



UNIVERSIDAD DON BOSCO

Facultad de Estudios Tecnológicos

Desafío 3

Asignatura:

Desarrollo de Aplicaciones Web con Software Interpretados en el Cliente

(DAW)

Docente:

Hugo Dimas

GRUPO TEÓRICO: 03T

Integrantes:

Escobar Pineda, Wendy Beatriz	EP161427
Fuentes Cornejo, Meybel Tatiana	FC150457
Gutiérrez Fonseca, Oscar Roberto	GF161487
Ramírez Díaz, Marlon Josué	RD151447

FECHA:

DOMINGO 6 DE NOVIEMBRE 2016.

LUGAR:

SAN SALVADOR, EL SALVADOR.

INDICE

¿Qué es un Framework?	3
Arquitectura de un framework	3
Modelo.....	4
Vista.....	4
Controlador.....	4
¿Qué es Angular JS?	5
AngularJS a vista de pájaro	6
Vistas	6
Controladores:	6
Modelo de la vista	6
Conclusiones.....	8
Bibliografía	9
Anexos.....	11
Url de ejemplos.....	11

Investigación

✚ ¿Qué es un Framework?

La palabra inglesa "framework" (infraestructura, armazón, marco) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.



En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

Arquitectura de un framework

Dentro de este aspecto, podemos basarnos en el modelo-vista-controlador o MVC (Controlador => Modelo => Vista), ya que debemos fragmentar nuestra programación. Tenemos que contemplar estos aspectos básicos en cuanto a la implementación de nuestro sistema:

- *Modelo*

Este miembro del controlador maneja las operaciones lógicas, y de manejo de información (previamente enviada por su ancestro), para resultar de una forma explicable y sin titubeos. Cada miembro debe ser meticulosamente llamado, con su correcto nombre y en principio, con su verdadera naturaleza: el manejo de información, su complementación directa.

- *Vista*

Al final, a este miembro de la familia le corresponde dibujar, o expresar la última forma de los datos: la interfaz gráfica que interactúa con el usuario final del programa (GUI). Después de todo, a este miembro le toca evidenciar la información obtenida hasta hacerla llegar al controlador. Solo (e inicialmente), nos espera demostrar la información.

- *Controlador*

Con este apartado podemos controlar el acceso (incluso todo) a nuestra aplicación, y esto puede incluir: archivos, scripts, y/o programas; cualquier tipo de información que permita la interfaz. Así, podremos diversificar nuestro contenido de forma dinámica, y estática (a la vez); pues, solo debemos controlar ciertos aspectos (como se ha mencionado antes).

.NET Framework es una tecnología que soporta la compilación y ejecución de aplicaciones y servicios Web XML de última generación. El diseño de .NET Framework está enfocado a cumplir los siguientes objetivos:

Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.

- *Proporcionar un entorno de ejecución de código que minimiza los conflictos en el despliegue y versionado de software.*

- *Ofrecer un entorno de ejecución de código que promueva la ejecución segura del mismo, incluso del creado por terceros desconocidos o que no son de plena confianza.*
- *Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos.*
- *Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.*
- *Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.*

¿Qué es Angular JS?

Angular JS es el nombre que recibe el framework de JS creado por empleados de Google, lanzado en Octubre de 2010, el framework se basa en el patrón MVC (Model, View, Controller), con el objetivo de separar las capas de presentación, lógica y componentes de una aplicación.



AngularJS es Javascript. Es un proyecto de código abierto, realizado en Javascript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en este caso sobre el lenguaje Javascript con programación del lado del cliente.

→ **Mejoras del HTML**

Este Javascript pretende que los programadores mejoren el HTML que hacen. Que puedan producir un HTML que, de manera declarativa, genere aplicaciones que sean fáciles de entender incluso para alguien que no tiene conocimientos profundos de informática. El objetivo es producir un HTML altamente semántico, es decir, que cuando lo leas entiendas de manera clara qué es lo que hace o para qué sirve cada cosa.

Lógicamente, AngularJS viene cargado con todas las herramientas que los creadores ofrecen para que los desarrolladores sean capaces de crear ese HTML enriquecido. La palabra clave que permite ese HTML declarativo en AngularJS es "directiva", que no es otra cosa que código Javascript que mejora el HTML. Puedes usar el que viene con AngularJS y el que han hecho terceros desarrolladores, puesto que muchas personas están contribuyendo con pequeños proyectos -independientes del propio framework- para enriquecer el panorama de directivas disponibles. Hasta este punto serás un "consumidor de directivas", y finalmente cuando vayas tomando experiencia serás capaz de convertirte en un "productor de directivas", enriqueciendo tú mismo las herramientas para mejorar tu propio HTML.

→ AngularJS a vista de pájaro

Patrón que se usa en Angular, el conocido Modelo, Vista, Controlador.

Vistas: Será el HTML y todo lo que represente datos o información.

Controladores: Se encargarán de la lógica de la aplicación y sobre todo de las llamadas "Factorías" y "Servicios" para mover datos contra servidores o memoria local en HTML5.

Modelo de la vista: En Angular el "Modelo" es algo más de aquello que se entiende habitualmente cuando te hablan del MVC tradicional, osea, las vistas son algo más que el modelo de datos. En modo de ejemplo, en aplicaciones de negocio donde tienes que manejar la

contabilidad de una empresa, el modelo serían los movimientos contables. Pero en una pantalla concreta de tu aplicación es posible que tengas que ver otras cosas, además del movimiento contable, como el nombre de los usuarios, los permisos que tienen, si pueden ver los datos, editarlos, etc. Toda esa información, que es útil para el programador pero que no forma parte del modelo del negocio, es a lo que llamamos el "Scope" que es el modelo en Angular.

Conclusiones

- Una framework te permite desarrollar más rápido mediante el uso de código base y tareas comunes, de modo que no tienes que empezar desde cero.
- Una framework ya viene programada para que el resultado (es decir, tu aplicación o sitio web) final luzca similar en todos los navegadores, de modo que puedes concentrarte en personalizar y crear contenido en vez de preocuparte en modificar la apariencia base.
- A diferencia de la mayoría de sistemas de 'templates' y/o frameworks Angular usa un sistema en el que vista y modelo están en relación constante, se considera el modelo como 'Single-Source-of-Truth'. Gracias a esto, se logra que todo cambio visual, se actualice a tiempo real en el modelo y viceversa, evitando que sea el desarrollador el encargado de lograr la sincronía entre modelo y vista, como es el caso de otros frameworks.

Bibliografía

Año de publicación: **2016**

Título de la página: [https://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)

Nombre del sitio web: Msdn.microsoft.com

URL: [https://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)

Fecha de acceso: **5 de Noviembre 2016**

Año de publicación: **2016**

Título de la página: Framework

Nombre del sitio web: Es.wikipedia.org

URL: <https://es.wikipedia.org/wiki/Framework>

Fecha de acceso: **5 de Noviembre 2016**

Año de publicación: **2016**

Título de la página: Qué es AngularJS

Nombre del sitio web: DesarrolloWeb.com

URL: <http://www.desarrolloweb.com/articulos/que-es-angularjs-descripcion-framework-javascript-conceptos.html>

Fecha de acceso: **5 de Noviembre 2016**

Año de publicación: **2016**

Título de la página: AngularJS — Superheroic JavaScript MVW Framework

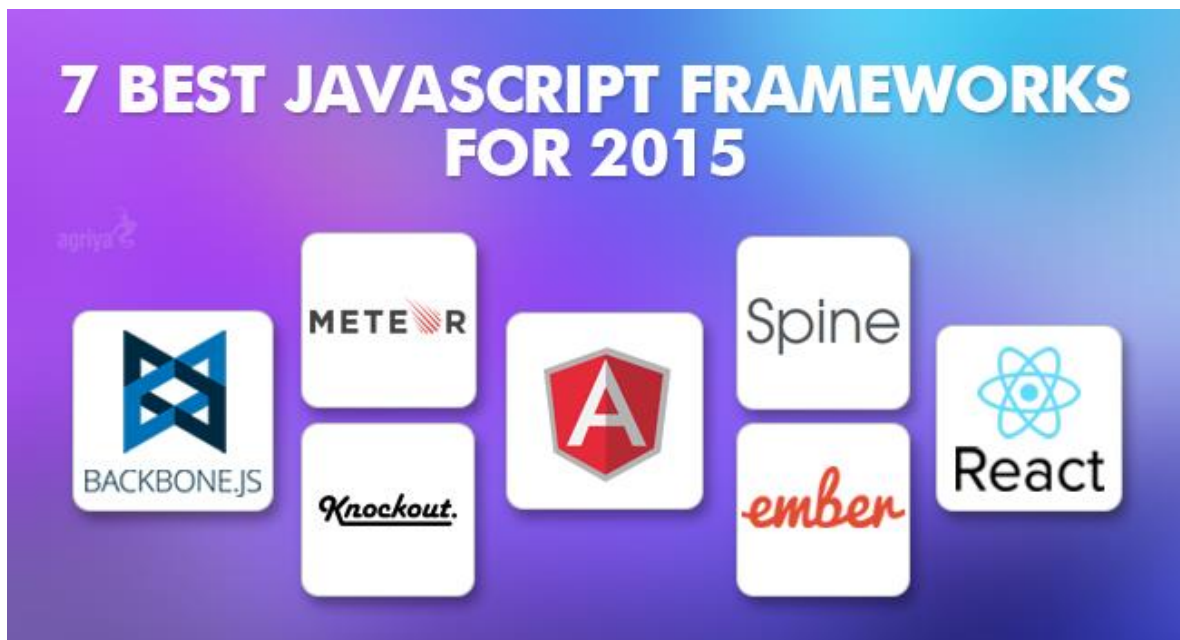
Nombre del sitio web: Angularjs.org

URL: <https://angularjs.org/>

Fecha de acceso: **5 de Noviembre 2016**

Anexos

Url de ejemplos: https://github.com/GF161487/Desafio3_DAW



```

(function () {
    angular
        .module('application')
        .controller('ContactCtrl', ['$scope', '$uibModalInstance', 'origin', 'Service', 'notify', '$translate',
            function ($scope, $uibModalInstance, origin, Service, notify, $translate) {
                $scope.data = {
                    origin: origin
                };
                $scope.send = function () {
                    Service.Contact($scope.data.name + ' ' + $scope.data.surname,
                        $scope.data.email,
                        $scope.data.phone,
                        $scope.data.origin,
                        $scope.data.message)
                        .then(function () {
                            $translate('info.contactMessageOK')
                                .then(function (message) {
                                    notify({
                                        message: message,
                                        classes: 'alert-info',
                                        templateUrl: 'views/common/notify.html'
                                    });
                                    $uibModalInstance.close();
                                }, function (err) {
                                    });
                            }, function (err) {
                                $translate('info.genericMessage')
                                    .then(function (message) {
                                        notify({
                                            message: message,
                                            classes: 'alert-danger',
                                            templateUrl: 'views/common/notify.html'
                                        });
                                        $uibModalInstance.close();
                                    }, function (err) {
                                        });
                                });
                            });
                });

                $scope.cancel = function () {
                    $uibModalInstance.dismiss('cancel');
                };
            }
        ]
    );
}

```