

## **Práctica 1**

**DESARROLLO WEB EN  
ENTORNO SERVIDOR**



## Índice

<b>Práctica 1</b>	<b>1</b>
<b>1. Diferencia entre páginas web, aplicación de escritorio y aplicaciones web.</b>	<b>3</b>
<b>2. Instalar el software XAMPP.</b>	<b>4</b>
<b>3. Configurar XAMPP para su correcto funcionamiento.</b>	<b>6</b>
<b>4. Crea tu primera página con PHP.</b>	<b>7</b>
<b>5. Instalar Node.js.</b>	<b>9</b>
<b>6. Instalar Typescript.</b>	<b>11</b>
<b>7. Crea un servidor web con Node.js.</b>	<b>13</b>
<b>8. Bibliografía</b>	<b>15</b>



# Práctica 1.

## 1. Diferencia entre páginas web, aplicación de escritorio y aplicaciones web.

Una página web es un conjunto de páginas **estáticas** que entregan información a las que podemos acceder desde un navegador web. La información que nos entregan puede ser dinámica o estática. Y no requieren instalación en nuestro equipo ya que se ejecutan en un servidor remoto.

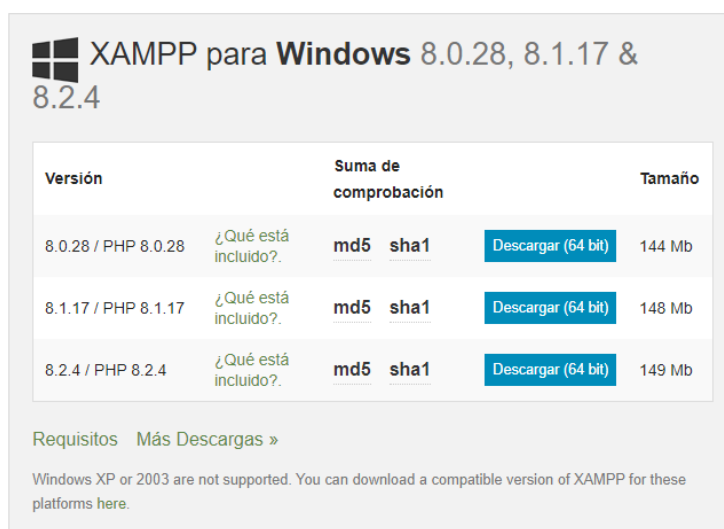
Sin embargo, las aplicaciones web son plataformas principalmente interactivas o **dinámicas** que se centran en que los usuarios puedan realicen acciones. Es decir, son programas que se ejecutan en un navegador web, similar a las páginas web, pero con funcionalidad más avanzada.

Esta sería la diferencia fundamental entre una página web y una aplicación web.

Por otra parte, una aplicación de escritorio es aquella que está diseñada para ser **instalada y ejecutada en un dispositivo**, esta podría no necesitar conexión a internet algo que es necesario en las páginas web y aplicaciones web. Una de sus mayores ventajas es la posibilidad de implementar cualquier función al contrario que una aplicación web.

## 2. Instalar el software XAMPP.

Primero nos descargamos xampp desde la página oficial de Xampp



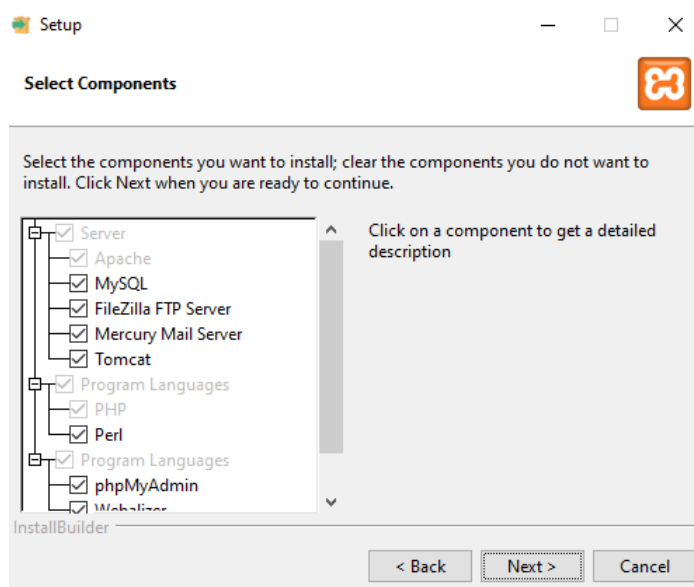
XAMPP para Windows 8.0.28, 8.1.17 & 8.2.4

Versión	¿Qué está incluido?	Suma de comprobación	Tamaño
8.0.28 / PHP 8.0.28	<a href="#">¿Qué está incluido?</a>	md5 sha1	144 Mb
8.1.17 / PHP 8.1.17	<a href="#">¿Qué está incluido?</a>	md5 sha1	148 Mb
8.2.4 / PHP 8.2.4	<a href="#">¿Qué está incluido?</a>	md5 sha1	149 Mb

[Requisitos](#) [Más Descargas »](#)

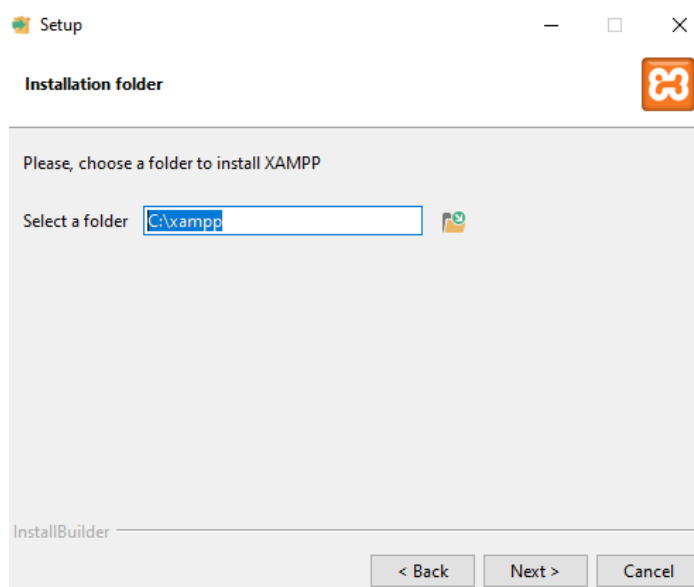
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Elegimos los complementos que queremos instalar junto con XAMPP

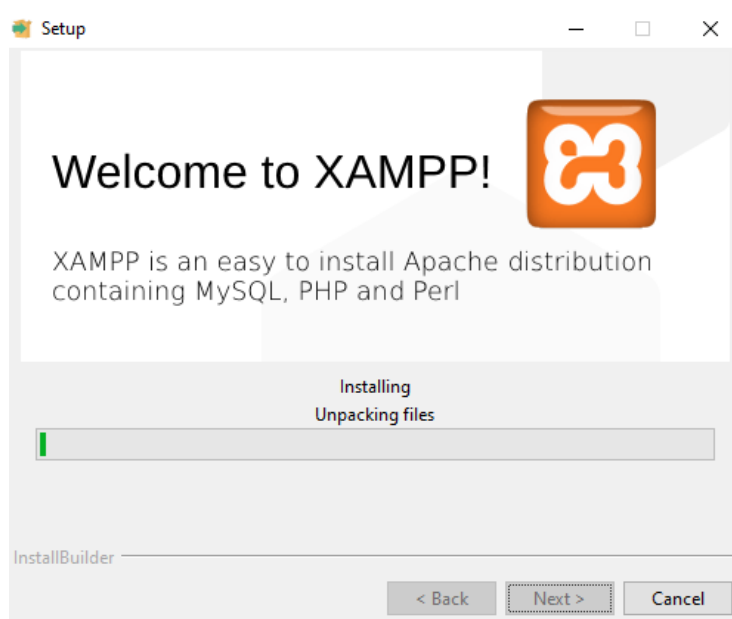




A continuación elegiremos la ruta de nuestra instalación



Después de esto comenzará nuestra instalación

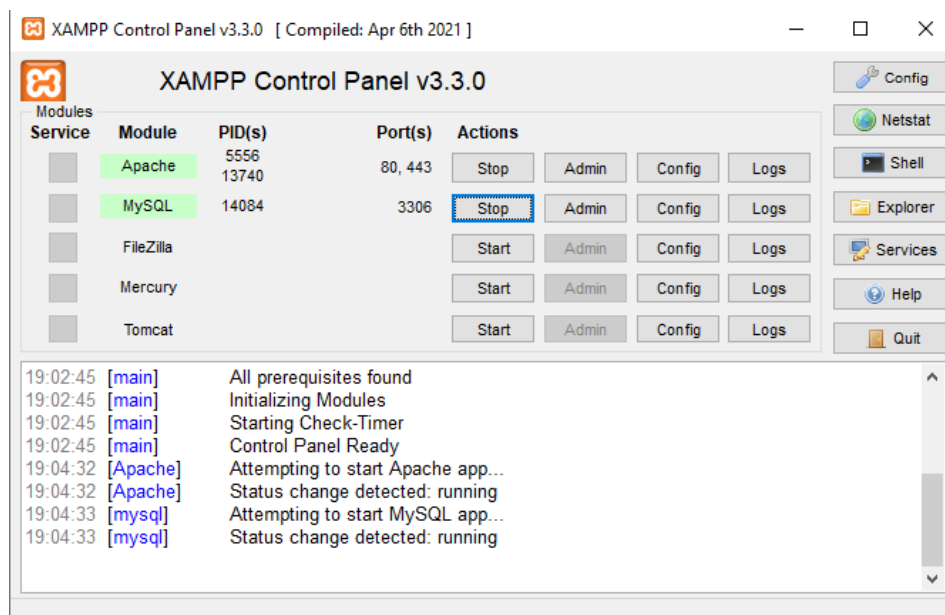


Una vez termine ya tendremos instalada nuestra aplicación.

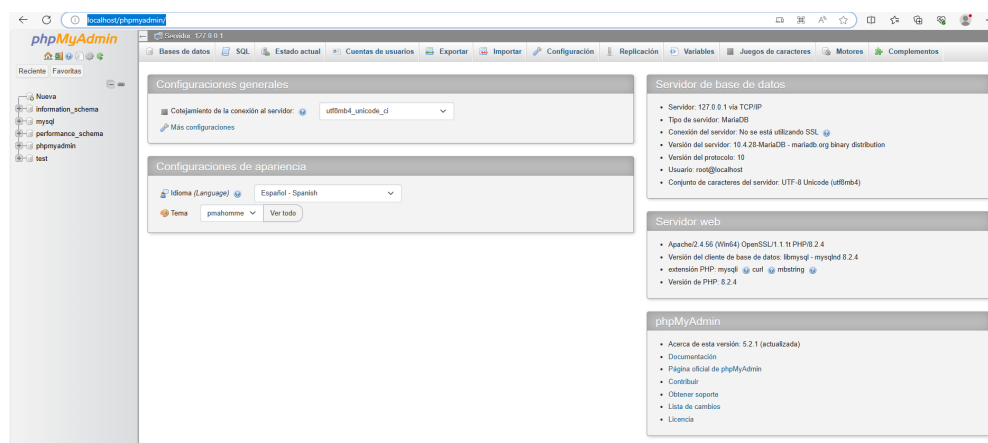


### 3. Configurar XAMPP para su correcto funcionamiento.

Ejecutamos XAMPP como administrador y ejecutamos los servicios de Apache y MySQL



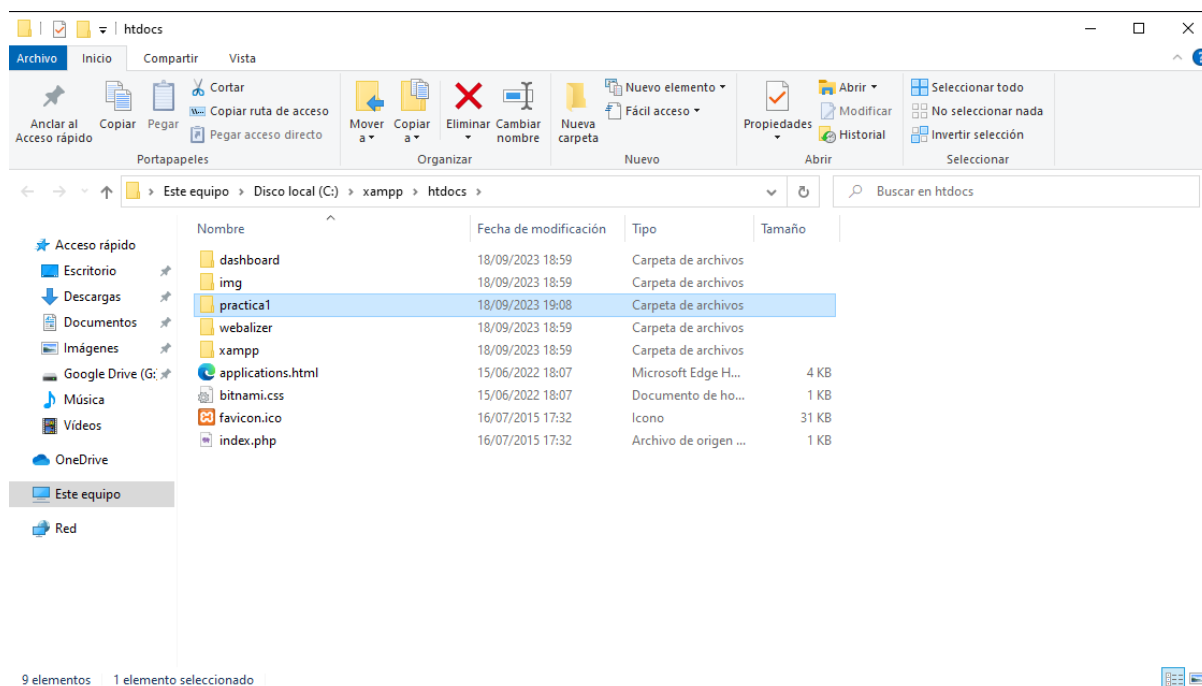
Ahora mismo ya podremos utilizar la url localhost/phpmyadmin y nos saldrá nuestro panel de administración para php, en el cual podremos configurar xampp





## 4. Crea tu primera página con PHP.

Nos iremos a la ruta donde hemos instalado XAMPP y entraremos en htdocs, dentro crearemos una carpeta para nuestro proyecto.

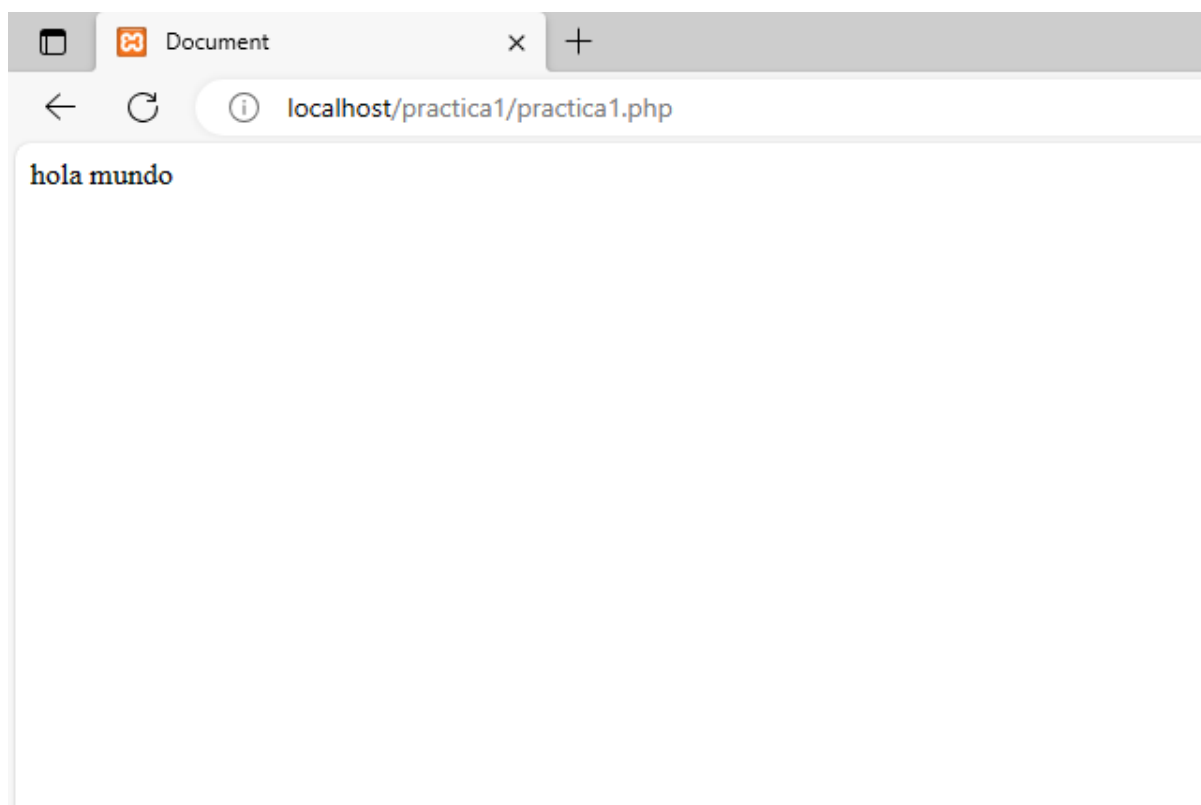




Ahora crearemos un fichero .php muy básico para nuestro ejemplo.

```
practica1.php X
htdocs > practica1 > practica1.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <?php
10         echo "hola mundo";
11     ?>
12 </body>
13 </html>
14
```

Si lo ejecutamos en nuestro navegador tendremos el siguiente resultado.





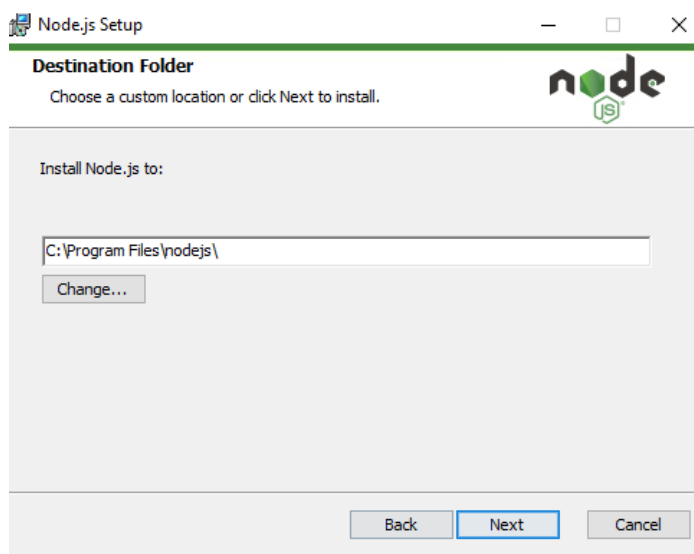


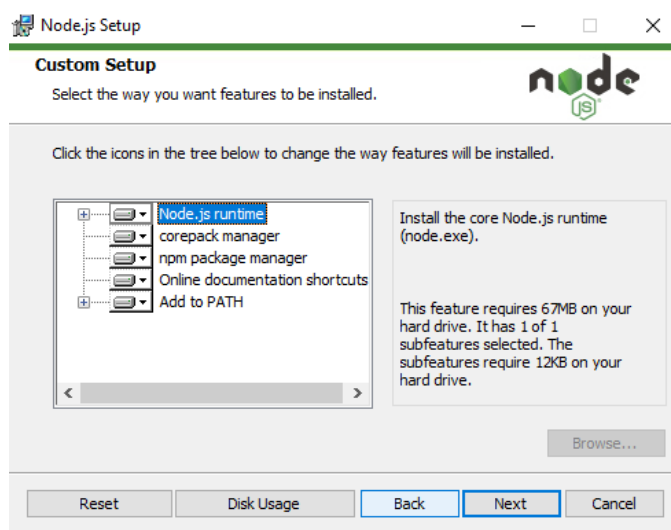
## 5. Instalar Node.js.

Primero nos descargamos nodejs desde la pagina oficial

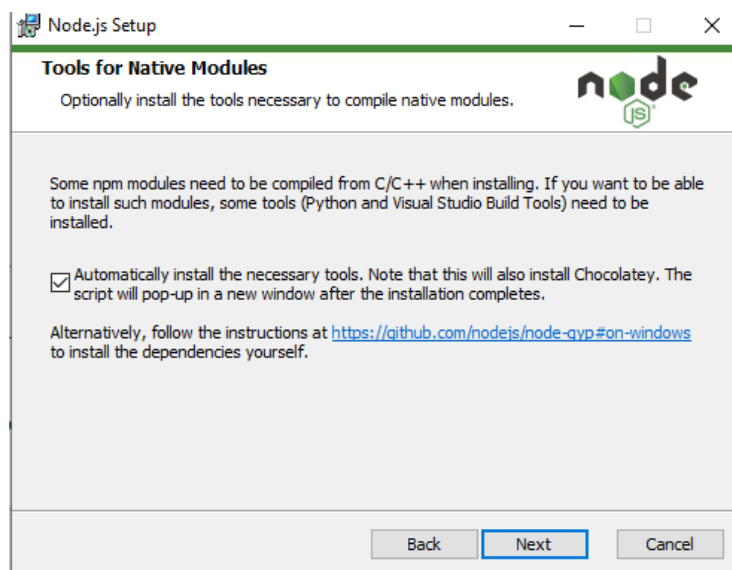


Ejecutamos nuestro archivo y elegimos la ruta donde deseamos instalar nuestro programa





Podemos instalar también otros módulos necesarios para el funcionamiento de nodejs, así como chocolatey que facilitan el funcionamiento de nodejs.



Luego solo tendremos que darle a siguiente hasta que finalice la instalación.



Con `node -v` podemos comprobar que nodejs funciona correctamente.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS G:\Mi unidad> node -v
v18.17.1
PS G:\Mi unidad> 
```

## 6. Instalar Typescript.

Voy a instalarlo mediante comando de bash siguiendo los pasos que vienen la página oficial de typescript.

via npm

TypeScript is available as a [package on the npm registry](#).  
available as `"typescript"`.

You will need a copy of [Node.js](#) as an environment to run the package. Then you use a dependency manager like [npm](#), [yarn](#) or [pnpm](#) to download TypeScript into your project.

```
npm install typescript --save-dev
```

npm yarn pnpm

All of these dependency managers support lockfiles, ensuring that everyone on your team is using the same version of the language. You can then run the TypeScript compiler using one of the following commands:

```
npx tsc
```

npm yarn pnpm



Ejecutamos el comando que nos aparece en la guía y veremos como se nos ha instalado typescript y la versión del mismo.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

AlumnoT@DESKTOP-RMV28MC MINGW64 /c/xampp/htdocs
$ npm install typescript --save-dev

added 1 package in 14s
npm notice
npm notice New major version of npm available! 9.6.7 -> 10.1.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.1.0
npm notice Run npm install -g npm@10.1.0 to update!
npm notice

AlumnoT@DESKTOP-RMV28MC MINGW64 /c/xampp/htdocs
$
```



## 7. Crea un servidor web con Node.js.

Nos ubicamos en la carpeta donde vamos a crear nuestro servidor de nodejs

```
AlumnoT@DESKTOP-RMV28MC MINGW64 /g/Mi unidad
$ cd 2º\ DAW/

AlumnoT@DESKTOP-RMV28MC MINGW64 /g/Mi unidad/2º DAW
$ cd ES/

AlumnoT@DESKTOP-RMV28MC MINGW64 /g/Mi unidad/2º DAW/ES
$ cd Práctica1/

AlumnoT@DESKTOP-RMV28MC MINGW64 /g/Mi unidad/2º DAW/ES/Práctica1
$ cd servidornodejs/

AlumnoT@DESKTOP-RMV28MC MINGW64 /g/Mi unidad/2º DAW/ES/Práctica1/servidornodejs
$
```

Creamos el archivo con el código de nuestro servidor.

```
AlumnoT@DESKTOP-RMV28MC MINGW64 /g/Mi unidad/2º DAW/ES/Práctica1/servidornodejs
$ touch servidorpractica1.js
```

Agregaremos el módulo http a nuestro archivo de servidor con nano servidorpractica1.js

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
GNU nano 7.2 servidorpractica1.js Modified
const http = require("http");
```

Lo siguiente será agregar el host y el puerto de nuestro servidor en este archivo.

```
GNU nano 7.2
const http = require("http");
const host = 'localhost';
const port = 8000;
```

Ahora, añadiremos una función especial, que en Node.js llamamos una escucha de solicitudes. Esta función está destinada a gestionar una solicitud HTTP entrante y devolver una respuesta HTTP. Yo le he puesto el mensaje Mi primer servidor para comprobar que funciona correctamente posteriormente.

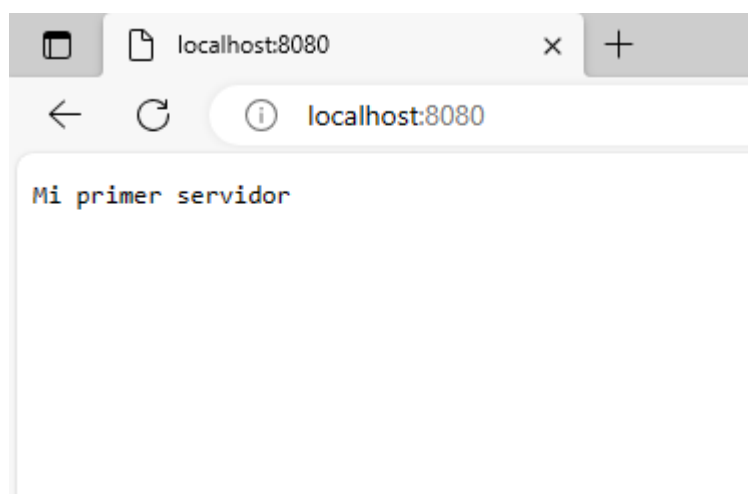
```
GNU nano 7.2
const http = require("http");
const host = 'localhost';
const port = 8080;

const requestListener = function (req, res) {
  res.writeHead(200);
  res.end("Mi primer servidor");
};
```

Agregaremos la función de server para lanzar arriba el servidor.

```
const server = http.createServer(requestListener);
server.listen(port, host, () => {
  console.log(`Server is running on http://${host}:${port}`);
});
```

Podemos comprobar que el servidor funciona correctamente.





## 8. Bibliografía

Aplicación de escritorio o aplicación web: pros y contras. (2021). *AppMaster - ultimate all-in no-code platform*. <https://appmaster.io/es/blog/aplicacion-de-escritorio-o-aplicacion-web-pros-y-contras>

Abuse, S. (2020). Cómo crear un servidor web en Node.js con el módulo HTTP. *DigitalOcean*. <https://www.digitalocean.com/community/tutorials/how-to-create-a-web-server-in-node-js-with-the-http-module-es>