

# 13° Meetup AngularJS México

Angular, Three.js y WebSockets

# ¿Qué es Three.js?

- Una biblioteca en Javascript para utilizar WebGL, aunque de ser necesario utiliza Canvas o SVG.
- Está orientada a objetos y oculta gran parte de los detalles complicados de WebGL, sin limitar el acceso para los programadores que los requieran.

# ¿Qué es WebSockets?

- Es un protocolo que provee canales de comunicación full-duplex sobre una misma conexión TCP.
- Existen varias bibliotecas para manejarlo, la más conocida es Socket.io, es necesario implementarlo tanto en cliente como en servidor.

# Planeación

Three.js está contenida en el objeto global THREE, por buenas prácticas y cuestiones de pruebas es mejor inyectarlo a AngularJS o a la biblioteca que lo utilice.

# Planeación

Las tendencias actuales son hacia los componentes, por lo que es mejor usar directivas y evitar controladores sueltos en medida de lo posible. Esto hace posible una web más semántica (para programadores al menos).

# No recomendable

```
ngjismx.controller('visorCtrl', ['$window', 'sailsIO', '$scope', function(window, ws, $) {  
  ws.createRoom();  
  
  $.reset = function() {  
    ws.clearRoom();  
  };  
}])
```

# Recomendable

```
ngjsmx.directive('deviceState', ['$window', function(window) .{  
    var deviceStateObject = .{  
        restrict: 'A',  
        require: '^deviceManager',  
        link: function($scope, element, attrs, devManager) .{  
            $scope.state = 'No Conectado';  
            $scope.id = null;  
  
            var socketReady = function (ev, data) .{  
                $scope.state = 'Conectado';  
                $scope.id = data.id;  
            };  
  
            $scope.$on('idReady', socketReady);  
        },  
        templateUrl: '../ngTemplates/state.html'  
    };  
  
    return deviceStateObject;  
}]);
```

# Nota

Las directivas facilitan la comunicación entre componentes mediante sus respectivos controladores.



# Eventos

AngularJS tiene un event hub incluido, es más sencillo usarlo que crear o incluir otro.

Hay que pensar los eventos jerárquicamente para evitar terminar con código spaghetti.

# Event Hub

```
var joinCb = function( res, rwjson ) {  
    self.socketId = res.id;  
    s.$emit('idReady', {id: self.socketId});  
};
```

```
var socketReady = function (ev, data) {  
    s.state = 'Conectado';  
    s.id = data.id;  
};  
  
s.$on('idReady', socketReady);
```

# \$apply

Para que Angular note los cambios (y actualice las variables correspondientes) es necesario avisar al scope, esto se hace mediante `$apply(fn)`.

Es particularmente importante cuando se usan servicios y APIs externos.

# \$apply

```
var handleOrientation = function (ev) {  
    s.$apply(function() {  
        s.orientation.absolute = ev.absolute;  
        s.orientation.alpha = ev.alpha;  
        s.orientation.beta = ev.beta;  
        s.orientation.gamma = ev.gamma;  
    });  
};  
  
window.addEventListener('deviceorientation', handleOrientation);
```

# \$apply

```
var onDataFn = function(data) {  
    // We are currently ignoring position data  
    ThreeModel.setMarkerPosition(data.id, data.orientation);  
  
    s.$apply(function () {  
        s.refreshId = data.id;  
    });  
};  
  
ws.setDeviceDataFn(onDataFn);
```

# Gracias

Críticas, sugerencias y comentarios en:  
[migsar.navarro@gmail.com](mailto:migsar.navarro@gmail.com)