Tooling


npm

npm

Gestor de dependencias de Node.js

- Compartir
- Reutilizar
- Instalación automática
- Gestión de versiones
- Enlazado con Git

npm

`package.json`

Sirve:

- Documentación de las dependencias de tu proyecto
- Para especificar la versión de código de tu proyecto
- Indicar repositorio
- Unificación de desarrollo y entregables
- Scripts de proyecto

Eslint

Eslint

- Un *analizador sintáctico*
- Revisa el código de nuestro proyecto
- Se asegura que se está siguiendo el conjunto de **reglas** que tenemos activas

Eslint

- Entiende la sintaxis, pero no la semántica
- Que toda la base de código siga un **estilo homogéneo**
- Puede detectar errores simples
- NO es ninguna garantía de funcionamiento

Eslint

```
npm install eslint
```

Eslint

```
./node_modules/.bin/eslint --init
```

Eslint

```
./node_modules/.bin/eslint .
```

Eslint

```
console.log('hola')
```

redradix

Eslint

```
./node_modules/.bin/eslint .
```

Eslint

```
/home/redradix/projects/jspro/test.js
  1:1   warning  Unexpected console statement  no-console
  1:20  error    Missing semicolon             semi

✖ 2 problems (1 error, 1 warning)
```

redradix

Eslint

- Nuestro código **no cumple las reglas** de estilo
- Tenemos dos opciones:
  - modificar el código
  - modificar las reglas

Eslint

```
/home/redradix/projects/jspro/test.js
  1:1   warning  Unexpected console statement  no-console
  1:20  error    Missing semicolon             semi

✖ 2 problems (1 error, 1 warning)
```

redradix

Eslint

```
/home/redradix/projects/jspro/test.js
  1:1   warning  Unexpected console statement  no-console
  1:20  error    Missing semicolon             semi

✖ 2 problems (1 error, 1 warning)
```

redradix

Eslint

```
/home/redradix/projects/jspro/test.js
  1:1   warning  Unexpected console statement  no-console
  1:20  error    Missing semicolon             semi

✖ 2 problems (1 error, 1 warning)
```

redradix

Eslint

```
/home/redradix/projects/jspro/test.js
  1:1   warning  Unexpected console statement  no-console
  1:20  error    Missing semicolon             semi

✖ 2 problems (1 error, 1 warning)
```

redradix

Eslint

```
/home/redradix/projects/jspro/test.js
  1:1   warning  Unexpected console statement  no-console
  1:20  error    Missing semicolon             semi

✖ 2 problems (1 error, 1 warning)
```

redradix

Eslint

```
console.log('hola');
```

redradix

Eslint

```
/home/redradix/projects/jspro/test.js
  1:1   warning  Unexpected console statement  no-console

✖ 1 problems (1 warning)
```

redradix

Eslint

```
/* eslint-disable */
console.log('hola');
```

redradix

Eslint

```
/* eslint-disable */
console.log('hola');
/* eslint-enable */
```

redradix

Eslint

```
// eslint-disable-next-line
console.log('hola');
```

redradix

Eslint

```
/* eslint-disable no-console */
console.log('hola');
```

Eslint

- Eslint sigue las reglas definidas en el fichero `.eslintrc`
- Podemos modificar el fichero para introducir modificaciones permanentes

Eslint

```
{
    "extends": "airbnb-base",
    "plugins": [
        "import"
    ]
}
```

redradix

Eslint

```
{
    "extends": "airbnb-base",
    "plugins": [
        "import"
    ]
}
```

redradix

Eslint

```
{
  "extends": "airbnb-base",
  "plugins": [
    "import"
  ],
  "rules": {
    "no-console": "off"
  }
}
```

Eslint

```
{
  "extends": "airbnb-base",
  "plugins": [
    "import"
  ],
  "rules": {
    "no-console": "off"
  }
}
```

redradix

Eslint

http://eslint.org/docs/rules/

Eslint

¡Intégralo con tu **editor**!

redradix

Webpack

redradix

Webpack

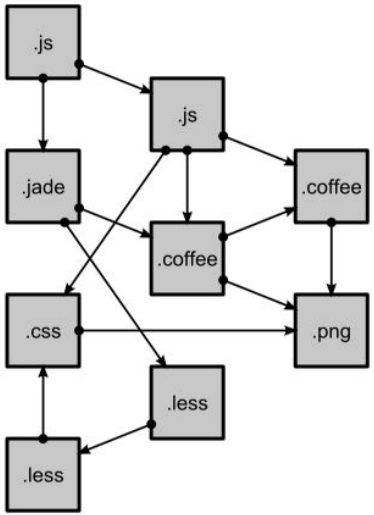
- Un empaquetador de módulos
- Analiza el árbol de código y sus dependencias
  - código
  - assets (estilos, imágenes, json, ...)
- “Empaqueta” lo que puede

Webpack

- Excelente soporte para código
  - Concatena, minifica, comprime...
  - Dead code elimination
  - Múltiples puntos de entrada

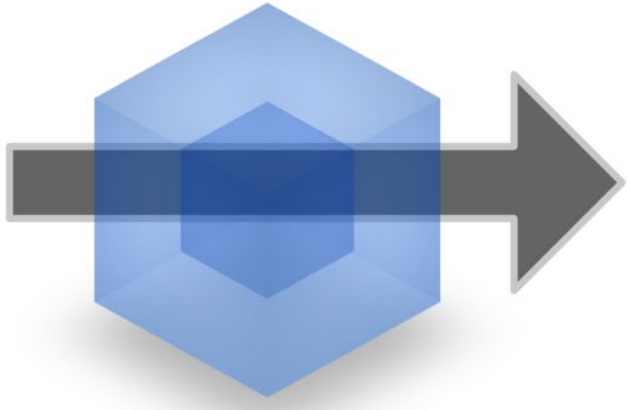
Webpack

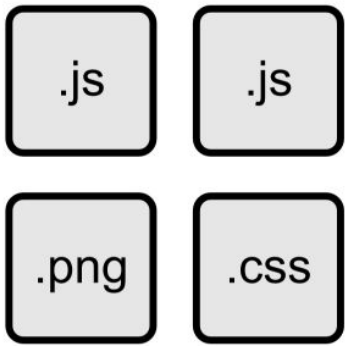
- Loaders (plugins) para muchos formatos
  - Sass / stylus / less / ...
  - JSON
  - glsl
  - ....



modules
with dependencies

webpack
MODULE BUNDLER





static
assets

Webpack

```
npm install webpack
```

Webpack

- Configuración en webpack.config.js
  - entry point
  - output
  - sources
  - loaders

```
var path = require('path');

module.exports = {
  entry: path.join(__dirname, 'src', 'main.js'),
  output: {
    path: path.join(__dirname, 'dist'),
    filename: 'bundle.js',
    publicPath: '/'
  },
  resolve : {
    root       : path.join(__dirname, 'src'),
    extensions : ['', '.js', '.jsx', '.json']
  },
  module: { loaders: [] }
};
```

```
var path = require('path');

module.exports = {
  entry: path.join(__dirname, 'src', 'main.js'),
  output: {
    path: path.join(__dirname, 'dist'),
    filename: 'bundle.js',
    publicPath: '/'
  },
  resolve : {
    root       : path.join(__dirname, 'src'),
    extensions : ['', '.js', '.jsx', '.json']
  },
  module: { loaders: [] }
};
```

```
var path = require('path');

module.exports = {
  entry: path.join(__dirname, 'src', 'main.js'),
  output: {
    path: path.join(__dirname, 'dist'),
    filename: 'bundle.js',
    publicPath: '/'
  },
  resolve : {
    root       : path.join(__dirname, 'src'),
    extensions : ['', '.js', '.jsx', '.json']
  },
  module: { loaders: [] }
};
```

```
var path = require('path');

module.exports = {
  entry: path.join(__dirname, 'src', 'main.js'),
  output: {
    path: path.join(__dirname, 'dist'),
    filename: 'bundle.js',
    publicPath: '/'
  },
  resolve : {
    root       : path.join(__dirname, 'src'),
    extensions : ['', '.js','.json']
  },
  module: { loaders: [] }
};
```

```
var path = require('path');

module.exports = {
  entry: path.join(__dirname, 'src', 'main.js'),
  output: {
    path: path.join(__dirname, 'dist'),
    filename: 'bundle.js',
    publicPath: '/'
  },
  resolve : {
    root       : path.join(__dirname, 'src'),
    extensions : ['', '.js','.json']
  },
  module: { loaders: [] }
};
```

```
loaders: [
   { test: /\.js$/,
     exclude: [/node_modules/],
     loader: 'babel-loader',
     query: {
       presets: ['es2015']
     },
     include: [path.join(__dirname, 'src')]
   },
   { test: /\.json$/,
     exclude: [/node_modules/],
     loader: 'json-loader'
   }
 ]
```

```
loaders: [
  { test: /\.js$/,
    exclude: [/node_modules/],
    loader: 'babel-loader',
    query: {
      presets: ['es2015']
    },
    include: [path.join(__dirname, 'src')]
  },
  { test: /\.json$/,
    exclude: [/node_modules/],
    loader: 'json-loader'
  }
]
```

```
loaders: [
   { test: /\.js$/,
     exclude: [/node_modules/],
     loader: 'babel-loader',
     query: {
       presets: ['es2015']
     },
     include: [path.join(__dirname, 'src')]
   },
   { test: /\.json$/,
     exclude: [/node_modules/],
     loader: 'json-loader'
   }
 ]
```

```
loaders: [
   { test: /\.js$/,
     exclude: [/node_modules/],
     loader: 'babel-loader',
     query: {
       presets: ['es2015']
     },
     include: [path.join(__dirname, 'src')]
   },
   { test: /\.json$/,
     exclude: [/node_modules/],
     loader: 'json-loader'
   }
 ]
```

```
loaders: [
   { test: /\.js$/,
     exclude: [/node_modules/],
     loader: 'babel-loader',
     query: {
       presets: ['es2015']
     },
     include: [path.join(__dirname, 'src')]
   },
   { test: /\.json$/,
     exclude: [/node_modules/],
     loader: 'json-loader'
   }
]
```

Webpack

webpack

redradix

Webpack

```
webpack -w
```

Todo junto

Vamos a crear nuestro primer paquete npm

Usaremos

- npm
- Webpack
- Eslitn
- Mocha

Todo junto

- npm
- npm init
- Configuramos package.json
- Empezamos a instalar dependencias
- Añadimos fichero de configuración webpack
- Añadimos dependencias (webpack, loaders...)
- Setear linter