

www.cesarguerra.mx



ENTORNOS MODERNOS DE DESARROLLO

Automatizadores, Preprocesadores, Vite y Linters

CONTENIDO



César Guerra

www.cesarguerra.mx

01 AUTOMATIZADORES
¿Qué son? ¿Para qué sirven?

02 PREPROCESADORES
Transpilación de código

03 VITE
¿Qué es y por qué usarlo?

04 COMENZANDO CON VITE
Creación de nuevos proyectos

05 LINTERS
¿Qué son y cómo configurarlos?



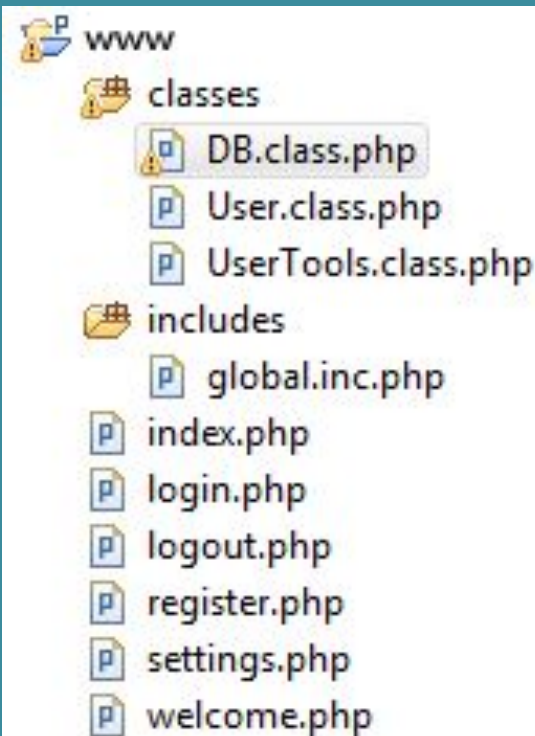
01

AUTOMATIZADORES



¿CÓMO TRABAJABAMOS EN DESARROLLO?

En los inicios del desarrollo web, para comenzar un sitio web **se debía crear manualmente una estructura de carpetas para alojar el proyecto** y sus distintas secciones (css, js, assets, etc.), configurar un servidor web, y añadir y vincular las diferentes librerías de CSS y/o Javascript necesarias.





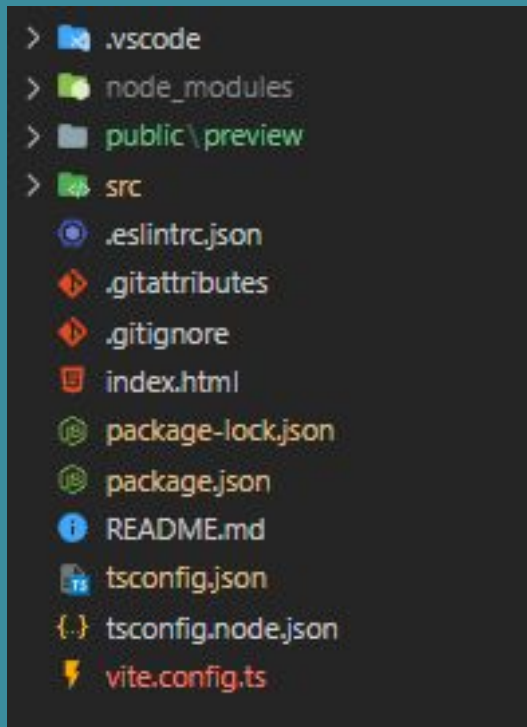
MALA PRÁCTICA

**ANTERIORMENTE
TRABAJABAMOS
DIRECTAMENTE SOBRE LOS
ARCHIVOS DE PRODUCCIÓN O
QUÉ IRÍAN A PRODUCCIÓN**



EVOLUCIÓN DE LA FORMA DE TRABAJO

Con el avance en el desarrollo web, **actualmente se utilizan técnicas de transpilación en las que se escribe código fuente en una carpeta específica (src) y luego es procesado y convertido en código ejecutable (en la carpeta dist) para ser leído por los navegadores.** Sin la automatización, el proceso de mantener el código fuente y el código ejecutable sincronizado se vuelve tedioso e incluso imposible, por lo que es indispensable el uso de herramientas.





AUTOMATIZADORES

Actualmente en el desarrollo de sitios web y aplicaciones con JavaScript, existen varios automatizadores que nos facilitan ciertas tareas:

- Creación del proyecto y su gestión (gestionado con NPM).
- Actualización y mantenimiento de dependencias (gestionado con NPM).
- Servidor web local de desarrollo.
- Minificación de código.
- Optimización de imágenes.
- Preprocesar o transpilar código fuente.
- Despliegue o subida del código final al servidor.

Año	Automatizador	Descripción
2012	Browserify	Automatizador centrado en traducir <code>require()</code> (<code>Node</code>) al navegador.
2013	Gulp	Automatizador Javascript de tareas basado en pipes y flujos.
2014	Webpack	Bundler enfocado en controlar todo y empaquetar aplicaciones JS.
2015	Rollup	Bundler empaquetador de aplicaciones con buen soporte ESM.
2016	Grunt	Automatizador enfocado en gestión de tareas.
2017	Parcel	Bundler de aplicaciones web enfocado en sencillez de uso.
2019	Snowpack	Automatizador enfocado en técnicas modernas con buen soporte ESM.
2020	Esbuild	Bundler Javascript ultra rápido, centrado en el rendimiento.
2021	Vite	Automatizador de aplicaciones web basado en Rollup y centrado en ESM.

02

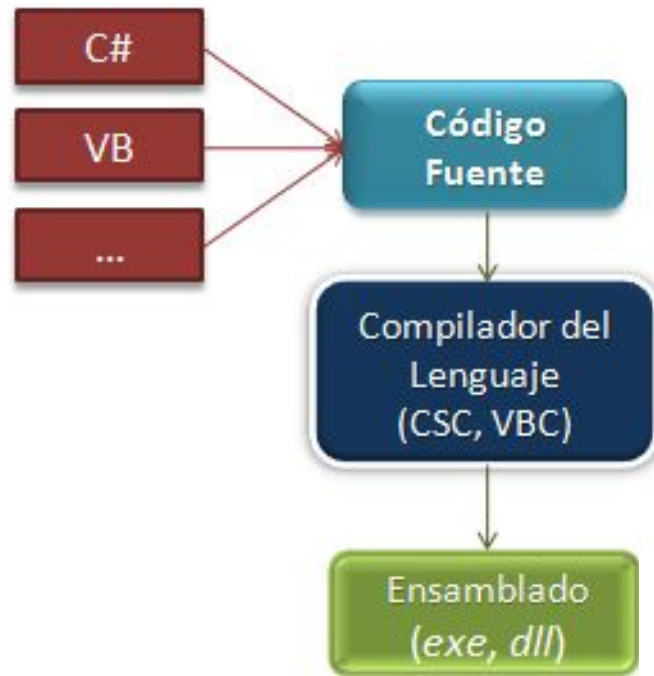
PREPROCESADORES



¿QUÉ ES COMPILAR?

Compilar es el proceso de transformar un programa informático escrito en un lenguaje en un conjunto de instrucciones en otro formato o lenguaje. Un compilador es un programa de computadora que realiza dicha tarea.

Normalmente, un compilador transforma código escrito en un lenguaje de alto nivel como C++, Rust o Java en código ejecutable — llamado código binario o código máquina.



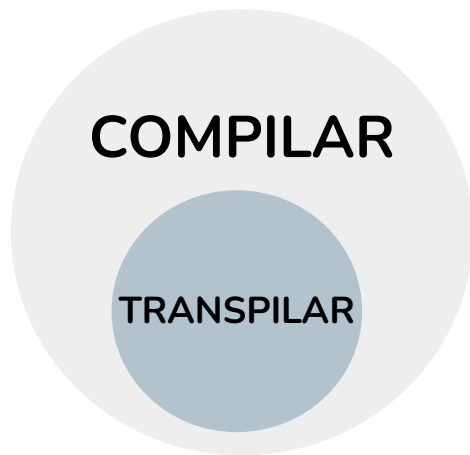


TRANSPILACIÓN

Muchos conocemos el término “Compilar”, pero no tanto el de transpilar.

Si el compilador traduce código entre dos lenguajes que están al mismo nivel de abstracción entonces, estamos ante un transpilador.

Si traduce código entre lenguajes de diferente nivel de abstracción (por ejemplo de alto a bajo nivel) entonces no lo es.



Transpilación



Compilación



```
Code:
0: iconst_2
1: istore_1
2: iload_1
3: sipush 1000
6: if_icmpge 44
9: iconst_2
10: istore_2
11: iload_2
12: iload_1
13: if_icmpge 31
16: iload_1
17: iload_2
18: irem                # remainder
19: ifne 25
22: goto 38
```

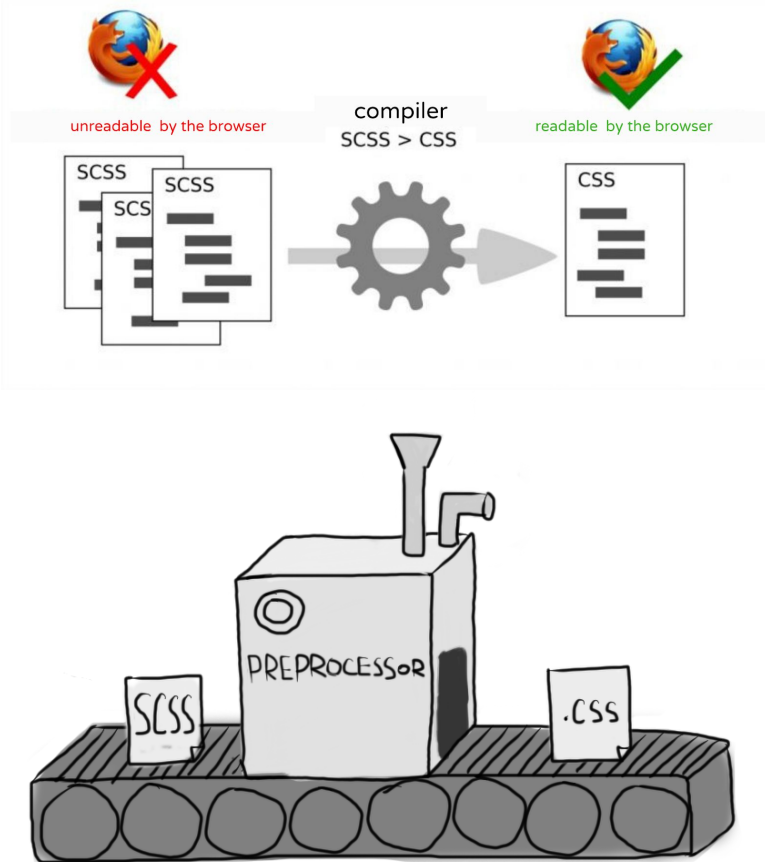


¿QUÉ ES UN PREPROCESADOR?

En programación, **es una herramienta que transpila código** (“Traduce” código de una sintaxis definida a otra) lo cual **le permite optimizar y estandarizar dicho código**.

Son:

- Herramientas altamente ocupadas en la industria.
- Proveen de mejores prácticas, shortcuts y facilidades al desarrollador.



ALGUNOS PREPROCESADORES

JS

BABEL



CSS

Sass

{less}

stylus

HTML



Haml 





BABEL

Babel es un "compilador" (o transpilador) para JavaScript.

Básicamente permite transformar código escrito con las últimas y novedosas características de JavaScript (ECMAScript) y transformarlo en un código que sea entendido por navegadores más antiguos.

BABEL

```
// Babel Input: ES2015 arrow function
[1, 2, 3].map( (n) => n+1 );

// Babel Output: ES5 equivalent
[1, 2 ,3].map(function(n) {
  return n+1;
});
```



03

VITE



¿QUE ES VITE?

Vite es una herramienta de compilación para proyectos web modernos que tiene como objetivo proporcionar una experiencia de desarrollo más rápida y ágil.

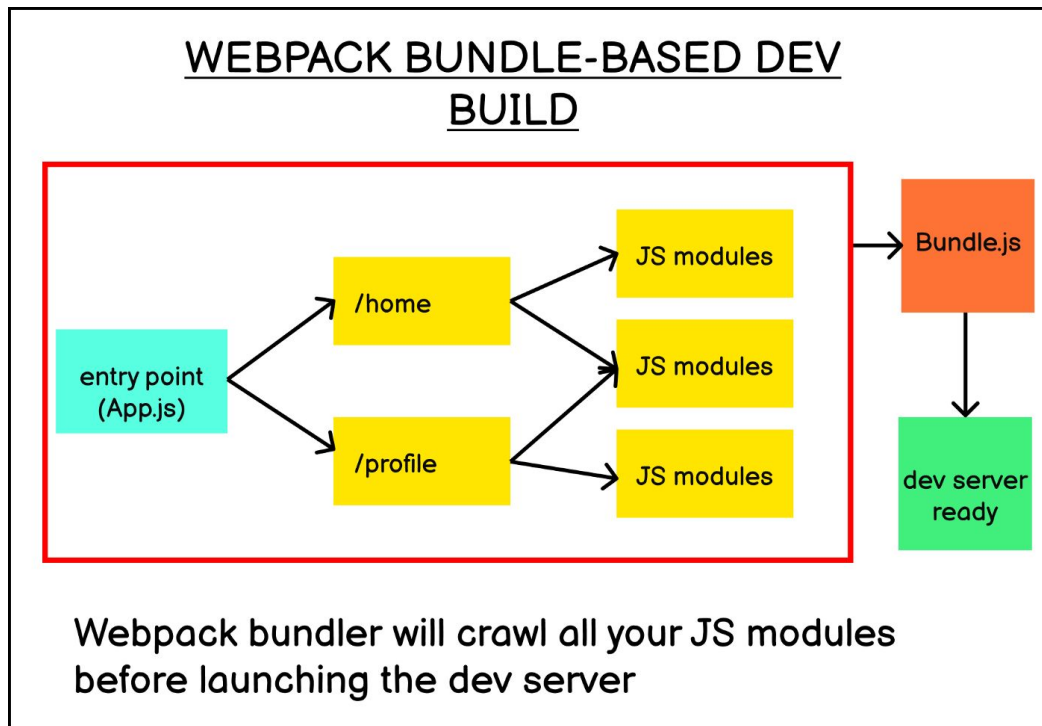
Consta de dos partes principales:

1. **Un servidor de desarrollo** que proporciona mejoras en funcionalidades sobre módulos ES nativos, como Hot Module Replacement (HMR) extremadamente rápido.
2. **Un comando de compilación que empaqueta el código** con Rollup, preconfigurado para generar recursos estáticos altamente optimizados para producción.

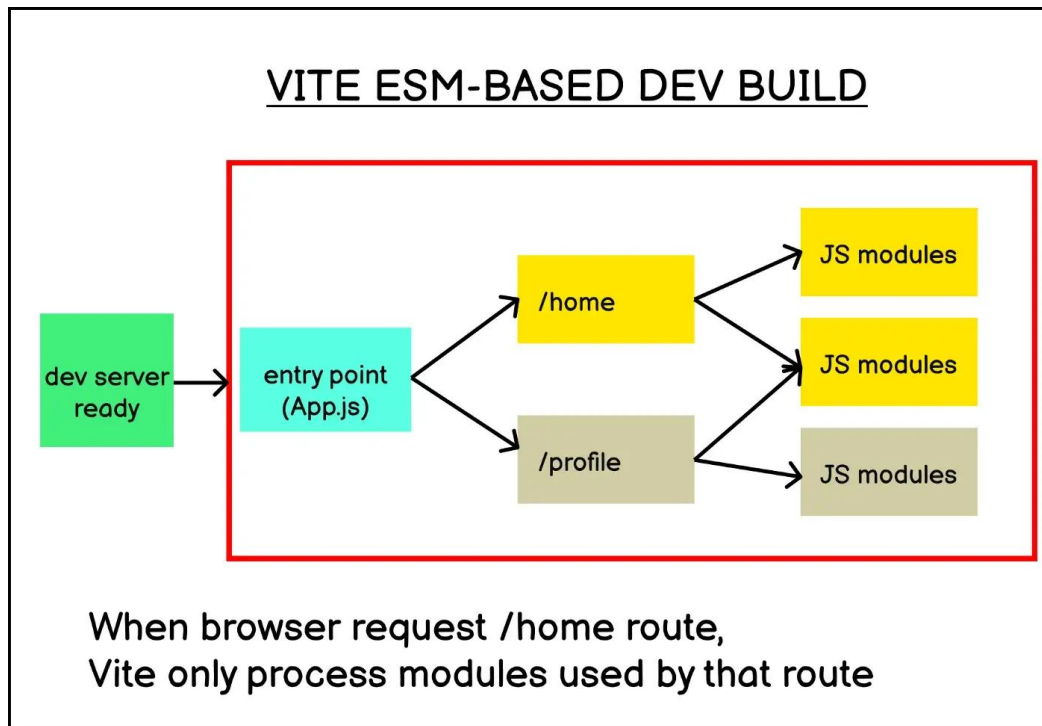


<https://es.vitejs.dev/>

FORMA DE TRABAJO DE WEBPACK



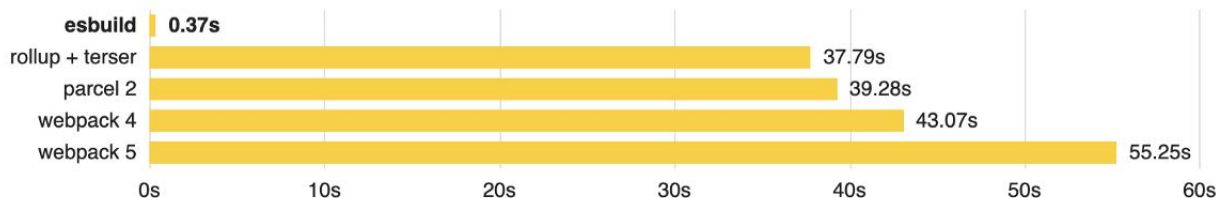
FORMA DE TRABAJO DE VITE



ESBUILD VS WEBPACK

esbuild

An extremely fast JavaScript bundler



Above: the time to do a production bundle of 10 copies of the [three.js](#) library from scratch using default settings, including minification and source maps. More info [here](#).

04

COMENZANDO CON VITE



www.cesarguerra.mx

INSTALANDO VITE: COMANDO PARA INSTALAR

Desde la terminal, comenzamos la instalación de Vite con el siguiente comando (npm):

```
npm create vite@latest
```

Monta tu primer proyecto Vite

Nota de compatibilidad

Vite requiere [Node.js](#) versión 14.18+, y 16+. Sin embargo, algunas plantillas requieren una versión superior de Node.js para funcionar, por favor actualiza si tu gestor de paquetes te advierte sobre ello.

Con NPM:

```
$ npm create vite@latest
```

bash

Con Yarn:

```
$ yarn create vite
```

bash

Con PNPM:

```
$ pnpm create vite
```

bash

¡Entonces sigue las instrucciones!





INSTALANDO VITE

OPCIONES DE INSTALACIÓN

Después nos hará una serie de preguntas:

Project name: Nombre del proyecto, en minúsculas, sin espacios ni caracteres especiales (será el nombre también de la carpeta del proyecto)

Select a Framework: Vite es compatible con varios frameworks de JavaScript.

Select a Variant: Debemos decidir si JavaScript puro o usamos TypeScript (tipado).

Al finalizar, deberemos entrar a la carpeta del proyecto, instalar las dependencias y entonces podremos correr nuestro servidor de desarrollo.

```
> npm create vite@latest
Need to install the following packages:
  create-vite@4.1.0
Ok to proceed? (y) y
✓ Project name: ... mi-primer-proyecto-vite
? Select a framework: » - Use arrow-keys. Return to submit.
> Vanilla
  Vue
  React
  Preact
  Lit
  Svelte
  Others
? Select a variant: » - Use arrow-keys. Return to submit.
> JavaScript
  TypeScript
Done. Now run:

  cd mi-primer-proyecto-vite
  npm install
  npm run dev
```





ESTRUCTURA DE PROYECTO (VANILLA)

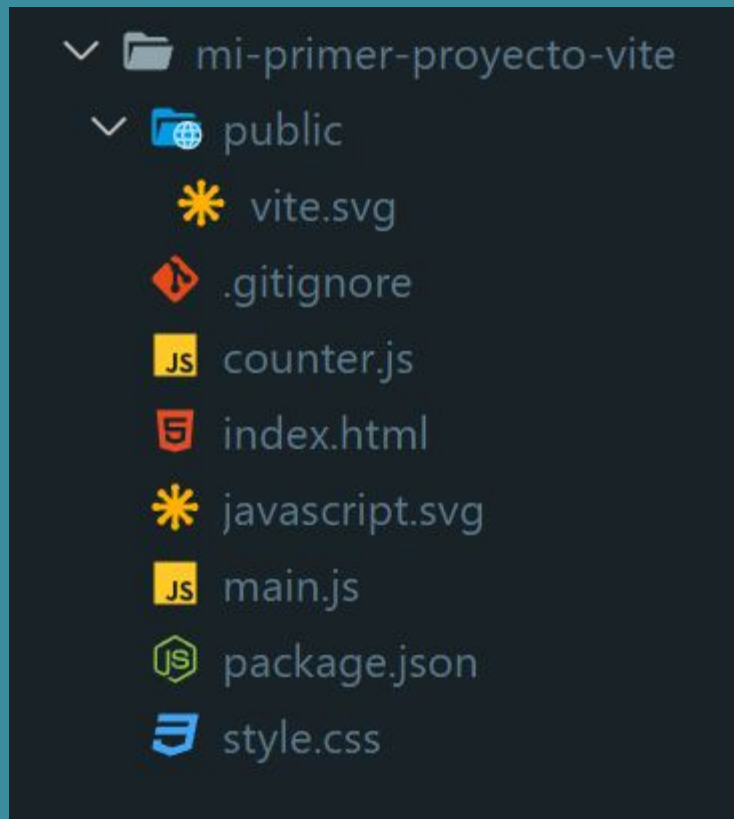
Los archivos importantes a tomar cuenta:

index.html: Es el archivo principal de nuestro proyecto, este mandará a llamar a main.js.

main.js: Archivo principal de JavaScript del proyecto, se ejecutará al inicio de la aplicación y podrá importar contenido de otros archivos.

style.css: Hoja de estilos globales del proyecto.

package.json: Contiene la información del proyecto, los scripts que se pueden ejecutar y también las dependencias insertadas por el manejador de paquetes (npm, yarn, pnpm, etc).





SCRIPTS DE VITE



```
npm run dev
```

Corre la aplicación en modo development en **http://localhost:5173**
Se autorecarga al detectar cambios.



```
npm run build
```

Construye la aplicación para producción y coloca el contenido en la carpeta dist.



```
npm run preview
```

Previsualización de la aplicación en modo producción.

```
package.json > ...
1  {
2    "name": "01-pokedex-vite",
3    "private": true,
4    "version": "0.0.0",
5    "type": "module",
6    "scripts": {
7      "dev": "vite",
8      "build": "vite build",
9      "preview": "vite preview"
10   },
11   "devDependencies": {
12     "vite": "^4.1.0"
13   }
14 }
```

package.json

DEMOSTRACIÓN



www.cesarguerra.mx

05

LINTERS



<https://www.cesarguerra.mx/configuracion-rapida-de-eslint-con-standard-js-para-proyectos-de-javascript-y-de-react-con-vite-js/>

www.cesarguerra.mx