiliza para el desarrollo de aplicaciones web y es de los más utilizados en el entorno de desarrolladores de apps. En otras palabras, es una herramienta para desarrolladores para crear aplicaciones en PHP.

Una de las principales ventajas de Symfony es que posee una licencia MIT, o lo que es lo mismo que una licencia de software libre permisivo, que se puede utilizar dentro del software del propietario y al no poseer Copyright permite su modificación.

Para poder trabajar con esta herramienta es necesario disponer de un servidor web, una consola de comandos del sistema operativo y cualquier versión de PHP5 o PHP7 si se quiere utilizar las últimas versiones.

Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Descarga:

[Download Symfony Framework and Components](#)

- Zend Framework

La colección de paquetes PHP profesionales integrados en este framework representa una herramienta de gran ayuda que ha llegado a impresionar a los desarrolladores más experimentados. Las funciones básicas se integran rápidamente y sin necesidad de grandes esfuerzos de programación, lo que permite que el usuario se pueda concentrar en la implementación de las funcionalidades de su propia aplicación.

La instalación estándar de Zend Frameworks contiene más de 60 paquetes y, con la ayuda de Composer y de una línea de código simple, es fácilmente ejecutable. Si el paquete de instalación no está incluido en el gestor de paquetes de tu sistema operativo, puedes recurrir a su página web.

- 1- Autenticación / Administración de usuarios
- 2- Acceso a la base de datos (CRUD)
- 3- Sistema de correo electrónico
- 4- Localización
- 5- Indexación de búsqueda
- 6- Almacenamiento en caché

Descarga:

[Home - Zend Framework](#)

- Laravel

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el «código espagueti» y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP.

- 1- Sistema de ruteo, también RESTful

- 2- Blade, Motor de plantillas
- 3- Peticiones Fluent
- 4- Eloquent ORM
- 5- Basado en Composer
- 6- Soporte para el caché
- 7- Soporte para MVC
- 8- Usa componentes de Symfony

Descarga:

[Laravel - The PHP Framework For Web Artisans](#)

- Django

Django es un framework de aplicaciones web gratuito y de código abierto (open source) escrito en Python. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente. Te ofrece una manera de manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para tu sitio web, formularios, una forma de subir archivos, etc.

En la función de view (vista) se hacen todas las cosas interesantes. La vista puede comprobar si tienes permiso para hacerlo, actualizar la descripción de tu trabajo y devolver un mensaje: "¡Hecho!". Luego la vista genera una respuesta y Django puede enviarla al navegador del usuario.

Descarga:

[The web framework for perfectionists with deadlines | Django \(djangoproject.com\)](#)

- Ruby on Rails

Ruby es un lenguaje de programación surgido en 1993 de la mano de Yukihiro Matsumoto. Fue creado por la complejidad de otros lenguajes como Java y C++. Es flexible y preciso, lo cual da la posibilidad de organizar de una forma más eficiente el código escrito. La sintaxis, por su parte, está fundada en bloques de código, por lo que se evitan los errores al cambiar de línea de instrucción. Sin embargo, el lenguaje Ruby no saltó a la fama de

forma internacional hasta 15 años después, en 2008, cinco años después de ser combinado con un framework denominado Rails. Esta combinación dio como resultado el marco de trabajo conocido como Ruby on Rails (RoR), creado por David Heinemeier Hasson en 2003 y muy popular entre los desarrolladores en los últimos años por su sencillez y su fundamento en las metodologías ágiles.

Descarga:

[Ruby on Rails | A web-application framework that includes everything needed to create database-backed web applications according to the Model-View-Controller \(MVC\) pattern.](#)

Frameworks para JavaScript

- angular.js
- d3
- nodeJS
- jquery
- html5-boilerplate (H5BP)
- reveal.js
- impress.js
- three.js
- backbone
- meteor
- todomvc
- express
- Bootstrap
- jqueryUI
- mean.js

Diferencia entre librería y frameworks

Los frameworks como su nombre lo indica nos crea un entorno de trabajo donde podremos desarrollar aplicaciones, por otro lado las librerías solo nos permiten solucionar un problema en concreto.

- Arreglos y métodos

Un arreglo es una colección o agrupación de elementos en una misma variable, cada uno de ellos ubicado por la posición que ocupa en el array. En Javascript, se pueden definir de varias formas

Ejemplo:

```
// Forma tradicional
const array = new Array("a", "b", "c");

// Mediante literales (preferida)
const array = ["a", "b", "c"]; // Array con 3 elementos
const empty = []; // Array vacío (0 elementos)
const mixto = ["a", 5, true]; // Array mixto (string, number, boolean)
```

Un método es una función la cual es propiedad de un Objeto. Existen dos tipos de métodos: Métodos de Instancia los cuales son tareas integradas realizadas por la instancia de un objeto, y los Métodos Estáticos que son tareas que pueden ser llamadas directamente en el constructor de un objeto.

- Clases

Las clases de javascript, introducidas en ECMAScript 2015, son una mejora sintáctica sobre la herencia basada en prototipos de JavaScript. La sintaxis de las clases no introduce un nuevo modelo de herencia orientada a objetos en JavaScript. Las clases de JavaScript proveen una sintaxis mucho más clara y simple para crear objetos y lidiar con la herencia.

Ejemplo:

```
class Rectangulo {
  constructor(alto, ancho) {
    this.alto = alto;
    this.ancho = ancho;
  }
}
```

- Funciones

Las funciones son uno de los bloques de construcción fundamentales en JavaScript. Una función en JavaScript es similar a un procedimiento un conjunto de instrucciones que realiza una tarea o calcula un valor, pero para que un procedimiento califique como función, debe tomar alguna entrada y devolver una salida donde hay alguna relación obvia entre la entrada y la salida. Para usar una función, debes definirla en algún lugar del ámbito desde el que deseas llamarla.

- Control (objetos)

JavaScript está diseñado en un paradigma simple basado en objetos. Un objeto es una colección de propiedades, y una propiedad es una asociación entre un nombre (o clave) y un valor. El valor de una propiedad puede ser una función, en cuyo caso

la propiedad es conocida como un método. Además de los objetos que están predefinidos en el navegador, puedes definir tus propios objetos. Este capítulo describe cómo usar objetos, propiedades, funciones y métodos; y cómo crear tus propios objetos.

- Eventos

Evento	Descripción	Elementos para los que está definido
<code>onblur</code>	Un elemento pierde el foco	<code><button></code> , <code><input></code> , <code><label></code> , <code><select></code> , <code><textarea></code> , <code><body></code>
<code>onchange</code>	Un elemento ha sido modificado	<code><input></code> , <code><select></code> , <code><textarea></code>
<code>onclick</code>	Pulsar y soltar el ratón	Todos los elementos
<code>ondblclick</code>	Pulsar dos veces seguidas con el ratón	Todos los elementos
<code>onfocus</code>	Un elemento obtiene el foco	<code><button></code> , <code><input></code> , <code><label></code> , <code><select></code> , <code><textarea></code> , <code><body></code>
<code>onkeydown</code>	Pulsar una tecla y no soltarla	Elementos de formulario y <code><body></code>
<code>onkeypress</code>	Pulsar una tecla	Elementos de formulario y <code><body></code>
<code>onkeyup</code>	Soltar una tecla pulsada	Elementos de formulario y <code><body></code>
<code>onload</code>	Página cargada completamente	<code><body></code>
<code>onmousedown</code>	Pulsar un botón del ratón y no soltarlo	Todos los elementos
<code>onmousemove</code>	Mover el ratón	Todos los elementos
<code>onmouseout</code>	El ratón "sale" del elemento	Todos los elementos
<code>onmouseover</code>	El ratón "entra" en el elemento	Todos los elementos
<code>onmouseup</code>	Soltar el botón del ratón	Todos los elementos
<code>onreset</code>	Inicializar el formulario	<code><form></code>
<code>onresize</code>	Modificar el tamaño de la ventana	<code><body></code>
<code>onselect</code>	Seleccionar un texto	<code><input></code> , <code><textarea></code>
<code>onsubmit</code>	Enviar el formulario	<code><form></code>
<code>onunload</code>	Se abandona la página, por ejemplo al cerrar el navegador	<code><body></code>

Propiedad/Método	Devuelve	Descripción
<code>altKey</code>	Boolean	Devuelve <code>true</code> si se ha pulsado la tecla <code>ALT</code> y <code>false</code> en otro caso
<code>bubbles</code>	Boolean	Indica si el evento pertenece al flujo de eventos de bubbling
<code>button</code>	Número entero	El botón del ratón que ha sido pulsado. Posibles valores: 0 – Ningún botón pulsado 1 – Se ha pulsado el botón izquierdo 2 – Se ha pulsado el botón derecho 3 – Se pulsan a la vez el botón izquierdo y el derecho 4 – Se ha pulsado el botón central 5 – Se pulsan a la vez el botón izquierdo y el central 6 – Se pulsan a la vez el botón derecho y el central 7 – Se pulsan a la vez los 3 botones
<code>cancelable</code>	Boolean	Indica si el evento se puede cancelar
<code>cancelBubble</code>	Boolean	Indica si se ha detenido el flujo de eventos de tipo <i>bubbling</i>
<code>charCode</code>	Número entero	El código unicode del carácter correspondiente a la tecla pulsada
<code>clientX</code>	Número entero	Coordenada X de la posición del ratón respecto del área visible de la ventana
<code>clientY</code>	Número entero	Coordenada Y de la posición del ratón respecto del área visible de la ventana
<code>ctrlKey</code>	Boolean	Devuelve <code>true</code> si se ha pulsado la tecla <code>CTRL</code> y <code>false</code> en otro caso
<code>currentTarget</code>	Element	El elemento que es el objetivo del evento
<code>detail</code>	Número entero	El número de veces que se han pulsado los botones del ratón
<code>eventPhase</code>	Número entero	La fase a la que pertenece el evento: 0 – Fase capturing 1 – En el elemento destino 2 – Fase bubbling
<code>isChar</code>	Boolean	Indica si la tecla pulsada corresponde a un carácter
<code>keyCode</code>	Número entero	Indica el código numérico de la tecla pulsada
<code>metaKey</code>	Número entero	Devuelve <code>true</code> si se ha pulsado la tecla <code>META</code> y <code>false</code> en otro caso
<code>pageX</code>	Número entero	Coordenada X de la posición del ratón respecto de la página

- Expresiones Regulares

Las expresiones regulares son patrones que se utilizan para hacer coincidir combinaciones de caracteres en cadenas. En JavaScript, las expresiones regulares también son objetos. Estos patrones se utilizan con los métodos `exec()` y `test()` de `RegExp`, y con `match()`, `matchAll()`, `replace()`, `replaceAll()` (en-US), `search()` y `split()` métodos de `String`. Este capítulo describe las expresiones regulares de JavaScript.

- Librerías

- 1- jQuery
- 2- React.js
- 3- D3.js
- 4- Underscore.js
- 5- Lodash
- 6- Algolia Places
- 7- Anime.js
- 8- Animate On Scroll (AOS)
- 9- Bideo.js
- 10- Chart.js
- 11- Cleave.js
- 12- Choreographer.js
- 13- Glimmer
- 14- Granim.js
- 15- fullPage.js
- 16- Leaflet
- 17- Multiple.js
- 18- Moment.js
- 19- Masonry
- 20- Omniscient
- 21- Parsley
- 22- Popper.js
- 23- Three.js
- 24- Screenfull.js
- 25- Polymer
- 26- Voca
- 27- Bootstrap
- 28- Aurelia
- 29- Vue.js
- 30- Ember.js

- Animaciones

Las animaciones de JavaScript son imperativas, cuando las escribes de manera integrada como parte de un código. También puedes encapsularlas dentro de otros objetos.

Ejemplo:

```
var target = document.querySelector('.box');
var player = target.animate([
  {transform: 'translate(0)'},
  {transform: 'translate(100px, 100px)'}
], 500);
player.addEventListener('finish', function() {
  target.style.transform = 'translate(100px, 100px)';
});
```

Las animaciones de JavaScript pueden manejar cosas que CSS no puede. Por ejemplo, moverse a lo largo de una ruta compleja, con una función de sincronización diferente a las curvas de Bézier, o una animación en un canvas.

- 1- Limpiar el canvas
- 2- Guardar el estado del canvas
- 3- Dibujar formas animadas
- 4- Restaurar el estado del canvas