

Vamos a revisar Atom y algunos de los Atom plugins más importantes en este artículo. Estos últimos años he estado usando mucho SublimeText como editor de texto pero estoy pasándome poco a poco a **Atom** ya que me esta pareciendo un editor superior y es OpenSource , Atom pertenece a **Github**.

Atom plugins

Vamos a irlos abordando poco a poco:

Git-plus: Al ser Atom un editor que ha sido creado por la gente de Github es evidente que su integración con Git debe ser buena. Este plugin es uno de los que más me ha convencido por su sencillez al usar GitHub desde Atom.

Linter: Es otro de los paquetes imprescindibles y provee a Atom de un API general para la validación de código y la generación de avisos de errores que luego otros plugins más especializados usan.

Linter-jslint : Complemento al plugin anterior si desarrollamos en JavaScript hoy por hoy es imprescindible instalar al menos este que es realmente potente y práctico.

```
112     ....this.controller.login('member', {~
113     ....    email: 'foo@bar.com'~
● 114     ....    password: 'foobarpasword',~
● 115     ....    remember: false~
● 116     ....});~
● 117     ....this.server.respond([200, {}, '']);~
● 118     ....this.spy.should.have.been.calledWith('auth:login:ok');~
● 119     ....});~
120 ~
● 121     ....it('should send an event nok on login error', function () {~
● 122     ....    var spy = sinon.spy(vent, 'trigger');~
```

jshint Expected 'password' to have an indentation at 7 instead at 9.
jshint Expected '}' to match '{' from line 112 and instead saw 'password'.
jshint Expected '}' and instead saw '.'.
jshint Missing semicolon.
jshint Expected 'foobarpasword' to have an indentation at 7 instead at 19.

test/spec/modules/auth/controllers/api.spec.js* 114,29 JavaScript

Terminal-Plus: Uno de mis favoritos nos permite abrir dentro del propio editor varias pestañas de terminal con personalizaciones varias y una forma muy cómoda y elegante de ocultarlas y que no nos molesten.



MiniMap: Este sí que parece un copy/paste de uno de los menús de SublimeText. Nos muestra en una barra lateral a la derecha todo el contenido de nuestro fichero de forma compacta.

Open-in-browser : Atom es tan modular, tan modular que incluso hay que instalar un plugin de este estilo para poder abrir los ficheros de forma cómoda en un navegador.

Atom-beautify: No podía faltar el típico plugin que nos maqueta todo el código con la ventaja que tiene de cubrir desde JavaScript, a Java pasando por C# hasta TypeScript.

```

274         data: JSON.stringify({
275             long: longCoord,
276             lat: latCoord,
277             time: timeStampFormatted,
278             email: window.localStorage.getItem(
279                 "email"),
280             categorysp: window.localStorage
281                 .getItem("sport"),
282             categorydk: window.localStorage
283                 .getItem("diskussion"),
284             categoryoefAr: window.localStorage
285                 .getItem(
286                     "öffentlichkeitsarbeit"
287                 ),
288             categorydpe: window.localStorage
289                 .getItem("publicEvent"),
290             categoryk: window.localStorage.getItem(
291                 "kinder")
292         }),
293         dataType: "json",

```

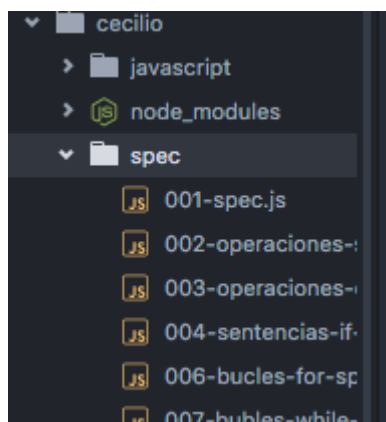
Atom-ternjs : Intellisense para JavaScript que cubre tecnologías como jQuery ,Node y Angular otro de los que no nos puede faltar.

```
document.querySelector()
```

```
querySelector(selectors: string) : NodeList
```

Emmet : Este es uno de los paquetes más clásicos ya que nos permite construir código html de una forma muy rápida a través de sus famosas abreviaciones.

File-icons: Uno de mis favoritos ya que asigna un icono a cada uno de los ficheros que tienes , clarificando mucho más el contenido que uno tiene disponible.



Atom es un editor moderno que esta en plena expansión y cada día aparecerán más Atom plugins interesantes según la comunidad vaya creciendo.

Otros artículos relacionados: [JavaScript Pure Functions](#) , [JavaScript Namespace](#) , [JavaScript Memorization](#)