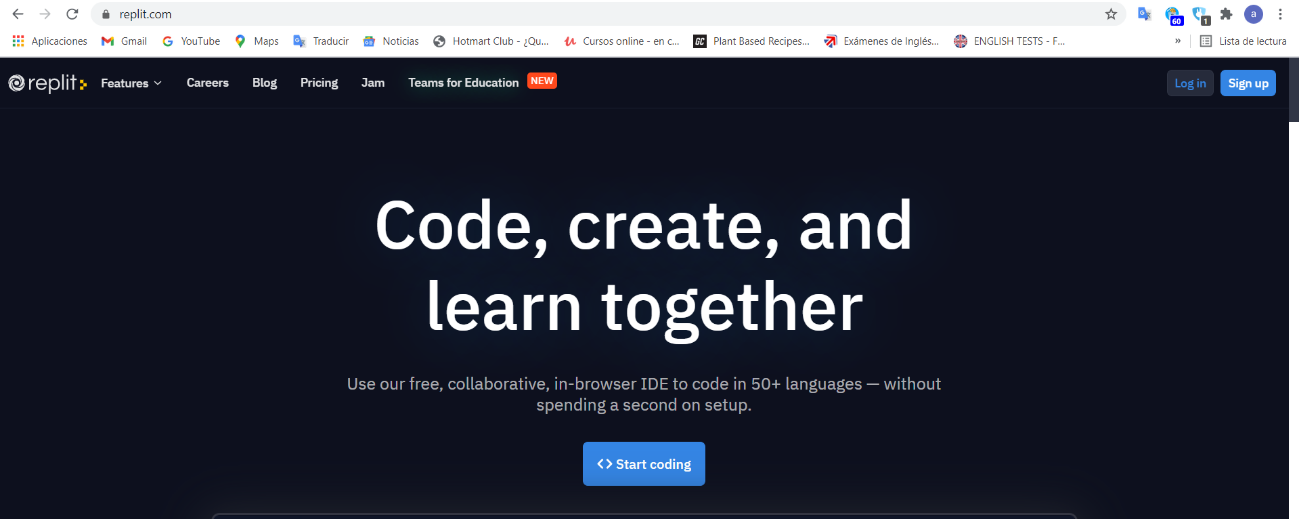
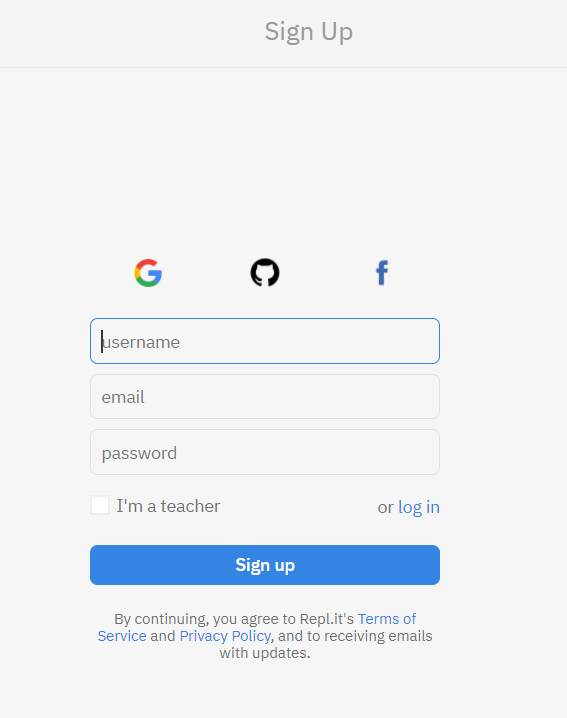
**Configuración de un entorno de desarrollo Node en Repl.it**

1. **REPL.IT:**

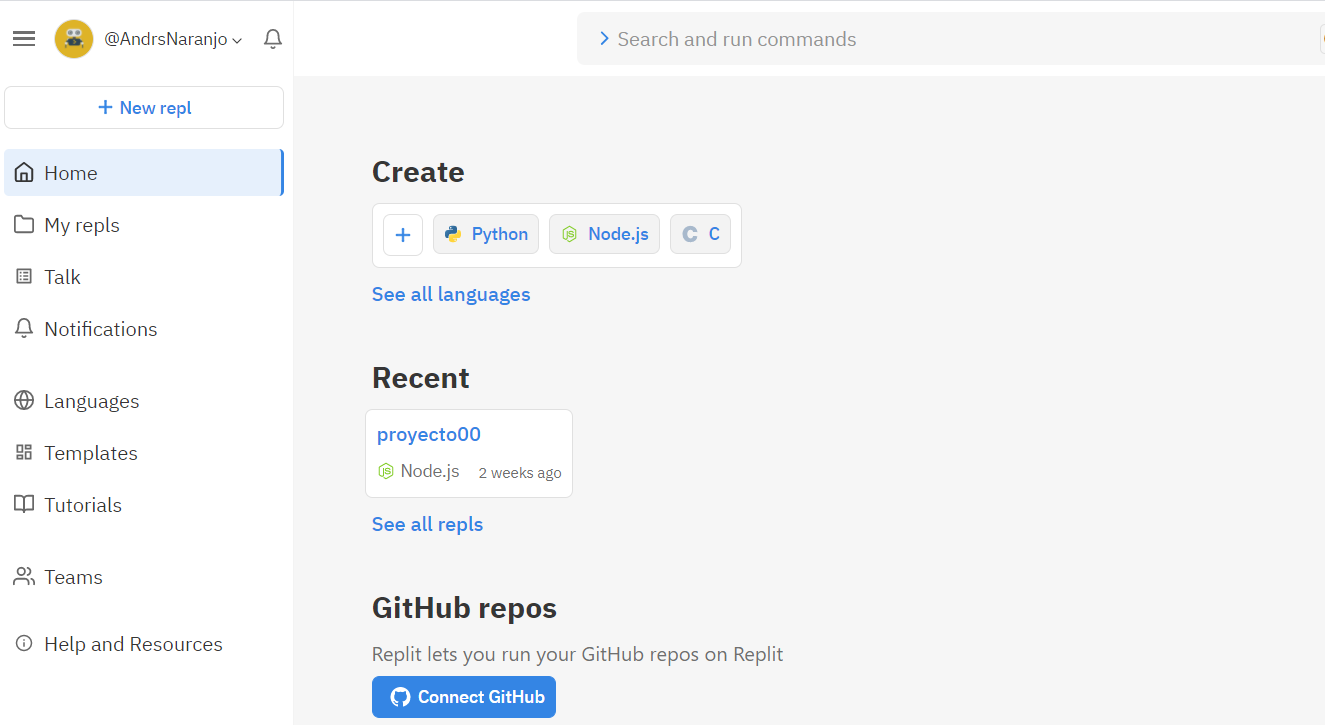
Repl.it es un IDE (entorno de desarrollo integrado) que permite a los usuarios escribir sus propios programas y códigos en numerosos lenguajes diferentes. Mientras que los IDE tradicionales son softwares que se descargan en nuestro sistema local, Repl.it es un IDE basado en plataforma web y se puede acceder a el desde cualquier dispositivo conectado a internet.



Para crear una cuenta en Repl.it solo tenemos que clicar en Sign Up y seleccionar la manera de registro que más nos convenga.



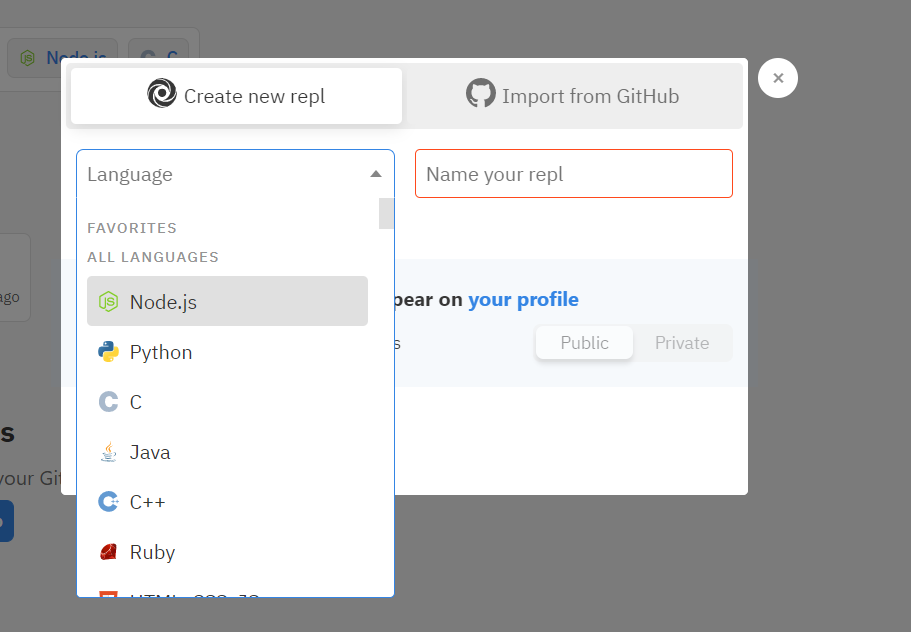
Como se ve en la imagen, además del método de registro habitual, la plataforma nos permite usar un token de registro usando la información de cualquiera de nuestras cuentas de Facebook, GitHub y Google Account. Para una mayor comodidad y rapidez se ha elegido la opción de Google.

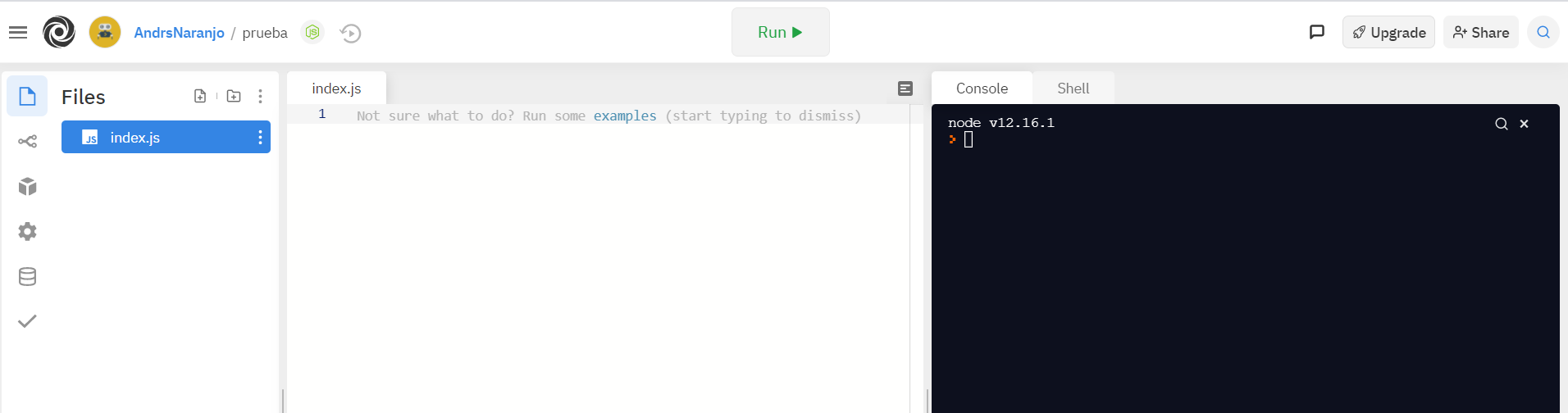


Para crear un nuevo repl solo hay que clicar en New Repl. En el menú de la izquierda tenemos diferentes opciones, como la de Home (página principal), My repls (todos nuestros repls), talk (noticias) y notificaciones.

Mas abajo tenemos lenguajes, donde nos aparecen la multitud de lenguajes soportados por la plataforma, plantillas, y tutoriales.

1. **Crear un Repl**

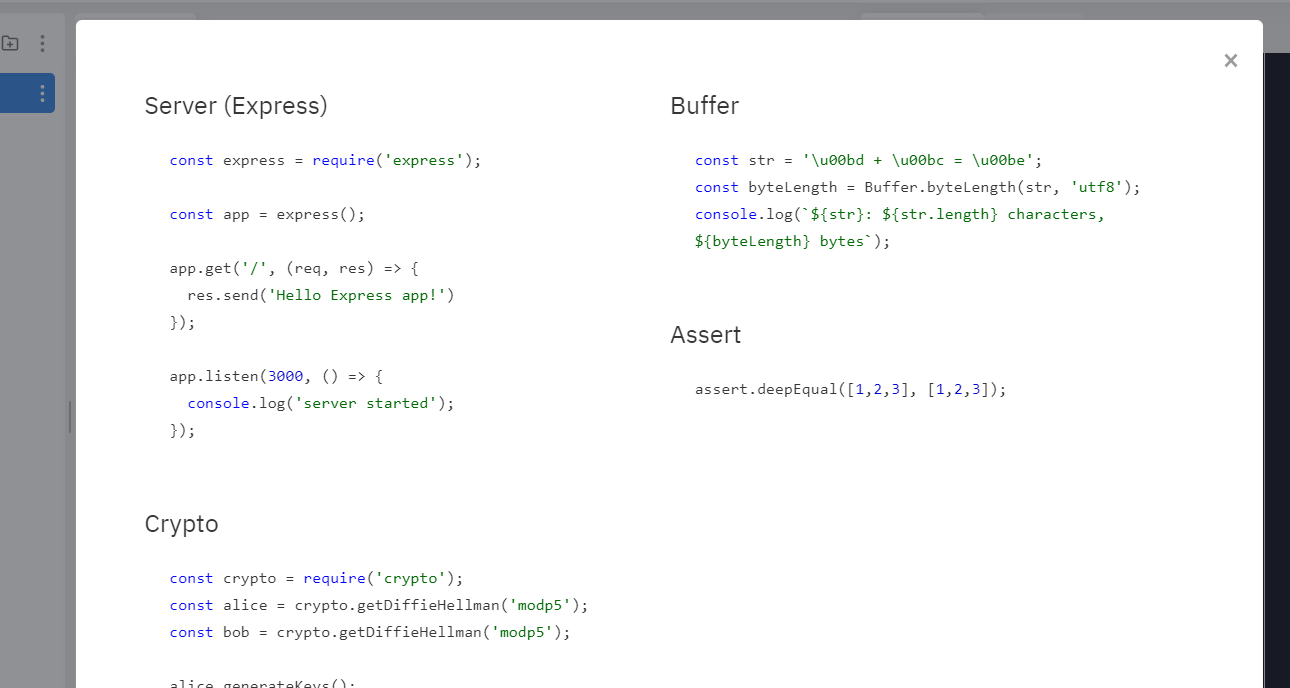
En esta asignatura se va a trabajar con Node.js. asique esa será la opción que elegiremos a la hora de la creación del nuevo Repl.



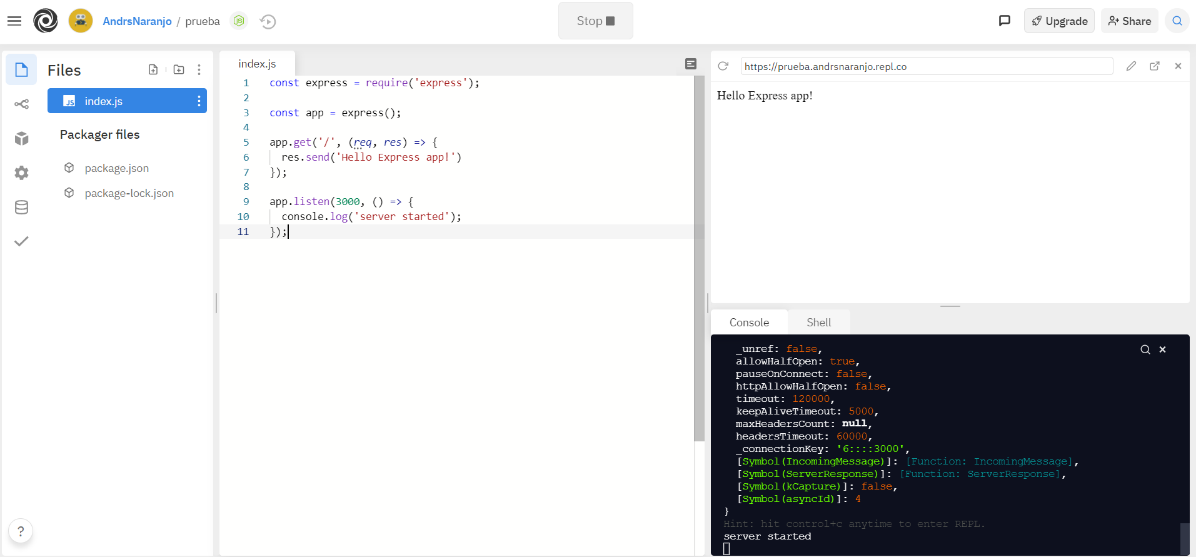
Una vez que creamos el repl, lo primero que nos aparecerá es lo que nos muestra la imagen anterior. Para que nuestra API empiece a funcionar debemos instalar el ambiente de desarrollo EXPRESS.

El ambiente de desarrollo EXPRESS incluye una instalación de Nodejs, el NPM administrador de paquetes y el generador de aplicaciones de Express en el PC.

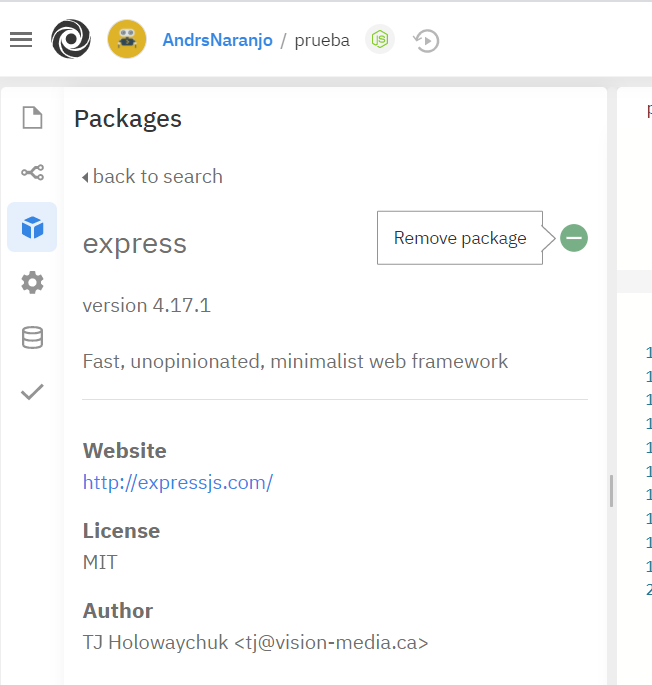
Para instalar Express podemos dirigirnos a “examples”.



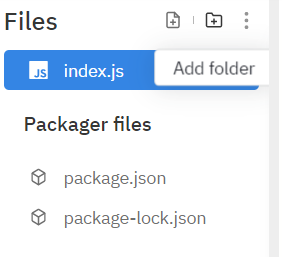
Clicamos sobre server Express y nos lo instalará.

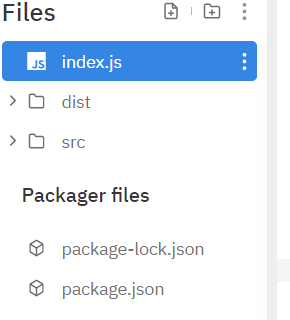


\*\*Otra opción es clicar sobre el cubo “packages”, escribir “express” y añadir paquete.

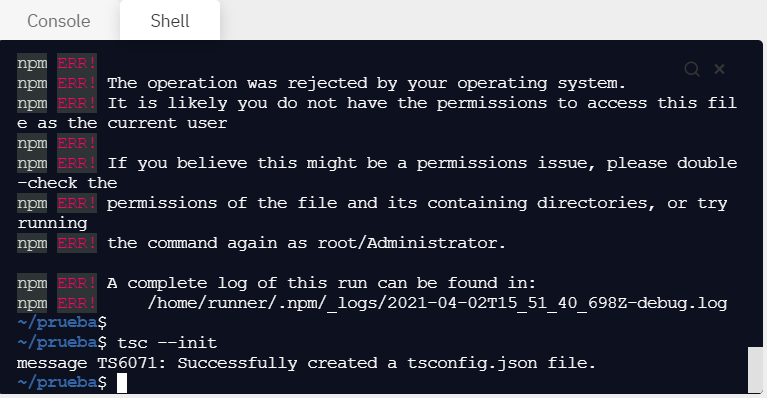


Para cualquier trabajo en repl es necesario crear dos carpetas: DIST y SRC.

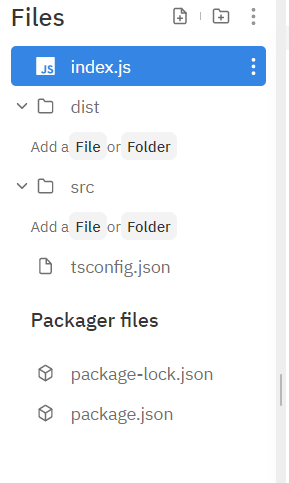


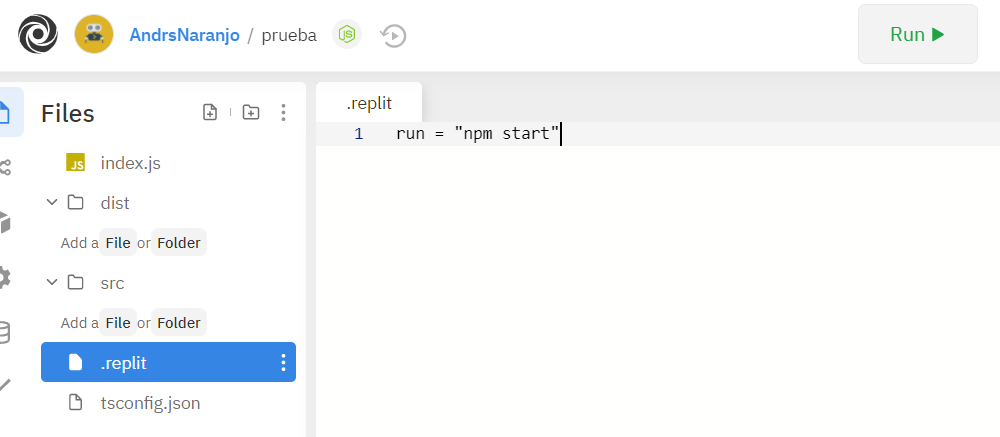


Como vamos a trabajar con el lenguaje TypeScript, un derivado de JAVA, necesitamos cargarlo a nuestro entorno de desarrollo, ya que no viene por defecto. Para ejecutarlo debemos poner en la Shell el siguiente comando.



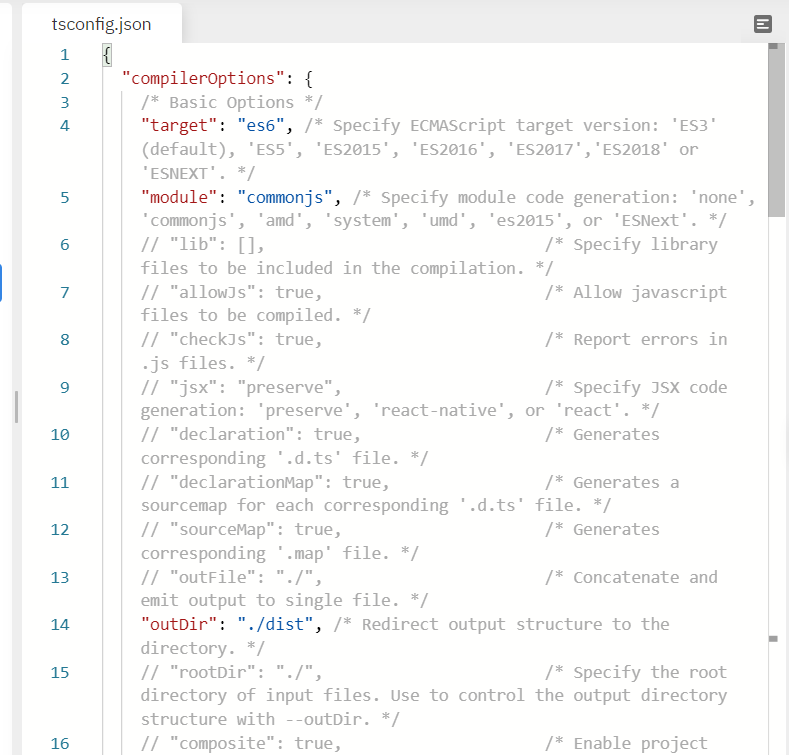
Si lo hemos hecho correctamente, nos saldrá en nuestro menú, el archivo tsconfig.json, lo que nos indica que TypeScript está listo para usarse.





Para la configuración del botón RUN, debemos crear un archivo propio del entorno. Para ello clicamos en nuevo archivo y le ponemos el nombre .replit. Dentro del archivo debemos de introducir la orden que queremos que se ejecute al presionar el botón Run. En este caso será una orden para ejecutar nuestro programa.

Dentro del archivo tsconfig.json debemos de realizar algunos cambios, ya que la versión que se nos crea por defecto no contiene las suficientes características que necesitamos.



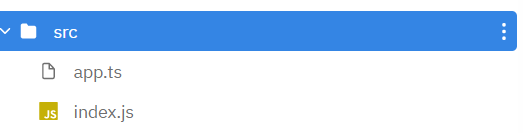
Los parámetros que guarda este archivo son los siguientes:

* **module**: Especifica el método de generación de código del módulo. Node utiliza commonjs.
* **target**: Especifica el nivel de lenguaje de salida.
* **moduleResolution**: Esto ayuda al compilador a averiguar a qué se refiere una importación. El valor node imita al mecanismo de resolución del módulo de Node.
* **outDir**: Esta es la ubicación para los archivos .js tras la transpilación. En este tutorial, lo guardará como dist.

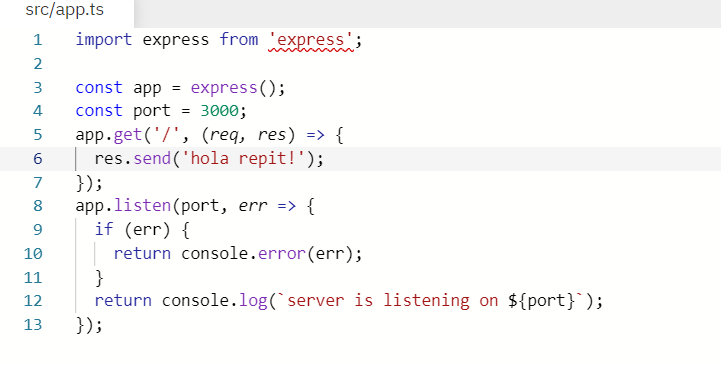
1. **Arrancar una aplicación**

Para proceder a arrancar una aplicación, es decir que el código que introducimos se ejecute y nos salga una respuesta por pantalla, necesitamos crear un código y ejecutarlo.

La creación del código se hará dentro de la capeta SRC anteriormente creada. Para ello, se creará un archivo .ts



Este archivo contendrá el script, programa o aplicación que queramos ejecutar. En este caso va a ser una muy sencilla.



\*\*\*IMPORTANTE, para ejecutar la aplicación o realizar cambios en ella, hay que inicializar tsc -w en la Shell, para que vaya compilando los cambios, si no, el programa creará un archivo transcrito al que no se le enviaran los cambios realizados.

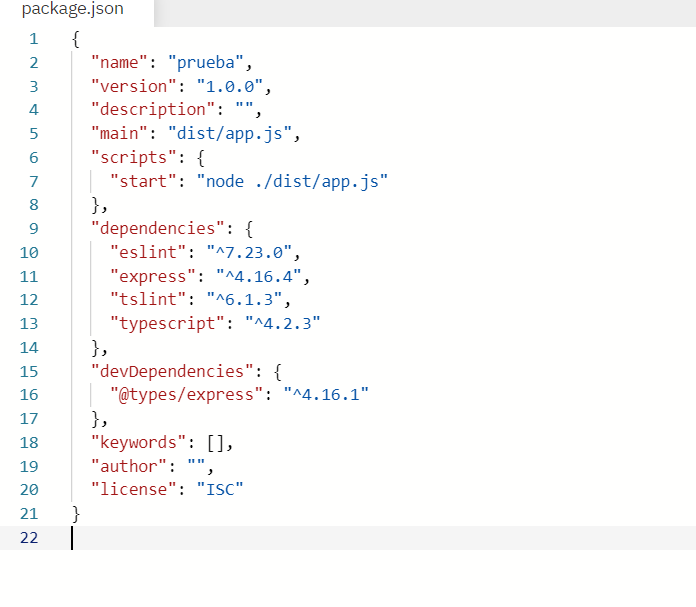
Otros procesos que podemos arrancar es nodemon**. Nodemon** es una herramienta que ayuda a las aplicaciones basadas en node.js a reiniciarse automáticamente cuando el archivo es modificado.

Para activar todos estos servicios podemos usar la ruta:

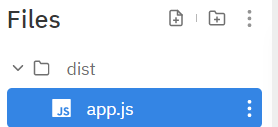
**Npm i -D typescript ts-node nodemon @types/node @types/express**

Los dos ultimo son ayudas para el entorno de desarrollo como por ejemplo brindar funciones de autocompletado.

Para ejecutar el programa, debemos dirigirnos a al archivo package.json y añadirle los scripts de inicialización. También, tenemos que indicar la ruta en el apartado “main”. En este caso vamos a crear el archivo dentro de la capeta DIST, para eso la creamos antes.

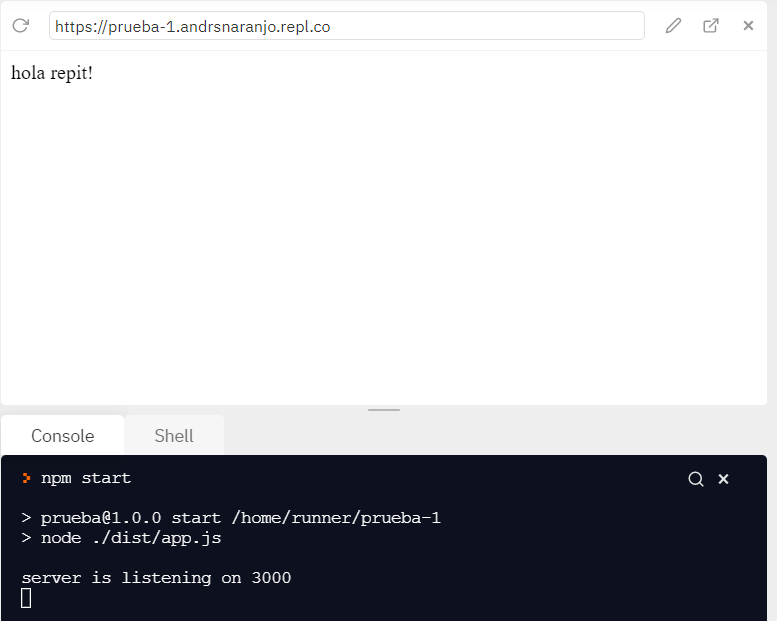


Una vez hecho esto, procedemos a ejecutar el programa en el botón RUN configurado anteriormente. Después de que el servidor cargue nuestro programa, nos generara un archivo transcrito en javascript desde typescript en la carpeta DIST.

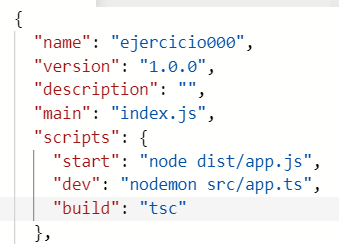




Como resultado, nos aparecerá tanto el mensaje por pantalla, como por consola.



\*\* configuración del script extendida:



Además del script de start, podemos configurar mas funciones dependiendo de los módulos que tenemos instalados.

En este caso, en el dev, hemos ejecutado el proceso nodemon

En build, hemos puesto el comando de tsc, que arranca el tsconfig.json