Actividad: Paso a Paso Aplicación Nodejs

Objetivo

Crear una aplicación backend simple usando VSCode, NodeJs y NPM.

Consigna

Crear una aplicación **adivina_numero_secreto** que contenga el código del juego:

- Genera un número aleatorio entre 1 y 100 para que el usuario lo adivine.
- Pide al usuario que ingrese un número y compáralo con el número generado.
- Proporciona retroalimentación al usuario si el número es demasiado alto o demasiado bajo.
- Continúa solicitando números al usuario hasta que adivine el número correcto.
- Muestra un mensaje de felicitaciones cuando el usuario adivina el número.

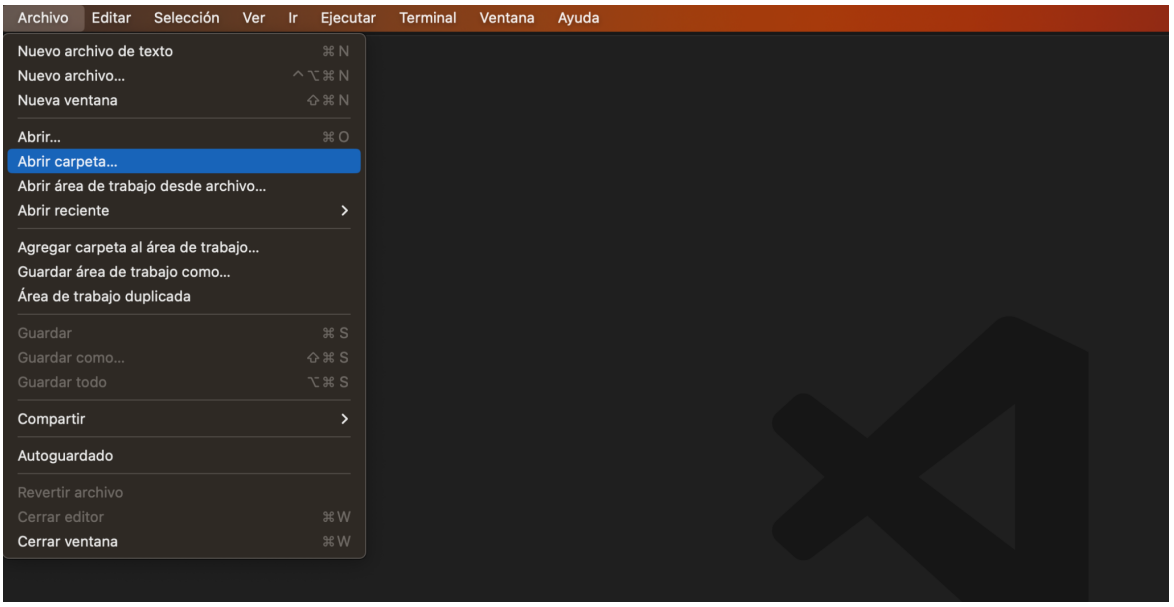
Utilizaremos el paquete npm **readline-sync** que nos permitirá interactuar con el usuario a través de la consola.

Consideraciones:

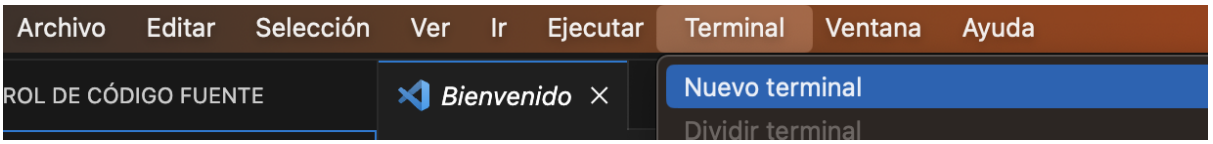
- Tanto los comandos como el código fuente de este paso a paso se encuentra disponible para que lo puedan copiar y pegar, y así no escribirlo completamente.

Desarrollo Paso a Paso

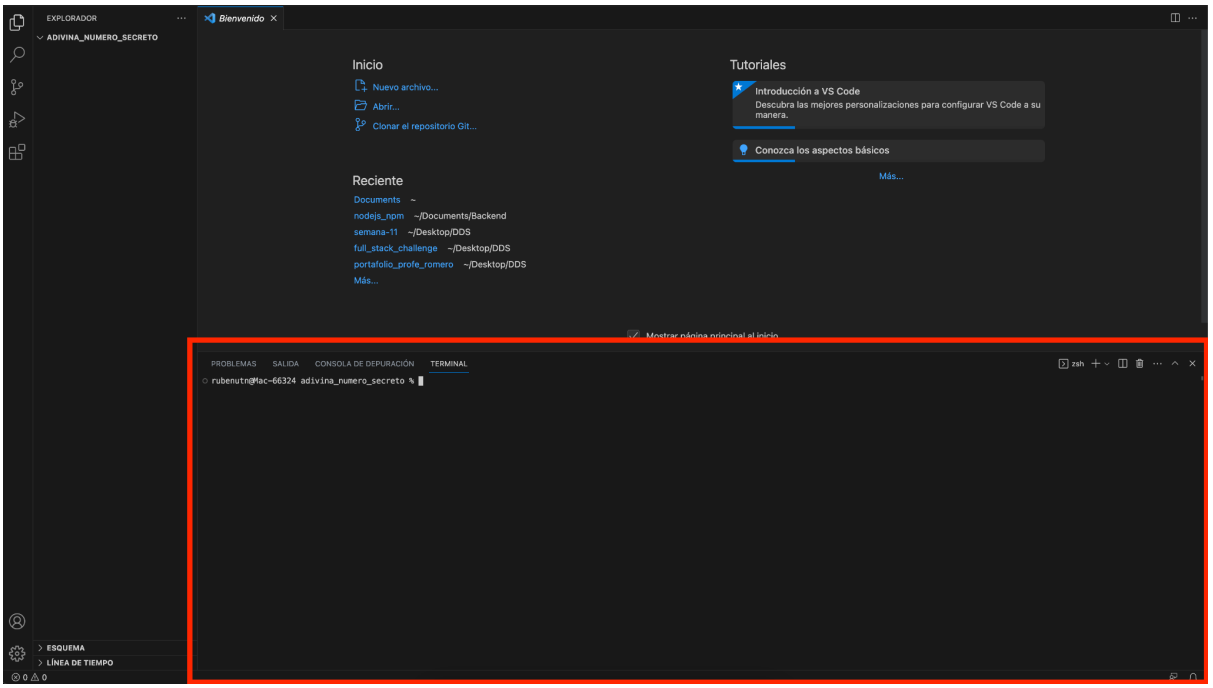
1. Iniciamos la aplicación VSCode.
2. Ir al menú Archivo -> Abrir Carpeta



3. Del paso anterior se abre una ventana de búsqueda de carpetas, aquí buscamos la carpeta Documentos, seleccionamos y luego elegimos la opción “Nueva Carpeta”, indicando el nombre de la carpeta **adivina_numero_secreto**.
4. Selecciona la carpeta **adivina_numero_secreto** para que se abra en VSCode.
5. En VSCode buscamos en el menú la opción **Terminal -> Nuevo Terminal**



6. En la siguiente imagen podemos ver en VSCode la sección de Terminal:



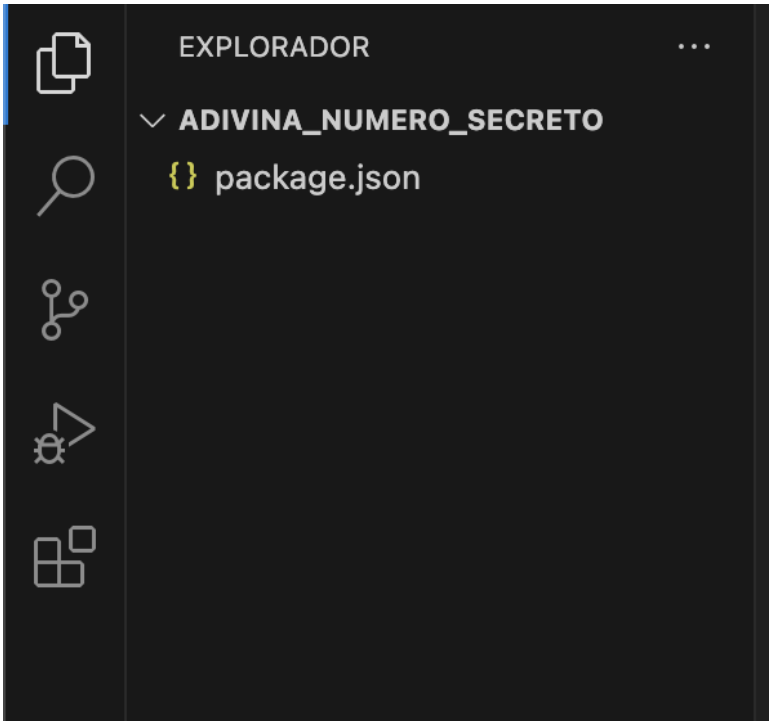
7. En el Terminal vamos a escribir el siguiente comando y luego apretar la tecla enter (para que se ejecute el comando):

```
npm init -y
```

Resultado esperado del comando:

```
> npm init -y
Wrote to /Users/rubenutn/Documents/adivina_numero_secreto/package.json:

{
  "name": "adivina_numero_secreto",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

8. El comando anterior creará un archivo **package.json** en la carpeta como se ve a continuación:



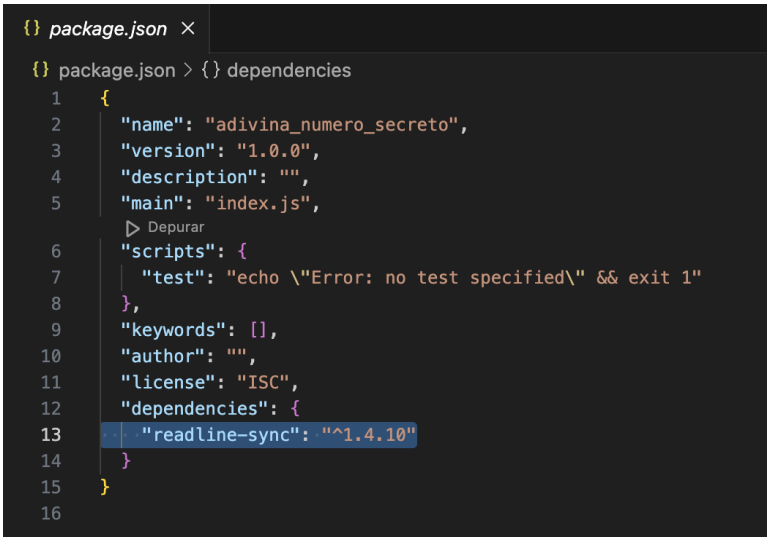
9. Ahora vamos a utilizar npm para instalar el paquete **readline-sync**, que nos permitirá interactuar con el usuario a través de la consola:

```
npm install readline-sync
```

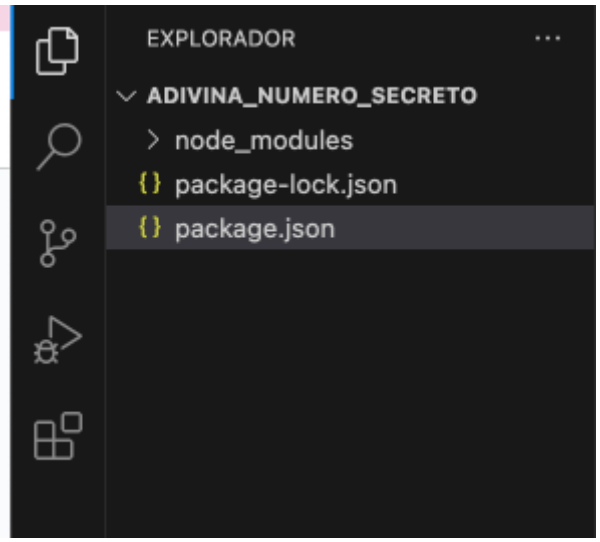
Resultado esperado:

```
> npm install readline-sync

added 1 package, and audited 2 packages in 2s

found 0 vulnerabilities
```

10. Luego de ejecutar el comando anterior podemos verificar en el archivo package.json que el paquete se agregó correctamente:



También podemos ver que npm creó automáticamente un archivo package-lock.json y la carpeta node_modules:



11. Continuamos creando un archivo **adivinanza.js** con el siguiente código con un módulo que vemos a importar en el siguiente paso:

```
const generarNumeroAleatorio = () => {
  return Math.floor(Math.random() * 100) + 1;
};

const verificarAdivinanza = (numeroSecreto, numeroAdivinado) => {
  if (numeroAdivinado === numeroSecreto) {
    console.log('¡Felicitaciones! ¡Adivinaste el número secreto!');
  } else if (numeroAdivinado > numeroSecreto) {
    console.log('El número secreto es menor. ¡Sigue intentando!');
  } else {
    console.log('El número secreto es mayor. ¡Sigue intentando!');
  }
};

module.exports = {
  generarNumeroAleatorio,
  verificarAdivinanza
};
```

12. Ahora creamos un archivo **app.js** donde utilizamos el módulo del punto anterior usando la sentencia **require('./adivinanza')**:

```
const readlineSync = require('readline-sync');
const { generarNumeroAleatorio, verificarAdivinanza } = require('./adivinanza');

const obtenerNumeroUsuario = () => {
  return readlineSync.question('Ingresa un número: ');
};

const juegoAdivinanza = () => {
  const numeroSecreto = generarNumeroAleatorio();
  let numeroAdivinado = 0;

  console.log('¡Bienvenido a Adivina el número secreto!');
  console.log('Intenta adivinar el número del 1 al 100.\n');

  while (numeroAdivinado !== numeroSecreto) {
    numeroAdivinado = obtenerNumeroUsuario();
    verificarAdivinanza(numeroSecreto, numeroAdivinado);
  }
};

juegoAdivinanza();
```

13. Como podemos ver en el archivo app.js, utilizamos la sentencia iterativa “while” para que la aplicación continúe ejecutándose hasta que el usuario adivine el número.

Sentencia Iterativa "while":

```
while (condición) {
  // Código a ejecutar mientras se cumpla la condición
}
```

- Permite ejecutar un bloque de código repetidamente mientras una condición se cumpla.
- La condición se evalúa antes de cada iteración. Si la condición es verdadera, se ejecuta el bloque de código. Si la condición es falsa, se sale del bucle y continúa con la ejecución del código siguiente.

Nota: *Si queremos que la aplicación termine sin adivinar el número podemos usar la combinación de teclas Ctrl + C para finalizar la aplicación en la línea de comandos.*

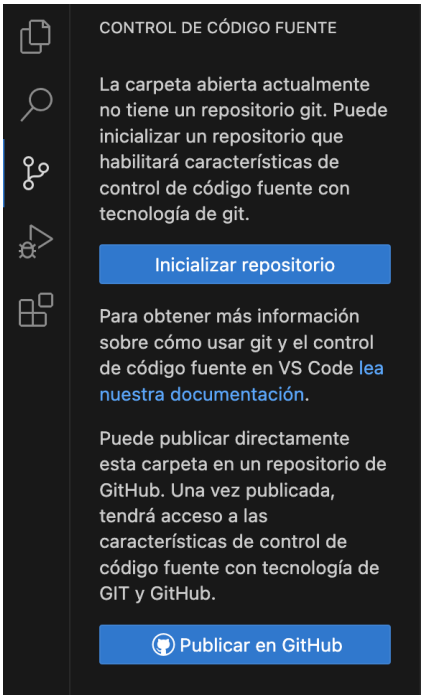
14. Ejecutamos el juego utilizando el Terminal de VSCode ejecutando el siguiente comand

```
node app.js
```

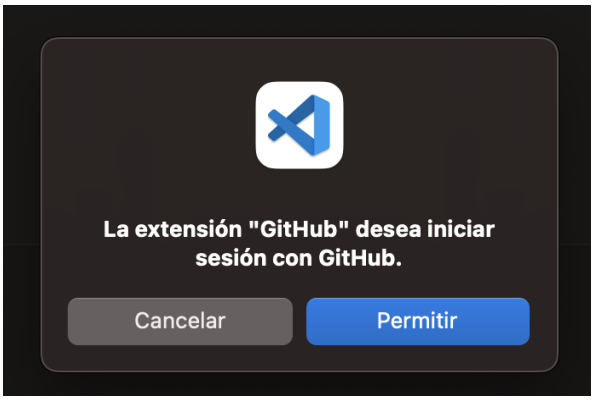
Resultado esperado:

```
> node app.js
¡Bienvenido a Adivina el número secreto!
Intenta adivinar el número del 1 al 100.

Ingresa un número: 50
El número secreto es mayor. ¡Sigue intentando!
Ingresa un número: 75
El número secreto es menor. ¡Sigue intentando!
Ingresa un número: 65
El número secreto es menor. ¡Sigue intentando!
Ingresa un número: 55
El número secreto es menor. ¡Sigue intentando!
Ingresa un número: 52
El número secreto es mayor. ¡Sigue intentando!
Ingresa un número: 54
El número secreto es menor. ¡Sigue intentando!
Ingresa un número: 53
¡Felicitaciones! ¡Adivinaste el número secreto!
```

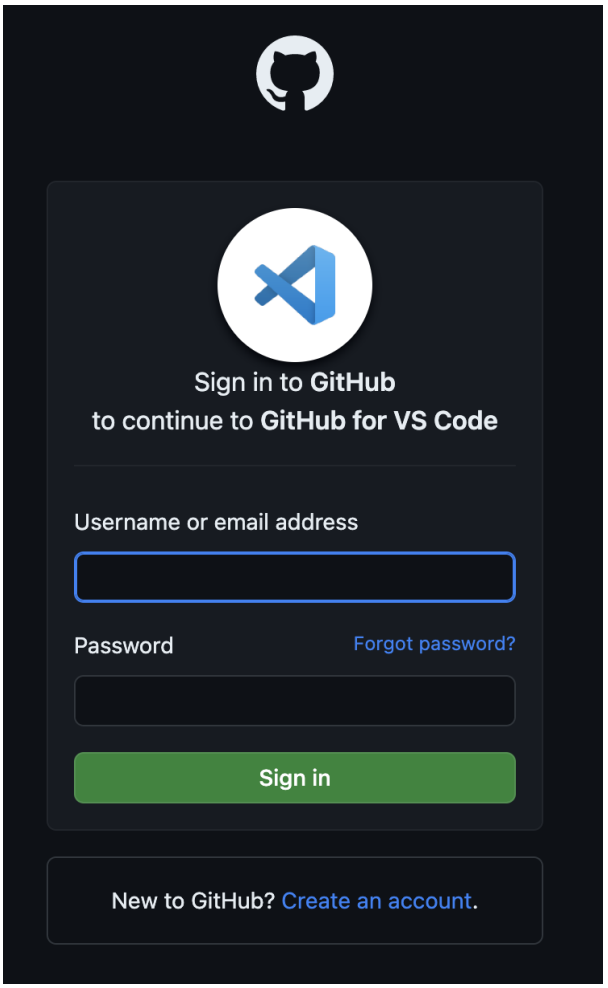
15. Ahora vamos a llevar a GitHub el código fuente generado usando la herramienta de git de VSCode. Hacemos click en la opción “Publicar en GitHub”:



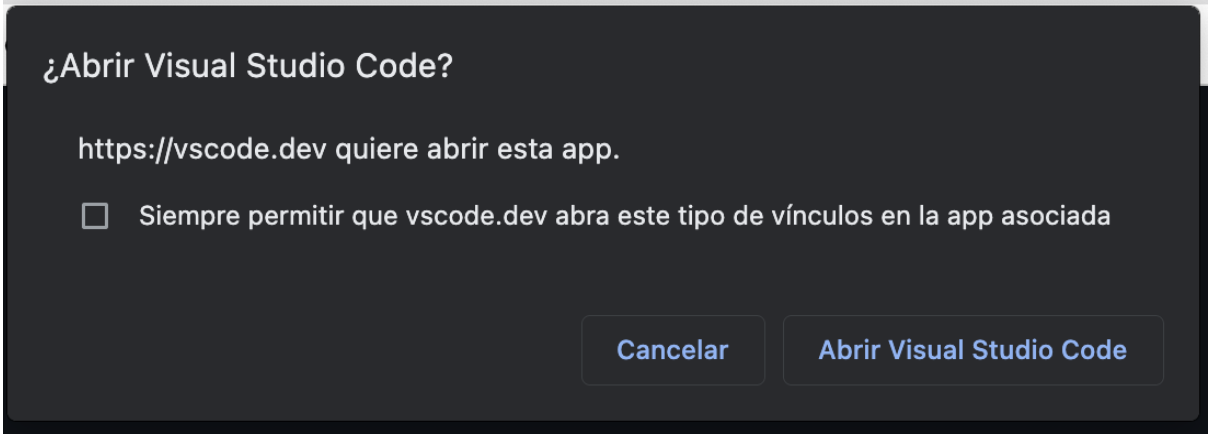
16. VSCode muestra este mensaje para que le darle permisos a iniciar sesión en GitHub a traves de un navegador:



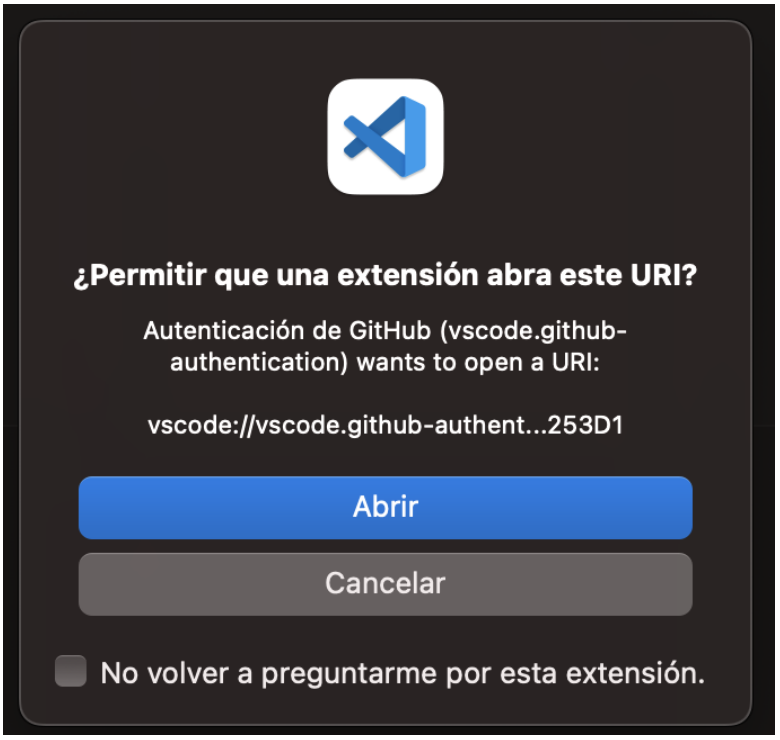
17. VSCode nos lleva al navegador web que tengamos por defecto y nos pide que ingresemos los datos de nuestra cuenta GitHub (el que no tenga cuenta se crea una nueva con su correo electrónico):



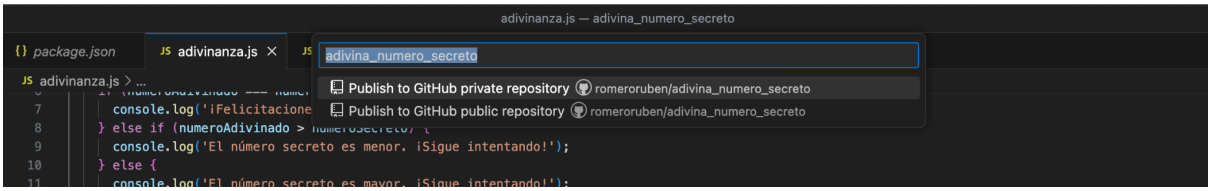
18. El navegador solicita que permitamos abrir VSCode, seleccionamos la opcion “Abrir Visual Studio Code”



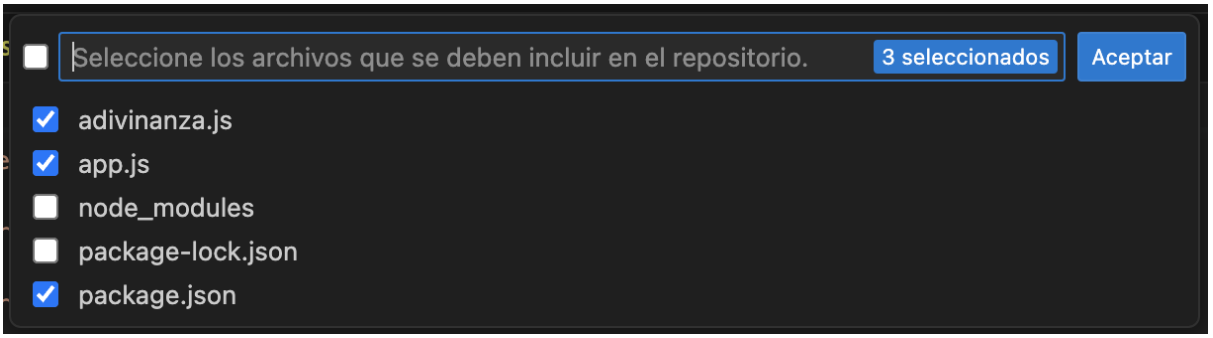
19. VSCode nos pide permisos para abrir una url, hacemos click en la opción “Abrir”:



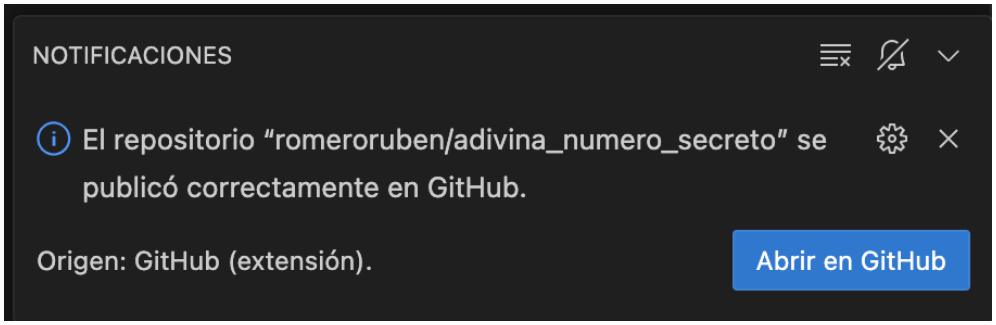
20. Ahora debemos seleccionar que tipo de repositorio queremos crear, seleccionar “Publish to GitHub **public** repository”, esto significa que vamos a publicar nuestro código en forma pública.



21. A continuación seleccionamos los archivos y carpetas que queremos “versionar” y llevar a GitHub.

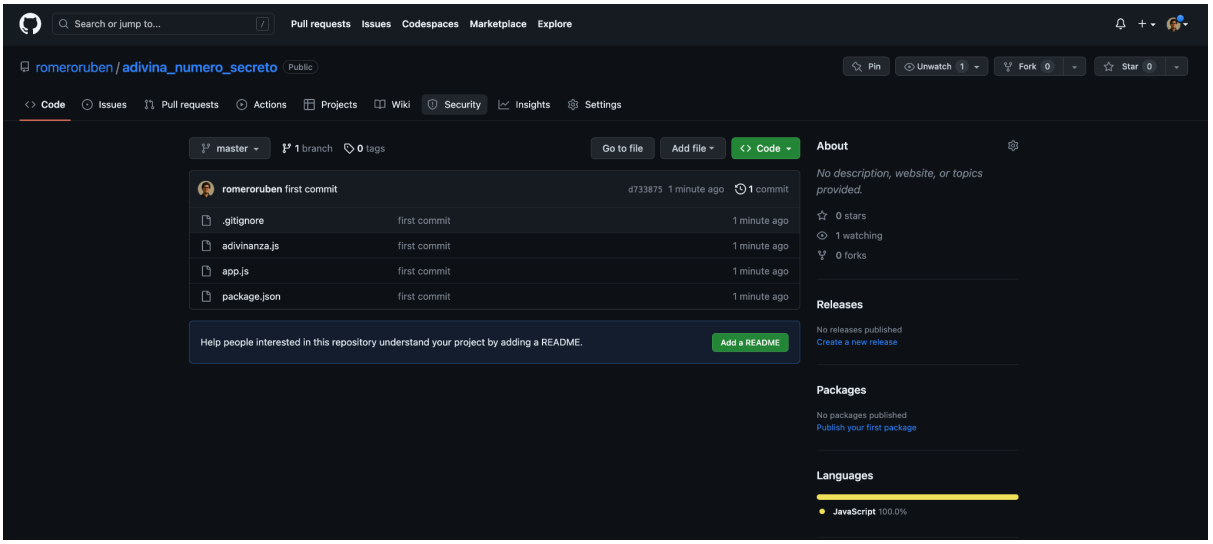


22. Esperamos unos minutos hasta que en VSCode podemos ver la siguiente notificación (abajo a la derecha), hacemos click en “Abrir en GitHub”



:

23. Se abre un navegador con el repositorio de GitHub disponible para que cualquiera lo pueda acceder:



24. IMPORTANTE! Para dar por finalizada la actividad compartir la url de GitHub en la actividad del aula virtual en https://uve.frc.utn.edu.ar/.