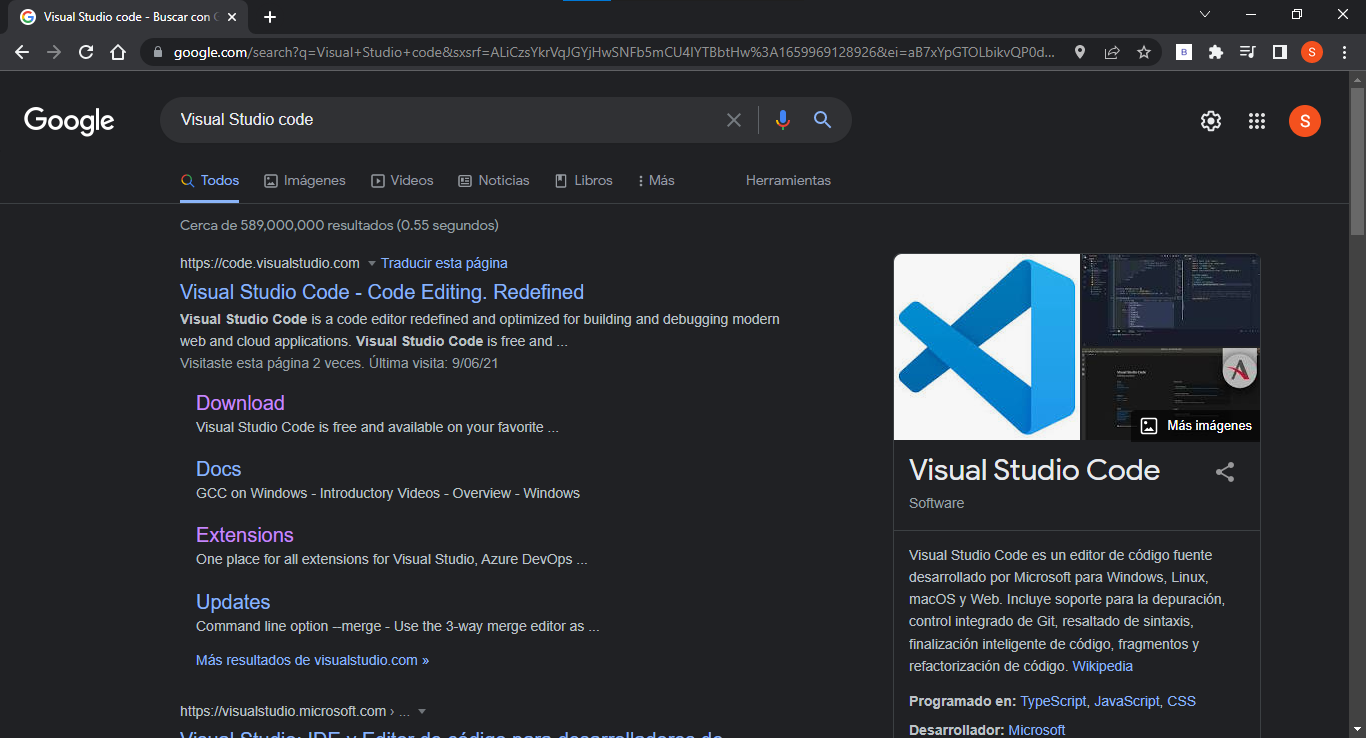
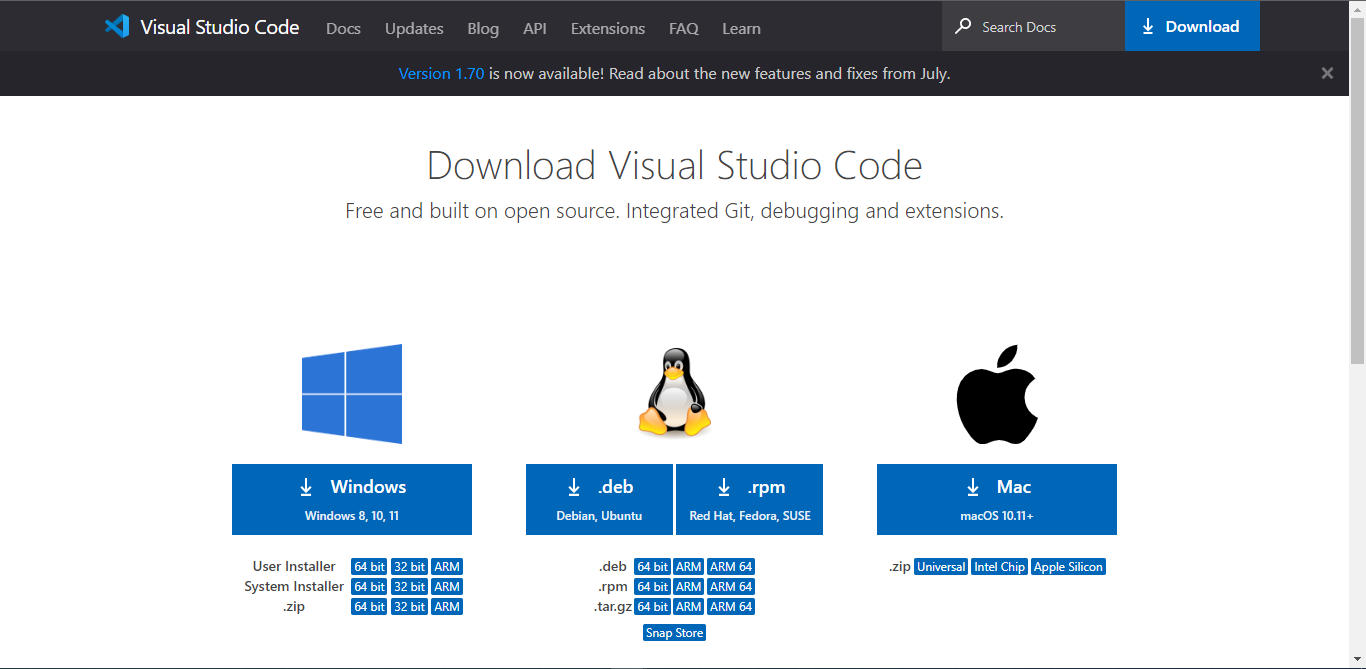
**Proceso instalación Visual Studio Code**

**Paso 1:** Escribimos en Google Visual Studio Code y seleccionamos donde dice “Download”.



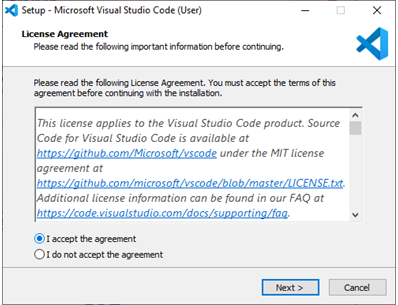
**Paso 2:** Seleccionamos el sistema operativo que tenemos y lo descargamos.



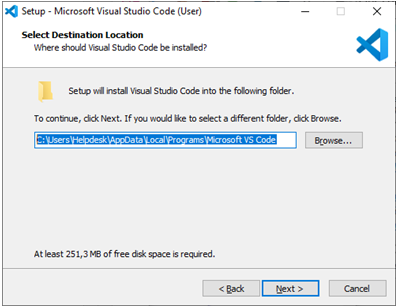
**Paso 3:** Al darle clic nos descargará un .exe, al cual le daremos clic encima.



**Paso 4:** Lee y acepta el acuerdo de licencia. Haz clic en Next para continuar.



**Paso 5:** Puedes cambiar la ubicación de la carpeta de instalación o mantener la configuración predeterminada. Haz clic en Next para continuar.



**Paso 6:** Elige si deseas cambiar el nombre de la carpeta de accesos directos en el menú Inicio o si no deseas instalar accesos directos en absoluto. Haz clic en Next.



**Paso 7:** Selecciona las tareas adicionales, por ej. crear un icono en el escritorio o añadir opciones al menú contextual de Windows Explorer. Haz clic en Next.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

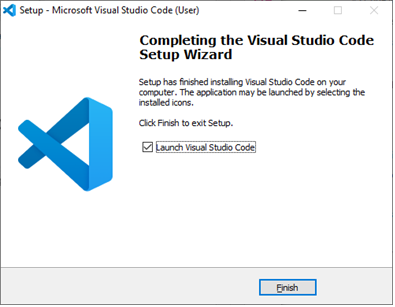
Descripción generada automáticamente

**Paso 8:** Haz clic en Install para iniciar la instalación.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 9:** El programa está instalado y listo para usar. Haz clic en Finish para finalizar la instalación y lanzar el programa.



**Proceso instalación Node JS**

**Paso 1:** Escribimos en Google node js y seleccionamos la primera opción.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 2:** Ahora vamos a seleccionar la opción “LTS” ya que es más estable que la nueva

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media

**Paso 3:** Buscaremos en las descargas el archivo que hemos descargado y le damos doble click.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 4:** Le damos click en “Next”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 5:** Lee los términos y condiciones le damos click en aceptar y le damos “next”.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

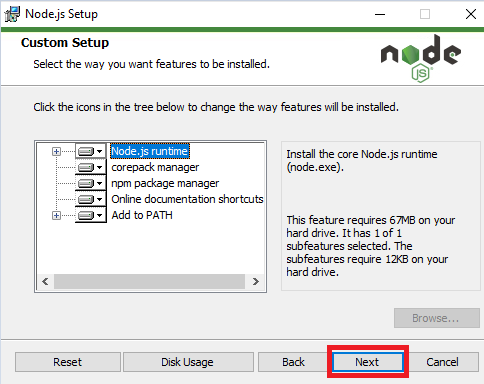
Descripción generada automáticamente

**Paso 6:** Le damos click en “Next” de nuevo

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 7:** Ahora de acá no vamos a mover nada y simplemente daremos “Next”



**Paso 8:** No vamos a chulear la opción que nos da y luego vamos a dar “Next”

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Paso 9:** Ahora damos click en “Install”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 10:** Con esto hemos instalado “Node js”.

Interfaz de usuario gráfica, Texto, Aplicación

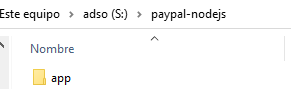
Descripción generada automáticamente

**Creación de archivos y carpetas**

**Paso 1:** Creamos una carpeta raíz. En nuestro caso la llamaremos “paypal-nodejs”.



**Paso 2:**  Dentro de nuestra carpeta ría vamos a crear una nueva carpeta llamada “app”



**Paso 3:** Dentro de la carpeta “app” vamos a crear 4 nuevas carpetas llamadas “config” “controller”, “routes”, “view”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 4:** Dentro de la carpeta “config” vamos a crear la última carpeta llamada “enviroment” que, aunque no la va amos a utilizar en esta ocasión es bueno que la tengas por estructura de nodejs.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**Pagos Paypal**

**Paso 1:** Vamos a comenzar yendo a nuestra carpeta raíz “paypal-nodejs” y en la barra de búsqueda vamos a poner “cmd” para abrir la consola y damos enter

**Texto

Descripción generada automáticamente con confianza baja**

**Paso 2:** Se nos va a abrir la consola y vamos a colocar “node -v” para ver si esta node instalado y también pondremos “npm -v” para saber si está instalado, y por último vamos a abrir visual studio node con ”code .”

Texto

Descripción generada automáticamente

**Paso 3:** Ahora en la carpeta raíz vamos a crear un archivo llamado “index.js”

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 4:** Ahora que vamos a mirar en la barra superior y buscaremos “terminal” y vamos a darle click

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 5:** Se nos abrirá un nuevo menú y daremos click en “new terminal”

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

**Paso 6:** Se abrirá la terminal y vamos a inicializar node con “npm init -y”

Texto

Descripción generada automáticamente

**Paso 7:** En caso de que se haya inicializado correctamente va a parecer un nuevo archivo llamado “package.json”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 8:** Ahora vamos a instalar dos paquetes, uno llamado express y otro llamado “morgan” con “npm i express morgan”

Texto

Descripción generada automáticamente con confianza media

**Paso 9:** Verificamos que se haya instalado

Texto

Descripción generada automáticamente

**Paso 10:** Ahora vamos a instalar cors en la consola colocando “npm i cors”

Texto

Descripción generada automáticamente con confianza media

**Paso 11:** Vemos que se haya instalado correctamente.

Texto

Descripción generada automáticamente

**Paso 12:** En caso de que se hayan instalado correctamente se creará tanto una carpeta llamada “node\_modules” y un archivo llamado “package-lock.json”

Una captura de pantalla de un celular

Descripción generada automáticamente

**Paso 13:** Ahora vamos a instalar en la consola nodemon con “npm i nodemon -D” para poder activar el servidor en google de forma más sencilla.

Interfaz de usuario gráfica, Texto

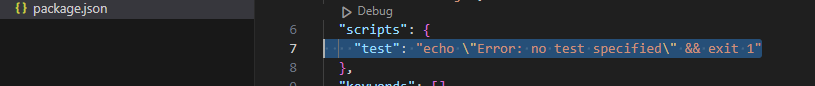
Descripción generada automáticamente

**Paso 14:** Vemos que se haya instalado correctamente.

Texto

Descripción generada automáticamente

**Paso 15:** En nuestro archivo llamado “package.json” vamos a “scripts” y vamos a cambiar el código de adentro (línea 7), comenzamos borrando lo de adentro



**Paso 16:** Y vamos a colocar (“dev”: “nodemon index.js”) (línea 7)

Texto

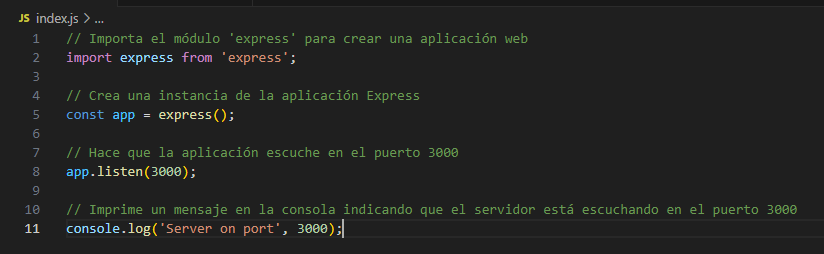
Descripción generada automáticamente

**Paso 17:** Y vamos a añadir la línea 6 (“type”: “module”) justo arriba de donde comienza “scripts”

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

**Paso 18:** Ahora iremos a nuestro archivo “index.js” y vamos a comenzar exportando el módulo “express” (línea 2) y crearemos una instancia (línea 5) y vamos a hacer un puerto y vamos a colocar el 3000 (línea 7) y cuando el servidor este activado vamos a imprimir un mensaje en la consola (línea 11)



**Paso 19:** Ahora en la terminar vamos a colocar “npm run dev”

PD: Este comando lo utilizaremos muchísimas veces a lo largo del trayecto ya que es donde esta nuestra aplicación y hace que funcione así que siempre debes estar actualizando este código en la consola.

Interfaz de usuario gráfica

Descripción generada automáticamente

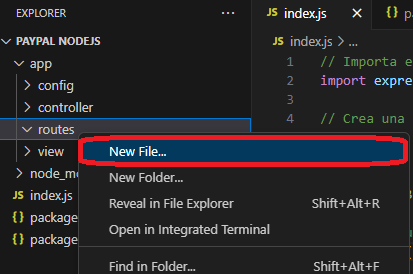
**Paso 20:** Ahora vemos como esta activado nuestro servidor en el puerto 3000

PD: Para cancelar el servidor y que la consola vuelva a la normalidad tienes que hacer (CTRL + C), recuerda siempre estar reiniciando el servidor.

Texto

Descripción generada automáticamente

**Paso 21:** Dentro de la carpeta “app” recordemos que creamos otra carpeta llamada “routes” allí vamos a crear un archivo

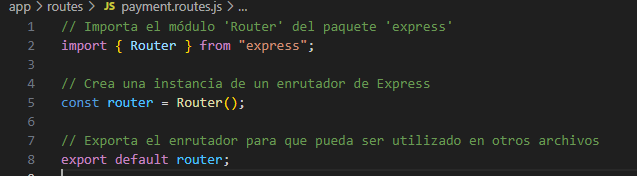


**Paso 22:** Al nuevo archivo lo llamaremos “payment.routes.js”

Interfaz de usuario gráfica

Descripción generada automáticamente

**Paso 23:** Ahora dentro del archivo recién creado vamos a importar el módulo “Router” de “express” (línea 2), también vamos a guardar el módulo en una constante (línea 5), y vamos a exportarlo para que se pueda utilizar en otros archivos (línea 8)



**Paso 24:** Ahora iremos a nuestro archivo “index.js” y vamos a importar el módulo express (línea 2) y también vamos a importar el enrutador desde nuestro archivo (línea 4), ten mucho cuidado al crear la ruta.

Interfaz de usuario gráfica, Texto, Sitio web

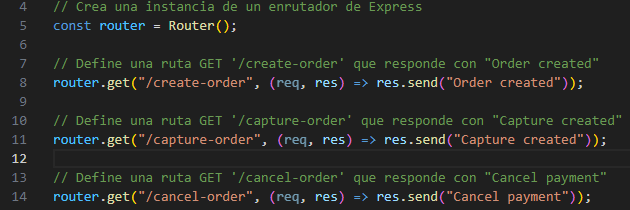
Descripción generada automáticamente

**Paso 25:** Vamos a crear una instancia en la aplicación “express” (línea 7), y vamos a utilizar el enrutado de la aplicación (línea 10)

Texto

Descripción generada automáticamente

**Paso 26:** Ahora iremos de nuevo a nuestro archivo “payment.routes.js” vamos a crear unas rutas justo antes de exportar “router” (línea 8) y vamos a crear 3 rutas una que servirá para crear la petición (línea 8), otra para recibirla (línea 11) y la última para cancelar la petición (línea 14)



**Paso 27:** Ahora iremos a nuestro navegar y vamos a la parte superior donde podremos colocar una dirección.



**Paso 28:** Vamos a buscar “localhost:3000/create-order”

PD: Para visualizar nuestro proyecto siempre vas a requerir tener el servidor activado.



**Paso 29:** Podemos ver que en la ruta (create-order) sale el mensaje que hemos puesto

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 30:** También funciona en la ruta (capture-order)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente **Paso 31:** También funciona en (cancel-order)

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 32:** Ahora vamos a regresar a visual studio code y vamos a buscar la carpeta “controller” y vamos a crear un nuevo archivo.

Interfaz de usuario gráfica

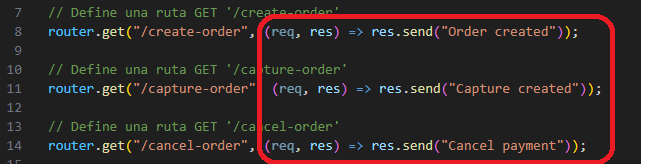
Descripción generada automáticamente

**Paso 33:** El archivo lo llamaremos “payment.controller.js”

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 34:** Volvemos a nuestro archivo “payment.routes.js” y vamos a quitar lo que esta seleccionado para colocarlo en un controlador.



**Paso 35:** Así debe de quedar.

Texto

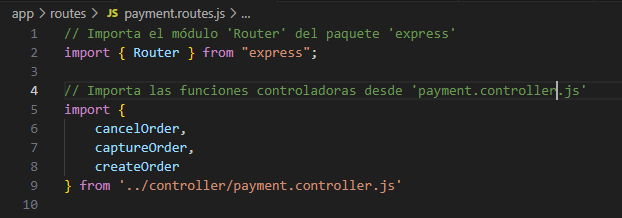
Descripción generada automáticamente

**Paso 36:** Ahora iremos a nuestro archivo “payment.controller.js” y vamos a exportar las rutas como funciones (línea 2, 5 y 8) recuerda que una ruta es para crear (línea 2) otra para capturar la orden (línea 5) y la última para cancelar el pago (línea 8)

Texto

Descripción generada automáticamente

**Paso 37:** Ahora iremos a nuestro archivo de rutas de nuevo “payment.routes.js” y vamos a importar las funciones que acabamos de crear (línea 5 hasta la 9), recuerda tener cuidado con la ruta (línea 9)



**Paso 38:** Ahora vamos a llamar las funciones (línea 15, 18 y 21)

Texto

Descripción generada automáticamente

**Paso 39:** Podemos ver que sigue funcionando correctamente.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

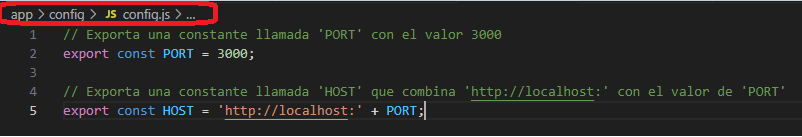
Descripción generada automáticamente

**Paso 40:** Ahora dentro de nuestra carpeta “config” vamos a crear un nuevo archivo llamado “config.js”

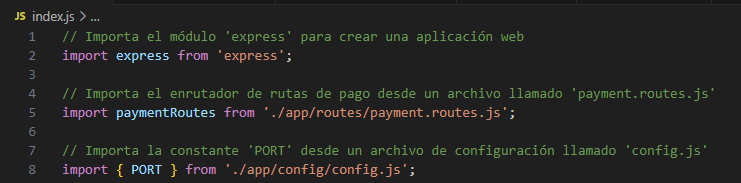
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 41:** Ahora vamos a asegurarnos de que estemos dentro de la carpeta “config” y no dentro de la carpeta “enviroment”, además vamos a exportar una constante llamada “PORT” que será “3000” (línea 2) y vamos a exportar otra constante que va a combinar la ruta con el puerto (línea 5), la llamaremos “HOST”



**Paso 42:** Ahora iremos a nuestro archivo “index.js” y vamos a importar la constante “PORT” (línea 8) como siempre cuidado con la ruta.

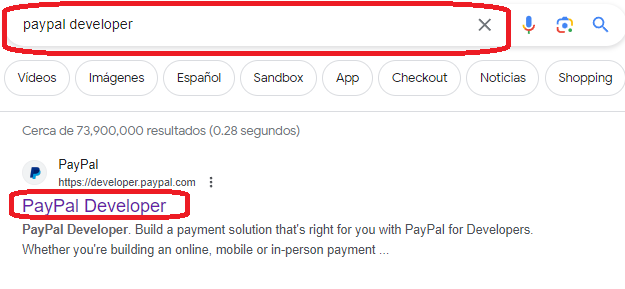


**Paso 43:** Ahora donde antes teníamos el numero ser puerto vamos a llamar “PORT” (línea 17 y línea 20)

Captura de pantalla de computadora

Descripción generada automáticamente

**Paso 44:** Ahora iremos a google y buscaremos “paypal developer” escogiendo la primera opción



**Paso 45:** Arriba a la derecha daremos click en “log in to dashboard”

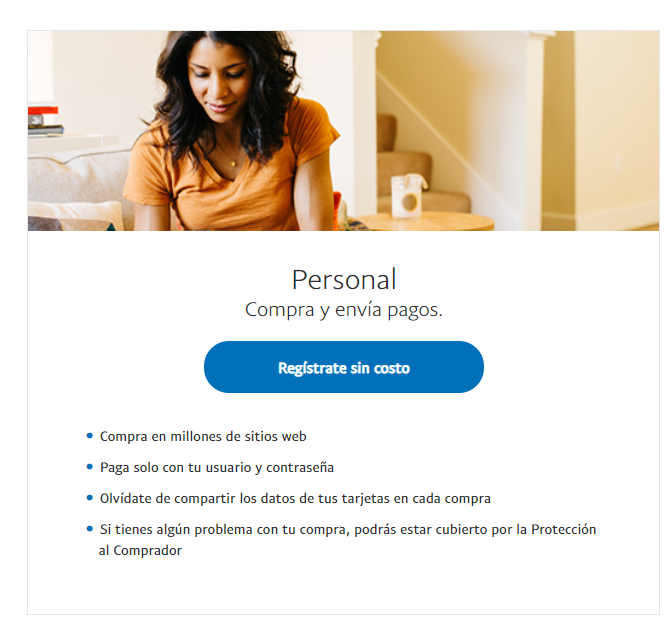
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente **Paso 46:** Ahora nos pedirá iniciar sesión en caso de que ya tengas cuenta en paypal, simplemente inicia sesión y salta hasta el paso (56) de lo contrario dale a registrarse.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 47:** Vamos a escoger la opción de “Personal” y daremos click en “Regístrate sin costo”



**Paso 48:** Ahora nos pedirá registrar un correo, lo registramos y damos “siguiente”

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 49:** Ahora nos pedirá un número de teléfono, lo vamos a registrar y daremos “siguiente”

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 50:** Se nos va a enviar un código al número de teléfono, lo vamos a colocar

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 51:** Ahora vamos a crear una contraseña y daremos “siguiente”

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 52:** Nos pedirá información personal, la rellenamos y daremos “siguiente.”

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 53:** Nos más información la rellenamos y damos “siguiente”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 54:** Ahora nos pedirá la dirección de nuestra casa, vamos a rellenar la información solicitada y vamos a leer los términos y condiciones y vamos a chulear que las aceptamos, luego damos click en “aceptar y abrir cuenta”

Interfaz de usuario gráfica, Texto, Aplicación

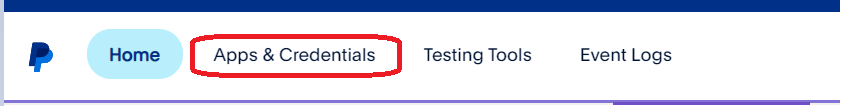
Descripción generada automáticamente

**Paso 55:** Ahora nos mostrará este mensaje, lo vamos a ignorar y vamos a buscar de nuevo en google “paypal developer” he iniciaremos sesión ahora

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 56:** Una vez hemos iniciado sesión daremos click en “Apps y credentials”

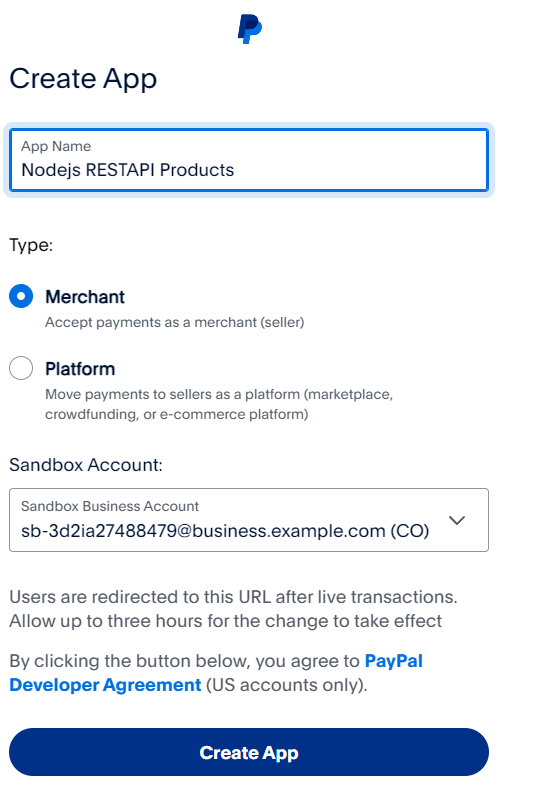
****

**Paso 57:** Arriba a la derecha nos saldrá la opción “create app” daremos click allí.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 58:** Ahora vamos a crear la app y de nombre de pondremos “Nodejs RESTAPI Products” dejaremos el tipo como “merchant” y damos click en create app.



**Paso 59:** Ahora que hemos creado la app se nos mostrará el “Client ID” el cual vamos a guardar.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 60:** Ahora volveremos a visual studio code y vamos a la terminal donde vamos a instalar dotenv usando “npm i dotenv”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 61:** Vemos que se haya instalado correctamente.

Texto

Descripción generada automáticamente

**Paso 62:** Ahora en la carpeta raíz vamos a crear un nuevo archivo.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**Paso 63:** El archivo llamaremos “.env”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 64:** Dentro del archivo “.env” crearemos una variable de entorno llamada “PAYPAL\_API\_CLIENT” y la haremos igual al Client ID el cuándo conseguimos en nuestro paypal developer.(Paso 59)

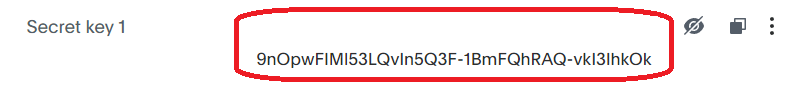


**Paso 65:** Ahora de nuevo en paypal developer vamos a buscar la “secret key” y daremos click al ojo de la derecha para que se nos muestre la contraseña

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 66:** Vamos a copiar la contraseña



**Paso 67:** Ahora de nuevo en visual studio code vamos a crear una nueva variable llamada “PAYPAL\_API\_SECRET” donde vamos a colocar la contraseña que hemos conseguido

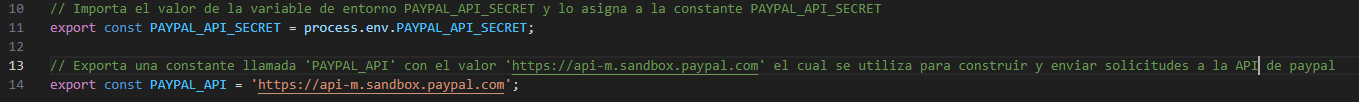
Texto

Descripción generada automáticamente

**Paso 68:** Ahora en el visual studio code vamos a nuestro archivo “config.js” y vamos a importar el valor de la variable de entorno tanto de PAYPAL\_API\_CLIENT (línea 8) y PAYPAL\_API\_SECRET (línea 11)



**Paso 69:** Ahora vamos a exportar una constante llamada “PAYPAL\_API” que va a contener la ruta para consumir la api de paypal “<https://api-m.sandbox.paypal.com>” (línea 14)



**Paso 70:** Ahora vamos a importar el módulo config arriba (línea 2) y vamos a cargar las variables de entorno (línea 5)

Una captura de pantalla de un celular

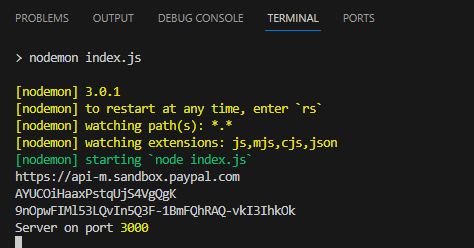
Descripción generada automáticamente

**Paso 71:** Nos vamos a asegurar que se estén tomando los datos correctamente imprimiendo los valores en la consola (línea desde la 23 hasta la 25)

Texto

Descripción generada automáticamente

**Paso 72:** Ahora vamos a depurar la consola de nuevo y allí se verán todos los datos en caso de que todo este correcto (recuerda que es npm run dev)

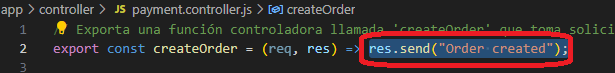


**Paso 73:** Como no queremos estar mostrando esos datos todo el rato vamos a borrar todo desde la línea (22 hasta la línea 25)

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 74:** Ahora iremos a nuestro archivo “payment.controller.js” y vamos a modificar la función “createOrder” borrando lo señalado



**Paso 75:** Ahora vamos a abrir la función (línea 2) y vamos a crear un objeto llamado “order” (línea 5)

Texto

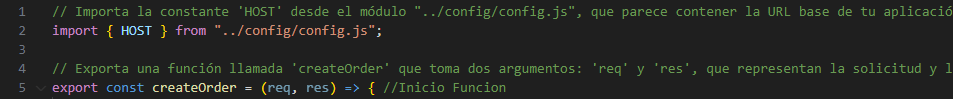
Descripción generada automáticamente

**Paso 76:** Dentro del objeto “order” vamos a hacer un intent “CAPTURE” (línea 6) y vamos a hacer representar la orden de compra (línea 7) dentro colocamos un “amount” (línea 9) y dentro vamos a poner que el pago será en dólares (línea 10) y la cantidad del monto (línea 11)

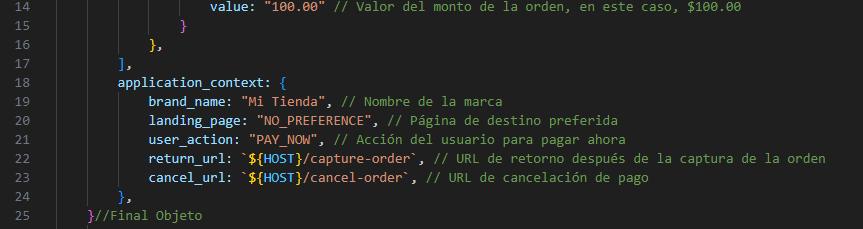
Texto

Descripción generada automáticamente

**Paso 77:** Ahora vamos a importar la constante “HOST” de nuestro “config.js” (línea 2)



**Paso 78:** Ahora dentro de nuestro objeto vamos a colocar el cómo se verá nuestra página colocando “application\_context” (línea 18) y colocaremos el nombre de nuestra tienda (línea 19), vamos a colocar la página de destino preferida que en este caso no hay (línea 20) y la acción de pagar (línea 21), también colocaremos la ruta de captura de orden en caso de que se le dé a pagar (línea 22) y la ruta en caso de que se cancele el pago (línea 23)



**Paso 79:** Ahora en la terminal vamos a instalar axios con “npm i axios”

Texto

Descripción generada automáticamente con confianza media

**Paso 80:** Verificamos que se instaló correctamente.

Texto

Descripción generada automáticamente

**Paso 81:** Ahora arriba vamos a importar el resto de las constantes de “config.js” (línea 2) y además vamos a importar el módulo “axios” (línea 5) en nuestro archivo “payment.controller.js”

Texto

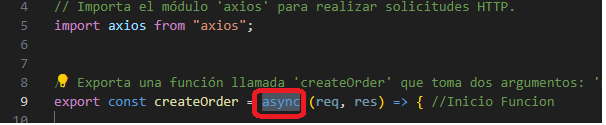
Descripción generada automáticamente

**Paso 82:** Ahora al fuera de nuestro objeto “order” vamos a utilizar axios para realizar una solicitud post a la api (línea desde la 32 hasta la 35)

Texto

Descripción generada automáticamente

**Paso 83:** Ahora vamos a nuestra función “createOrder” (línea 9) y la vamos a convertir a async



**Paso 84:** Ahora volvemos al código donde utilizamos axios y lo seleccionamos

Interfaz de usuario gráfica, Texto

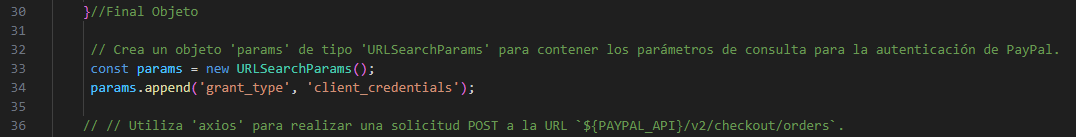
Descripción generada automáticamente

**Paso 85:** Y vamos a comentar el código que hemos seleccionado de momento (desde la línea 32 hasta la 36) (CTRL+K+C) para comentarlo rápidamente.

Texto

Descripción generada automáticamente

**Paso 86:** Arriba de lo que acabamos de comentar, pero fuera de nuestro objeto vamos a crear un objeto “params” para contener los parámetros de la autenticación de Paypal (línea 33 y 34)



**Paso 87:** Ahora vamos a realizar una solicitud post para obtener el token de acceso de Paypal (línea 37) y vamos a realizar la autenticación (línea 38 a 41)

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Paso 88:** Ahora justo debajo de lo que hemos comentado en el paso (85) vamos a devolver la respuesta Json al cliente con el texto “capture order” (línea 54)

Texto

Descripción generada automáticamente

**Paso 89:** Ahora si vamos a nuestra ruta “créate-order” para que nos muestre información en la terminal.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 90:** En la terminal luego de abrir nuestra ruta vemos bastante información.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 91:** Ahora vamos a borrar lo que muestra los datos en la consola porque no nos interesa estar mostrando eso (línea 45)

Texto

Descripción generada automáticamente

**Paso 92:** Ahora vamos a capturar nada más el token de acceso (línea 37) y además vamos a quitar los comentarios que hicimos en el paso (85)

(desde la línea 44 hasta la línea 48)

Captura de pantalla de computadora

Descripción generada automáticamente

**Paso 93:** Ahora donde lo que acabamos de quitar los comentarios vamos a añadir lo que esta señalado ya que vamos a esperar una respuesta del servidor (línea 45), además vamos a añadir dentro de “header” el token de acceso en los encabezados (línea 47)

Texto

Descripción generada automáticamente

**Paso 94:** Justo al terminar vamos a imprimir en la consola la respuesta (línea 52)

Texto

Descripción generada automáticamente

**Paso 95:** Ahora en nuestra página vamos a refrescar la página para que se muestre en nuestra consola (lo mejor es que reinicies el servidor antes de hacer esto)

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 96:** Ahora vemos en la consola la información.

Texto

Descripción generada automáticamente

**Paso 97:** En los links vemos como 3 de los 4 links tienen “checkout” mientras que solo 1 tiene “checkoutnow”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 98:** Ese enlace lo copiaremos.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 99:** En nuestro navegador lo pegaremos y buscaremos.

Texto

Descripción generada automáticamente

**Paso 100:** Nos pedirá iniciar sesión, sin embargo, no inicies sesión aún.

Interfaz de usuario gráfica

Descripción generada automáticamente

**Paso 101:** Iremos de nuevo a paypal developer y buscaremos la opción que dice “testing tools” y escogemos la opción llamada “Sandbox Accounts”

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 102:** Ahora vamos a darle click en “create account” que servirá como una cuenta de testeo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 103:** Ahora la pondremos como personal y damos click en “create account”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 104:** Veremos la cuenta que acabamos de crear y daremos click a los 3 puntos de la derecha

Interfaz de usuario gráfica

Descripción generada automáticamente

**Paso 105:** Damos click en “View/Edit account”

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 106:** Se nos mostrará la información de la cuenta de pruebas y daremos click en el ojo de “password” para mostrar la contraseña

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 107:** Copiaremos la información del “Email” y el “Password”

Interfaz de usuario gráfica, Texto, Correo electrónico

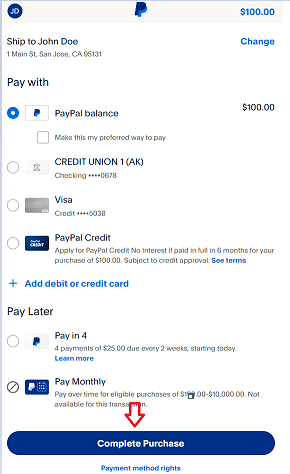
Descripción generada automáticamente

**Paso 108:** Ahora iremos de nuevo al paso (100) he iniciamos sesión con la información de la cuenta de pruebas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 109:** Esto es lo que verán las personas que entren, daremos click en “Complete purchase”

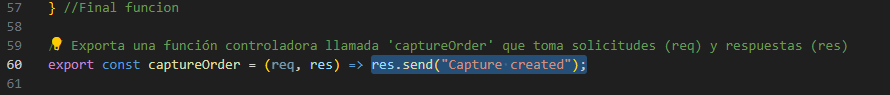


**Paso 110:** Veremos que todo esté funcionando correctamente significa que la orden se está enviando (recuerda que debes tener el servidor abierto)

Texto

Descripción generada automáticamente con confianza media

**Paso 111:** Ahora iremos de nuevo a visual studio code al archivo “payment.controller.js” y vamos a la función “captureOrder” y vamos a quitar lo que esta seleccionado para hacerlo una función flecha.

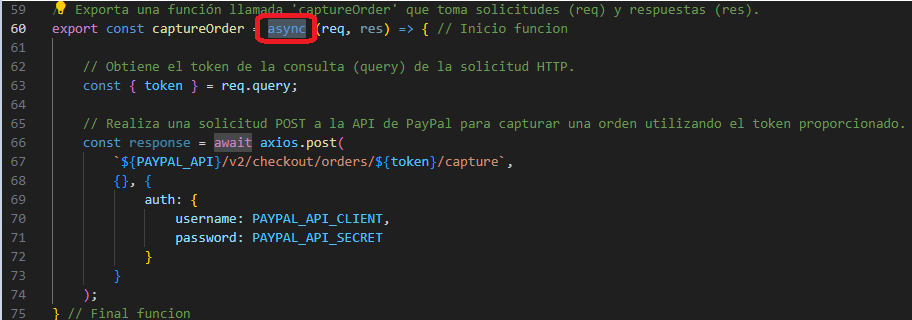


**Paso 112:** Así tiene que haber quedado.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 113:** Ahora convertimos la función a async (línea 60) y dentro de la función vamos a obtener el token de la consulta (línea 63), también realizaremos una solicitud POST a la API mediante una constante que llamada “response” que va a esperar la respuesta (línea 66), dentro de la constante vamos a capturar el token (línea 67 y 68) y vamos a autentificar la información de PAYPAL (línea desde la 69 hasta la 72)

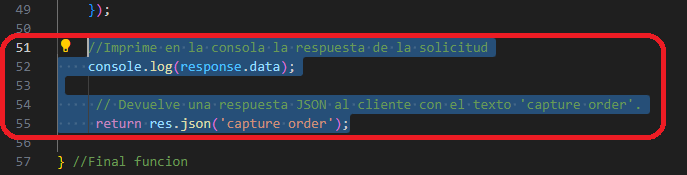


**Paso 114:** Ahora antes de terminar la función, pero después de la solicitud POST vamos a imprimir en la consola los datos de la respuesta de paypal (línea 76) y vamos a enviar una respuesta al cliente informándole que ya pago (línea 79)

Texto

Descripción generada automáticamente

**Paso 115:** Ahora iremos a la función “createOrder” y justo antes de terminar vamos a borrar desde la (línea 51 hasta la 55)



**Paso 116:** Y solo vamos a retornas la respuesta JSON al cliente con los datos de la respuesta (línea 52)

Texto

Descripción generada automáticamente

**Paso 117:** Ahora al ir de nuevo a nuestra página muestra la información que hemos solicitado

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

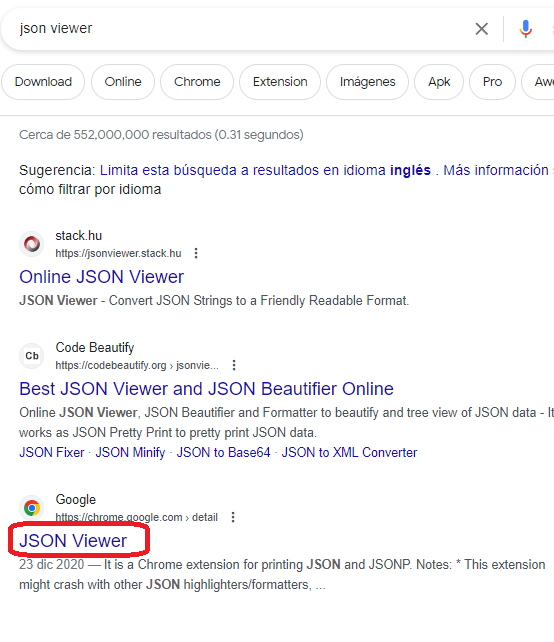
Descripción generada automáticamente

**Paso 118:** Para ver todo más organizado ya que es un formato JSON hay 2 opciones, en caso de que tengas la opción de arriba que dice “Dar formato al texto” puedes chulear la opción, si es así puedes saltarte hasta el paso (123) de lo contrario continua.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 119:** Buscaremos en google “json viewer” y buscaremos la opción de google para instalar la extensión



**Paso 120:** Ahora daremos click en “Agregar a Chrome”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 121:** Nos saldrá está pestaña y daremos click en “añadir extensión”

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 122:** Y refrescamos la página para que se agregue el formato.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 123:** Ahora buscaremos de nuevo el link de “checkoutnow” y lo copiaremos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 124:** En caso de que nos pida iniciar sesión de nuevo, vamos a iniciar sesión otra vez con la cuenta de pruebas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 125:** Ahora daremos click en “complete purchase”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

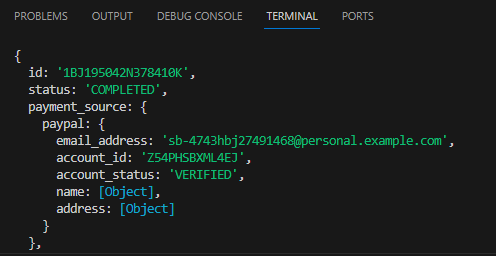
Descripción generada automáticamente

**Paso 126:** Se nos mostrará la pestaña de que se pago

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 127:** Ahora se nos mostrará en la consola información en caso de que haya salido todo bien.

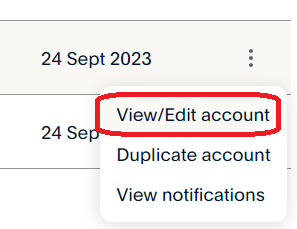


**Paso 128:** Ahora vamos a nuestras cuentas de nuevo (paso 101 en caso de que hayas olvidado como ver las cuentas) y buscaremos la cuenta de tipo “businees” y daremos click en los 3 puntos de la derecha.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 129:** Daremos click en “View/Edit account”



**Paso 130:** Ahora vemos la información de la cuenta “business”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 131:** Iremos a google y buscaremos “sandbox paypal” y escogeremos la primera opción

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 132:** Vamos a iniciar sesión con la información de la cuenta “business”

Interfaz de usuario gráfica, Texto, Aplicación

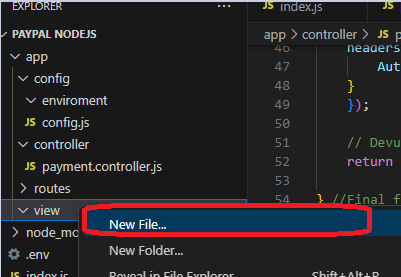
Descripción generada automáticamente

**Paso 133:** Vemos como se realizaron los pagos que hemos hecho anteriormente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 134:** Ahora iremos de nuevo a visual studio code y en la carpeta “view” vamos a crear dos nuevos archivos



**Paso 135:** Los archivos los llamaremos “index.html” y otro que se llamará “payed.html”

Texto

Descripción generada automáticamente con confianza media

**Paso 136:** Ahora iremos a nuestro archivo “index.js” y vamos a importar el módulo “path” que sirve para trabajar con rutas de archivos (línea 5)

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 137:** Ahora en el mismo archivo “index.js” vamos a configurar el express para que reciba archivos de la carpeta “view” (línea 20)

Texto

Descripción generada automáticamente

**Paso 138:** Ahora iremos a nuestro archivo antes creado llamado “index.html” y vamos a colocar “!” y seleccionamos la primera opción.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 139:** Vamos a cambiar el tittle por “Products” (línea 6)

Texto

Descripción generada automáticamente

**Paso 140:** Dentro del body vamos a crear un encabezado que diga “products” (línea 10)

Texto

Descripción generada automáticamente

**Paso 141:** Ahora en el mismo body vamos a crear un botón con el id “checkout” (línea 13) y dentro del botón va a decir pay (línea 15)

Texto

Descripción generada automáticamente

**Paso 142:** Ahora fuera del botón vamos a crear un script (línea 19) y dentro del script vamos a capturar el elemento del botón con su id (línea 21) y crearemos un evento de escucha click (línea 24) el cual realizará una petición “POST” a la ruta (create-order) (línea 26) y vamos a convertir la respuesta a json (línea 30) y vamos a redirigir a la persona al “checkoutnow” (línea 32)

Captura de pantalla de un celular

Descripción generada automáticamente

**Paso 143:** Ahora iremos a nuestro archivo “payed.html” y vamos a colocar “!” y escogemos la primera opción

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Paso 144:** Ahora cambiaremos el titulo por “Payed” (línea 6)

Texto

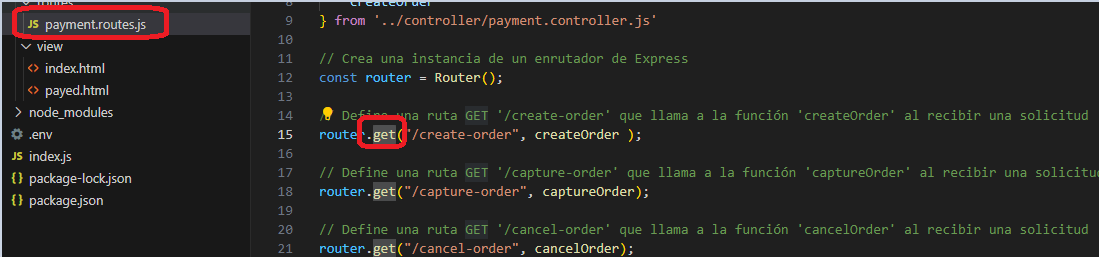
Descripción generada automáticamente

**Paso 145:** Ahora vamos a añadir un encabezado en el body (línea 10)

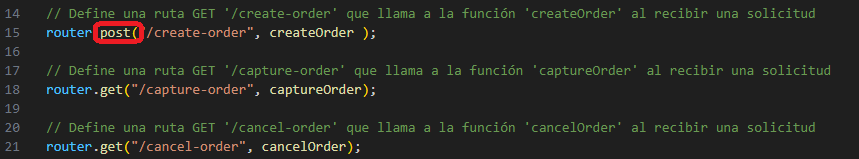
Texto

Descripción generada automáticamente

**Paso 146:** Ahora iremos a nuestro archivo “payment.routes.js” y cambiaremos la ruta “get” por “post” en la ruta de “create-order”



**Paso 147:** Así tiene que haber quedado.

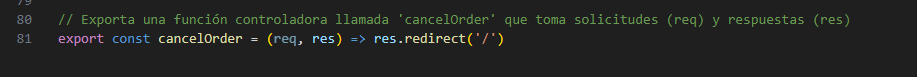


**Paso 148:** Ahora iremos al archivo llamado “payment.controller.js” y debajo del todo de ese archivo está la función “cancelOrder”

Texto

Descripción generada automáticamente

**Paso 149:** Vamos a cambiar el “res.send” por “res.redirect(/)” para que en caso de que se cancele la petición nos envié a la página principal.



**Paso 150:** Ahora iremos a la ruta base (localhost:300) y damos click en “pay”

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Paso 151:** Nos pedirá iniciar sesión y la iniciamos con la información de la primera cuenta que creamos (paso 107, en caso de que lo hayas olvidado)

PD: En caso de que no te deje iniciar sesión y te envié un error es debido a que abriste la cuenta de business (132), solo ve de nuevo, cierra la sesión, reinicia el servidor y vuelve a hacer el paso (150)

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 152:** Ahora de nuevo nos solicitará realizar el pago, damos en “complete purchase”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 153:** Ahora se nos mostrará que se pago

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 154:** Ahora vamos a hacer de nuevo el paso (150 y 151) y en caso de que queramos cancelar la compra no le damos a “complete purchase” sino que le damos “cancel and return to Mi tienda”

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 155:** Y vemos como nos envía de nuevo a la página principal. Y con esto hemos terminado nuestro proyecto donde aprendimos a consumir la API de paypal para realizar peticiones en nodejs.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente