

# AureliaJS

Jesse Javier Cogollo Alvarez

Developer by passion

*email: [cogollo87@gmail.com](mailto:cogollo87@gmail.com)*

*MedellinJS*

August 10, 2015

# Contenido

AureliaJS

DEMO

## Que es Aurelia?

Aurelia is a next generation JavaScript client framework that leverages simple conventions to empower your creativity.

<http://aurelia.io/>

Rob Eisenberg

[robeisenberg.com](http://robeisenberg.com)

# Características

## 1. Forward-thinking

Escrito con ES6 y ES7. se integra con web components. no tiene dependencias externas (excepto polyfills).

# Características

1. **Forward-thinking**
2. **Modern Architecture**

Aurelia esta compuesto por pequenos modulos. Utiliados juntos como un completo framework. Pero tambien permite personalizarlo con diferentes modulos.

# Características

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**

Aurelia tiene una poderosa two-way binding. el cual utiliza tecnicas adaptativas para utilizar el mecanismo mas eficiente para observar cada propiedad.  
Object.observer

# Características

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**

La extensibilidad HTML de Aurelia permite crear elementos HTML personalizados, añadir nuevos comportamientos para elementos ya existentes.

# Características

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**
5. **Routing and UI Composition**

Router con acoplamiento activo, patrones de ruta dinamica, activacion de pantallas asincronicamente, etc.



# Características

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**
5. **Routing and UI Composition**
6. **MV\* with Conventions**

Quieres invertir tiempo en escribir código de configuración? Aurelia maneja una convención simple.

## Características

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**
5. **Routing and UI Composition**
6. **MV\* with Conventions**
7. **Broad Language Support**

Utiliza ES5, ES6, TypeScript, AtScript, CoffeeScript.

## Características

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**
5. **Routing and UI Composition**
6. **MV\* with Conventions**
7. **Broad Language Support**
8. **Testable**

combinando modulos ES6 con contenedor inyeccion de dependencias. se hace mas facil crear codigo altamente cohesivo y bajo acoplamiento. haciendo las pruebas unitarias algo facil.

## Como empezar - Estructura

Descargamos la ultima version del skeleton de aurelia.

<https://github.com/aurelia/skeleton-navigation/releases/tag/0.15.1>

descomprimos y ya tenemos una estructura lista para utilizar:

- build (configuracion gulp).
- doc (documentacion).
- src (codificacion).
- styles (estilos CSS).
- test (pruebas).
- README.md (instrucciones para utilizar AureliaJS).

# Modulos

- **logging**

Esta libreria es parte de la plataforma aurelia y contiene un "Appender" para el logging.

uso: welcome.js

# Modulos

- **Plugins**

Aurelia permite crear e implementar plugins que se adaptan a Aurelia. Estos plugins se configuran en: `index.html` uso: `animation-main.js`

# Vistas y vistas modelos

- **DI inyeccion de dependencias**

Los AureliaElements son creados como clases los cuales son instanciados por el framework usando contenedor de inyeccion de dependencias.

uso: flickr.js

## Vistas y vistas modelos

El motor de plantillas de aurelia es el encargado de cargar las vistas y sus recursos, compilando su HTML para un optimo desempeno y renderizando el UI en la pantalla.

- **Plantillas**

`¡template¿` `¡/template¿`. Todo lo que este adentro de esas etiquetas sera administrado por aurelia.

uso: `welcome.html`



# Databinding

El Databinding nos permite enlazar el estado y comportamiento en un objeto(js) y una vista(html).

- **Que es**

Cualquier cambio es enlazado y/o sincronizado en una o varias direcciones. Se puede diferenciar en la vista por que el elemento va seguido de un punto en el atributo con el valor (bind, one-way, two-way o one-time).

implementacin: welcome.html

# Bindings

- **delegate, trigger and call**

TODO

# Bindings

- **string interpolation**

TODO

# Bindings

- **ref**

TODO

# Bindings

- **select elements**

TODO

# Bindings

- **radios**

TODO

# Bindings

- **checkboxes**

TODO

# Bindings

- **InnerHTML**

TODO



# Bindings

- **textContent**

TODO

# Bindings

- **style**

TODO

# Bindings

- **adaptive binding**

TODO

# Redes sociales

## 1. Meetup

[/MongoDB-Medellin](#)

<http://goo.gl/fw5Gyh>

# Redes sociales

1. Meetup
2. **Twitter**

@jessecogollo

<http://goo.gl/gdCAjF>

# Redes sociales

1. Meetup

2. Twitter

[/jessecogollo](#)

3. **Facebook**

<http://goo.gl/Q1JnXQ>

# Redes sociales

1. Meetup

2. Twitter

3. Facebook

4. Google Plus

5. **Lista de correo**

+ correo

<http://goo.gl/FJvrjT>

## Redes sociales

1. Meetup
2. Twitter
3. Facebook
4. Google Plus
5. Lista de correo
6. **Grupo de estudio**

Formulario grupo de estudio

<http://goo.gl/7ALdst>



# Donde aprender

## 1. Organizar un coding dojo

<http://johannesbrodwall.com/2011/12/18/how-to-start-a-coding-dojos/>

# Donde aprender

1. Organizar un coding dojo
2. **Cyber dojo**

<http://cyber-dojo.org/>

## Donde aprender

1. Organizar un coding dojo
2. Cyber dojo
3. **Codingdojo**

<http://www.codingdojo.org/>

## Donde aprender

1. Organizar un coding dojo
2. Cyber dojo
3. Codingdojo
4. **Ejercicios**

[http://rosettacode.org/  
wiki/Category:  
Programming\\_Tasks](http://rosettacode.org/wiki/Category:Programming_Tasks)

## Donde aprender

1. Organizar un coding dojo
2. Cyber dojo
3. Codingdojo
4. Ejercicios
5. **Mas ejercicios**

[http:  
//brendan.enrick.com/post/  
Coding-Katas-and-Exercises](http://brendan.enrick.com/post/Coding-Katas-and-Exercises)

# Preguntas

# Empecemos...

Gracias !!! =)