

En muchas ocasiones me encuentro con que al comenzar a trabajar con Angular.js a casi todo el mundo le resulta difícil entender el concepto de Inyección de Dependencia y Convención sobre configuración que el framework aplica a sus ejemplos más básicos. Vamos a ver un bloque de código de Angular.js y explicarlo un poco más a detalle.

```
<html ng-app="moduloFacturas">
<head>
<script type="text/javascript" src="factura.js">
</script>
<script type="text/javascript" src="angular.min.js">
</script>
<script type="text/javascript">

angular.module('moduloFacturas', []).controller("controladorFacturas", function($scope,$http) {

var peticion= $http.get("facturas");

peticion.success(function(listaFacturas) {

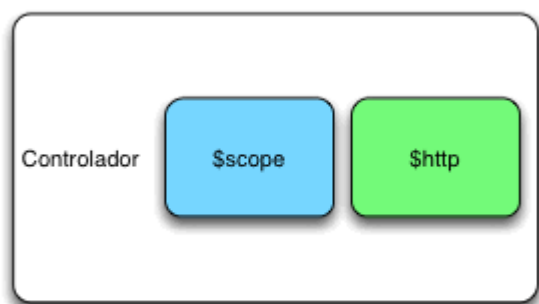
$scope.listaFacturas=listaFacturas;

});
});

</script>
</head>
```

```
<body ng-controller="controladorFacturas">
<table >
<tr ng-repeat="factura in listaFacturas">
<td>{{factura.id}}</td>
<td>{{factura.concepto}}</td>
<td>{{factura.importe}}</td>
</tr>
</table>
</body>
</html>
```

Aquí tenemos definido un controlador principal (controladorFacturas) que es el encargado de dibujarnos una lista de Facturas en nuestra web. Ahora bien el controlador recibe como parámetros dos objetos :\$scope y \$http. ¿Como funciona esto exactamente?.

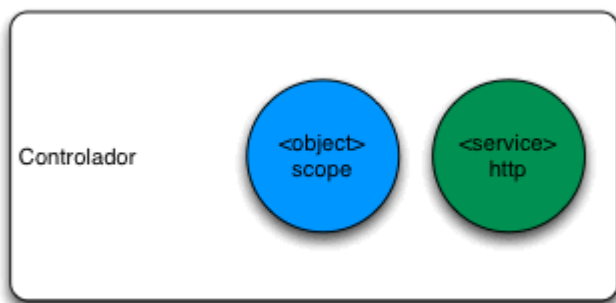


Angular y su Motor

El motor de Angular.js es capaz de a través de la declaración que acabamos de realizar obtener una cadena con el siguiente contenido :"\$scope,\$http" la cual procede a partir y sacar cada una de las literales.



Una vez hecho esto Angular se apoyará en el principio de Convención sobre Configuración(COC) y si la variable se denomina "\$scope" inyectará un objeto \$scope a la función que esta delimitado por el controlador en el que estamos ubicados. En segundo lugar si la variable se denomina \$http Angular será capaz de darse cuenta que hacemos referencia al servicio de \$http y lo inyectará.



De esta forma Angular inicializará los objetos que el controlador necesita. Así pues y al apoyarse en el principio de Convención sobre Configuración. El programa no nos funcionará si cambiamos el nombre de las variables por \$ambito y \$servicioAjax por poner un ejemplo algo que cuesta entender en un primer momento. Una vez hecho la aplicación arrancará sin ningún problema y nos mostrará una lista de facturas por pantalla.

A mac 1000

B Iphone 700

Otros artículos relacionados: [Batarang](#) , [Libro Angular](#)