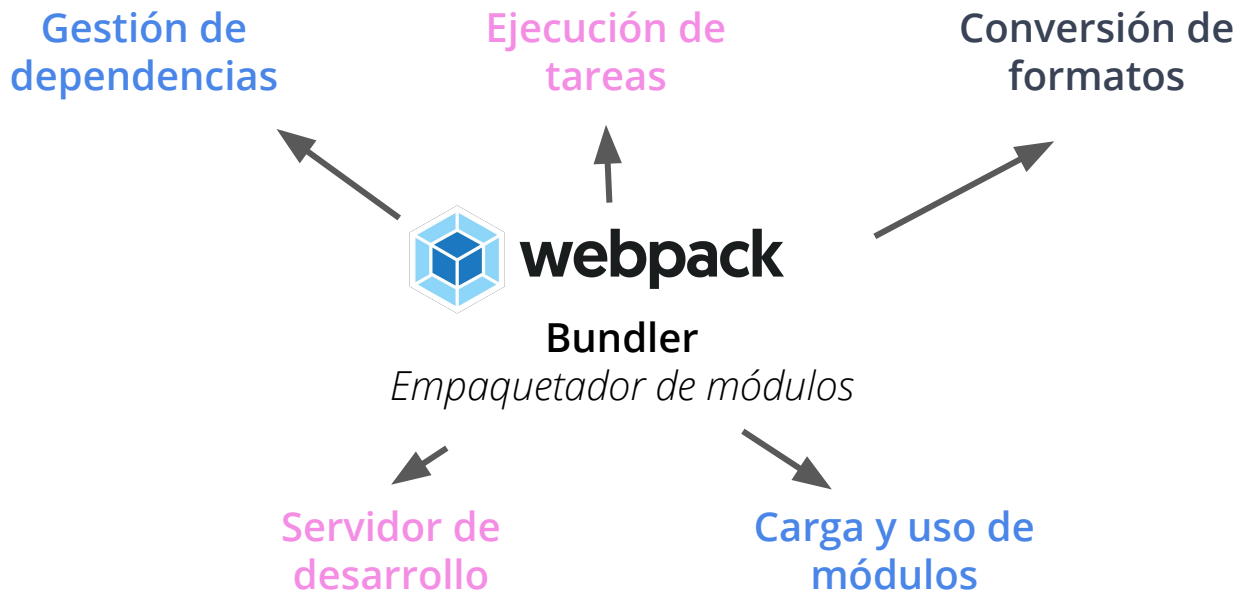


Laravel Mix



Desarrollado por *Jeffrey Way*, fundador de Laracast. **Laravel Mix** es un **envoltorio** sobre **Webpack** que nos facilita una **API** con la que afrontar el *80% por ciento** de los casos de uso.

(*) <https://laravel-mix.com/>



¿Qué papel tienen este tipo de herramientas en nuestro workflow?

Backend

Frontend

Drupal™

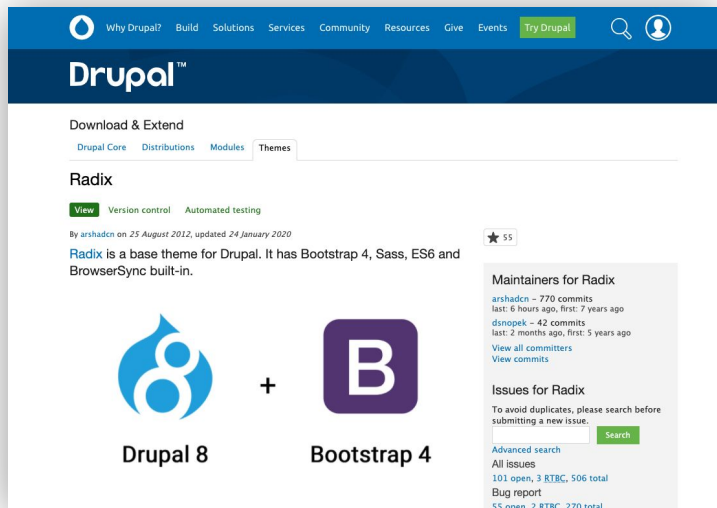


Backend

Frontend



Ejemplo de uso de **Laravel Mix** en **Drupal**



radix



Bootstrap 4



Laravel Mix

¿Qué definimos en el **package.json**?



Laravel Mix

```
{
  "name": "radix",
  "description": "Responsive base theme for Drupal.",
  "private": true,
  "scripts": {
    "postinstall": "find node_modules/ -name '*.info' -type f -delete",
    "dev": "cross-env NODE_ENV=development node_modules/webpack/bin/webpack.js --progress --hide-modules --config=node_modules/laravel-mix/setup/webpack.config.js",
    "watch": "cross-env NODE_ENV=development node_modules/webpack/bin/webpack.js --watch --progress --hide-modules --config=node_modules/laravel-mix/setup/webpack.config.js",
    "hot": "cross-env NODE_ENV=development webpack-dev-server --inline --hot --config=node_modules/laravel-mix/setup/webpack.config.js",
    "production": "cross-env NODE_ENV=production node_modules/webpack/bin/webpack.js --progress --hide-modules --config=node_modules/laravel-mix/setup/webpack.config.js"
  },
  "devDependencies": {
    "bootstrap": "^4.3.1",
    "browser-sync": "^2.26.7",
    "browser-sync-webpack-plugin": "^2.0.1",
    "cross-env": "^5.1",
    "laravel-mix": "^5.0.4",
    "node-sass": "^4.13.1",
    "popper.js": "^1.14.7"
  }
}
```

radix

¿Qué definimos en el **webpack.mix.json**?



```
const mix = require('laravel-mix')
const glob = require('glob')

mix.setPublicPath('assets').disableNotifications()

mix.browserSync({
  proxy: 'drupal.local',
  files: ['assets/js/**/*.js', 'assets/css/**/*.css'],
  stream: true
})

mix
  .sass('src/sass/radix.bootstrap.scss', 'css')
  .sass('src/sass/radix.style.scss', 'css')
```

Incluimos **Laravel Mix** y las herramientas de configuración necesarias.

```
const mix = require('laravel-mix')
const glob = require('glob')

mix.setPublicPath('assets').disableNotifications()

mix.browserSync({
  proxy: 'drupal.local',
  files: ['assets/js/**/*.js', 'assets/css/**/*.css'],
  stream: true
})

mix
  .sass('src/sass/radix.bootstrap.scss', 'css')
  .sass('src/sass/radix.style.scss', 'css')
```

Define el directorio `/assets` del tema como nuestro directorio público y desactiva las notificaciones de compilado

```
const mix = require('laravel-mix')
const glob = require('glob')

mix.setPublicPath('assets').disableNotifications()

mix.browserSync({
  proxy: 'drupal.local',
  files: ['assets/js/**/*.js', 'assets/css/**/*.css'],
  stream: true
})

mix
  .sass('src/sass/radix.bootstrap.scss', 'css')
  .sass('src/sass/radix.style.scss', 'css')
```

Configura **BrowserSync** para que detecte los cambios que se produzcan en los archivos javascript y css que se encuentran dentro de **assets/js/** y **assets/css**.

```
const mix = require('laravel-mix')
const glob = require('glob')

mix.setPublicPath('assets').disableNotifications()

mix.browserSync({
  proxy: 'drupal.local',
  files: ['assets/js/**/*.js', 'assets/css/**/*.css'],
  stream: true
})

mix
  .sass('src/sass/radix.bootstrap.scss', 'css')
  .sass('src/sass/radix.style.scss', 'css')
```

Compila el código **Sass** de **Bootstrap** y el propio de Radix en código **CSS**.

```
"scripts": {  
  "postinstall": "find node_modules/ -name '*.info' -type f -delete",  
  "dev": "cross-env NODE_ENV=development node_modules/webpack/bin/webpack.js --progress --hide-modules --config=node_modules/laravel-mix/setup/webpack.config.js",  
  "watch": "cross-env NODE_ENV=development node_modules/webpack/bin/webpack.js --watch --progress --hide-modules --config=node_modules/laravel-mix/setup/webpack.config.js",  
  "hot": "cross-env NODE_ENV=development webpack-dev-server --inline --hot --config=node_modules/laravel-mix/setup/webpack.config.js",  
  "production": "cross-env NODE_ENV=production node_modules/webpack/bin/webpack.js --progress --hide-modules --config=node_modules/laravel-mix/setup/webpack.config.js"  
},
```

npm run **dev**

npm run **watch**

Laravel Mix desde cero



<https://github.com/managume/taller-laravel-mix>

Laravel Mix desde cero

```
# Creamos un nuevo tema de drupal con Drupal Console
drupal generate:theme --theme tailmix

# Nos situamos en la carpeta del tema e inicializamos un package.json
cd tailmix
npm init -y

# Instalamos Laravel Mix y las dependencias que vamos a utilizar
npm i laravel-mix cross-env tailwindcss --save-dev

# Copiamos en nuestro entorno la plantilla que nos provee Laravel Mix
cp node_modules/laravel-mix/setup/webpack.mix.js ./
```

Laravel Mix desde cero

```
# Vamos a añadir TailwindCSS a nuestro proyecto
npm i --save tailwindcss

# Configuramos nuestro webpack.mix.json para que trabaje con Laravel Mix
code webpack.mix.json
```

Laravel Mix desde cero

```
# Compilamos nuestros cambios
node_modules/.bin/webpack --config=node_modules/laravel-mix/setup/webpack.config.js

# Para evitar el paso anterior vamos a generar una serie de scripts
code package.json

# Ahora ya podemos lanzar nuestros scripts de compilación
npm run dev
npm run watch
npm run production
npm run hot
```



Laravel Mix

