**YARGS**

Importamos el paquete en la app principal app.js

const argv = require('yargs').argv;

diferencia entre argv de yargs y el argv de process.argv

\*\*\*node app

{ \_: [], '$0': 'app' } es el argv de yargs

[

'C:\\Program Files\\nodejs\\node.exe', es el argv de process

'D:\\PROGRAMACION\\NODE FERNANDO\\03BASESNODE\\app'

const argv = require('yargs')

.command('listar','Imprime la tabla de multiplicar',{

    base:{

        demand:true,

        alias: 'b'

    }

})

.argv;

Para contruir el comadando temenos que concatenar en el require el método command “que nos va permitir poner la configuración del comando” como primer parámetro recibe el nombre del comando que vamos a usar, como segundo la descripción del comando y en el tercer parámetro recibe un objeto que tendrá la serie de flags con su abreviatura

Nuestro comando quería asi

Npm app listar -b 5

Npp app listar -–base 5

Si yo hago npm app listar –limite -b5

Ya va a ejecutar el 5

/\*\*\*\* se queremos ejecutar mand fglags hay que ponerlos como objetos json con su respectiva configuración

multiplicar',{

    base:{

        demand:true,

        alias: 'b'

    },

    limite:{

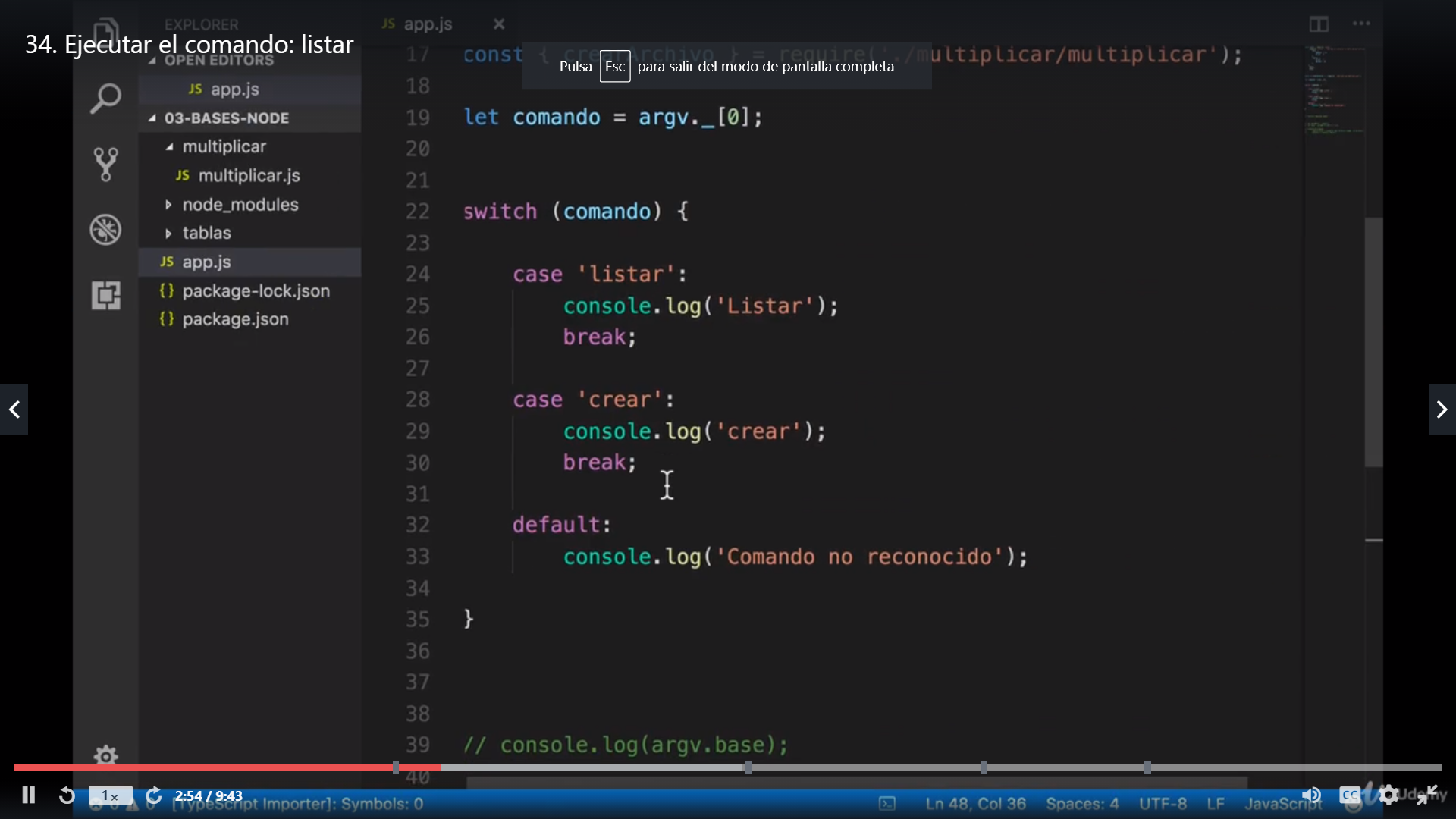
        alis: 'l',

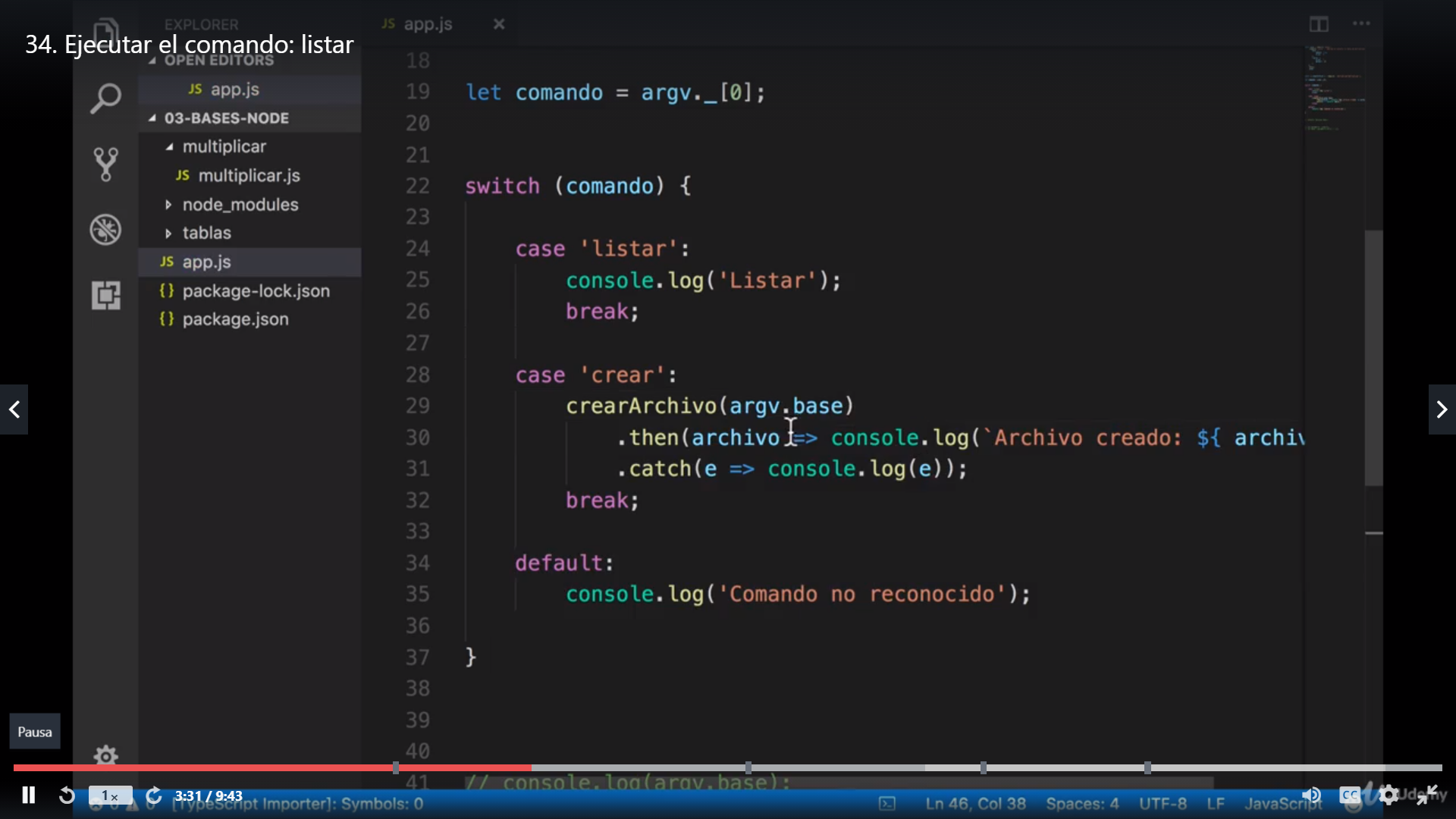
        default:10

    }

**CONFIGURAR EL COMANDO**

De esta manera configuramos los comandos argv.\_[0] en la posición que le pongamos en el array.





REPOSITORIO DE GIT HUB

En el proyecto en la consola git init

Git estatus muestra que archivos están cargados

Eligiendo la carpetas que no queremos subir (en ese ejercicio los modulos de node y la carpeta tablas)

Tengo que crear el archivo .gitignore en la raíz del proyecto (archivo que ignora)

node\_modules/

tablas/\*.txt (todos los archivos txt)

\*\*si le doy **git estatus** vamos a notar que ya no están las carpetas que seleccionamos

Después **git add** . para agregar todos los archivos a github

**Git comit -m “Commit inicial”**

Pra subirlo a git hub nos logeamos “new repository” ponemos el nombre del repositorio y lo creamos.

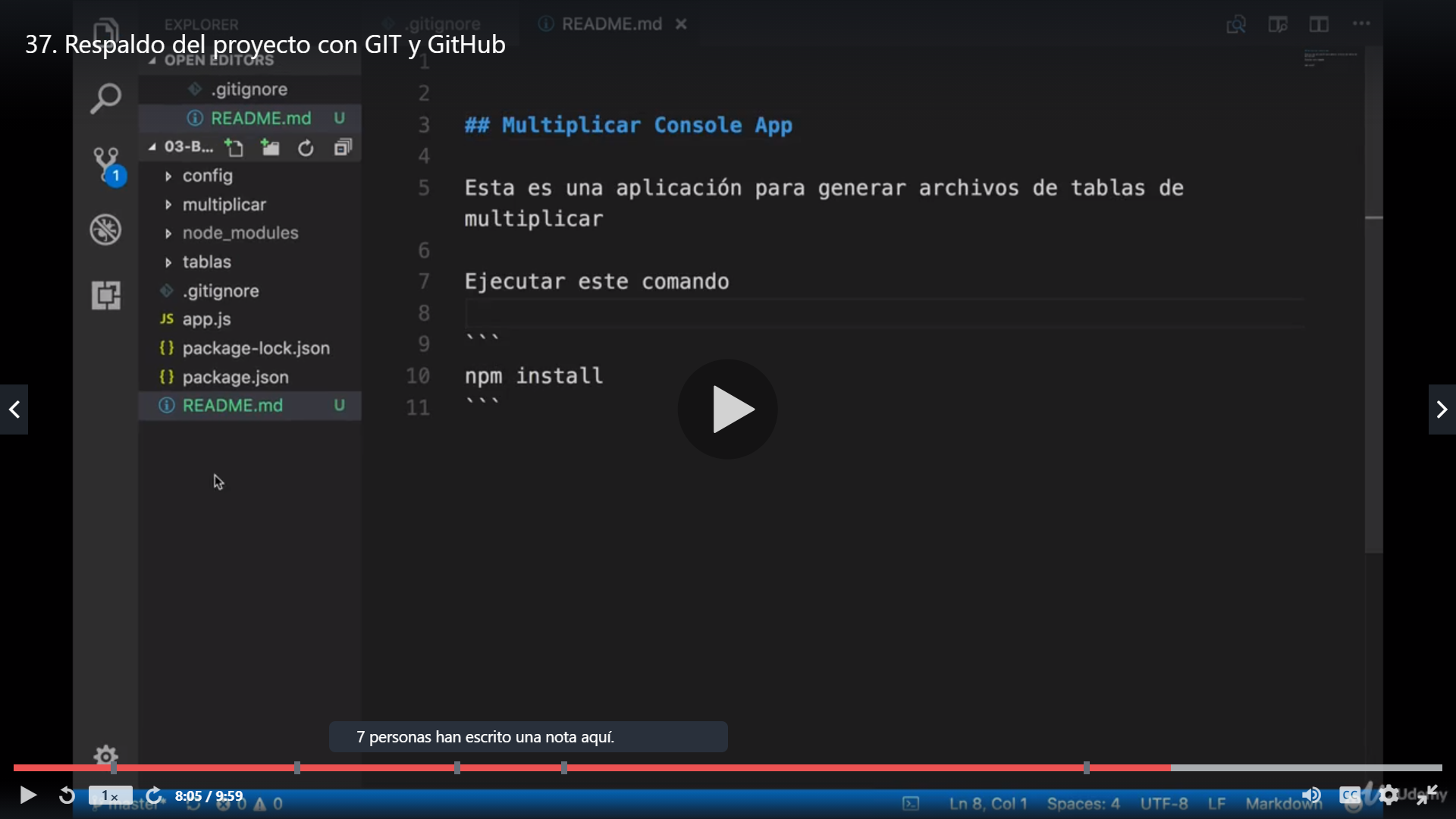
Y ejecutamos la línea de comando que nos da al crear el repositorio

**Git remote add origin** [**http://xxxxxx**](http://xxxxxx)

**Git push -u origin master**

Ya recargamos y esta listo

Podemos crear el archivo Readme



Cuando haya actualizaciones en los archivos

**Git status** para ver los cambios y archivos que no se guardaron.

Y pongo **git add .** para subir los archivos de nuevo.

Ponemos nuevo commit

**Git commit -m “”**

Y hacemos el push

**Git push**