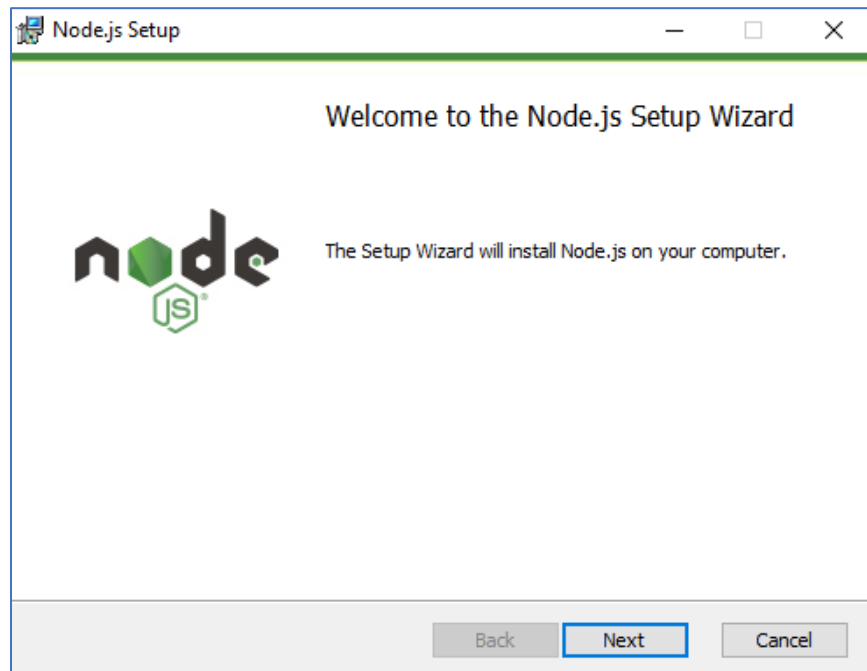
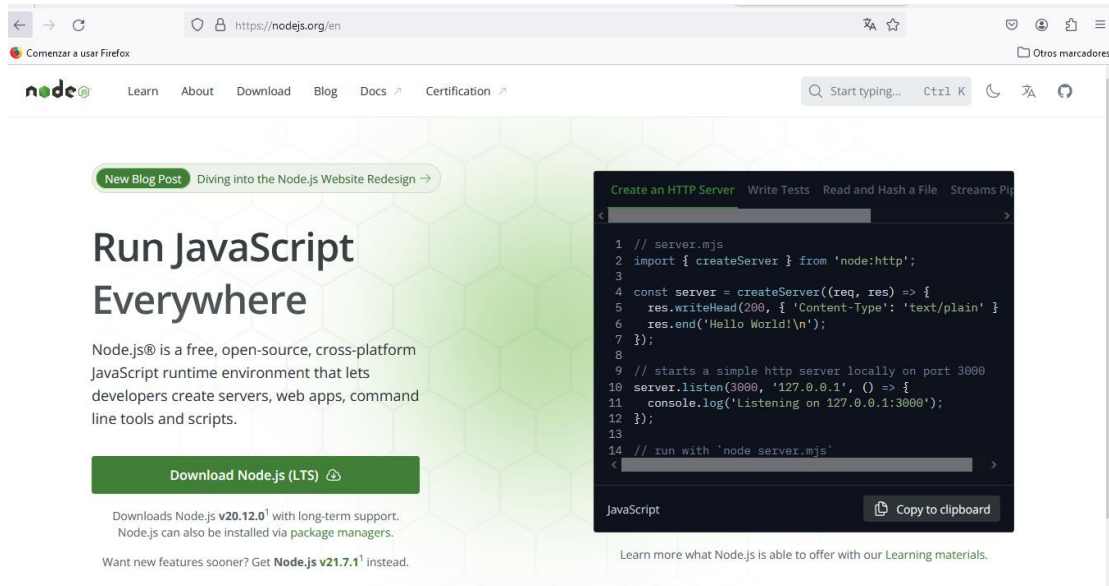
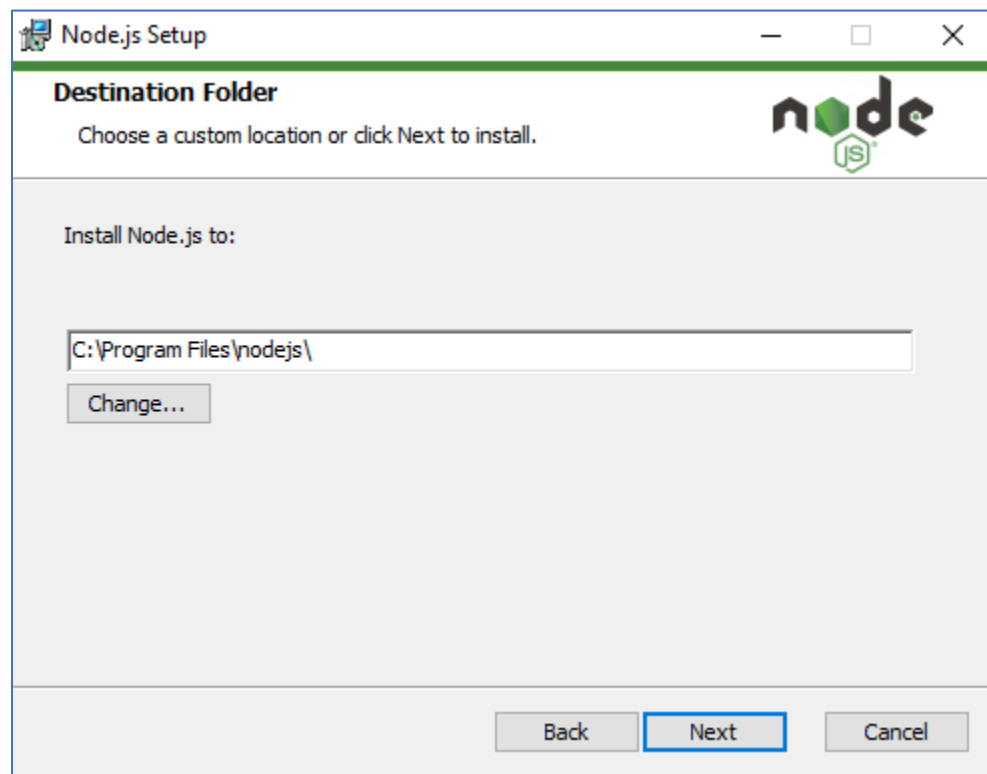
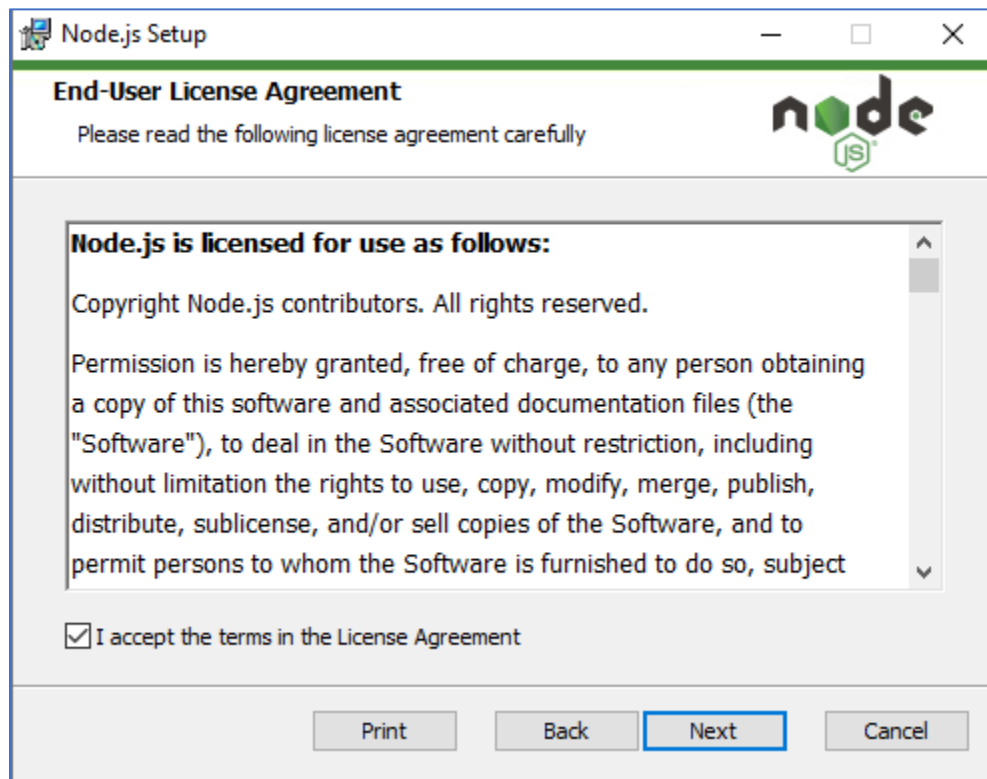


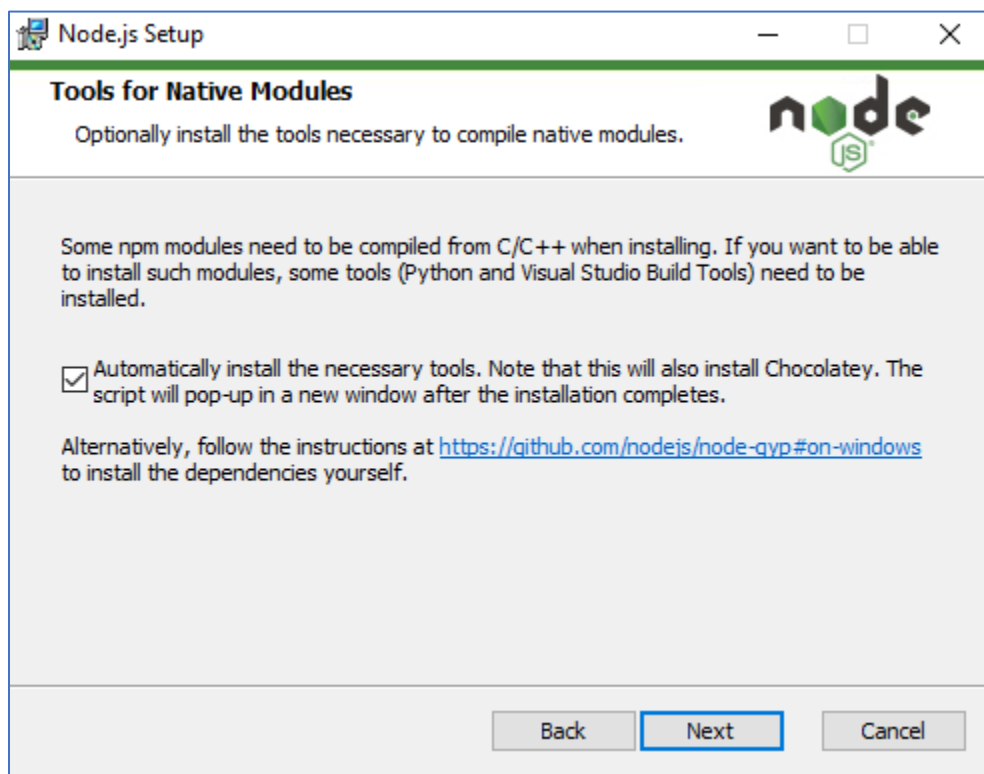
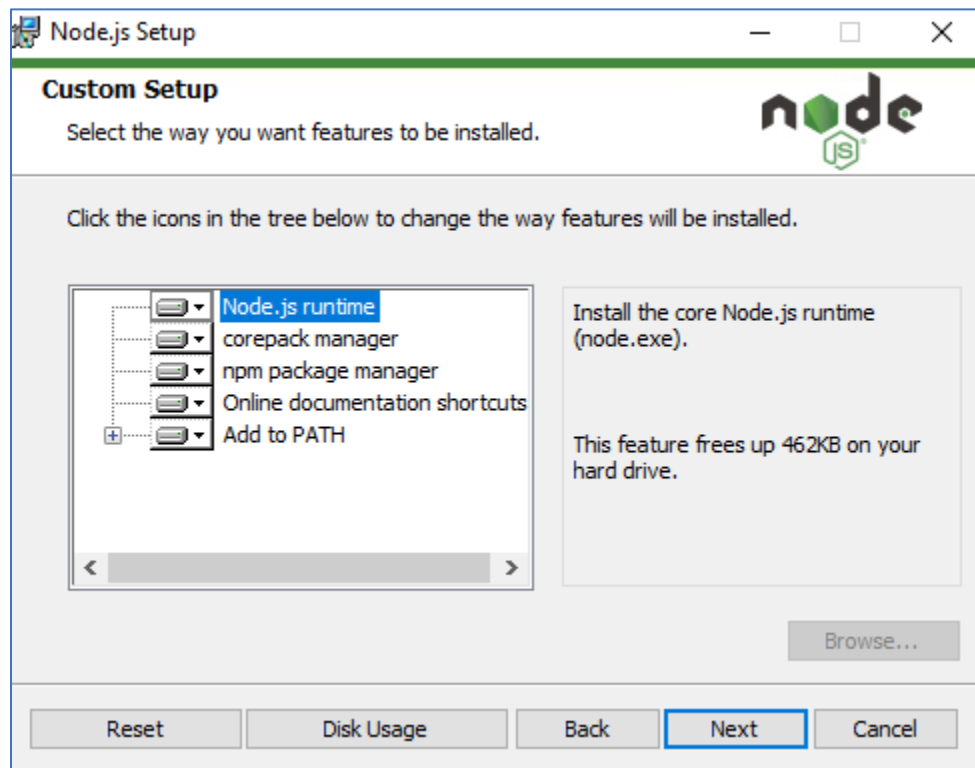


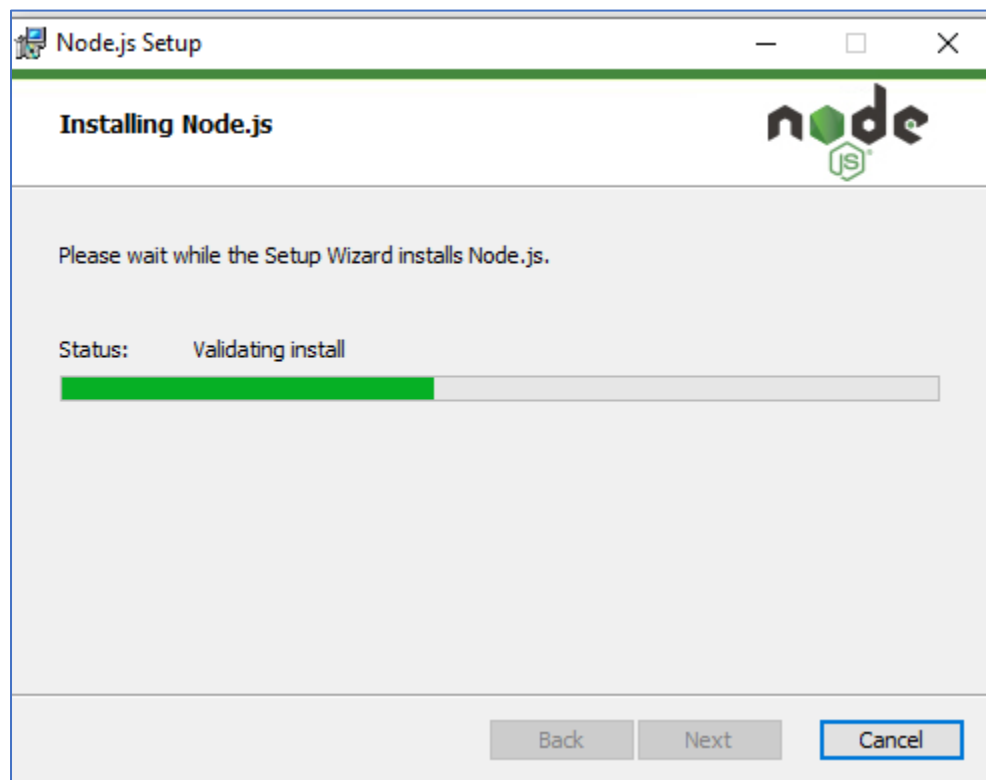
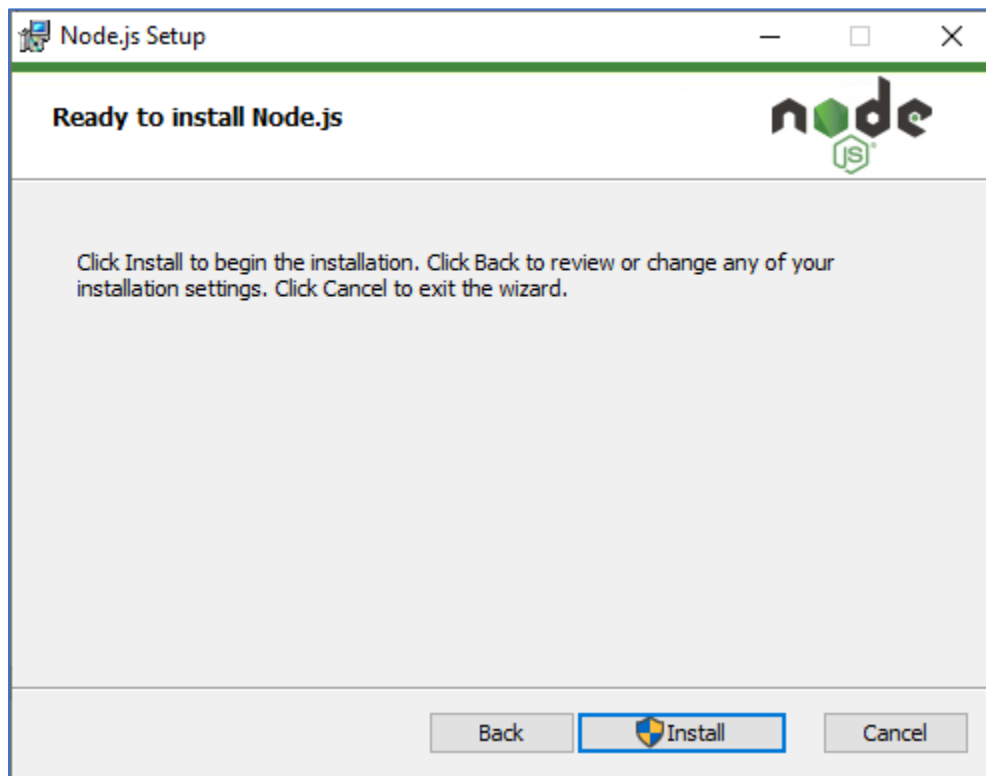
Instructivo de Svelte

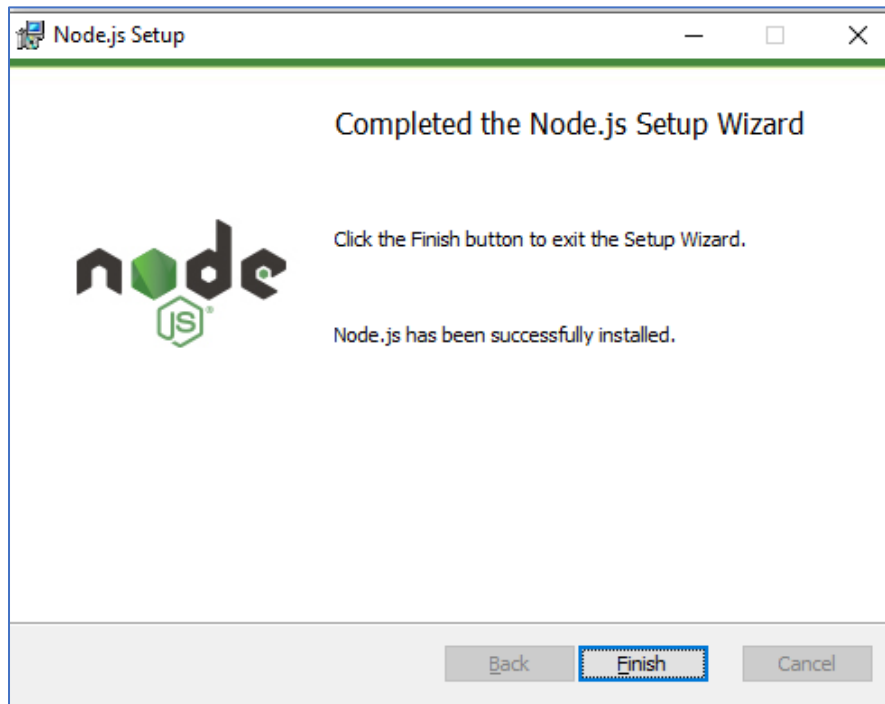
Primero nos dirigimos a la web <https://nodejs.org/en> para realizar la descarga de la última versión de **Node.js** este contiene el gestor de paquetes Node.js (**npm**) que necesitaremos para inicializar un proyecto de **Svelte**.











Iniciamos un proyecto de Svelte llamado app-inicial que en este caso será la interfaz inicial al crear un proyecto de Svelte con Vite.

Abrimos una terminal y verificamos que efectivamente hemos instalado la última versión de Node.js junto a su gestor de paquetes (**npm**).

Se comparte enlace de la web oficial de [Svelte](#) debido a que esta puede surtir cambios en algunas instrucciones para el futuro, pero en este caso trabajaremos con [Vite](#).

App-inicial

Como editor se trabajará con Visual Studio Code donde abrimos una terminal y tecleamos el siguiente comando.

```
npm create vite@latest app-inicial
```

Donde se nos solicita que seleccionemos un framework para el cual elijeremos [Svelte](#).

A screenshot of the Visual Studio Code (VS Code) interface. The Explorer panel on the left shows a file tree with a folder named "SVELTE" containing three files: "~\$structivo de Svelte.docx", "~WRL3844.tmp", and "Instructivo de Svelte.docx". The main editor area is empty. The bottom panel shows the "TERMINAL" tab, which contains the following text:

```
PS D:\svelte> npm create vite@latest app-inicial
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
  Vue
  React
  Preact
  Lit
> Svelte
  Solid
  Qwik
  Others
```

Se nos pide que seleccionemos una variante para trabajar en Svelte a lo que seleccionaremos Javascript.

A close-up screenshot of the VS Code terminal window. The terminal shows the output of the previous command, with "Svelte" selected. The prompt "? Select a variant:" is followed by a list of options: "TypeScript", "JavaScript", and "SvelteKit". The "JavaScript" option is highlighted with a green cursor.

```
PS D:\svelte> npm create vite@latest app-inicial
✓ Select a framework: » Svelte
? Select a variant: » - Use arrow-keys. Return to submit.
  TypeScript
> JavaScript
  SvelteKit ↗
```



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS D:\svelte> npm create vite@latest app-inicial
✓ Select a framework: » Svelte
✓ Select a variant: » JavaScript

Scaffolding project in D:\svelte\app-inicial...

Done. Now run:

  cd app-inicial
  npm install
  npm run dev

PS D:\svelte> █
```

Tecleamos el comando `cd app-inicial`

Luego `npm install`

```
PS D:\svelte> npm create vite@latest app-inicial
✓ Select a framework: » Svelte
✓ Select a variant: » JavaScript

Scaffolding project in D:\svelte\app-inicial...

Done. Now run:

  cd app-inicial
  npm install
  npm run dev

PS D:\svelte> cd app-inicial
PS D:\svelte\app-inicial> npm install

added 37 packages, and audited 38 packages in 12s

3 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\svelte\app-inicial> █
```



y por último `npm run dev`

```
PS D:\svelte\app-inicial> npm run dev

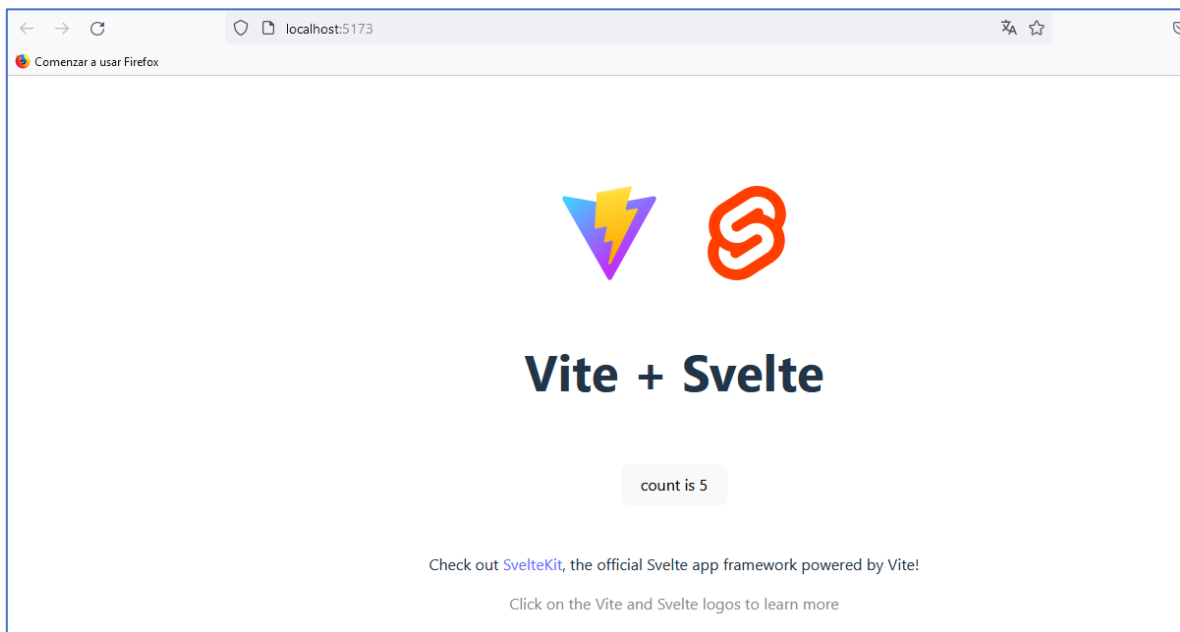
> app-inicial@0.0.0 dev
> vite

Forced re-optimization of dependencies

VITE v5.2.6 ready in 1652 ms

  → Local:   http://localhost:5173/
  → Network: use --host to expose
  → press h + enter to show help
```

Al dar seguir el vinculo <http://localhost:5173/> tenemos nuestra primera app en Svelte.



En el editor de código debería quedarnos una estructura similar a esta:



```
1 <script>
2   import svelteLogo from './assets/svelte.svg'
3   import viteLogo from '/vite.svg'
4   import Counter from './lib/Counter.svelte'
5 </script>
6
7 <main>
8   <div>
9     <a href="https://vitejs.dev" target="_blank" rel="noreferrer">
10       <img src={viteLogo} class="logo" alt="Vite Logo" />
11     </a>
12     <a href="https://svelte.dev" target="_blank" rel="noreferrer">
13       <img src={svelteLogo} class="logo svelte" alt="Svelte Logo" />
14     </a>
15   </div>
16   <h1>Vite + Svelte</h1>
17
18   <div class="card">
19     <Counter />
20   </div>
21
```

Como podemos ver en la imagen, o hay distinción de colores en el código para lo cual instalaremos la extensión **Svelte for VS Code**, también instalaremos **Svelte Intellisense**.

```
1 <script>
2   import svelteLogo from './assets/svelte.svg'
3   import viteLogo from '/vite.svg'
4   import Counter from './lib/Counter.svelte'
5 </script>
6
7 <main>
8   <div>
9     <a href="https://vitejs.dev" target="_blank" rel="noreferrer">
10       <img src={viteLogo} class="logo" alt="Vite Logo" />
11     </a>
12     <a href="https://svelte.dev" target="_blank" rel="noreferrer">
13       <img src={svelteLogo} class="logo svelte" alt="Svelte Logo" />
14     </a>
15   </div>
16   <h1>Vite + Svelte</h1>
17
18   <div class="card">
19     <Counter />
20   </div>
21
```

Podemos observar que nuestro código esta segmentado con colores.



App-par-impar

En la siguiente app identificaremos al capturar un numero si es par o impar, para lo cual iniciaremos un nuevo proyecto aplicando los pasos anteriores.

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS D:\svelte> npm create vite@latest app-par-impar
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
  Vue
  React
  Preact
  Lit
> Svelte
  Solid
  Qwik
  Others
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS D:\svelte> npm create vite@latest app-par-impar
✓ Select a framework: » Svelte
? Select a variant: » - Use arrow-keys. Return to submit.
  TypeScript
> JavaScript
  SvelteKit ↗
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS D:\svelte> npm create vite@latest app-par-impar
✓ Select a framework: » Svelte
✓ Select a variant: » JavaScript

Scaffolding project in D:\svelte\app-par-impar...

Done. Now run:

  cd app-par-impar
  npm install
  npm run dev

PS D:\svelte> █
```



```
cd app-par-impar
npm install
npm run dev

PS D:\svelte> cd app-par-impar
PS D:\svelte\app-par-impar> npm install

added 37 packages, and audited 38 packages in 12s

3 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\svelte\app-par-impar> █
```

Al ejecutar el proyecto en entorno de desarrollo tenemos el mismo proyecto del ejemplo anterior, pero sobre este es que montaremos nuestro nuevo proyecto.

```
PS D:\svelte\app-par-impar> npm run dev

> app-par-impar@0.0.0 dev
> vite

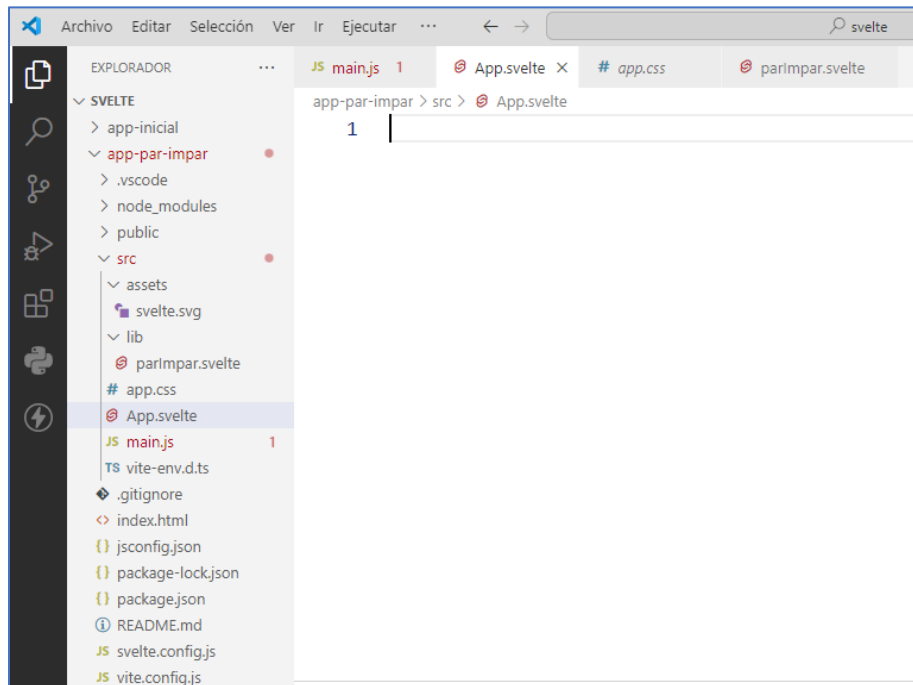
Forced re-optimization of dependencies

VITE v5.2.6 ready in 2365 ms

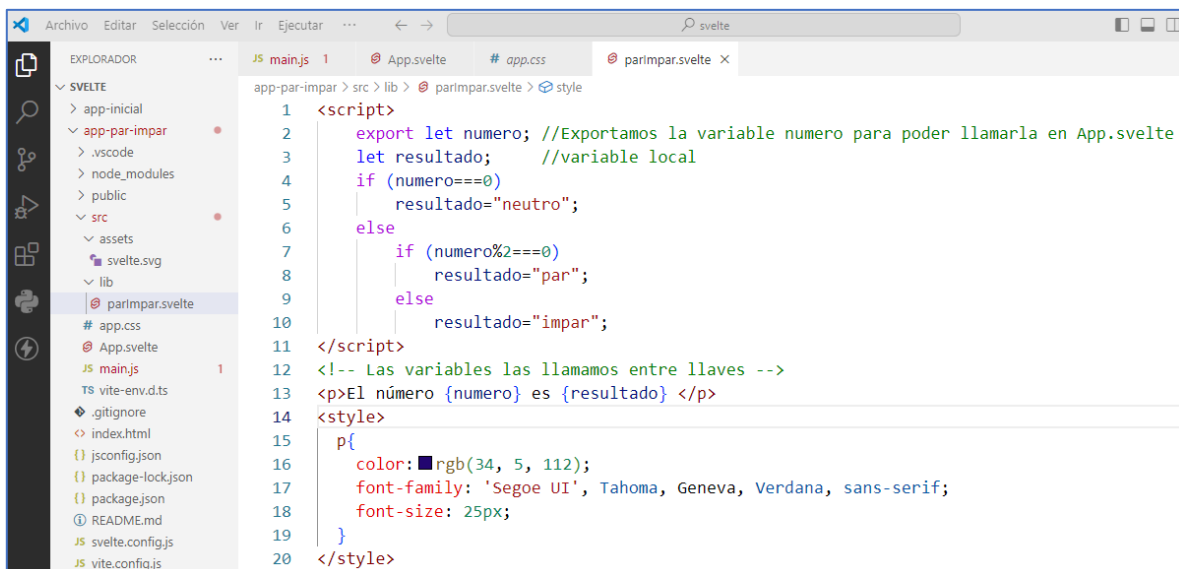
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
█
```

Para ello primero eliminamos la carpeta counter.Svelte que se encuentra en la carpeta lib.

Dentro de App.svelte limpiamos el contenido ubicado dentro de main y style debería quedarnos similar a la siguiente imagen:



Dentro de lib creamos nuestro primer componente llamado **parImpar.svelte** dentro ingresamos el siguiente contenido:



Dentro de App.svelte ingresamos el siguiente código.



```
1 <script>
2   //importamos el componente parImpar.svelte y lo llamamos ParImpar
3   import ParImpar from './lib/parImpar.svelte';
4 </script>
5 <main>
6   <!-- llamamos el componente ingresando un numero impar en este caso 5 -->
7   <ParImpar numero={5}/>
8   <!-- llamamos nuevamente el componente ingresando un numero par en este caso 20 -->
9   <ParImpar numero={20}/>
10  <!-- llamamos el componente ingresando un el número 0 -->
11  <ParImpar numero={0}/>
12 </main>
13
```

La salida de nuestro código es la siguiente:

El número 5 es impar

El número 20 es par

El número 0 es neutro

Agregaremos un input para nuestro ejemplo para ingresar datos por teclado

Por lo cual nuestra aplicación queda de la siguiente manera:



parImpar.svelte

```
JS main.js 1 App.svelte # app.css parImpar.svelte X
app-par-impar > src > lib > parImpar.svelte > script > $: if (numero===0) //se coloca el simbolo $: para ...
1 <script>
2   export let numero; //Exportamos la variable numero para poder llamarla en App.svelte
3   let resultado; //variable local
4   $: if (numero===0) //se coloca el simbolo $: para generar reactividad en el if para resultado
5     resultado="neutro";
6   else
7     if (numero%2===0)
8       resultado="par";
9     else
10      resultado="impar";
11 </script>
12 <!-- Las variables las llamamos entre llaves -->
13 <p>El número {numero} es {resultado} </p>
14 <style>
15   p{
16     color: rgb(34, 5, 112);
17     font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
18     font-size: 25px;
19   }
20 </style>
```

App.svelte

```
JS main.js 1 App.svelte X # app.css parImpar.svelte
app-par-impar > src > App.svelte > main
1 <script>
2   //importamos el componente parImpar.svelte y lo llamamos ParImpar
3   import ParImpar from '../lib/parImpar.svelte';
4   let valor=0;
5   // escuchamos el evento `input`
6   const listenInput = (event) => valor = event.target.value
7 </script>
8
9 <main>
10 <!-- llamamos el componente que inicialmente procesa el valor de 0 -->
11 <ParImpar numero={valor}/>
12 <!-- En el elemento input se coloca en value la variable valor que vale 0 -->
13 <!-- Tambien se llama el evento on:input para que reaccione al ingresar los números -->
14 <input value={valor} on:input={listenInput}>
15 </main>
```

Pantallazos del resultado

El número 8 es par



El número 0 es neutro

El número 9 es impar

App-paises-suramerica

Ahora elaboraremos una app con una breve descripción para los países de Suramérica por lo cual crearemos un objeto json con los datos mas importantes de los mismos

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS D:\svelte> npm create vite@latest app-paises-suramerica
? Select a framework: » - Use arrow-keys. Return to submit.
>  Vanilla
  Vue
  React
  Preact
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
  Vue
  React
  Preact
  Lit
>  Svelte
  Solid
  Qwik
  Others
```

```
✓ Select a framework: » Svelte
? Select a variant: » - Use arrow-keys. Return to submit.
  TypeScript
>  JavaScript
  SvelteKit ↗
```



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS D:\svelte> cd app-paises-suramerica
PS D:\svelte\app-paises-suramerica> npm install

added 37 packages, and audited 38 packages in 17s

3 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\svelte\app-paises-suramerica> npm run dev

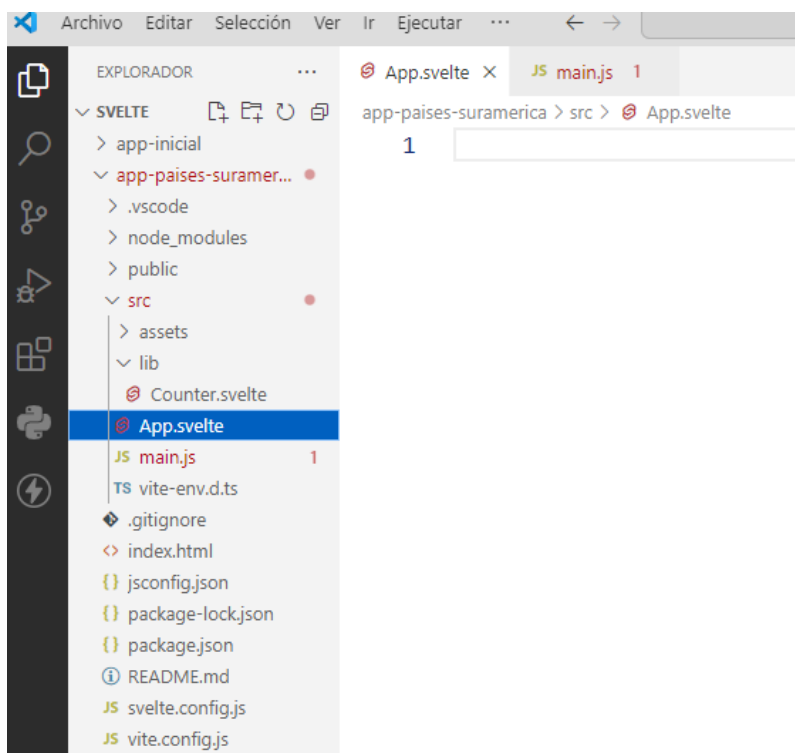
> app-paises-suramerica@0.0.0 dev
> vite

Forced re-optimization of dependencies

VITE v5.2.6 ready in 2319 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

Una vez con la estructura de archivos y carpetas eliminamos counter.svelte y limpiamos App.svelte





Dentro de lib creamos un componente llamado países.svelte y agregamos el siguiente código:

```
1 <script>
2   export let pais;
3   let p;
4   /*[0]=Nombre_pais [1]=capital [2]=poblacion [3]=moneda [4]=área [5]=bandera*/
5   const países={
6     "colombia":["Colombia","Bogota","51,52 millones","Peso colombiano","1,142 millones km²","Flag_of_Colombia.svg"],
7     "venezuela":["Venezuela","Caracas","28,2 millones","Bolivar"," 916.445 km²","Flag_of_Venezuela.svg"],
8     "ecuador":["Ecuador","Quito","17,8 millones","Dólar estadounidense","256.370 km²","Flag_of_Ecuador.svg"],
9     "peru":["Peru","Lima","33,72 millones","Sol peruano","1.285.215 km²","800px-Flag_of_Peru_(state).svg.webp"],
10    "brasil":["Brasil","Brasilia","214,3 millones","Real brasileño","8,51 millones km²","Flag_of_Brazil.svg"]
11  }
12  p=países[pais];
13 </script>
14
15 <div class="main">
16   <div class="col1">
17     
18   </div>
19   <div class="col2">
20     <div class="linea"><strong>País:</strong> {p[0]}</div>
21     <div class="linea"><strong>Capital:</strong> {p[1]}</div>
22     <div class="linea"><strong>Población:</strong> {p[2]}</div>
23     <div class="linea"><strong>Moneda:</strong> {p[3]}</div>
24     <div class="linea"><strong>Extensión:</strong> {p[4]}</div>
25   </div>
26 </div>
27
```

Continuación:

```
27
28 <style>
29   .main {
30     display: grid;
31     grid-template-columns: repeat(4, 1fr);
32     grid-gap: 5px;
33     grid-auto-rows: minmax(100px, auto);
34     border-radius: 10px;
35     border-style: solid;
36     border-color: rgb(186, 190, 182);
37     border-width: 2px;
38     margin: left;
39     width: 600px;
40     box-shadow: 5px 5px 5px black;
41     margin-right: 5px;
42     margin-bottom: 10px;
43   }
44   .col1{
45     grid-column: 1;
46     grid-row: 1;
47     margin: 5px;
48     padding: 5px;
49   }
50   .col2{
51     grid-column: 2;
52     grid-row: 1;
53     margin: 5px;
54     padding: 5px;
55     border-color: black;
56     width: max-content;
57   }

```



Continuación:

```
58     strong{
59         color: blue;
60         font-weight: bolder;
61     }
62     .linea{
63         font-size: 20px;
64         margin-bottom: 18px;
65     }
66 </style>
```

Adicionalmente en la carpeta src/assets debemos agregar las imágenes de las banderas de los países suramericanos, este ejemplo esta creado con 5 países pero debe ampliarse a los demás

En App.svelte agregamos el siguiente código:

```
1 <script>
2   import Países from "../lib/paises.svelte";
3   const lista=["colombia","venezuela","ecuador","peru","brasil"]
4 </script>
5 <div class="main">
6   {#each lista as pais}
7     <div>
8       <Países {pais}/>
9     </div>
10  {:else}
11    No existen elementos
12  {/each}
13 </div>
14 <svelte:head>
15   <title>Países de suramerica</title>
16 </svelte:head>
17 <style>
18   .main {
19     display: grid;
20     grid-template-columns: repeat(2, 1fr);
21     grid-gap: 10px;
22     grid-auto-rows: minmax(100px, auto);
23   }
24 </style>
```



La salida debería quedar de la siguiente manera:

	Pais: Colombia Capital: Bogota Población: 51,52 millones Moneda: Peso colombiano Extensión: 1,142 millones km ²		Pais: Venezuela Capital: Caracas Población: 28,2 millones Moneda: Bolivar Extensión: 916.445 km ²
	Pais: Ecuador Capital: Quito Población: 17,8 millones Moneda: Dólar estadounidense Extensión: 256.370 km ²		Pais: Peru Capital: Lima Población: 33,72 millones Moneda: Sol peruano Extensión: 1.285.215 km ²
	Pais: Brasil Capital: Brasilia Población: 214,3 millones Moneda: Real brasileño		