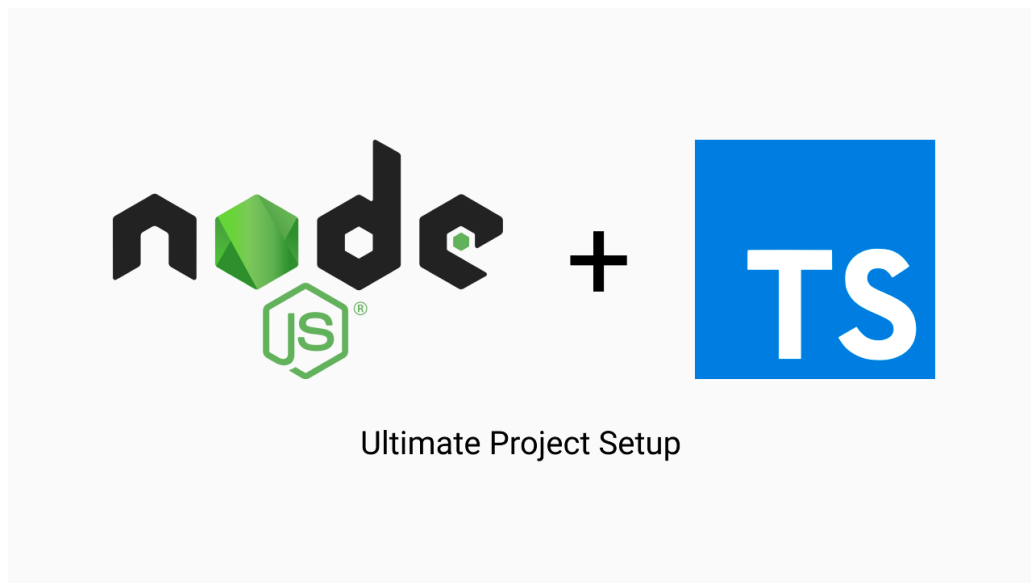


## Node y TypeScript



### Node.js:

[Node.js](#) es un entorno de código abierto en tiempo de ejecución multiplataforma, para la capa del servidor JavaScript. Es asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de google.

### TypeScript:

[TypeScript](#) es un lenguaje de programación libre y de código abierto, desarrollado y mantenido por Microsoft. Tiene la base de JavaScript, añadiéndole los tipos estáticos y objetos basados en clases.

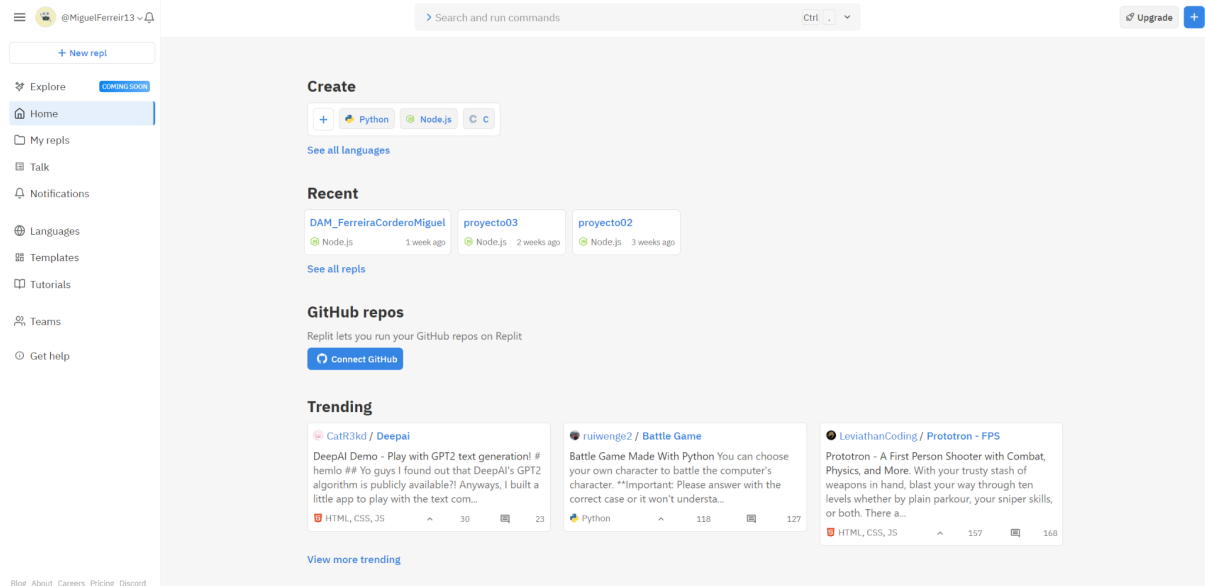
### Replit:

[Repl.it](#) es un lenguaje de programación que permite a los usuarios escribir código y crear aplicaciones y sitios web a través de un navegador.

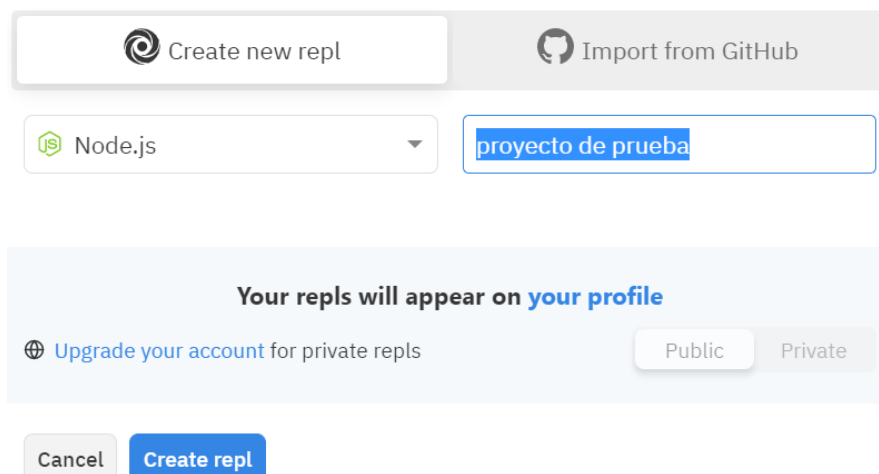
El sitio también tiene varias características de colaboración, incluida la capacidad para la edición multiusuario en tiempo real con chat para la comunicación en tiempo real. Admite más de 50 lenguajes de programación y marcado, incluidos Java, Python y HTML. El sitio está integrado con GitHub, proporcionando una forma de importar y ejecutar proyectos desde este. Los usuarios también pueden importar proyectos desde Glitch, que da un servicio similar al de Replit.

## Primeros pasos con Repl.it

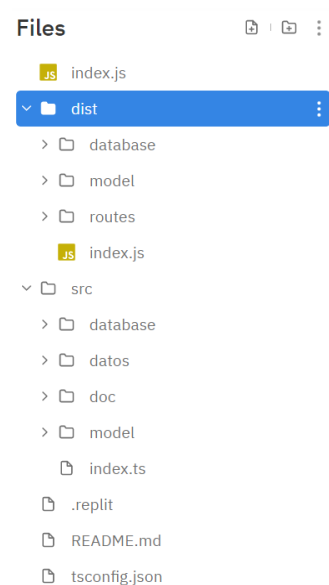
Lo primero que tendremos que hacer es crearnos una cuenta en repl.it. Si ya la tenemos iniciamos sesión y se nos redirigirá la página principal. Una vez dentro crearemos un proyecto pinchando en el botón + de arriba a la izquierda, tal y como se muestra en la siguiente imagen:



Decimos que queremos crearlo con el lenguaje Node.js y le ponemos nombre al repl:



Luego, para tener el proyecto bien ordenado y estructurado tendremos que crear la siguiente estructura de directorios:



A continuación nos dirigiremos a la consola e introduciremos los siguientes comandos:

```
npm i express mongoose  
npm i typescript -D  
npm i @types/express @types/mongoose @types/node@14.14.30  
npm install @types/cors  
tsc --init
```

Luego nos dirigimos al archivo `.replit` y escribimos en él `run = "npm start"`.  
En el archivo `index.ts` estará todo el código.

Para obtener los paquetes json clickamos en packages y seleccionamos json. Así, se crearán dos archivos `.json`.

Ahora en el archivo `"tsconfig.json"`, que es el cual nos permitirá trabajar con el lenguaje Typescript y en él introducimos:

```
"target": "es5"  
"outDir": "./dist"  
"rootDir": "src"
```

Por último en el `package.json` introducimos de nuevo:

```
"ts": "tsc -w"  
"dev": "nodemon ./dist/index.js"  
"start": "node ./dist/index.js"
```

Hecho todo lo anterior estaremos listos para empezar a programar.

Una vez programado lo que quedamos, tendremos que escribir en la shell el comando `"tsc"` para que el programa se compile antes de ejecutarlo, o si vamos a hacer varios cambios y queremos ver los fallos en tiempo real podremos compilar en watch mode, introduciendo `"tsc -w"`.