



Protractor



KARMA

stylus



ASP.NET Core



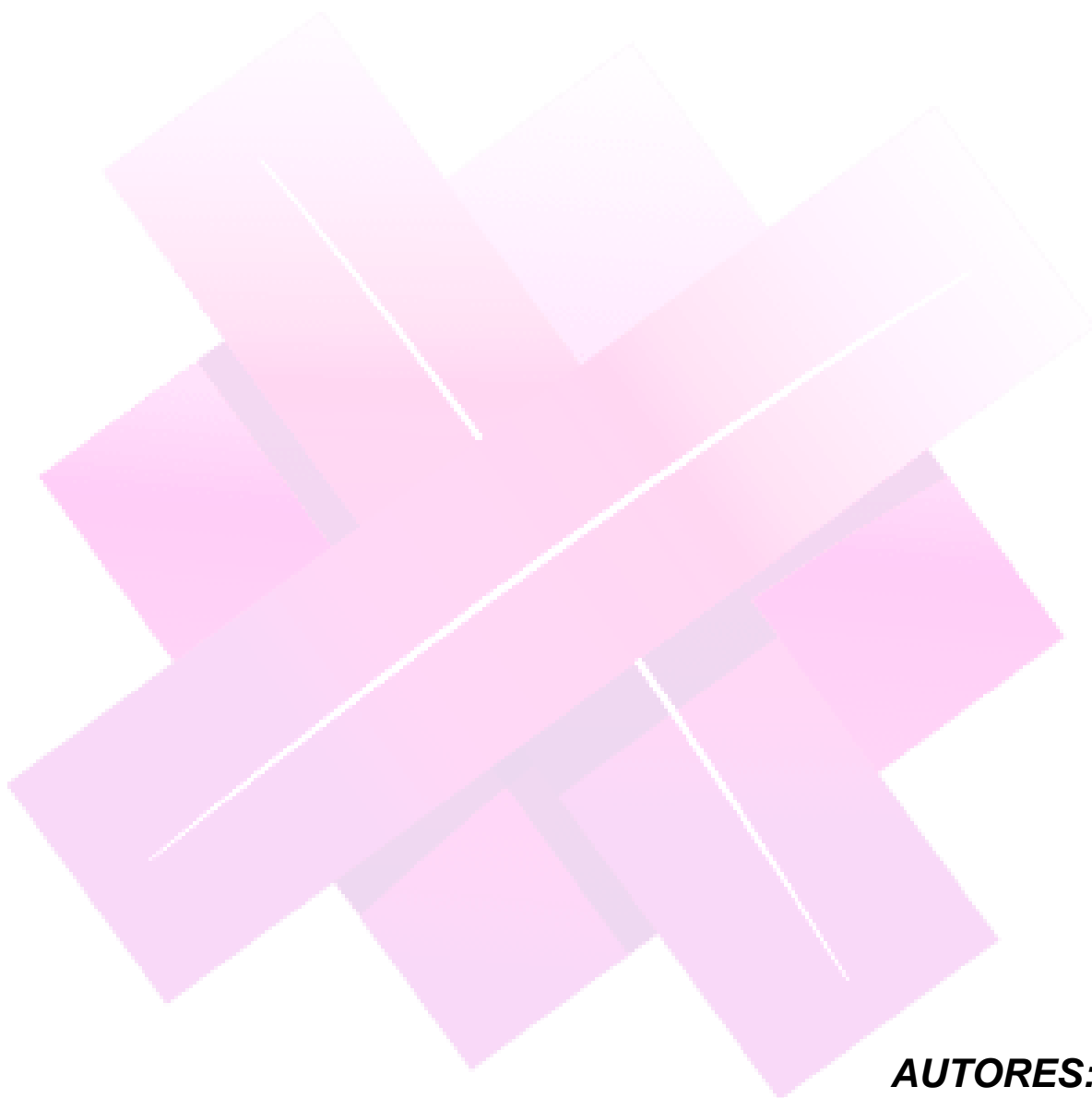
webpack

Sass



aurelia

MANEJANDO AURELIA



AUTORES:

SERGIO STIVES BARRIOS BUITRAGO 1151747
LEYNER JOSE ORTEGA ARIAS 1151725

TABLA CONTENIDO

INTRODUCCION.....	4
AURELIA.....	5
¿QUE ES AURELIA?	5
¿POR QUE ELIGIR AURELIA?	6
CARACTERISTICAS DE AURELIA.....	9
VENTAJAS AL USAR AURELIA	10
ENTORNO PRERREQUISITOS E INSTALACION	11
ENTORNO	11
PRERREQUISITOS	11
INSTALACION DE LA CLI	13
CREANDO UN NUEVO PROYECTO AURELIA	15
RUTA DE NUESTRO PRIMER PROYECTO AURELIA	16
ESTRUCTURA DE UN PROYECTO AURELIA.....	18
.vscode	18
aurelia_project	18
node_modules.....	19
src	19
static.....	20
test	20
otros.....	20
EJECUTANDO TU APLICACION EN AURELIA	22
MI APLICACIÓN CON AURELIA.....	24
¿COMO DESINSTALAR AURELIA?	35
¿QUE PASA SI NECESITO AYUDA?	36
CONCLUSIONES.....	37
REFERENCIAS BIBLIOGRAFICAS.....	38

INTRODUCCION

En el siguiente documento les vamos a presentar un framework de java Script que no es muy conocido en el mercado del desarrollo web, pero que cada vez toma más relevancia para los desarrolladores, pues aurelia es un framework que nos permite crear aplicaciones para móvil escritorio y la web que aprovecha las convenciones simples para potenciar su creatividad. En el desarrollo del contenido las vamos a hacer un manual sobre como instalarlo y crear una aplicación paso a paso y de fácil comprensión.



AURELIA

¿QUE ES AURELIA?

Aurelia es una **colección de módulos modernos de JavaScript** *que*, cuando se usan juntos, funcionan como **una plataforma poderosa para crear aplicaciones** de navegador, de escritorio y móviles, todas de código abierto y basadas en estándares web abiertos.

Además, nos permite crear increíbles interfaces de usuario no solo eso, sino que lo hace de una manera que es fácil de mantener, testear y escalar. Aurelia es como si fuese una variante de angular, pero más sencilla de entender

Una colección de módulos modernos de JavaScript

En lugar de ser un marco monolítico, Aurelia se divide en una colección de módulos orientados a funciones. Los ejemplos de módulos de funciones incluyen metadatos, inyección de dependencia, enlace, plantillas, enrutador y más. Cada módulo se escribe utilizando ECMAScript (también conocido como JavaScript) o TypeScript (un súper conjunto estricto de JavaScript que agrega la verificación de tipos en tiempo de compilación). Muchos de estos módulos se pueden usar individualmente en cualquier tipo de proyecto de JavaScript, incluido Node.js.

Una potente plataforma para crear aplicaciones

Si bien los módulos de Aurelia se pueden usar para muchos propósitos, su verdadero poder radica en usarlos juntos como una plataforma de aplicaciones front-end. Si su objetivo final es crear experiencias ricas y atractivas para sus clientes, cumpliendo o superando lo que se encuentra en las aplicaciones nativas modernas, Aurelia proporciona los medios. A través de su rico modelo de componentes, composición dinámica de UI, enrutamiento y un extenso conjunto de complementos, Aurelia proporciona un conjunto integral de capacidades y herramientas para construir cualquier experiencia de usuario que pueda imaginar, ya sea que esté apuntando al navegador, al móvil o al escritorio.

Fuente abierta

Con toda la potencia y la capacidad que ofrece Aurelia, puede esperar que tenga un modelo de licencia costoso o que sea de código cerrado, pero tampoco es cierto. Aurelia es gratuita y su código es de código

abierto bajo la Licencia MIT , una licencia muy permisiva utilizada por muchos proyectos web populares en la actualidad.

¿POR QUE ELIGIR AURELIA?

Hay muchos tipos de framework para elegir hoy en día. Pero creemos que Aurelia aporta un enfoque fresco y emocionante al desarrollo de aplicaciones para el usuario con una potencia y flexibilidad que no tiene comparación con otros framework.

Aunque reconocemos que cada proyecto y cada equipo tienen necesidades diferentes. Pero es posible que Aurelia sea la opción correcta para un proyecto amigable, flexible y que sea cómodo al momento de realizarlo.

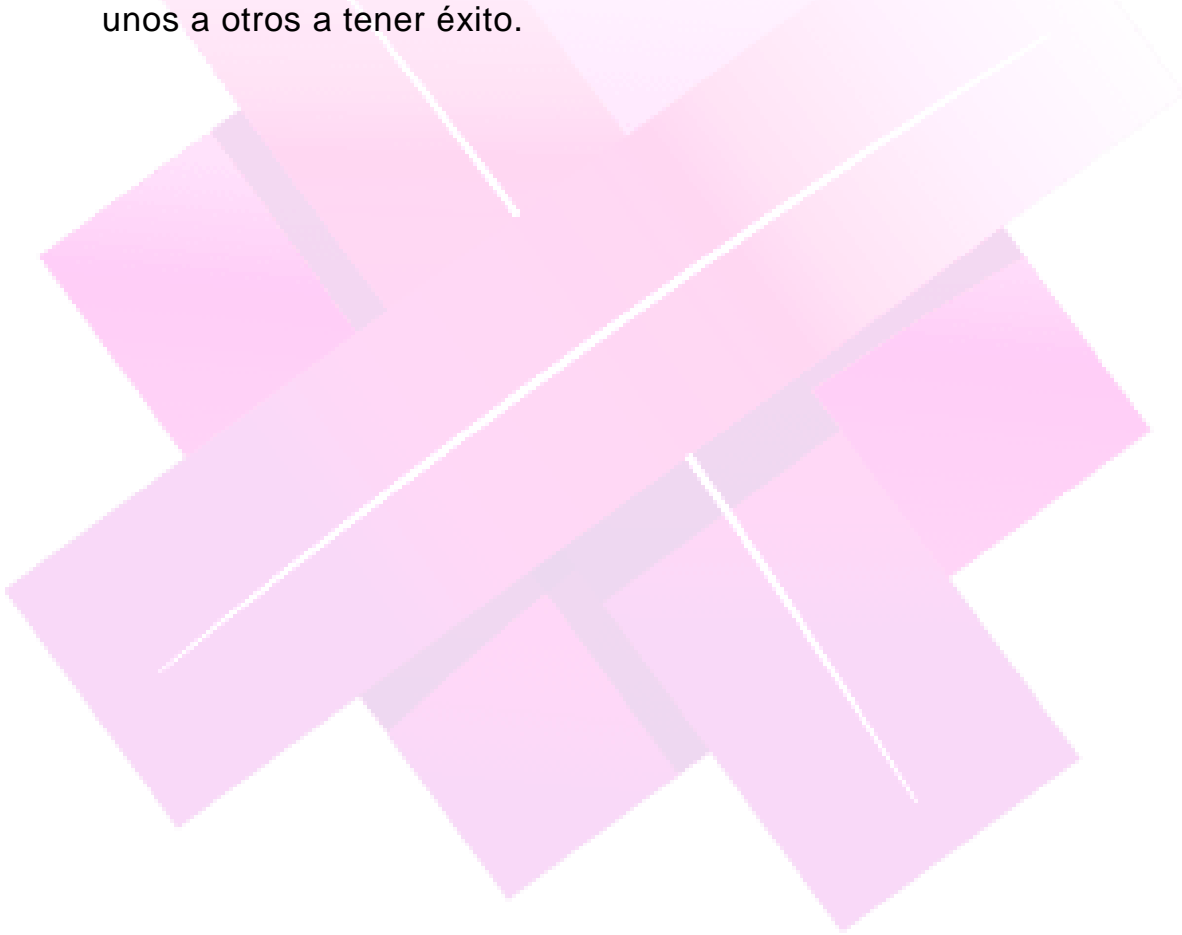
Es posible que Aurelia sea la opción correcta para ti si...

- **Desea una solución todo en uno:** Aurelia proporciona capacidades básicas como inyección de dependencias, plantillas, enrutamiento y publicación / sub, por lo que no tiene que juntar un montón de bibliotecas para crear una aplicación. Además de este rico núcleo, Aurelia también proporciona una serie de complementos adicionales para la internacionalización, validación, diálogos modales, virtualización de IU y mucho más. Tampoco tienes que improvisar un montón de herramientas diferentes. Aurelia proporciona un CLI para generar y crear proyectos, un complemento de navegador para la depuración y un complemento de Código de VS también. Sin embargo, no está obligado a usar ninguno de estos, ya que Aurelia está estructurada para permitirle intercambiar cualquier detalle, incluso hasta el motor de plantillas / encuadernación, para garantizar la máxima flexibilidad.
- **Necesita una velocidad de renderización deslumbrante y una gran eficiencia de memoria:** en los puntos de referencia de terceros como DB Monster, Aurelia rinde más rápido que cualquier otro framework hoy en día. Debido a su representación por lotes y agrupación de objetos observables, Aurelia también utiliza menos memoria y causa menos pérdida de GC que otros marcos.
- **Necesita la seguridad del flujo de datos unidireccional, pero necesita la productividad del enlace de datos:** Aurelia cuenta con un sistema de enlace basado en observaciones que utiliza el flujo

de datos unidireccional de forma predeterminada, lo que coloca los datos de su modelo en su vista a través de un Muy eficiente, mecanismo de procesamiento de lotes DOM. El enlace bidireccional también puede aprovecharse para los controles de formulario HTML, lo que permite aumentar la productividad del desarrollador, sin sacrificar la seguridad del flujo unidireccional o la de la encapsulación de componentes.

- **Deseas la estabilidad de la API en medio de un paisaje turbulento de JavaScript:** Aurelia sigue a Semver y trabaja arduamente para no hacer cambios importantes en sus API. Nos enorgullece decir que hemos seguido innovando y avanzando en la plataforma sin que se hayan producido cambios importantes en las API del marco central desde nuestra versión 1.0 el 27 de julio de 2016.
- **Usted valora los altos estándares de cumplimiento:** centrada en los componentes web ES2015 + y W3C mientras evita las abstracciones innecesarias, Aurelia proporciona el modelo de componentes más limpio y compatible con los estándares que encontrará en cualquier lugar.
- **Piensa que un marco debería "salirse de su camino":** Aurelia es el único marco que le permite crear componentes con JavaScript / TypeScript simple y de vainilla. El marco se mantiene fuera de su camino para que su código permanezca limpio y fácil de evolucionar con el tiempo.
- **A usted le gustan los modelos de programación que son fáciles de aprender y recordar:** debido a su diseño simple y consistente, los desarrolladores pueden aprender un conjunto muy pequeño de patrones y APIs de Aurelia a la vez que desbloquean posibilidades ilimitadas. Las convenciones simples ayudan a los desarrolladores a seguir patrones sólidos y reducir la cantidad de código que tienen que escribir y mantener. Todo esto se traduce en menos problemas con el marco y más enfoque en la aplicación.
- **Prefiere una plataforma que se integre bien con otros marcos y bibliotecas.** Debido al diseño extensible de Aurelia y su estricto cumplimiento con los estándares web, es fácil integrar Aurelia con cualquier biblioteca o marco de terceros, incluidos [jQuery](#), [React](#), [Polymer](#), [Bootstrap](#), [Materialize](#), CSS y muchos más.

- **Usted ama o quiere ser parte del código abierto:** Aurelia es de código abierto bajo la licencia MIT y no agrega ni elimina cláusulas o condiciones especiales de la licencia. Estamos orgullosos del trabajo que nuestra comunidad ha realizado en conjunto y nos encantaría que participe y nos ayude a mejorar Aurelia para todos.
- **Le encanta ser parte de una comunidad acogedora:** con uno de los canales de desarrolladores más grandes y más activos, un gran número de colaboradores y un gran equipo central activo, Aurelia tiene una comunidad increíble. A nuestro equipo central y a nuestra comunidad les encanta dar la bienvenida a los nuevos desarrolladores y todos trabajamos arduamente para ayudarnos unos a otros a tener éxito.



CARACTERISTICAS DE AURELIA

- I. Forward-thinking, ya que está escrito en EcmaScript y que no tiene dependencias externas.
- II. Modern Architecture, debido a que en lugar de una arquitectura monolítica Aurelia se compone de pequeños módulos.
- III. Extensible HTML. Te permite crear elementos customizados o añadirles a elementos existentes atributos customizados.
- IV. Routing & UI Composition. Facilita el Routing a través de un pipeline específico que tienen creado, lo que simplifica esta tarea mucho.
- V. MV* with Conventions. Utiliza convenciones para facilitar la creación de arquitecturas MVC o MVVM
- VI. Broad Language Support: ES5, ES 2015, ES 2016 and TypeScript.
- VII. Comprobable: Al combinar módulos ES 2015 con un contenedor de inyección de dependencias muy potente, facilitan la creación de código poco acoplado y, por ende, el testing unitario.

VENTAJAS AL USAR AURELIA

- I. You want an all-in-one solution- Aurelia te proporciona ya elementos como la inyección de dependencias, templates, enrutamiento, etc... Que te permiten crear una aplicación rápidamente sin preocuparte de librerías externas. También tiene plugins para internalización, validación, ventanas modales y más cosas.
- II. Es un framework con una curva de aprendizaje muy corta y además es muy fácil de integrar con otras librerías o frameworks. También te permite realizar aplicaciones con Vainilla JavaScript.
- III. Tiene lo que denominan una data binding adaptativo.
- IV. Tiene una alta velocidad de renderizado y una gran eficiencia de memoria.
- V. Está compuesto por módulos.

ENTORNO PRERREQUISITOS E INSTALACION

ENTORNO

- El framework Aurelia se puede utilizar en cualquier entorno con sistema operativo Windows, Linux y Mac
- El framework funciona con Node.js, npm, gulp como servidor de JavaScript y se puede visualizar en un navegador nuevo

PRERREQUISITOS

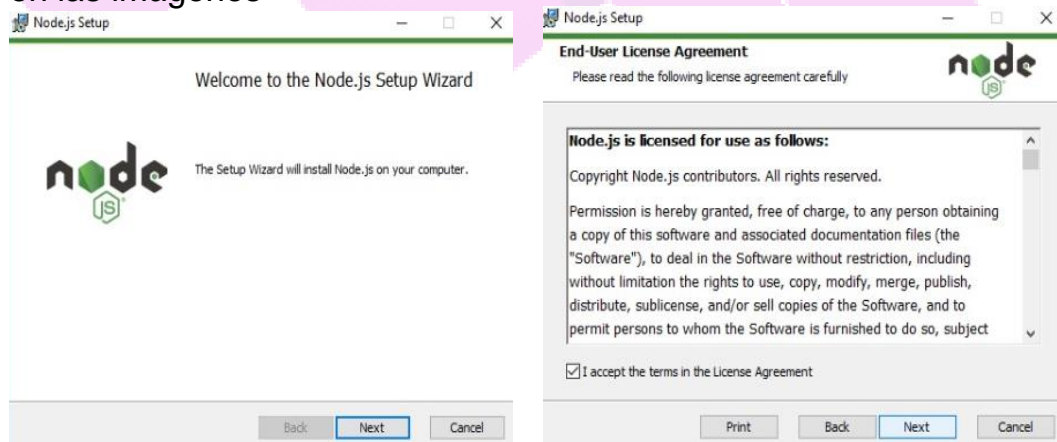
Aurelia CLI es una aplicación basada en [Node.js](#), así que asegúrese de instalarla antes de continuar. Necesitará Node.js 4.4.7 o posterior.

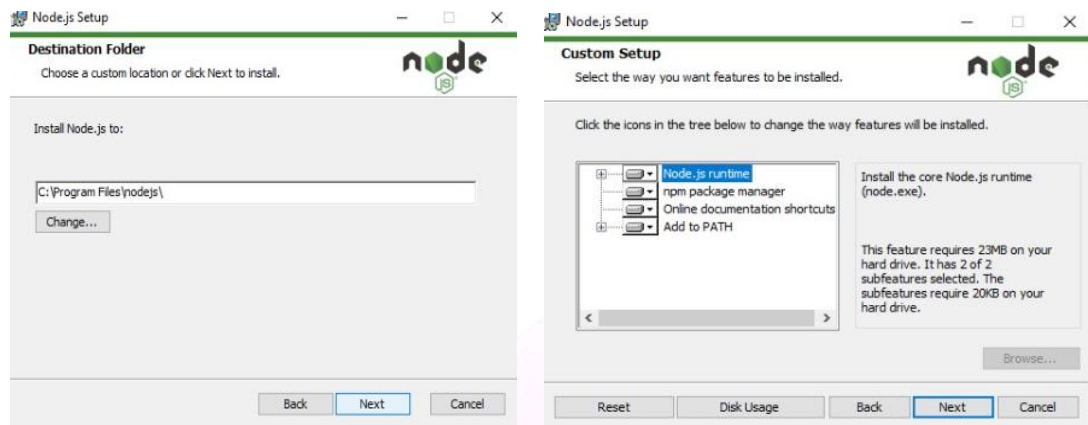
Si no tienes instalado Node.js, estos son los pasos:

1. Descargar [Node.js](#) en la página oficial, puedes descargar la versión que usted quiera en este caso use la recomendada por la mayoría.

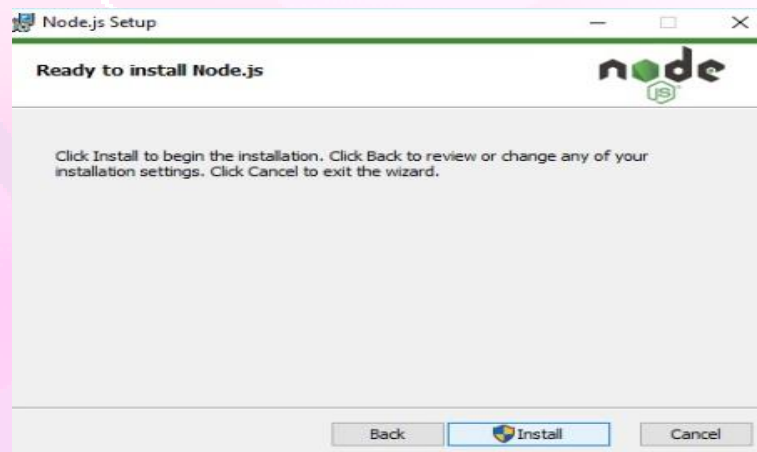


2. Ejecutamos el archivo descargado y seguimos los pasos mostrados en las imágenes





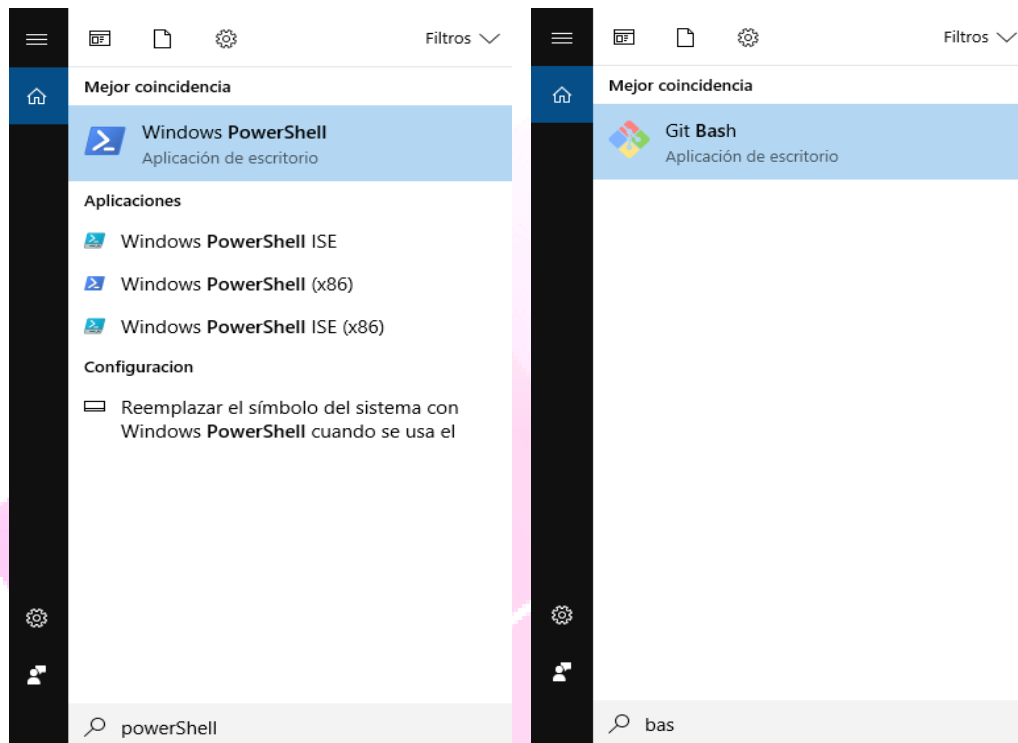
3. Instalamos



También necesitarás un cliente Git instalado. Algunas excelentes opciones incluyen: [Github Desktop](#) , [Sourcetree](#) y [Git SCM](#).

INSTALACION DE LA CLI

Siempre que haya instalado Node.js y Npm correctamente, abra una Terminal de comando / **PowerShell** o **Bash**.



Instale la CLI globalmente con el siguiente comando:

	JavaScript
<pre>npm install aurelia-cli -g</pre>	

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Sergio Barrios> npm install aurelia-cli -g
```

Comenzará a instalarse el CLI, espera que se termine de instalar.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Sergio Barrios> npm install aurelia-cli -g
[.....] | fetchMetadata: sill locate-path@2.0.0 checking installable status
```

Ejecutamos el comando `au -v` para asegurarse de que Aurelia CLI se haya instalado correctamente.

Debería mostrarnos la versión de **Aurelia CLI** que tenemos instalada si nos muestra la versión de **aurelia-cli** es porque ya fue instalada con éxito y si no tienes que volver hacer los pasos anteriores.

```
PS C:\Users\Sergio Barrios> au -v
Global aurelia-cli v1.0.0-beta.13
PS C:\Users\Sergio Barrios>
```

CREANDO UN NUEVO PROYECTO AURELIA

Ahora que tiene la CLI instalada, para crear un nuevo proyecto, ejecute el siguiente comando y siga las instrucciones en pantalla:

au new

```
PS C:\Users\Sergio Barrios> au new
Global aurelia-cli v1.0.0-beta.13
aurelia CLI
Please enter a name for your new project below.
[aurelia-app]>
```

Después le vamos a dar nombre a nuestro proyecto en este caso yo la llame **aurelia2.0** usted le puede colocar el nombre que usted quiera.

```
Please enter a name for your new project below.
[aurelia-app]> aurelia2.0
```

Podrás elegir entre diferentes formatos y cargadores, para simplificar las cosas, simplemente seleccione los valores predeterminados. En este caso voy a seleccionar los valores predeterminados, a medida que se familiarice con la CLI, puede configurar estas opciones para que se ajusten a sus necesidades.

Para poder seleccionar los valores predeterminados vamos a digitar en las tres ventanas que nos va a salir la opción 1, como se muestra en las imágenes anexadas.

```
[aurelia-app]> aurelia2.0
Would you like to use the default setup or customize your choices?
1. Default ESNext (Default)
   A basic web-oriented setup with Babel and Webpack for modern JavaScript development.
2. Default TypeScript
   A basic web-oriented setup with TypeScript and Webpack for modern JavaScript development.
3. Custom
   Select bundlers (built-in or webpack), loaders (requirejs or systemjs), transpilers, CSS pre-processors and more.
[Default ESNext]> 1
```



```

Project Configuration
Name: aurelia2.0
Platform: Web
Bundler: Webpack
Loader: None
Transpiler: Babel
Markup Processor: Minimal Minification
CSS Processor: None
Unit Test Runner: Jest
Integration Test Runner: None
Editor: Visual Studio Code
Features: None

Would you like to create this project?
1. Yes (Default)
   Creates the project structure based on your selections.
2. Restart
   Restarts the wizard, allowing you to make different selections.
3. Abort
   Aborts the new project wizard.

[Yes]> 1

Project structure created and configured.

Would you like to install the project dependencies?
1. Yes (Default)
   Installs all server, client and tooling dependencies needed to build the project.
2. No
   Completes the new project wizard without installing dependencies.

[Yes]> 1

```

Espera hasta que se termine de instalar y después ya tendrás tu primer proyecto creado con **Aurelia CLI**

```
Installing project dependencies.
```

RUTA DE NUESTRO PRIMER PROYECTO AURELIA

Para conocer la ruta donde está creado nuestro proyecto debemos escribir el comando

```
cd (nombre que le diste a la aplicación sin los paréntesis)
```

En mi caso sería:

```
cd aurelia2.0
```

```
PS C:\Users\Sergio Barrios> cd aurelia2.0
```

Después que estemos parados en nuestro proyecto debemos digitar el comando:

```
ls
```

```
PS C:\Users\Sergio Barrios\aurelia2.0> ls

Directorio: C:\Users\Sergio Barrios\aurelia2.0

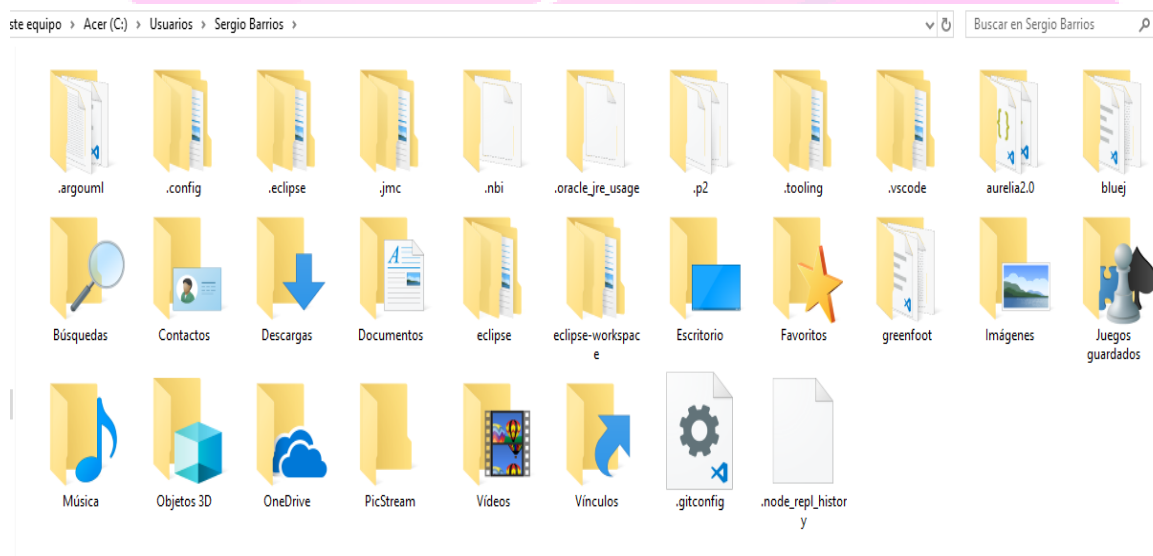
Mode                LastWriteTime         Length Name
----                -
d-----          15/03/2019   1:11 p.m.         .vscode
d-----          15/03/2019   1:11 p.m.       aurelia_project
d-----          15/03/2019   1:11 p.m.         dist
d-----          15/03/2019   1:19 p.m.       node_modules
d-----          15/03/2019   1:11 p.m.         src
d-----          15/03/2019   1:11 p.m.        static
d-----          15/03/2019   1:11 p.m.         test
-a-----          15/03/2019   1:11 p.m.         751 .babelrc.js
-a-----          15/03/2019   1:11 p.m.         257 .editorconfig
-a-----          15/03/2019   1:11 p.m.         148 .eslinttrc.json
-a-----          15/03/2019   1:11 p.m.        1037 .gitignore
-a-----          15/03/2019   1:11 p.m.       15086 favicon.ico
-a-----          15/03/2019   1:11 p.m.         390 index.ejs
-a-----          15/03/2019   1:11 p.m.         171 jsconfig.json
-a-----          15/03/2019   1:19 p.m.      487079 package-lock.json
-a-----          15/03/2019   1:19 p.m.        2689 package.json
-a-----          15/03/2019   1:11 p.m.        8133 webpack.config.js

PS C:\Users\Sergio Barrios\aurelia2.0>
```

Y nuestro proyecto está creada en la dirección:

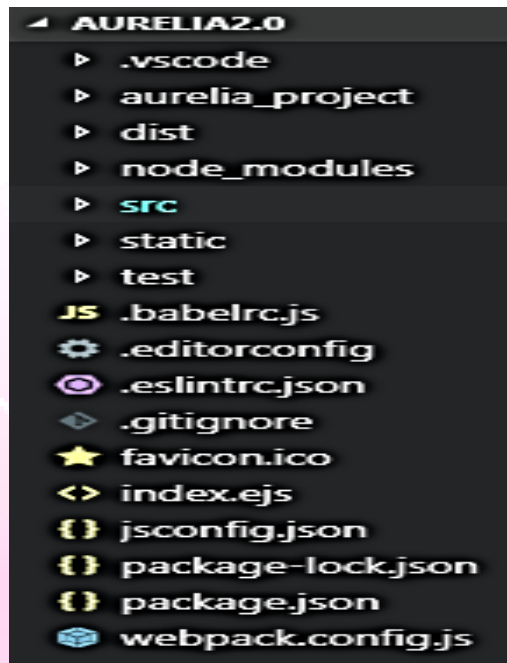
Directorio: C:\Users\Sergio Barrios\aurelia2.0

Podemos abrir nuestro Explorador de archivo y confirmar si nuestro proyecto está en esa dirección.

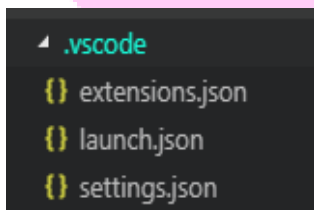


Ya podemos abrir nuestro proyecto con cualquier editor de texto en mi caso voy a utilizar **Visual Studio Code**.

ESTRUCTURA DE UN PROYECTO AURELIA

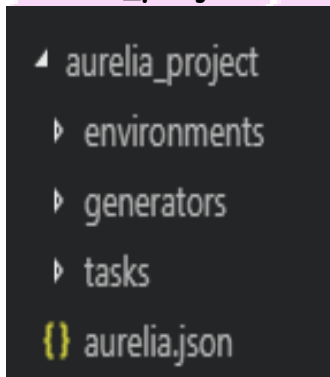


.vscode



.vscode: es una extensión con la cual se espera brindar una experiencia de desarrollo agradable al crear aplicaciones Aurelia utilizando [Visual Studio Code](#).

aurelia_project



aurelia_project: Este directorio contiene lo relacionado con la construcción y empaquetado de la aplicación mediante el CLI

environments: Viene preparada por defecto para desarrollo, stage y producción

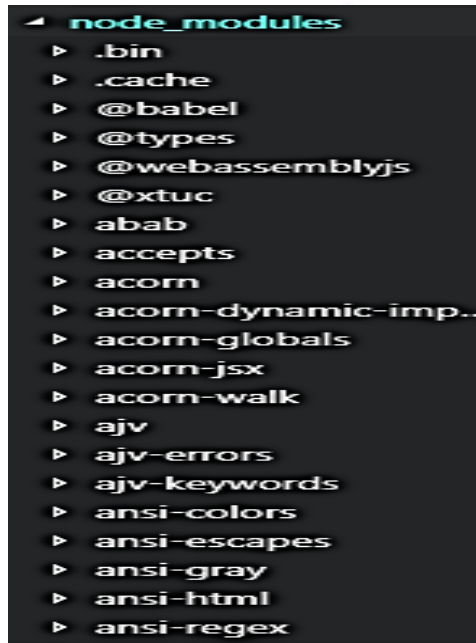
generators: : La ejecución **au generate** <resource> ejecuta un generador para armar estructuras típicas de Aurelia. Las opciones para los recursos son: element, attribute, value-converter, binding-behavior, tasky generator.

tasks: El CLI de Aurelia le permite ejecutar tareas de Gulp, y se proporcionan algunas tareas de manera inmediata que le ayudarán a comenzar. Para cualquier proyecto nuevo,

necesita una tarea para compilar, ejecutar y probar la aplicación, por lo que la CLI le proporciona estas tareas. Estas tareas se pueden encontrar en el **aurelia_project/tasksdirectorio**.

aurelia.json: Este archivo es la configuración centralizada para todas las tareas trilladas como compilación y prueba.

node_modules



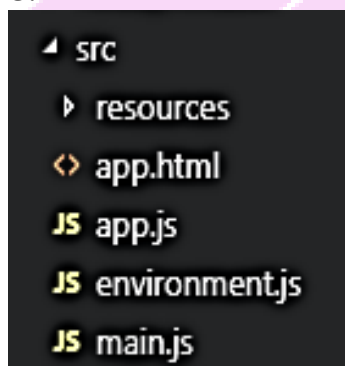
node_modules: Proporciona paquetes npm a todo el espacio de trabajo.

Node Package Manager o simplemente npm es un gestor de paquetes, el cual hará más fáciles nuestras vidas al momento de trabajar con Node, ya que gracias a él podremos tener cualquier librería disponible con solo una línea de código, npm nos ayudará a administrar nuestros módulos, distribuir paquetes y agregar dependencias de una manera sencilla.

Cuando instalamos nuevos paquetes lo que hace npm es instalarlo de manera local en nuestro proyecto dentro de la carpeta

node_modules, pero nosotros podemos decirle que lo instale de manera global de ser necesario.

src



src: aquí es donde se va a trabajar

resources: ponga sus imágenes / fuentes / iconos aquí.

app.js: El modelo de vista predeterminado

app.html: la vista predeterminada para una aplicación de Aurelia. El punto de entrada principal

environment.js: Contiene opciones de configuración de compilación para entornos de destino particulares. Por defecto, hay un entorno de desarrollo estándar

sin nombre y un entorno de producción ("prod"). Puede definir configuraciones de entorno de destino adicionales.

main.js: El principal punto de entrada para tu aplicación. Compila la aplicación con el compilador JIT y arranca el módulo raíz de la aplicación (AppModule) para ejecutarse en el navegador. También puede usar el compilador AOT sin cambiar ningún código agregando la `--aot` bandera a la CLI buildy los servecomandos.

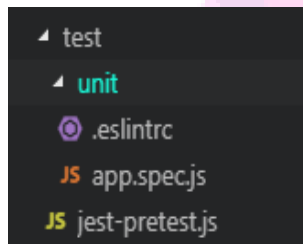
static



static: La palabra clave **static** define un método estático para una clase, Los métodos estáticos son llamados sin instanciar su clase. Se usan para crear funciones para una aplicación.

favicon.ico: Es un pequeño icono (normalmente de 16x16 píxeles o de 32x32 px) que se utiliza para identificar un sitio web en las pestañas de un navegador, en un listado de favoritos, o en cualquier otra parte que requiera una identificación pequeña.

test

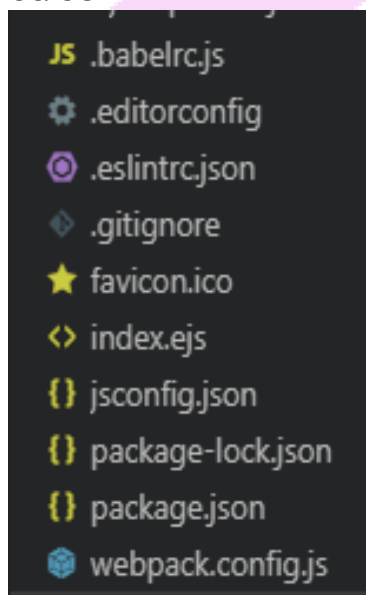


test: El directorio que contiene las aplicaciones de pruebas automatizadas. El punto de entrada principal para sus pruebas unitarias, con algunas configuraciones específicas de Angular. Normalmente no es necesario editar este archivo.

Unit: unidad de pruebas

app.spec.js: Define una prueba unitaria para la raíz AppComponent.

otros



JS .babelrc.js: el archivo de configuración para babel, que utiliza el CLI para trasladar nuestro código de ESNEXT de aplicaciones en ES5 por lo que la mayoría de los navegadores puede ejecutar **.editorconfig:** ayuda a mantener estilos de codificación consistentes para múltiples desarrolladores que trabajan en el mismo proyecto en varios editores e IDE. El proyecto EditorConfig consiste en un formato de archivo para definir estilos de codificación y una colección de complementos de editor de texto que permiten a los editores leer el formato de archivo y adherirse a los estilos definidos. Los archivos EditorConfig son fáciles de leer y funcionan bien con los sistemas de control de versiones.

.eslintrc.json: se usa para definir la estructura de configuración. Los archivos JSON de ESLint también permiten comentarios de estilo JavaScript.

.gitignore: Los archivos ignorados se rastrean en un archivo especial llamado .gitignore que se registra en la raíz de su repositorio. No hay un comando explícito de git ignore: en su lugar, el .gitignore archivo debe editarse y confirmarse a mano cuando tiene archivos nuevos que desea ignorar. .gitignore los archivos contienen patrones que coinciden con los nombres de los archivos en su repositorio para determinar si deben ignorarse o no.

favicon.ico: Un icono para usar para esta aplicación en la barra de marcadores.

index.ejs: La página HTML principal que se sirve cuando alguien visita su sitio. La CLI agrega automáticamente todos los archivos JavaScript y CSS al crear su aplicación, por lo que normalmente no necesita agregar ninguna etiqueta <script> o <link> etiquetas aquí manualmente.

package-lock.json: Proporciona información de versión para todos los paquetes instalados node_modules por el cliente npm. Consulte la documentación de npm para más detalles. Si usa el cliente de hilo, este archivo será yarn.lock en su lugar.

package.json: Configura las dependencias de paquetes npm que están disponibles para todos los proyectos en el área de trabajo. Consulte la documentación de npm para conocer el formato y los contenidos específicos de este archivo.

webpack.config.js: Fuera de la caja, el paquete web no requiere que uses un archivo de configuración. Sin embargo, asumirá que el punto de entrada de su proyecto es src/index y dará como resultado el resultado en dist/main.js minificación y optimizado para la producción

EJECUTANDO TU APLICACION EN AURELIA

Para ejecutar su aplicación Aurelia, abra una Terminal de comando / **PowerShell** o **Bash** y escribe el comando:

```
cd (nombre aplicación)
```

Después ejecutamos el comando:

```
au run
```

```
PS C:\Users\Sergio Barrios> cd aurelia2.0
PS C:\Users\Sergio Barrios\Aurelia2.0> au run
Local aurelia-cli v1.0.0-beta.13
Starting 'configureEnvironment'...
Finished 'configureEnvironment'
Starting 'runWebpack'...
Project is running at http://localhost:8080
webpack output is served from /
Content not from webpack is served from C:\Users\Sergio Barrios\Aurelia2.0\dist
404s will fallback to /index.html
Finished 'runWebpack'
 @wdm@: Hash: 372d87f9fc76fd13a2f5
Version: webpack 4.29.6
Time: 5671ms
Built at: 2019-03-15 14:33:11

    Asset      Size      Chunks             Chunk Names
app~d0ae3f07.372d87f9fc76fd13a2f5.chunk.js 11.3 KiB    app~d0ae3f07    [emitted]
                                [emitted]
                                [emitted]
                                [emitted]
runtime~app.372d87f9fc76fd13a2f5.bundle.js 6.05 KiB    runtime~app    [emitted]
vendors~01be3b92.372d87f9fc76fd13a2f5.chunk.js 448 KiB    vendors~01be3b92    [emitted] [big] vendors~01be3b92
vendors~02227409.372d87f9fc76fd13a2f5.chunk.js 472 KiB    vendors~02227409    [emitted] [big] vendors~02227409
vendors~253ae210.372d87f9fc76fd13a2f5.chunk.js 496 KiB    vendors~253ae210    [emitted] [big] vendors~253ae210
vendors~41983590.372d87f9fc76fd13a2f5.chunk.js 707 KiB    vendors~41983590    [emitted] [big] vendors~41983590
vendors~50e8d500.372d87f9fc76fd13a2f5.chunk.js 272 KiB    vendors~50e8d500    [emitted] [big] vendors~50e8d500
vendors~5196d596.372d87f9fc76fd13a2f5.chunk.js 46 KiB     vendors~5196d596    [emitted] vendors~5196d596
vendors~556c66f2.372d87f9fc76fd13a2f5.chunk.js 259 KiB    vendors~556c66f2    [emitted] [big] vendors~556c66f2
vendors~5ea1390f.372d87f9fc76fd13a2f5.chunk.js 43.5 KiB   vendors~5ea1390f    [emitted] vendors~5ea1390f
vendors~72fd3f32.372d87f9fc76fd13a2f5.chunk.js 72.6 KiB   vendors~72fd3f32    [emitted] vendors~72fd3f32
vendors~ccfd30e1.372d87f9fc76fd13a2f5.chunk.js 61.2 KiB   vendors~ccfd30e1    [emitted] vendors~ccfd30e1
vendors~db300d2f.372d87f9fc76fd13a2f5.chunk.js 269 KiB    vendors~db300d2f    [emitted] [big] vendors~db300d2f
vendors~de47f514.372d87f9fc76fd13a2f5.chunk.js 61.8 KiB   vendors~de47f514    [emitted] vendors~de47f514
vendors~ec8c427e.372d87f9fc76fd13a2f5.chunk.js 500 KiB    vendors~ec8c427e    [emitted] [big] vendors~ec8c427e
vendors~ecff2e3d.372d87f9fc76fd13a2f5.chunk.js 361 KiB    vendors~ecff2e3d    [emitted] [big] vendors~ecff2e3d
Entrypoint app [big] = runtime~app.372d87f9fc76fd13a2f5.bundle.js vendors~253ae210.372d87f9fc76fd13a2f5.chunk.js vendors~556c66f2.372d87f9fc76fd13a2f5.chunk.js vendors~72fd3f32.372d87f9fc76fd13a2f5.chunk.js vendors~50e8d500.372d87f9fc76fd13a2f5.chunk.js vendors~5ea1390f.372d87f9fc76fd13a2f5.chunk.js vendors~ecff2e3d.372d87f9fc76fd13a2f5.chunk.js vendors~02227409.372d87f9fc76fd13a2f5.chunk.js vendors~01be3b92.372d87f9fc76fd13a2f5.chunk.js vendors~41983590.372d87f9fc76fd13a2f5.chunk.js vendors~db300d2f.372d87f9fc76fd13a2f5.chunk.js vendors~ccfd30e1.372d87f9fc76fd13a2f5.chunk.js vendors~ec8c427e.372d87f9fc76fd13a2f5.chunk.js vendors~de47f514.372d87f9fc76fd13a2f5.chunk.js vendors~5196d596.372d87f9fc76fd13a2f5.chunk.js app~d0ae3f07.372d87f9fc76fd13a2f5.chunk.js
[0] multi aurelia-webpack-plugin/runtime/empty-entry aurelia-webpack-plugin/runtime/pal-loader-entry webpack-dev-server/client aurelia-webpack-plugin/runtime/empty-entry aurelia-webpack-plugin/runtime/pal-loader-entry aurelia-bootstrapper 8
8 bytes {app~d0ae3f07} [built]
[+gfil] (webpack)-dev-server/client/overlay.js 3.59 KiB {vendors~5196d596} [built]
```


Miramos en que puerto tenemos el **localhost**

```
Project is running at http://localhost:8080
```

Copiamos y pegamos la dirección en nuestro navegador web y nos mostrara nuestra primera aplicación “**Hello World**”.

El servidor de desarrollo de CLI estará disponible en <http://localhost:8080>



Hello World!

MI APLICACIÓN CON AURELIA

A continuación, les vamos a mostrar el paso a paso de cómo crear una aplicación en Aurelia de una manera sencilla.

El resultado final es una aplicación que nos va a agregar y eliminar los elementos que nos nosotros queramos con la única condición que estén agregados.

The diagram illustrates the state of an Aurelia application at two different points in time. Both states are represented by a light gray rectangular area containing the text "Manejando Aurelia".

Initial State (Top): Below the header, there is a white input field followed by a dark gray button labeled "Agregar". Below this is a table with two columns: "NOMBRE" and "ELIMINAR".

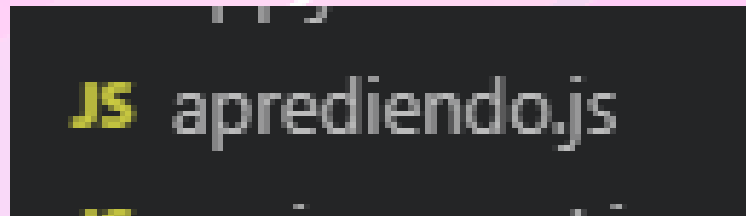
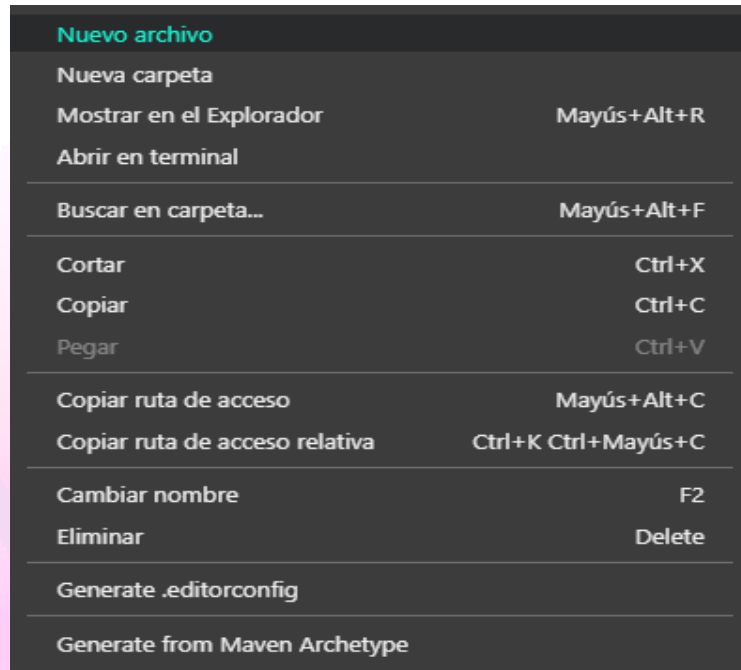
NOMBRE	ELIMINAR
--------	----------

Final State (Bottom): This state is identical to the initial state, but with an additional row added to the table. The first column of this row contains the text "sergio", and the second column contains a dark gray button labeled "Eliminar".

NOMBRE	ELIMINAR
sergio	Eliminar

Pasos:

1. Creamos un archivo en la carpeta **src** con el nombre que se le desea poner por ejemplo “**aprendiendo.js**” de la siguiente manera:



2. En el archivo creado vamos a exportar una clase, creamos un constructor y declaramos las propiedades de la siguiente manera

```
JS aprendiendo.js x
1  export class Aprendiendo {
2      constructor(descripcion) {
3          this.descripcion = descripcion;
4          this.done = false;
5      }
6  }
```

3. Una vez hecho esto lo importamos en nuestra carpeta **app.js**

```
JS app.js x
1  import {Aprendiendo} from './aprendiendo';
2
```

4. A continuación agregamos **this.aprendiendos= [];** y **this.aprendiendoDescripcion="";** a nuestro constructor

```
JS app.js x
1  import {Aprendiendo} from './aprendiendo';
2
3
4  export class App {
5      constructor() {
6          this.message = 'Manejando Aurelia';
7          this.aprendiendos = [];
8          this.aprendiendoDescripcion = "";
9      }
10 }
```

5. Creamos un método para agregar datos le puedes dar cualquier nombre a este método en mi caso lo voy a llamar “**add**”.

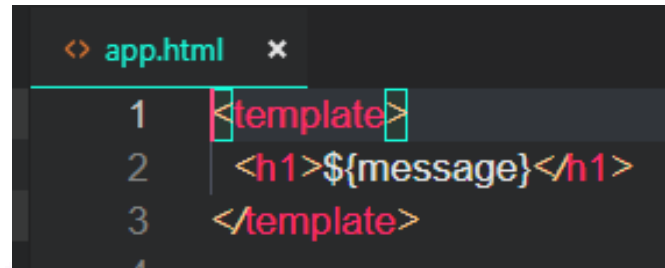
```
JS app.js x
10
11 add() {
12   if (this.aprendiendoDescripcion) {
13     this.aprendiendos.push(new Aprendiendo(this.aprendiendoDescripcion));
14     this.aprendiendoDescripcion = "";
15   }
16 }
```

6. También creamos un método para poder borrar los datos que se insertan, que recibe el elemento que se va a eliminar

```
remove(aprendiendo) {
  let index = this.aprendiendos.indexOf(aprendiendo);
  if (index !== -1) {
    this.aprendiendos.splice(index, 1);
  }
}
}
```

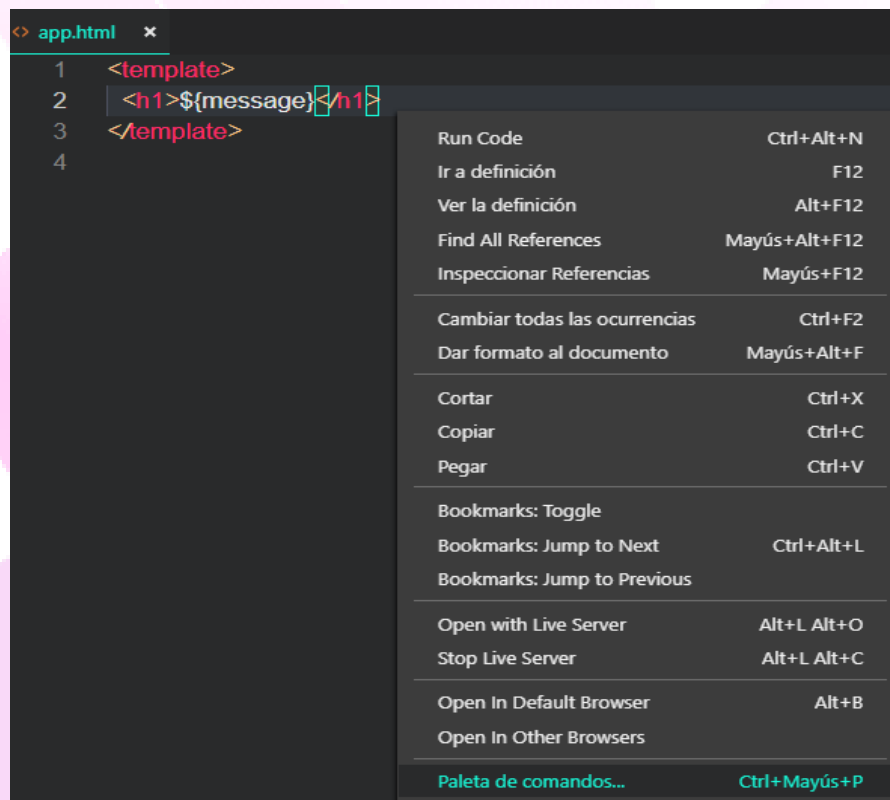
au run --watch periódicamente deja de ver los cambios. Noté esto particularmente con los archivos .html, y más en las rutas de carpeta más profundas. Lamentablemente la reproducción de este problema no es confiable. He hecho un montón de console.log en varias bibliotecas y creo que tengo una causa.

Vamos a ejecutar nuestra aplicación para esto debemos estar en el file app.html



```
<> app.html x
1  <template>
2    <h1>${message}</h1>
3  </template>
```

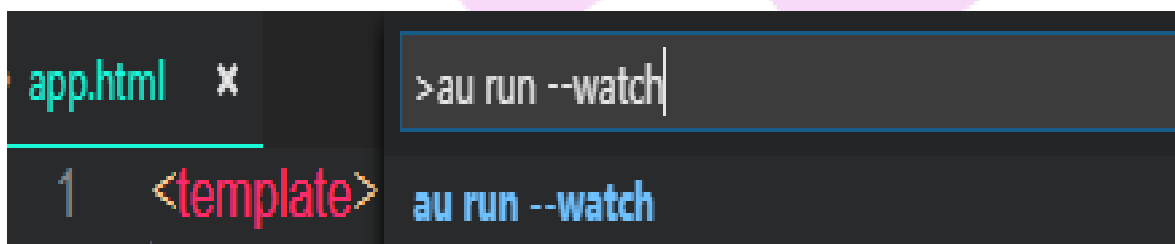
Después presionamos click derecho y seleccionamos paleta de comandos



```
<> app.html x
1  <template>
2    <h1>${message}</h1>
3  </template>
4
```

Run Code	Ctrl+Alt+N
Ir a definición	F12
Ver la definición	Alt+F12
Find All References	Mayús+Alt+F12
Inspeccionar Referencias	Mayús+F12
Cambiar todas las ocurrencias	Ctrl+F2
Dar formato al documento	Mayús+Alt+F
Cortar	Ctrl+X
Copiar	Ctrl+C
Pegar	Ctrl+V
Bookmarks: Toggle	
Bookmarks: Jump to Next	Ctrl+Alt+L
Bookmarks: Jump to Previous	
Open with Live Server	Alt+L Alt+O
Stop Live Server	Alt+L Alt+C
Open In Default Browser	Alt+B
Open In Other Browsers	
Paleta de comandos...	Ctrl+Mayús+P

En la parte superior ejecutamos el comando **au run --watch**



```
<> app.html x
1  <template>
```

>au run --watch

au run --watch

Y por último abrimos este vínculo de **localhost**



```
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Sergio Barrios\lurelia2.0> cd c:\Users\Sergio Barrios\lurelia2.0
Set-Location : No se encuentra ningún parámetro de posición que acepte el argumento '
En línea: 1 Carácter: 1
+ cd c:\Users\Sergio Barrios\lurelia2.0
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Set-Location], ParameterBindingException
+ FullyQualifiedErrorId : PositionalParameterNotFound, Microsoft.PowerShell.Commands.SetLocationCommand

PS C:\Users\Sergio Barrios\lurelia2.0> au run --watch
Local aurelia-cli v1.0.0-beta.13
Starting 'configureEnvironment'...
Finished 'configureEnvironment'
Starting 'runWebpack'...
Project is running at http://localhost:8080
webpack output is served from /
Content not from webpack is served from C:\Users\Sergio Barrios\lurelia2.0\dist
404s will fallback to /index.html
Finished 'runWebpack'
```

Ya podremos ver el título que es lo único que hemos mostrado no cerramos esta pestaña para más adelante mirar los cambios que hagamos.



Manejando Aurelia

7. Nos vamos a el archivo **app.html** que es donde vamos a mostrar el contenido que por defecto muestra `<h1>${message}</h1>` que es para mostrar el título

Seguidamente creamos un `<form submit.trigger="add ()">` donde se puede evidenciar que se llama al método `add` creado anteriormente precisamente para agregar contenido al sitio web

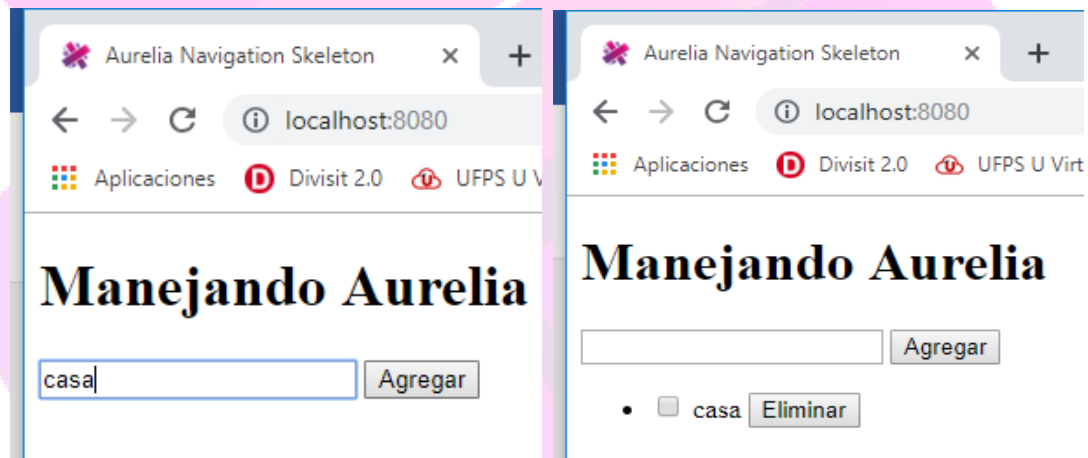
En la siguiente línea creamos un `<input type="text" value.bind="aprendiendoDescripcion">` y creamos un botón como se puede ver en la siguiente imagen


```

1 <template>
2
3 <h1>${message}</h1>
4 <form submit.trigger="add()">
5   <input type="text" value.bind="aprendiendoDescripcion">
6   <button type="submit">Agregar </button>
7 </form>
8
9 <ul>
10  <li repeat.for="aprendiendo of aprendiendos">
11    <input type="checkbox" checked.bind="aprendiendo.done">
12    <span css="text-decoration: ${aprendiendo.done?'line-through':'none'}">
13      ${aprendiendo.descripcion}
14    </span>
15  </li>
16 </ul>
17 </template>
18

```

Volvemos hacer los pasos para mirar si nos agrega y vemos que nuestro método que agregar ya funciona



8. Ahora mostramos y ponemos a funcionar el método remove de misma manera como se puede ver en la imagen, claro está que poniéndole un formato diferente

```

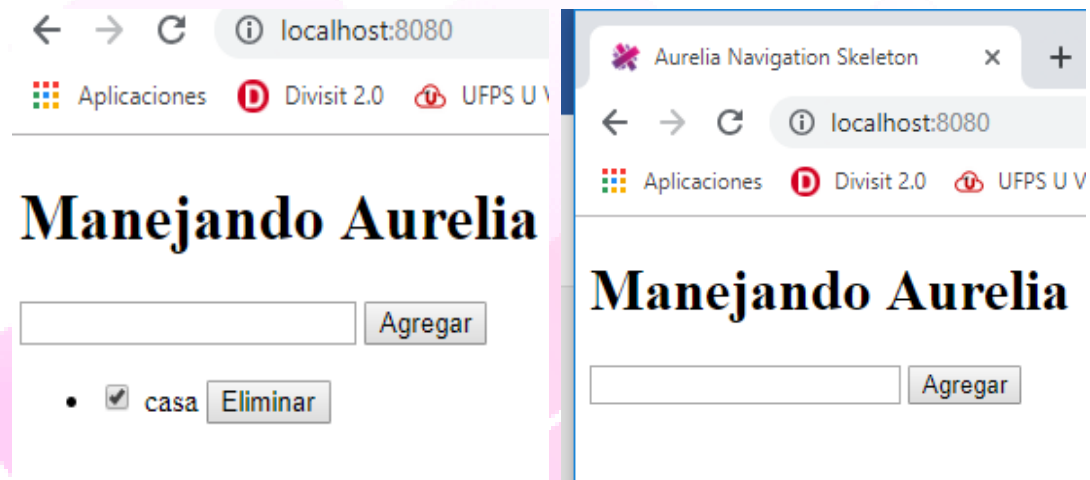
<button click.trigger="remove(aprendiendo)">Eliminar</button>

```

agregamos ese botón debajo de nuestra etiqueta

```
14     </span>
15     <button click.trigger="remove(aprendiendo)">Eliminar</button>
16   </li>
17 </ul>
```

Y con esto ya nos debe funcionar el botón de eliminar



9. Ahora quiero mostrar los resultados en una tabla

```
</form>
<table>
  <thead>
    <tr>
      <th>Selector</th>
      <th>Nombre</th>
      <th>Eliminar</th>
    </tr>
  </thead>
  <tbody repeat.for="aprendiendo of aprendiendos">
    <tr>
      <td> <input type="checkbox" checked.bind="aprendiendo.done"> </td>
      <td> <span css="text-decoration: ${aprendiendo.done?'line-through':'none'}"
        <span>
          ${aprendiendo.descripcion}
        </span>
      </td>
      <td> <button click.trigger="remove(aprendiendo)">Eliminar</button> </td>
    </tr>
  </tbody>
</table>
```

quedaría así:



10. Lo siguiente es ponerle un poco de estilo a la página, empezaremos usando bootstrap y la importamos en la capeta test en el archivo **index.ejs**.

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
```

```
index.ejs x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title><%- htmlWebpackPlugin.options.metadata.title %></title>
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <base href="<%- htmlWebpackPlugin.options.metadata.baseUrl %>">
8     <!-- imported CSS are concatenated and added automatically -->
9     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggO
10  </head>
11  <body aurelia-app="main">
12  </body>
13 </html>
```

11. Después mejoramos el h1 de la página:

```
<> app.html x JS app.js
1 <template>
2   <div class="jumbotron">
3     <div class="row">
4       <div class="col-md-12 text-center">
5         <h1>${message}</h1>
6       </div>
7     </div>
8   </div>
9
```

nos mostraría esto:

Manejando Aurelia

Selector Nombre Eliminar

12. Por ultimo le vamos a dar estilo a lo que nos falta

```

7  </div>
8  </div>
9  <div class="text-center">
10   <form submit.trigger="add()">
11     <input type="text" value.bind="aprendiendoDescripcion">
12     <button type="submit" class="btn btn-secondary">Agregar</button>
13   </form>
14 </div>
15 <div>
16   <table class="table table-bordered text-center" >
17     <thead>
18       <tr class="table-primary">
19         <th scope="col">NOMBRE</th>
20         <th scope="col">ELIMINAR</th>
21       </tr>
22     </thead>
23     <tbody repeat.for="aprendiendo of aprendiendos">
24       <tr class="table-active">
25         <td> <span css="text-decoration :${aprendiendo.done?'line-through':'none'}">
26           ${aprendiendo.descripcion}
27         </span>
28       </td>
29       <td> <button click.trigger="remove(aprendiendo)" class="btn btn-secondary">Eliminar</button> </td>
30     </tr>
31   </tbody>
32 </table>
33 </div>
34 </template>

```

Y así quedaría nuestra aplicación

Manejando Aurelia

Agregar

NOMBRE	ELIMINAR
--------	----------

Manejando Aurelia

Agregar

NOMBRE	ELIMINAR
sergio	Eliminar

¿COMO DESINSTALAR AURELIA?

Abrimos una Terminal de comando / **PowerShell** o **Bash** y escribe el comando y lo ejecutamos.

```
npm uninstall aurelia-cli -g
```

Esperamos a que se complete la desinstalación de Aurelia

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Sergio Barrios> npm uninstall aurelia-cli -g
[ ..... ] / rollbackFailedOptional: verb 8486b15c38eb34d5
```

Por ultimo comprobamos si fue desinstalado correctamente con el comando:

```
cd aurelia
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Sergio Barrios> npm uninstall aurelia-cli -g
removed 491 packages in 27.52s
PS C:\Users\Sergio Barrios> cd aurelia
cd : No se encuentra la ruta de acceso 'C:\Users\Sergio Barrios\auralia' porque no existe.
En línea: 1 Carácter: 1
+ cd aurelia
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\Sergio Barrios\auralia:String) [Set-Location], ItemNotFoundExc
ption
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLocationCommand
```

¿QUE PASA SI NECESITO AYUDA?

Hay dos maneras de solucionar tus dudas en Aurelia

1. Debemos abrir una Terminal de comando / **PowerShell** o **Bash**. Después debes introducir el comando:

```
cd (nombre aplicación)
```

Cuando estemos en nuestro proyecto ejecutamos el comando:

```
au help
```

2. También podemos contactarnos en el [foro Aurelia Discourse](#).

Si no está seguro de qué versión de la CLI está ejecutando, puede ejecutar

```
au -v
```

Si desea conocer más sobre Aurelia [ingrese aquí](#).

CONCLUSIONES

Se concluye que Aurelia es un framework fácil de usar con muchas utilidades que no tiene dependencias externas, te permite crear elementos customizados o añadirles a elementos existentes atributos customizados, que te permiten crear una aplicación rápidamente sin preocuparte de librerías externas, también tiene plugins para internalización, validación, ventanas modales y más cosas que hacen que sea un muy buen framework a la hora de hacer desarrollo web



REFERENCIAS BIBLIOGRAFICAS

<https://aurelia.io/docs/cli/basics#introduction>

<https://er1x.github.io/2016/02/empezar-desarrollo-spa-con-aurelia/>

<https://riptutorial.com/Download/aurelia-es.pdf>

<https://riptutorial.com/es/aurelia/example/19295/instalacion-del-complemento-aurelia-i18n>

<https://books.google.com.co/books?id=L8zcDgAAQBAJ&pg=PA15&lpg=PA15&dq=como+desinstalar+aurelia&source=bl&ots=qf6c-LnYzE&sig=ACfU3U2QkSoy60hJ4D0mODuLnLwgUz-Z-A&hl=es&sa=X&ved=2ahUKEwjB7Y6mnonhAhVxpIkKHbdnCBUQ6AEwBHoECAIQAQ#v=onepage&q=como%20desinstalar%20aurelia&f=false>

<https://medium.com/canariasjs/descubriendo-aurelia-8c7cbb7a49c>