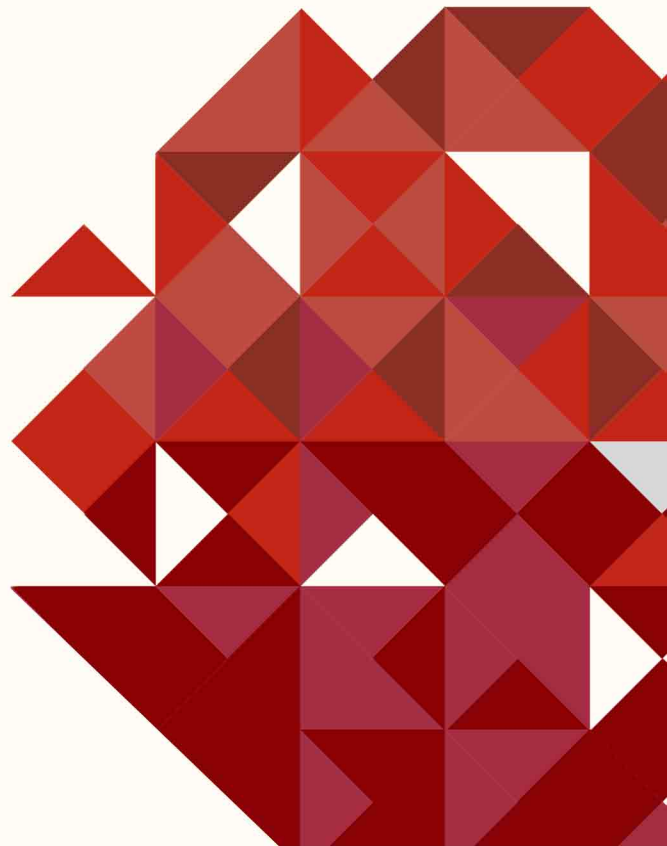




Manual del Guerrero

ANGULARJS



Carlos Solis

Manual del Guerrero: AngularJS

Carlos Solis

©2015 Carlos Solis

A los que fueron, a los que vienen y en especial, al loco que pensaba en estrellas lejanas.

...."Après moi, le déluge"

Índice general

Presentación	1
Prerrequisitos	1
¿Cómo leer este libro?	2
Primeros pasos	2
Creando aplicaciones	2
Manipulación de datos	2
Expandiendo AngularJS	3
Materiales adicionales	3
Mantente en contacto	3
SECCION 1: PRIMEROS PASOS	4
Introducción a AngularJS	5
¿Qué es AngularJS?	5
El Patrón MVW	6
Data binding	6
Inyección de dependencias	6
Directivas	7
¿Porqué usar AngularJS?	7
Herramientas de Desarrollo	9
Sublime Text 2	9
Configurar Sublime Text para AngularJS	10
Instalar Package Control	10
Instalar soporte para AngularJS	11
Instalar detector de errores JavaScript	12
Adobe Brackets	13
Configurar Brackets para AngularJS	13
Adobe Dreamweaver	15
Creando un Proyecto con AngularJS	16
Descargar AngularJS	16
Versión	17
Build	17

ÍNDICE GENERAL

CDN	18
Librerías adicionales	18
Instalar AngularJS en un documento HTML	19
Configurar una aplicación AngularJS	20
Alcance de ng-app	20
Inicializar la aplicación	21
Cargando módulos adicionales	21
Ejemplo: Hola Mundo	22
SECCION 2: CREANDO APLICACIONES	25
Expresiones	26
Definir expresiones	26
Plantillas	28
Representar valores dinámicos	28
Representar objetos	31
Representar resultados	32
Filtros en expresiones	35
Representar Divisas	35
Modificando las mayúsculas	39
Representando números	42
Formato de Fechas	44
Encadenar filtros	47
Filtros personalizados	49
Filtros de Datos	55
Filtrando datos	55
Ordenar colecciones de datos	57
Controladores	61
Creando un controlador	61
Activar el controlador en un documento	62
El objeto \$scope	63
Alcance de un controlador	65
Controladores múltiples	67
Métodos	69
Directivas	75
Agregar Directivas	75
Directivas con valores	77
Directivas AngularJS y estándares W3C	79
Directivas Comunes	81

ÍNDICE GENERAL

ng-app	81
ng-controller	82
ng-model	82
ng-bind	82
Directivas personalizadas	83
Creando una directiva personalizada	84
Aplicando directivas personalizadas en un documento	85
Agregando valores a las directivas personalizadas	86
Modificar estilos con AngularJS	89
Manipular propiedades CSS	91
Asignar clases	93
Ocultar y mostrar elementos	94
Modificar imágenes	97
Mostrar elementos condicionalmente	98
Eventos e interacción	101
Clicks	101
Eventos del mouse	104
Eventos para móviles	106
Simulando una pantalla táctil en Chrome	109
Eventos de teclado	113
SECCION 3: MANIPULACION DE DATOS	116
Representar colecciones de datos	117
Crear listas con arreglos	118
Agregar elementos a una lista	119
Asignar eventos en listas dinámicas	120
Crear listas con objetos	121
Acceder datos externos	124
Preparando los datos externos	125
Consideraciones de seguridad y permisos de lectura	127
Cargar información desde servidores externos	128
Cargando un archivo externo	128
Desplegar los datos cargados	131
SECCION 4: EXPANDIENDO ANGULAR	133
Integrando AngularJS con Bootstrap	134
¿Por qué usar Bootstrap?	134
Usando Bootstrap para hacer sitios responsivos	135
Contenedores	135
Matriz de columnas	135

ÍNDICE GENERAL

Soporte para diferentes resoluciones	136
Las clases	136
Creando una plantilla con Bootstrap	137
Agregando contenidos	138
Agregar Interactividad en Bootstrap	140
Instalar AngularJS en Bootstrap	141
Agregar controladores	142
Administrar plantillas y rutas	144
Indexación de AngularJS en motores de búsqueda	144
Instalando ngRoute	145
Preparando las plantillas	147
Usando includes	148
Vistas	149
Rutas	150
Definiendo las rutas	151
when	152
otherwise	153
Cargando controladores	154
Crear el archivo del controlador	155
Incluir el controlador dentro de la app	155
Relacionar el controlador con una ruta	156
Rutas dinámicas	157
Leer variables del URL	158
Conclusión y despedida	161
Próximos pasos	161
Documentación oficial	161
Angular 2	162
Cursos Online	162
Despedida	162

Presentación

Hay muchos tipos de desarrolladores, desde los que trabajan día y noche sin parar, hasta los artistas que usan el código como un medio de expresión, pero hay un tipo especial de desarrollador, uno que sin importar cuanto le pidan o cuanto cobre, siempre busca hacer mejor su código, el que pasa las madrugadas investigando y no tiene problema en cambiar horas de sueño por conocimiento.

Un guerrero no es un desarrollador común y corriente, es el que va mas allá, el que vive intensamente su pasión por el código perfecto y lucha por ganar cada línea de programación y reclamarla para la excelencia.

Es guerreros no quieren acabar con nadie, pero cada día se despiertan listos para luchar y darlo todo hasta quedar sin fuerzas. Un guerrero vive por su profesión y no hay espacio para el error, es la perfección o la muerte.

Este manual es para esos guerreros, no es para los que programan para vivir, sino para los que viven para programar.

Pre-Requisitos

Aunque el manual esta escrito de la manera mas simple y practica posible, puede ser que algunos conceptos te sean difíciles de comprender si no tienes algunos conocimientos previos. Antes de comenzar a leer este libro, necesitas tener nociones básicas de HTML y JavaScript.

Si nunca has trabajado estos lenguajes te recomiendo que leas algo de documentación sobre el tema o si quieres ahorrar tiempo y entrar a la vía rápida en el mundo del desarrollo web, puedes matricularte estos cursos online antes de continuar leyendo este libro.

HTML5 para Diseñadores Nivel 1

Impartido por Marlon Ceballos, este curso te enseñará las bases de HTML5 y a estructurar sitios de manera profesional.

<https://www.udemy.com/html5-disenadores-i/>

Introducción a JavaScript

Aprende las bases de la programación Javascript con este curso introductorio desde 0. Utiliza el código JSGRATISFTW para llevar este curso gratuitamente

<https://www.udemy.com/test368/>

Cómo leer este libro?

El manual del guerrero:AngularJS está ordenado progresivamente, cada capítulo ira profundizando cada vez mas en temas incrementalmente avanzados, si lo lees de forma lineal tendrás una experiencia fluida de aprendizaje. Pero yo también soy desarrollador y los desarrolladores somos impacientes, además muy posiblemente necesitas desarrollar ese sitio web en AngularJS con urgencia y solo tienes un par de días para hacerlo, por esta razón, cada capítulo fue pensado como una unidad independiente para que encuentres toda la información que necesitas de manera individual, así podrás encontrar únicamente el truco que buscas poner a andar tu componente o simplemente usarlo como referencia en unos meses cuando no recuerdes bien un tema.

Para facilitar tu lectura, este libro esta separado en cuatro unidades principales, cada una de ellas desarrolla un tema y necesidad distintos.

Primeros pasos

Comenzaremos el libro detallando todos los procedimientos de configuración de AngularJS y los programas necesarios para comenzar a crear tu código de inmediato. Examinaremos los principios fundamentales de AngularJS, las herramientas de desarrollo y depuración así como los elementos que vas a necesitar para trabajar durante todo el resto del proceso de aprendizaje. Si es la primera vez que trabajas con AngularJS, te recomiendo que leas con detalle esta primera parte en detalle para que te sientas con mas confianza y soltura durante el resto del libro, si por el contrario ya tienes alguna experiencia básica con AngularJS, puedes pasar directamente a la segunda sección y comenzar a experimentar con una aplicación.

Creando aplicaciones

En esta sección entraremos de lleno en el desarrollo de aplicaciones con AngularJS, revisaremos sus elementos fundamentales como las expresiones o directivas, aprenderemos a procesar datos con los controladores, modificar la apariencia de un documento con CSS, interactuar con el usuario a través de eventos y finalmente usaremos los servicios para conectar todo entre si para crear aplicaciones complejas.

Esta es la sección que utilizaras como referencia en el futuro, cuando seas todo un desarrollador de aplicaciones AngularJS, recuerda dejar muchos marcadores y notas, te servirán mas adelante para repasar los temas que puedas olvidar ;)

Manipulación de datos

Una sección particularmente importante para quienes se enfocan en la programación y desarrollo de aplicaciones sofisticadas, aquí examinaremos las formas de conectar tu aplicación con diferentes servicios de datos desde bases de datos locales hasta servicios externos.

Nos enfocaremos principalmente en la manipulación de datos en formato JSON que es el standard de facto en la industria, si aun no has utilizado este formato, creo que te encantará por su bajo uso de recursos y capacidad de almacenamiento, si ya tienes experiencia, aprenderás como aplicarlo eficientemente en un documento AngularJS.

Expandiendo AngularJS

Aquí aprenderemos a utilizar frameworks y herramientas complementarias para mejorar o expandir las capacidades de AngularJS. Debido a que AngularJS se enfoca únicamente en la lógica de las aplicaciones, te mostrare como integrarlo con Bootstrap para crear la interfaz de un sitio web profesional fácil y rápidamente.

Materiales adicionales

Puedes encontrar todos los ejercicios utilizados en este libro así como información sobre actualizaciones y promociones exclusivas para lectores en el sitio oficial:

<http://www.manualdelguerrero.com/angularjs>

Si prefieres utilizar Github puedes clonar el repositorio de ejercicios del libro en esta URL

<https://github.com/siddharta1337>

Mantente en contacto

La tecnología avanza a toda velocidad y ten por seguro que siempre tendremos actualizaciones sobre este y cualquier otra tecnología en la que trabajes. Si quieres estar completamente actualizado de cualquier cambio en AngularJS o en cualquier tema relacionado con el mundo del desarrollo, no te olvides de visitar mis redes sociales donde encontraras tutoriales, artículos gratuitos y puedes sugerir temas para mejorar este libro

Facebook: <https://www.facebook.com/revolucionmovil>

Twitter: https://twitter.com/revolucion_-mobi

Web: <http://revolucion.mobi>

SECCION 1: PRIMEROS PASOS

Introducción a AngularJS

La web de está era se ha transformado en un medio masivo, donde millones de usuarios se congregan para consumir contenido multimedia, realizar tramites, compartir información, aprender y divertirse. Son pocos los ámbitos de la sociedad que ya no utilizan la internet para interactuar con otras personas o incluso para trabajar.

Las aplicaciones web son posiblemente una de las formas mas comunes de interacción con los usuarios, no son las paginas estáticas de los 90's sino sitios que ofrecen servicios, procesan datos y realizan transacciones o cálculos complejos.

Hablar de aplicaciones web es referirse a los sitios mas populares de la web, como Gmail , Facebook o Amazon, todas ellas son aplicaciones web complejas que reciben y procesan enormes cantidades de datos y los administran para ofrecer sus servicios a un publico especializado.

Las aplicaciones web utilizan en muchas ocasiones servicios externos y para ofrecer una experiencia mas fluida a sus usuarios, por esta razón han surgido aplicaciones que descansan en AJAX para reducir el tiempo de carga y dar una experiencia consolidada.

Con la llegada de los móviles y el crecimiento de browsers , equipos y plataformas el trabajo de crear un sitio web se ha vuelto cada vez mas complejo y el tiempo de desarrollo es crucial. Muchos emprendimientos se han visto presa de su éxito precisamente porque aunque logran generar un producto inicial, pero no logran escalar su servicio a tiempo y terminan perdiendo a sus usuarios.

Para cumplir con los cronogramas de desarrollo cada vez mas ajustados, los programadores utilizan frameworks para ahorrar tiempo y cumplir con las demandas de calidad de sus proyectos. AngularJS es uno de esos frameworks y ofrece una alternativa de desarrollo rápida, escalable y sencilla de aprender.

Gracias a sus capacidades y velocidad, AngularJS es uno de los frameworks mas populares y uno de los favoritos de las grandes empresas y emprendimientos porque permite cumplir con la producción rápidamente y escalar el producto sin temor.

¿Qué es AngularJS?

AngularJS es, en síntesis, un framework de código abierto y gratuito desarrollado por Google. Esta basado en el popular lenguaje Javascript y su objetivo principal es crear aplicaciones web dinámicas y eficientes.

A diferencia de otros frameworks populares, AngularJS es un framework estructural, no depende ni esta compuesto por elementos gráficos, imágenes o CSS, solamente se enfoca en administrar la parte lógica de tu aplicación.

El Patrón MVW

El patrón MVC (modelo vista controlador) es uno de los patrones de programación mas populares para desarrollo de aplicaciones y permite administrar una aplicación, separando los datos, la interfaz e interactividad en diferentes capas independientes.

La mayoría de los frameworks JavaScript modernos implementan en alguna medida este patrón, pero requieren que separes todos tus archivos en capas solo para pedirte luego agruparlos de nuevo en tu aplicación, eso requiere gran cantidad de trabajo redundante. AngularJS por su parte implementa el patrón MVC pidiéndote que separes tu aplicación en diferentes capas, pero una vez que lo haces el framework se encarga del resto. AngularJS administra todos los módulos por ti y ademas funciona como el catalizador que los unifica a todos los elementos de tu aplicación.

Sin embargo, siendo estrictamente técnicos AngularJS utiliza una variación de este patrón llamado MVW un acrónimo de Model View Whatever (modelo-vista-lo que quieras) este nombre fue acuñado por uno de de sus desarrolladores para representar la libertad de desarrollo que ofrece este framework.

El concepto de MVW es ayudar a reducir al máximo el trabajo manual de crear una aplicación de gran escala sin comprometer la calidad y las buenas practicas de programación.

Data binding

Posiblemente una de las características mas populares de AngularJS es el data binding, que consiste en unir en tiempo real los datos de dos elementos, si el valor de uno de esos elementos cambia, el efecto se reflejará de inmediato en el otro elemento enlazado.

Esta técnica es sumamente útil para realizar cálculos o para representar gráficamente los cambios que realiza el usuario, tradicionalmente la mayoría de los frameworks pueden implementar esta conducta utilizando eventos y funciones adicionales que ocupan tiempo y depuración, en AngularJS el data binding esta integrado y no requiere ni siquiera de una línea de código, solo unas cuantas propiedades y tendrás un enlace en dos vías de datos.

Este tipo de enlaces se hacen en tiempo real y el usuario notará de inmediato el resultado de cualquier cambio o proceso que realice en la aplicación.

Inyección de dependencias

Desde el momento que comienzas a crear tu aplicación, no importa lo simple que sea, esta diseñada para crecer modularmente. La librería de AngularJS se mantiene únicamente con los

elementos básicos para funcionar, pero si en el futuro necesitas agregar nuevas funcionalidad, puedes usar librerías extra.

Para mantener esta modularidad AngularJS utiliza la inyección de dependencias, eso significa que una vez que importas o creas una librería solo tienes que inyectar una dependencia de esta en cualquier parte del código para que este disponible en tu aplicación, sin conflictos de instancias ni implementaciones complejas.

Tu aplicación puede crecer indefinidamente utilizando nuevos módulos progresivamente, AngularJS crecerá contigo. Al usar módulos puedes mantener tu código encapsulado y fácil de mantener, créeme, los futuros desarrolladores que trabajen en tu aplicación te agradecerán que no dejes un código mezclado a lo largo de un solo archivo imposible de leer o depurar.

Directivas

Las directivas son la firma con la que vas a reconocer una aplicación creada con AngularJS. En pocas palabras, las directivas te permiten darle poderes adicionales al código HTML regular.

Las directivas son elementos de programación que inyectamos en un documento web totalmente compatibles con la sintaxis HTML, son fáciles de recordar y permiten crear conductas o código complejo en un documento. Están pensadas para ahorrarte tiempo al agregar conductas avanzadas con solo incluir unas cuantas propiedades en el HTML.

Las directivas te permiten reutilizar funciones y tareas predefinidas en el código de AngularJS permitiéndote invocarlos una y otra vez en tu documento con solo incluir una etiqueta o propiedad especial.

Adicionalmente, aunque AngularJS incluye un amplio repertorio de variadas directivas de uso común, tienes total control para seguir extendiendo la colección con tus propias directivas personalizadas.

¿Porqué usar AngularJS?

Utilizar un framework para un nuevo proyecto puede ser una apuesta costosa, debes estar seguro no solo de que todo funcionara bien al inicio, también debes contar con que en el futuro puedas escalar tu aplicación y hacerla crecer según tus necesidades.

AngularJS es un excelente framework y uno de los favoritos de grandes empresas y emprendimientos precisamente por reunir las características necesarias para administrar desde una pequeño sitio web hasta aplicaciones masivas con millones de usuarios. Algunas de las características que convierten AngularJS en una excelente opción para crear tu proyecto son:

1. **Es extremadamente popular:** te será muy fácil encontrar materiales, foros y contratar desarrolladores que dominen el tema.

2. **No utiliza componentes gráficos:** tienes libertad total para personalizar tu aplicación hasta el mas mínimo detalle.
3. **Es Liviano y eficiente:** El framework completo mide apenas 105kb y esta optimizado para utilizar al mínimo los recursos del sistema.
4. **Escribes Menos Código:** Todo el framework esta diseñado para ahorrarte tiempo sin perder de vista la calidad y buenas practicas.
5. **Coexiste con otros frameworks:** Puedes utilizar AngularJS con otros frameworks y herramientas como jQuery, Bootstrap o PhoneGap sin temor a que aparezcan problemas de incompatibilidad.

Creando un Proyecto con AngularJS

Una aplicación AngularJS en su mínima expresión es, en síntesis, un documento HTML enlazado a un archivo javascript en el cual se encuentran un grupo de comandos e instrucciones.

Sin embargo, para implementar una aplicación AngularJS necesitas seguir algunos pasos, que aunque sencillos, son indispensables para que tu aplicación se ejecute correctamente.

Dedicaremos este capítulo a enumerar y revisar cada uno de los pasos y elementos que componen una aplicación AngularJS: cómo instalar, configurar e inicializar AngularJS. Al final de este capítulo aplicaremos todos los conceptos y técnicas para crear nuestra primera aplicación.

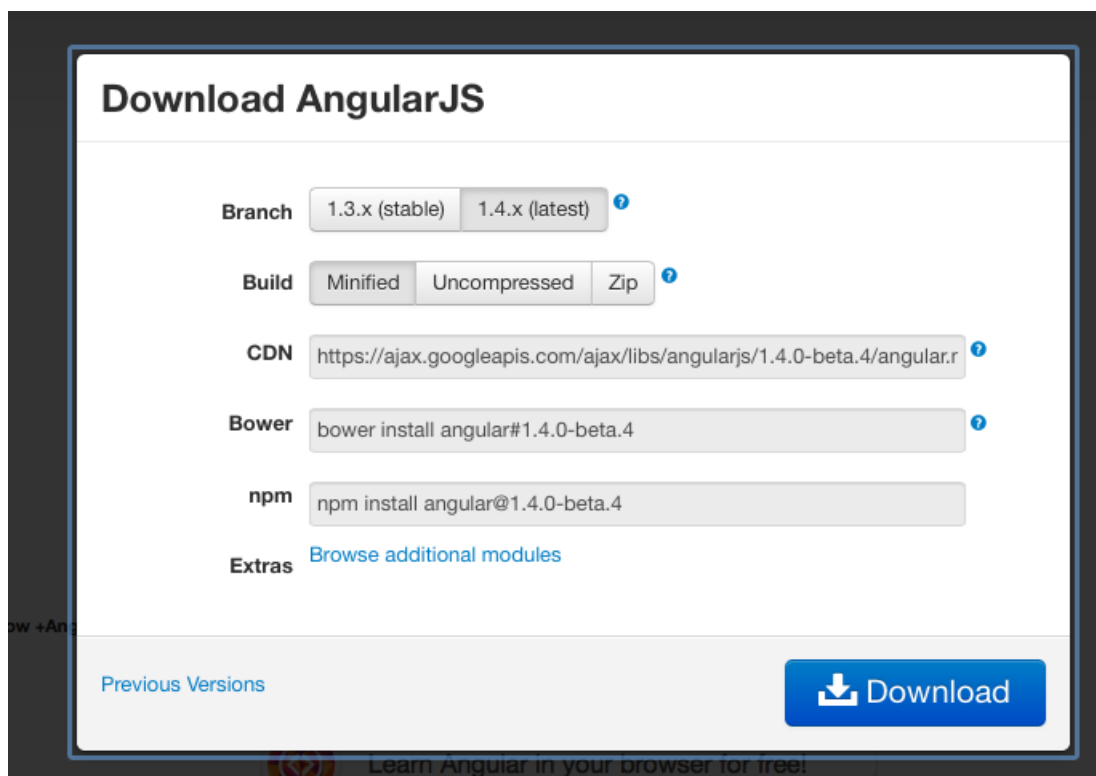
Descargar AngularJS

Sin duda el primer paso para comenzar a utilizar esta librería es tener acceso a su código, AngularJS nos ofrece varias formas de utilizar sus rutinas y procesos.

AngularJS se debe instalar desde el sitio oficial de Google, aunque puedes encontrarlo en algunos otros lugares, repositorios o enlaces, es recomendable que en la medida de lo posible utilices la versión del sitio oficial para estar 100% seguro de que tienes no solo una versión estable sino también un programa seguro sin ningún tipo de modificación de su código original. El sitio oficial es:

<https://angularjs.org>

En cuanto ingreses al sitio oficial encontrarás un botón que te invita a descargar el código de AngularJS, al hacer click en el encontraras una ventana con varias opciones para personalizar tu descarga



Todas estas opciones están pensadas para que puedas adaptar AngularJS a diferentes escenarios. Vamos a revisar cada uno de estos apartados para ayudarte a elegir la versión perfecta para tus necesidades.

Version

Encontrarás 2 opciones, la primera te permite descargar la versión antigua de AngularJS (1.3.x) elige esta opción si ya tienes un sitio que usa AngularJS, si ya has creado componentes personalizados o si has encontrado problemas de compatibilidad con la versión actual.

La segunda opción te permite descargar la última versión de AngularJS (1.4.x) Si estás leyendo este libro es probable que sea la primera vez que trabajas con AngularJS, te recomiendo que elijas esta opción porque te permite aprovechar lo mejor y más avanzado de este framework.

Build

Esta opción nos permite elegir el nivel de compresión y tamaño del archivo. La opción “Zip” descarga un archivo comprimido con el código fuente de AngularJS, utiliza esta versión si vas a trabajar en una aplicación de Intranet, si tu aplicación se ejecutara de manera local o con limitación de acceso a internet. La opción Uncompressed contiene el código fuente de AngularJS sin ninguna compresión, con comentarios y detalles sobre el funcionamiento del programa, esta opción es perfecta si contribuyes al código de AngularJS o si deseas

crear funciones adicionales. Finalmente, la opción Minified genera una versión del código completamente reducida y optimizada para tener el mínimo tamaño posible, te recomiendo que siempre que puedas uses esta versión para asegurarte los mejores resultados y la carga mas rápida posible.

CDN

Un CDN es un Content Delivery Network o red de distribución de contenidos, al utilizarlo tendrás varias ventajas como ahorro de ancho de banda y olvidarte de la necesidad de mantener tu código en un servidor.

Al utilizar un CDN tu archivo se relaciona con un vinculo dinámico que se ajusta al servidor mas cercano a la ubicación del usuario que descarga tu aplicación. Esto hace que el archivo javascript tenga la menor latencia y por ende se descargue mas rápido lo que hace tu sitio mas rápido y eficiente.

Librerías adicionales

Las librerías adicionales te permitirán incluir nuevas modalidades, están separadas de núcleo central de código para reducir el tamaño total de la instalación. Al utilizar un esquema modular tu aplicación nunca va a cargar código que no necesite, solamente lo importante y con causa será instalado en tu aplicación.

Una vez elegidas las opciones de tu archivo, procede al proceso de descarga. Si elegiste las opciones recomendadas, al presionar download la aplicación te enviara a un documento que posiblemente se vea similar a este

```

/*
AngularJS v1.3.9
(c) 2010-2014 Google, Inc. http://angularjs.org
License: MIT
*/
(function(M,Y,t){'use strict';function T(b){return function(){var a=arguments[0],c=c+"(b?b+":"")+a+"}
http://errors.angularjs.org/1.3.9/"+(b?b+"/":"")+a;for(a=1;a<arguments.length;a++){c=c+(1==a?" ":"&")+a+"(a-1)+"=";var
d=encodeURIComponent(e,e=arguments[a];e="function"==typeof e?e.toString().replace(/\s\S/g,""):e="undefined"==typeof
e?"undefined":"string"!=typeof e?JSON.stringify(e):e;c+=d(e)}return Error(c)}function Ta(b){if(null==b||!b)return!1;var a=b.length;return
b.nodeType===
o&a%10:F(b)||D(b)||0===a||"number"==typeof a&&0<a&a-1 in b}function s(b,a,c){var d,e;if(b)if(G(b))for(d in
b)"prototype"==d||"length"==d||"name"==d||b.hasOwnProperty&&b.hasOwnProperty(d)||a.call(c,b[d],d,b);else if(D(b)||Ta(b)){var
f="object"==typeof b;d=0;for(e=b.length;d<c;d++){f||d in b&&a.call(c,b[d],d,b)}else if(b.forEach&&b.forEach!==s)b.forEach(a,c,b);else for(d
in b)b.hasOwnProperty(d)&&a.call(c,b[d],d,b);return b}function Ed(b,a,c){for(var d=Object.keys(b).sort(),e=0;e<d.length;e++)a.call(c,
b[d[e]],d[e]);return d}function kc(b){return function(a,c){b(c,a)}}function Fd(){return+nb}function lc(b,a){a?b.$hashKey=a:delete
b.$hashKey}function z(b){for(var a=b.$hashKey,c=1,d=arguments.length;c<d;c++){var e=arguments[c];if(e)for(var
f=Object.keys(e),g=0,h=f.length;g<h;g++){var l=f[g];b[l]=e[l]}lc(b,a);return b}function ba(b){return parseInt(b,10)}function H(){function
pa(b){return b}function da(b){return function(){return b}}function A(b){return"undefined"==typeof b}function y(b){return"undefined"!
typeof b}function I(b){return null==b&&"object"==typeof b}function F(b){return"string"==typeof b}function V(b){return"number"==typeof
b}function qa(b){return"object Date"==Da.call(b)}function G(b){return"function"==typeof b}function ob(b){return"object
RegExp"==Da.call(b)}function Ua(b){return b&&b.window==b}function Va(b){return b&&b.$evalAsync&&b.$watch}function Wa(b)
{return"boolean"==typeof b}function mc(b){return!(!b||!(b.nodeName|b.prop&&b.attr&&b.find))}function Gd(b){var a={};
b=b.split(",");var c;for(c=0;c<b.length;c++){a[b[c]]=!0;return a}function ua(b){return Q(b.nodeName|b[0]&&b[0].nodeName)}function Xa(b,a){var
c=b.indexOf(a);0<c&&b.splice(c,1);return a}function Ea(b,a,c,d){if(Ua(b)||Va(b))throw Ka("cpws");if(a){if(b==a)throw Ka("cpi");c=c||[];d=d||
[];if(I(b)){var e=c.indexOf(b);if(-1==e)return d[e];c.push(b);d.push(a)}if(D(b))for(var
f=a.length-0;f<b.length;f++){e=Ea(b[f],null,c,d,I(b[f]))&&(c.push(b[f]),d.push(e)),a.push(e);else{var g=a.$hashKey;D(a)?a.length=
0:s(a,function(b,c){delete a[c]});for(f in b)b.hasOwnProperty(f)&&(e=Ea(b[f],null,c,d,I(b[f]))&&(c.push(b[f]),d.push(e)),a[f]=e);lc(a,g)}else
if(a=b)D(b)?a=Ea(b,[],c,d):qa(b)?a=new Date(b.getTime()).ob(b)?(a=new RegExp(b.source,b.toString().match(/[/\]/g)?$/))
[0],a.lastIndex=b.lastIndex):I(b)&&(e=Object.create(Object.getPrototypeOf(b)),a=Ea(b,e,c,d));return a}function ra(b,a){if(D(b)){a=a||
[];for(var c=0,d=b.length;c<d;c++)a[c]=b[c]}else if(I(b))for(c in a=a||[],b){if("$"!=c.charAt(0)||"$"!=c.charAt(1))a[c]=
b[c];return a}b}function fa(b,a){if(b==a)return!0;if(null==b||null==a)return!1;if(b!=b&&a!=a)return!0;var c=typeof b,d;if(c==typeof
a&&"object"==c){if(D(b)){if(!D(a))return!1;if((c=b.length)==a.length){for(d=0;d<c;d++){if(!fa(b[d],a[d]))return!1;return!0}else{if(qa(b))return
qa(a)?fa(b.getTime(),a.getTime()):!1;if(ob(b)&&ob(a))return b.toString()==a.toString();if(Va(b)||Va(a)||Ua(b)||Ua(a))return!1;for(d
in b)if("$"!=d.charAt(0)&&G(b[d])){if(!fa(b[d],a[d]))return!1;c[d]=!0}for(d in a)if(!c.hasOwnProperty(d)&&
"$"!=d.charAt(0)&&a[d]!=t&&G(a[d]))return!1;return!0}return!1}function Ya(b,a,c){return b.concat(Za.call(a,c))}function nc(b,a){var
c=2<arguments.length?Za.call(arguments,2):[];return!G(a)||a instanceof RegExp?a.c.length?function(){return arguments.length?
a.apply(b,Ya(c,arguments,0)):a.apply(b,c):function(){return arguments.length?a.apply(b,arguments):a.call(b)}}function Hd(b,a){var
c=a;"string"==typeof b&&"$"==b.charAt(0)&&"$"==b.charAt(1)?c=t:Ua(a)?c="$WINDOW":a&&Y==a?c="$DOCUMENT":Va(a)&&
(c="$SCOPE");return c}function $(a,b){if("undefined"==typeof b)return t;V(a)||a?a?2:null;return JSON.stringify(b,Hd,a)}function oc(b)
{return F(b)?JSON.parse(b):b}function va(b){b=B(b).clone();try{b.empty()}catch(a){var c=B("<div>").append(b).html();try{return
b[0].nodeType==pb?Q(c):c.match(/^(?>)/[1].replace(/^(?<[^\w-]+)/,function(a,b){return"<">Q(b)}))}catch(d){return Q(c)}}function pc(b)
{try{return decodeURIComponent(b)}catch(a){}}function qc(b){var a=c,d;s(b||"").split("&"),function(b){b&&

```

Lo que ves el el código fuente de AngularJS, solo debes copiar el URL que se encuentra en la parte superior, algo similar a esto:

<http://ajax.googleapis.com/ajax/libs/angularjs/1.3.9/angular.min.js>

Conserva a mano este URL, lo vas a necesitar en un minuto cuando instales AngularJS en un documento HTML.

Instalar AngularJS en un documento HTML

AngularJS depende directamente de tu código HTML, por eso, para instalarlo te recomiendo crear primero un documento HTML bien estructurado, con código validado y siguiendo los standards para asegurarte que todo esta perfecto desde el inicio. Puedes comenzar tu aplicación, utilizando este código base

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>AngularJS</title>
6  <meta name="description" content="">
7  <meta name="viewport" content="width=device-width">
8  </head>
9  <body>
10
11 </body>
12 </html>
```

Ahora que tienes el marco perfecto para tu aplicación es momento de instalarlo en tu documento. El proceso de instalación no puede ser mas sencillo, solo debes colocar esta línea de código justo antes de la etiqueta `</body>`

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.9/angular.min.js"></script>
```

Para este punto tienes un documento listo para utilizar AngularJS, sin embargo, aunque el documento tiene acceso a la librería no tiene ninguna indicación de como y donde usarla, para ello es necesario definir e inicializar una aplicación

Configurar una aplicación AngularJS

Uno de los elementos fundamentales de AngularJS son las directivas, elementos que podemos inyectar en el código HTML que nos permiten realizar procesos complejos. Mas adelante hablaremos de ellas en detalle, por ahora vamos a ver nuestra primera directiva y una de las mas importantes ya que al incluirla en nuestro documento, le indicaremos a AngularJS que puede comenzar a procesar esa parte del documento.

Hablamos de la directiva *ng-app* y la forma de utilizarla es incluyéndola dentro de nuestro documento en forma de un parámetro HTML.

Veamos un ejemplo: supongamos que yo deseo que el documento HTML en el que estamos trabajando se comporte como una aplicación angular, solo debo incluir esta directiva dentro de cualquier etiqueta HTML y definir el nombre de mi aplicación. En este caso, si mi aplicación se llama "miAplicacionAngular" la forma de inicializarla seria así

```
1  <body ng-app="miAplicacionAngular">
2
3  </body>
```

El valor que le estoy asignando a “ng-app” es el nombre con el que identificare a mi aplicación y tienes libertad de darle el nombre que desees.

Toma en cuenta que aunque es técnicamente posible incluir únicamente la directiva sin asignarle un nombre a la aplicación y de hecho encontraras muchos ejemplos en internet que declaran la directiva “ng-app” de esta forma:

```
1 <body ng-app >
2
3 </body>
```

Aunque en teoría una aplicación tiene la capacidad de funcionar con ese código, es mejor que siempre le definas un nombre desde el inicio. Conforme tu aplicación se vuelva mas compleja, tarde o temprano estarás obligado a asignarle un nombre, así que es mejor arrancar con un documento de buena calidad desde el primer momento. En este libro usaremos esta practica en todos los ejemplos y te recomiendo aplicarlo en tus propios documentos.

Alcance de ng-app

La directiva “ng-app” no solo define una aplicación AngularJS, también la delimita a un área de acción específica.

Puedes incluir la directiva “ng-app” en cualquier parte de tu documento, pero debes tomar en cuenta que las acciones de tu aplicación AngularJS tendrán efecto únicamente dentro de la etiqueta en que la asignes. Cualquier elemento que este fuera de la etiqueta elegida, será totalmente ignorado por la aplicación.

Tienes libertad total para usar la directiva “ng-app” en cualquier parte de tu documento y esta posición dependerá muchas veces de las necesidades especificas de tu aplicación, sin embargo te recomiendo que de ser posible incluyas esta directiva en las etiquetas <html> o <body> (solo en una de ellas a la vez) así tendrás una sola app para englobar todas las rutinas del sitio, te aseguras que todo el contenido será parte de la aplicación y tendrás control sobre todos los elementos visibles del documento HTML.

Inicializar la aplicación

Ahora que ya tienes declarada tu aplicación usando la directiva “ng-app” es momento de inicializarla. Para realizar este paso debes crear una variable donde declararemos el modulo correspondiente a tu nueva aplicación. Por ejemplo si tenemos una aplicación que fue declarada con:

```
1 ng-app='holamundo'
```

Debemos inicializarla con el codigo:

```
1 var miaplicacionAngular = angular.module('holamundo', [])
```

En este caso le asigno a la variable el nombre “miaplicacionAngular”, pero tu puedes usar cualquier nombre. El valor que almacena esta variable declara un nuevo modulo de angularJS en el cual estará tu aplicación.

Cargando modulos adicionales

Notaras tambien que agregué un arreglo vacio despues del nombre del modulo “[]” , la intencion de este arreglo es declarar dependencias adicionales que le añadiran funcionalidades nuevas a tu aplicacion. AngularJS nos permite incluir modulos adicionales a la hora de declarar nuestra aplicacion y asi mantener un codigo modular. Cuando descargas el framework tienes incluyen unicamente las funcionades mas basicas y frecuentes para mantener la carga rapida y eficiente, pero si necesitas servicios adicionales puedes descargar nuevos modulos e incorporarlos en este arreglo para que esten disponibles en tu aplicacion. Por ejemplo, en el caso de que quisieramos agregarle soporte para administrar los URL’s a traves del modulo adicional de routing, lo inyectariamos de esta forma:

```
1 var miaplicacionAngular = angular.module('holamundo', ['ngRoute'])
```

En este ejemplo no usaremos modulos adicionales, pero en los siguientes capitulos aprovecharemos esta funcionalidad para expandir las capacidades de tu aplicacion.

Ejemplo: Hola Mundo

Para este punto ya sabes como instalar, configurar y declarar AngularJS en un documento HTML. Vamos a poner todos los temas de este capitulo en practica para crear ¡Tu primera aplicación AngularJS! En este ejemplo vamos a aplicar los contenidos de este capitulo para crear una aplicación HTML básica.

Abre tu editor de código favorito y crea un nuevo documento HTML5, asígnale un titulo en la etiqueta <title> y agrega un titular <h1> dentro de la etiqueta <body>. Tu documento debería verse muy similar a este:

```
1 <!DOCTYPE html>
2 <html ng-app='holamundo'>
3
4 <head>
5   <meta charset="utf-8">
6   <title>AngularJS - Hola Mundo!</title>
7   <meta name="viewport" content="width=device-width">
8 </head>
9 <body>
10
11   <h1> Hola Mundo! </h1>
12
13 </body>
14
15 </html>
```

Guarda el documento con el nombre: “holamundo.html”. Ahora es momento de agregar AngularJS, primero debes incluir el enlace al framework:

```
1 <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.9/angular.min.js">
2 "></script>
```

Luego insertar la directiva ng-app con el nombre de tu aplicación, en este caso la pondremos en la etiqueta <html>

```
1 <html ng-app='holamundo'>
```

Finalmente declaramos el modulo de la aplicación para inicializarla

```
1 var miAppAngular = angular.module('holamundo', []);
```

Ahora tenemos una aplicación AngularJS lista para trabajar, vamos a adelantarnos al tema del próximo capítulo y usaremos una expresión para que pruebes lo genial que es AngularJS. Reemplaza el <h1> actual por este código:

```
1 <h1> {{ "Mi Aplicación AngularJS #" + 1 |uppercase }}</h1>
```

La expresión que hemos incluido esta mostrando ahora una concatenación y modificando el texto para que solo se despliegue en mayusculas.

El código completo de tu primera aplicación seria este:

```
1  <!DOCTYPE html>
2  <html ng-app='holamundo'>
3
4  <head>
5      <meta charset="utf-8">
6      <title>AngularJS - Hola Mundo!</title>
7      <meta name="viewport" content="width=device-width">
8  </head>
9  <body>
10
11      <h1> {{ "Mi Aplicación AngularJS #" + 1 |uppercase }}</h1>
12
13      <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.9/angular.min.js"></script>
14      <script>
15
16          var miAppAngular = angular.module('holamundo', []);
17
18      </script>
19  </body>
20 </html>
```

Guarda tu documento y ábrelo en el navegador, tendrás un resultado similar a este:



Como puedes ver, con solo unos ajustes y con muy poco esfuerzo ya tienes funcionando tu primera aplicación AngularJS ... ¿Genial no?

Esto es solo el inicio, en los siguientes capítulos aprenderás a crear aplicaciones completas a velocidad récord, muy pronto te convertirás en un todo un desarrollador de aplicaciones AngularJS

Extracto del "Manual del Guerrero:AngularJS"

Encuentra el libro completo en

www.manualdelguerrero.com