

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE
AREQUIPA



INGENIERÍA DE SOFTWARE II

Construcción Automática de Proyectos

Alumna :

- Chullunquía Rosas, Sharon Rossely

Profesor :

- Sarmiento Calisaya, Edgar

20 de octubre de 2020

Índice

1. Modelo de construcción de software	2
2. Tutorial	4
2.1. Herramienta : Webpack	4
2.2. Instalación	5
2.3. Construcción del proyecto	5

1. Modelo de construcción de software

```
JS output.js > x  
Laboratories > Laboratory 03 > Octree > dist > JS output.js > ...  
  
1 /*  
2 * ATTENTION: The "eval" devtool has been used (maybe by default in mode: "development").  
3 * This devtool is neither needed for production nor for readable output files.  
4 * It uses "eval()" calls to create a separate source file in the browser devtools.  
5 * If you are trying to read the output file, select a different devtool (https://webpack.js.org/  
configuration/devtool/)  
6 * or disable the default devtool with "devtool: false".  
7 * If you are looking for production-ready output files, see mode: "production" (https://webpack.js.org/  
configuration/mode/).  
8 */  
9 /******/ (() => { // webpackBootstrap  
10 /*****/ "use strict";  
11 /*****/ var __webpack_modules__ = ({  
12  
13 /***/ "./node_modules/three/build/three.module.js":  
14 |/*|*****|  
15 |*** ./node_modules/three/build/three.module.js ***|  
16 |*****|  
17 |/*! namespace exports */  
18 |! export ACESFilmicToneMapping [provided] [no usage info] [missing usage info prevents renaming] */  
19 |! export AddEquation [provided] [no usage info] [missing usage info prevents renaming] */  
20 |! export AddOperation [provided] [no usage info] [missing usage info prevents renaming] */  
21 |! export AdditiveAnimationBlendMode [provided] [no usage info] [missing usage info prevents renaming] */  
22 |! export AdditiveBlending [provided] [no usage info] [missing usage info prevents renaming] */  
23 |! export AlphaFormat [provided] [no usage info] [missing usage info prevents renaming] */  
24 |! export AlwaysDepth [provided] [no usage info] [missing usage info prevents renaming] */  
25 |! export AlwaysStencilFunc [provided] [no usage info] [missing usage info prevents renaming] */  
26 |! export AmbientLight [provided] [no usage info] [missing usage info prevents renaming] */  
27 |! export AmbientLightProbe [provided] [no usage info] [missing usage info prevents renaming] */  
28 |! export AnimationClip [provided] [no usage info] [missing usage info prevents renaming] */  
29 |! export AnimationLoader [provided] [no usage info] [missing usage info prevents renaming] */  
30 |! export AnimationMixer [provided] [no usage info] [missing usage info prevents renaming] */
```

Figura 1: Archivo de salida Output.js

```
webpack.config.js X
Laboratories > Laboratory 03 > Octree > webpack.config.js > ...
1  const path = require('path');
2
3  module.exports = {
4    entry: './src/init.js',
5    output: {
6      filename: 'output.js',
7      path: path.resolve(__dirname, 'dist'),
8    },
9  };
```

Figura 2: webpack.config.js

```
1 {
2   "name": "octree",
3   "version": "1.0.0",
4   "description": "",
5   "main": "main.js",
6   "directories": {
7     "lib": "lib"
8   },
9   "scripts": {
10    "test": "echo \\\"Error: no test specified\\\" && exit 1",
11    "build": "webpack --mode=development"
12  },
13  "repository": {
14    "type": "git"
15  },
16  "keywords": [],
17  "author": "",
18  "license": "ISC",
19
20  "devDependencies": {
21    "webpack": "^5.1.3",
22    "webpack-cli": "^4.1.0"
23  },
24  "dependencies": {
25    "three": "^0.121.1"
26  }
27 }
```

Figura 3: package.json

2. Tutorial

A continuación, vamos a realizar la construcción del proyecto Octree; el cual esta desarrollado en el lenguaje *JavaScript*, haciendo uso de la herramienta **Webpack**, una de las muchas que hay para el lenguaje *JavaScript*.

2.1. Herramienta : Webpack

Webpack es un paquete de módulos estáticos para aplicaciones JavaScript modernas. Cuando webpack procesa su aplicación, crea internamente un gráfico de dependencia que mapea cada módulo que su proyecto necesita y genera uno o más paquetes . [1]

- **Entrada :**

Un punto de entrada indica qué módulo webpack debe usar para comenzar a construir su gráfico de dependencia interna .Webpack averiguará de qué otros módulos y bibliotecas depende ese punto de entrada (directa e indirectamente).

De forma predeterminada, su valor es `./src/index.js`, pero puede especificar uno diferente (o varios puntos de entrada) estableciendo una `entry` propiedad en la configuración del paquete web. [2]

Por ejemplo:

- **Salida :**

La propiedad de salida le dice al paquete web dónde emitir los paquetes que crea y cómo nombrar estos archivos. El valor predeterminado es `./dist/main.js` para el archivo de salida principal y la `./dist` carpeta para cualquier otro archivo generado. [4]

Puede configurar esta parte del proceso especificando un `output` campo en su configuración:

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

module.exports = {
  entry: './path/to/my/entry/file.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'my-first-webpack.bundle.js'
  }
};
```

Figura 4: `webpack.config.js`

2.2. Instalación

■ Primer Paso : Instalación del módulo de Three npm

Primero instalamos el módulo de *Three npm*, abrimos una ventana de la terminal en la carpeta del proyecto y ejecutamos el siguiente comando:

```
npm install --save three
```

■ Segundo Paso : Inicialización de npm e instalación de webpack

Luego, inicialicemos npm, instalemos *webpack* localmente e instalemos *webpack-cli* (la herramienta utilizada para ejecutar webpack en la línea de comandos):

```
npm init -y  
npm install webpack webpack-cli --save-dev
```

2.3. Construcción del proyecto

■ Primer Paso : Definir estructura

Crearemos la siguiente estructura de directorios, archivos y su contenido, teniendo en cuenta como plantilla la estructura de abajo.

```
webpack-demo  
├─ package.json  
+ ─ index.html  
+ ─ /src  
+   ─ index.js
```

Figura 5: Estructura ejemplar

```
▼ Octree  
  ▼ dist  
    <> index.html  
    JS output.js  
  > lib  
  > node_modules  
  ▼ src  
    JS init.js  
    JS main.js  
    JS octree.js  
  {} package-lock.json  
  {} package.json  
  🌀 webpack.config.js
```

Figura 6: Estructura del proyecto

Nuestro *index.js* en este caso seria nuestro *init.js* donde realizamos importaciones de los archivos *main.js* , *octree.js* y de la libreria *Three.js* que se encuentra en la carpeta *node_modules*. El archivo *index.html* vendria ser el html a correr, donde se realiza el llamado al archivo *output.js*, el cual es generado por la herramienta *webpack*.

- **Segundo Paso : Crear el archivo webpack.config.js**

Después, creamos un archivo *webpack.config.js* en el cual podremos configurar a nuestro modo la entrada y salida del *webpack*.

```
1  const path = require('path');
2
3  module.exports = {
4    entry: './src/init.js',
5    output: {
6      filename: 'output.js',
7      path: path.resolve(__dirname, 'dist'),
8    },
9  };
```

Figura 7: Personalizando entrada y salida para webpack

- **Tercer Paso : Agregar un script npm a package.json**

Luego, modificamos nuestro *package.json* agregando un script npm ("build": "webpack --mode=development"), tal y como se muestra en la siguiente imagen:

```
1  {
2    "name": "octree",
3    "version": "1.0.0",
4    "description": "",
5    "main": "main.js",
6    "directories": {
7      "lib": "lib"
8    },
9    > Debug
10   "scripts": {
11     "test": "echo \\\"Error: no test specified\\\" && exit 1",
12     "build": "webpack --mode=development"
13   },
14   "repository": {
15     "type": "git"
16   },
17   "keywords": [],
18   "author": "",
19   "license": "ISC",
20   "devDependencies": {
21     "webpack": "^5.1.3",
22     "webpack-cli": "^4.1.0"
23   },
24   "dependencies": {
25     "three": "^0.121.1"
26   }
27 }
```

Figura 8: Agregando script npm "build"

■ Cuarto Paso : Ejecución del proyecto

Finalmente, corremos el siguiente comando desde la terminal; y abrimos nuestro archivo *index.html* .

```
sharon@r2d2:~/Documentos/3ro_B/Software-Engineering-II/Laboratories/Laboratory 03/Octree$ npm run build
> octree@1.0.0 build /home/sharon/Documentos/3ro_B/Software-Engineering-II/Laboratories/Laboratory 03/Octree
> webpack --mode=development

[webpack-cli] Compilation finished
asset output.js 1.37 MiB [compared for emit] (name: main)
runtime modules 668 bytes 3 modules
cacheable modules 1.17 MiB
  modules by path ./src/*.js 9.21 KiB
    ./src/init.js 2.97 KiB [built] [code generated]
    ./src/octree.js 4.83 KiB [built] [code generated]
    ./src/main.js 1.41 KiB [built] [code generated]
  modules by path ./node_modules/three/ 1.16 MiB
    ./node_modules/three/build/three.module.js 1.14 MiB [built] [code generated]
    ./node_modules/three/examples/jsm/controls/OrbitControls.js 24.8 KiB [built] [code generated]
webpack 5.1.3 compiled successfully in 1507 ms
sharon@r2d2:~/Documentos/3ro_B/Software-Engineering-II/Laboratories/Laboratory 03/Octree$
```

Figura 9: Corriendo el comando *npm run build*

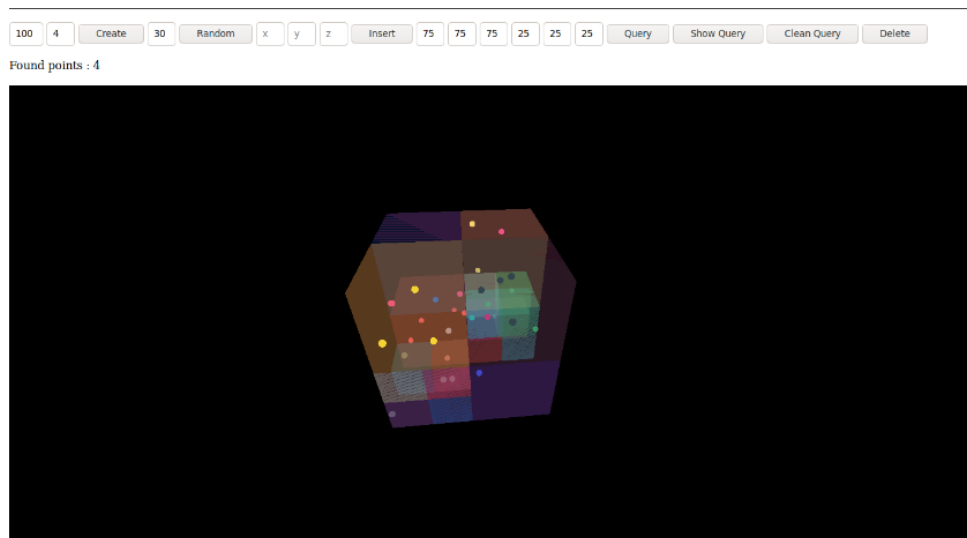


Figura 10: Proyecto en ejecución

Referencias

- [1] Webpack Documentation v5.1.3, *Concepts*. <https://webpack.js.org/concepts/>.
- [2] Webpack Documentation v5.1.3, *Entry*. <https://webpack.js.org/concepts/#entry>.
- [3] Webpack Documentation v5.1.3, *Output*. <https://webpack.js.org/concepts/#output>.
- [4] Repositorio en GitHub, <https://github.com/sharon1160/Software-Engineering-II/tree/master/Octree>.