**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

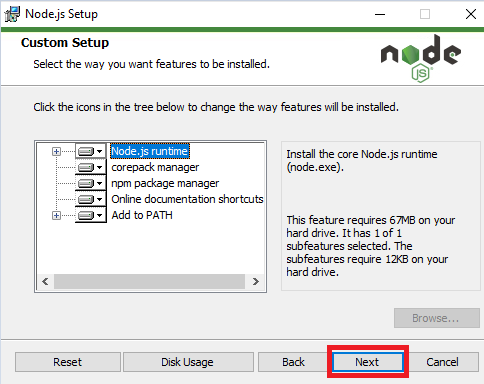
Descripción generada automáticamente

**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:** Ahora vamos a instalar algunos módulos de “npm” chuleando la opción que nos da y luego damos “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” en caso de que te salgan más pestañas como les suele pasar a algunos equipos para instalar simplemente a todo le das enter y se instalará correctamente

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 “calendariogoogle”.

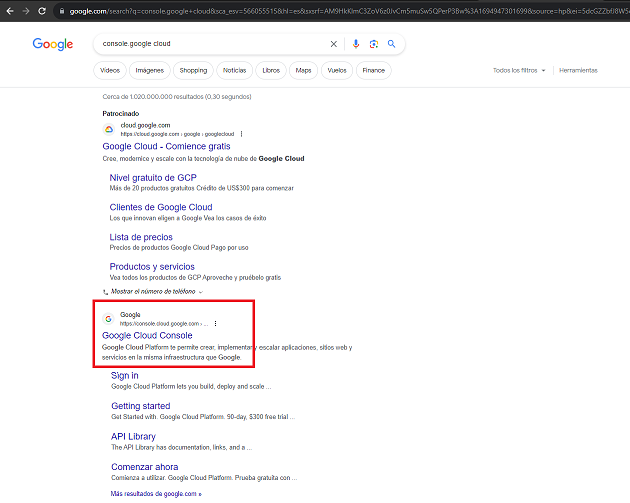


**Calendario De Google**

**Paso 1:** Vamos a comenzar yendo a google.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**Paso 2:** Buscaremos console.google cloud y escogemos la señalada por la imagen

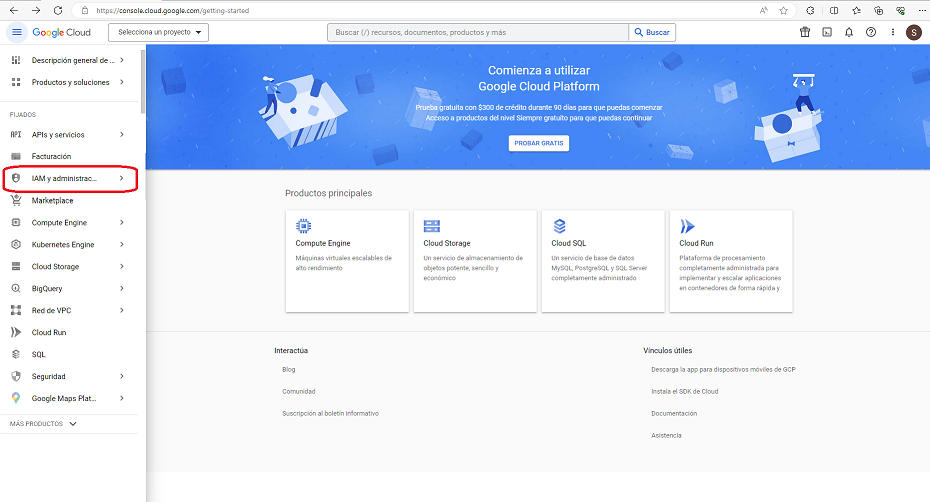


**Paso 3:** Nos pedirá iniciar sesión con Google así que lo haremos

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 4:** Ahora nos mostrara esta pestaña he iremos a “IAM y administración”



**Paso 5:** Y seleccionamos la opción de “Cuentas de servicio”

Interfaz de usuario gráfica, Aplicación

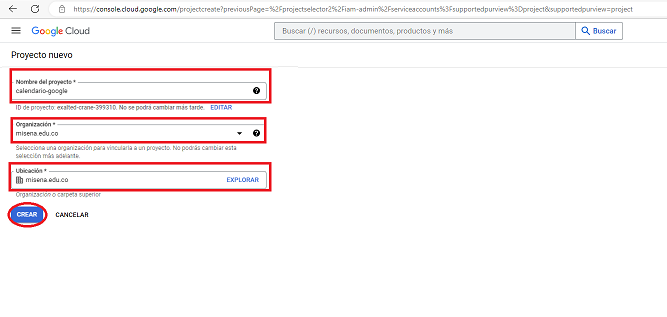
Descripción generada automáticamente

**Paso 6:** Nos mostrará esta pestaña y seleccionamos “crear proyecto”

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**Paso 7:** Llenamos la información solicitada y le damos “Crear”



**Paso 8:** Ahora nos sale esta pestaña y seleccionamos “crear cuenta de servicio”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 9:** Ahora llenamos de nuevo la información solicitada y seleccionamos “CREAR Y CONTINUAR”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 10:** Ahora vamos a seleccionar la opción de “selecciona un rol”

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

**Paso 11:** Escogemos “propietario”

Interfaz de usuario gráfica, Aplicación, Teams

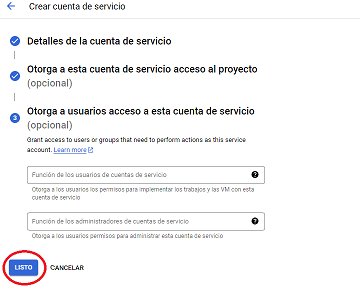
Descripción generada automáticamente

**Paso 12:** Ahora damos “continuar”

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

**Paso 13:** Ahora no llenamos nada de esta información y le damos “listo”



**Paso 14:** Ahora podemos ver como se ve la cuenta que acabamos de crear. Vamos a seleccionar los tres puntos de la derecha.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 15:** Ahora le damos “Administrar claves”

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 16:** Seleccionamos “Agregar clave”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 17:** Y escogemos “Crear clave nueva”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 18:** Escogemos que la clave será tipo “JSON” y damos click en “CREAR”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 19:** Una vez hemos creado la llave se te descargará un archivo en tu computador.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 20:** Ese archivo que se descargo vamos a cortarlo de la carpeta de descargas. Dándole click derecho al archivo y luego izquierdo en cortar

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 21:** Iremos a nuestra carpeta raíz “calendariogoogle” y pegamos ahí el archivo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 22:** Para abrir el Visual Studio Code, haremos lo siguiente: Dentro de la carpeta raíz, en la barra superior, escribiremos cmd y le damos enter.

Forma, Rectángulo

Descripción generada automáticamente

**Paso 23:** Esto nos abrirá la consola donde colocamos “code .” para abrir el Visual Studio Code

Texto

Descripción generada automáticamente

**Paso 24:** Ahora vemos el archivo que hemos puesto y lo seleccionamos

Texto

Descripción generada automáticamente

**Paso 25:** Así se ve actualmente nuestro archivo.

Texto

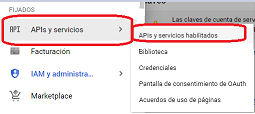
Descripción generada automáticamente con confianza media

**Paso 26:** Ahora iremos a Google de nuevo y seleccionamos las tres líneas de arriba a la izquierda

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 27:** Ahora escogemos “APIs y servicios” y luego “APIS y servicios habilitados”

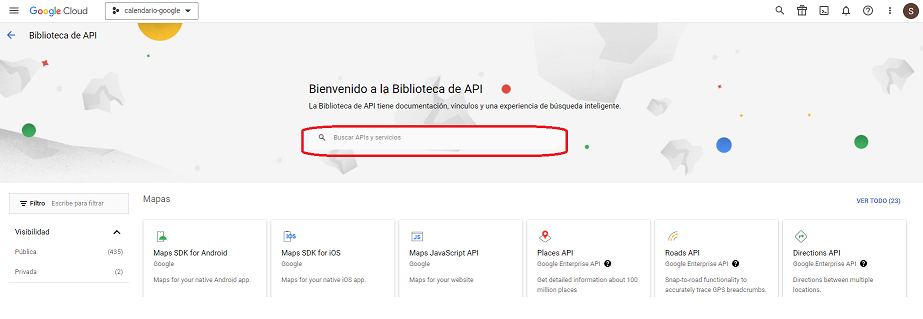


**Paso 28:** Nos aparece esta pestaña y seleccionamos “HABILITAR APIS Y SERVICIOS”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 29:** Ahora nos sale esta pestaña y vamos a la barra de búsqueda y colocamos “calendar”



**Paso 30:** Ahora seleccionamos la opción de Google Calendar API

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 31:** Nos sale esta api y le damos “habilitar” para habilitar la api en nuestro proyecto

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 32:** Una vez habilitada la Api volvemos a Google y buscamos “Google calendar” donde seleccionamos la primera opción

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 33:** Nos aparecerá un calendario y vamos a la izquierda donde dice “other calendars” y escogemos el “+”

Calendario

Descripción generada automáticamente con confianza baja

**Paso 34:** Ahora nos salen estás opciones y seleccionamos “Create new calendar”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 35:** Ahora vamos a rellenar la información solicitada y vamos a escoger “Create calendar” donde nos saldrá que fue creado correctamente.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

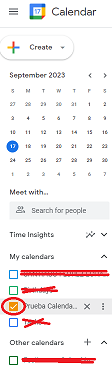
Descripción generada automáticamente

**Paso 36:** Ahora iremos a escogemos la flecha para ir atrás

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

Descripción generada automáticamente

**Paso 37:** Donde se ven todos los calendarios que tenemos creados vamos a quitar todos los que tenemos excepto el que acabamos de crear que lo vamos a chulear



**Paso 38:** Al calendario que acabamos de crear vamos a seleccionar los tres puntos de la derecha

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

**Paso 39:** Escogemos la opción “Setting and sharing”

Imagen que contiene Gráfico de burbujas

Descripción generada automáticamente  
**Paso 40:** Nos aparece esta página y bajamos

Interfaz de usuario gráfica, Aplicación

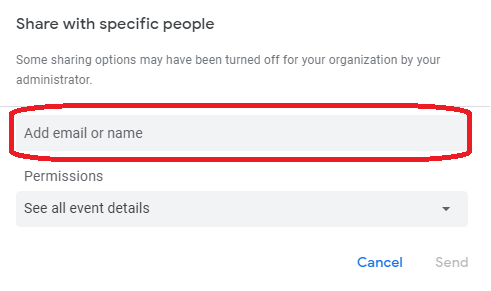
Descripción generada automáticamente

**Paso 41:** Donde dice esta opción llamada “Share with specific people or groups” aparecemos nosotros y le damos a “Add people and groups”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 42:** Ahora vemos que nos pide añadirlo, no añadimos nada de momento.



**Paso 43:** Volvemos al archivo del código y buscamos en (línea 6) la donde dice “client\_email” y copiamos lo que dice entre las “” quitando las comillas

Texto

Descripción generada automáticamente

**Paso 44:** Lo colocamos en la página y en permisos le damos click

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

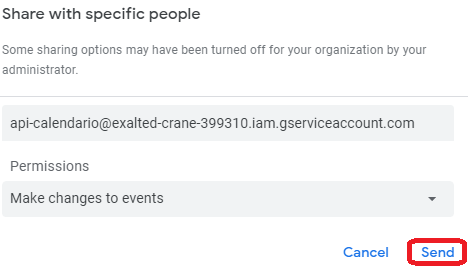
Descripción generada automáticamente

**Paso 45:** Vamos a seleccionar “Make changes to events”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 46:** Ahora que lo tenemos preparado le damos click en “Send”



**Paso 47:** Ahora vemos como se agregó.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 48:** Ahora vamos a abrir de nuevo la consola en nuestro proyecto y veremos si tenemos node instalado colocando “node -v”

Texto

Descripción generada automáticamente

**Paso 49:** En caso de que este instalado nos saldrá la versión de node que tenemos, también vamos a mirar si tenemos “NPM” instalado colocando “npm -v”

Texto

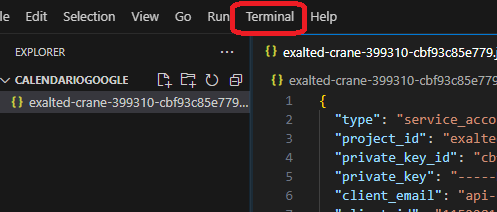
Descripción generada automáticamente

**Paso 50:** Y si está instalado nos sale la versión y vamos a abrir el código una vez más

Texto

Descripción generada automáticamente

**Paso 51:** Ahora en nuestro proyecto arriba vamos a seleccionar “Terminal”



**Paso 52:** Ahora seleccionamos “New Terminal” para abrir la terminal

Interfaz de usuario gráfica, Aplicación

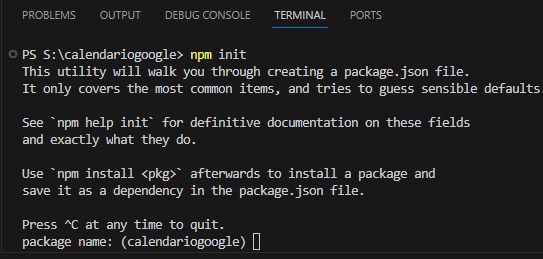
Descripción generada automáticamente

**Paso 53:** Gracias a que tenemos node js y npm instalados vamos a colocar en la termina “npm init”

Texto

Descripción generada automáticamente

**Paso 54:** Comenzará a crear los archivos json y nos pedirá información vamos a dar enter a todo excepto a el autor



**Paso 55:** En Author vamos a colocar nuestro nombre y daremos enter de nuevo hasta que se termine de instalar.

Texto

Descripción generada automáticamente

**Paso 56:** Nos mostrará los datos sobre el archivo y luego nos dirá si está bien, escribiremos “yes”

Texto

Descripción generada automáticamente

**Paso 57:** Ahora si se instalo correctamente vamos a ver que se creo un archivo llamado “package.json”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 58:** Limpiaremos la terminal con “clear”

Texto

Descripción generada automáticamente

**Paso 59:** Ahora vamos a instalar la biblioteca de Google apis y también vamos a instalar “dotenv” el cual es una biblioteca de node js que sirve para cargar variables de entorno desde un archivo llamado “.env” que crearemos más adelante. Y si tienes una versión de npm superior a 5.0.0 no es necesario colocar --save el cual sirve para que personas que clonen el proyecto o trabajen en el puedan instalar las dependencias fácilmente sin embargo luego de la versión 5.0.0 no es necesario ya que npm guarda automáticamente las dependencias en “package.json”

Texto

Descripción generada automáticamente con confianza media

**Paso 60:** Veremos si se instaló correctamente.

Texto

Descripción generada automáticamente

**Paso 61:** Ahora vamos a crear un nuevo archivo en nuestra carpeta raíz y lo vamos a llamar “google\_calendar.js”

Captura de pantalla de un celular

Descripción generada automáticamente

**Paso 62:** Así debe quedar nuestro archivo.

Interfaz de usuario gráfica, Texto

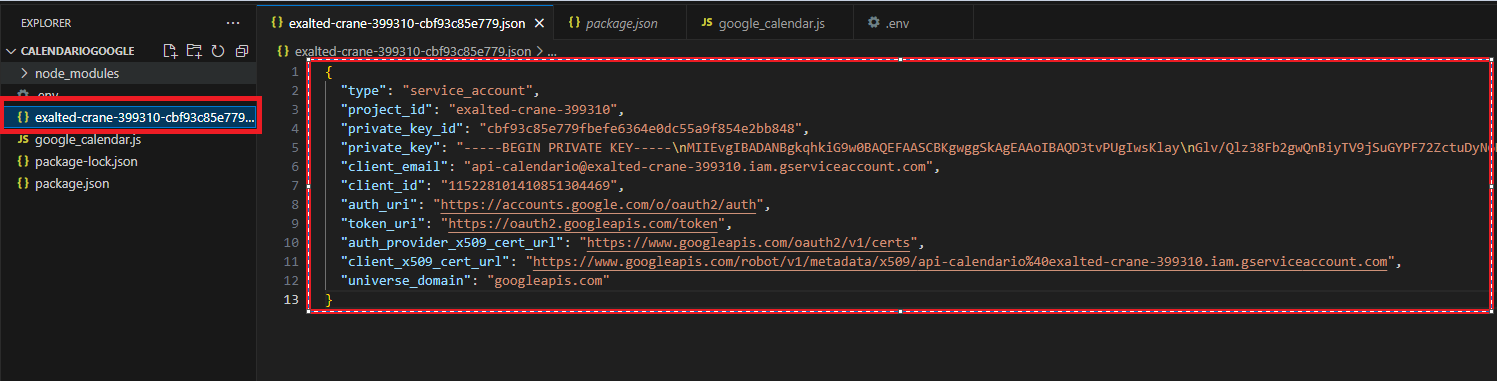
Descripción generada automáticamente

**Paso 63:** Ahora vamos a crear las variables de entorno. Creando un nuevo archivo llamado “.env”

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

Descripción generada automáticamente

**Paso 64:** Iremos a primer archivo el cual descargamos de la Google. Y copiamos todo lo que hay dentro.



**Paso 65:** Lo seleccionamos todo y damos click derecho.

Una captura de pantalla de una red social

Descripción generada automáticamente

**Paso 66:** Le damos “copy” o “copiar”

Captura de pantalla de computadora

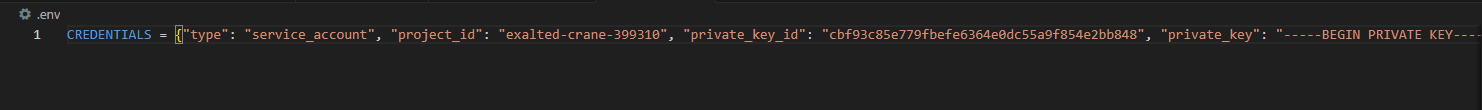
Descripción generada automáticamente con confianza media

**Paso 67:** Ahora iremos a nuestro archivo “.env” y vamos a crear una variable que contendrá todas las credenciales llamada “CREDENTIALS” y la haremos igual a lo que copiamos del archivo anterior. En este archivo de .env todas las variables deben ir todo en mayúscula

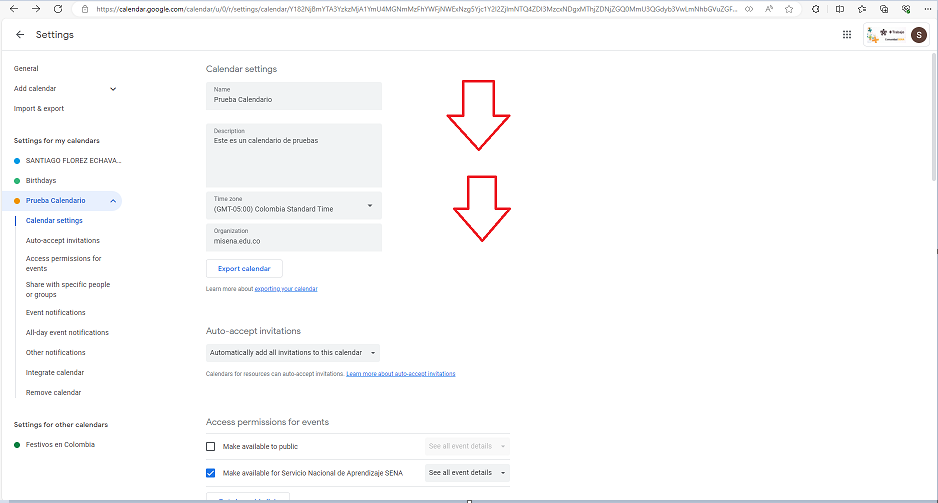
Captura de pantalla de un celular

Descripción generada automáticamente

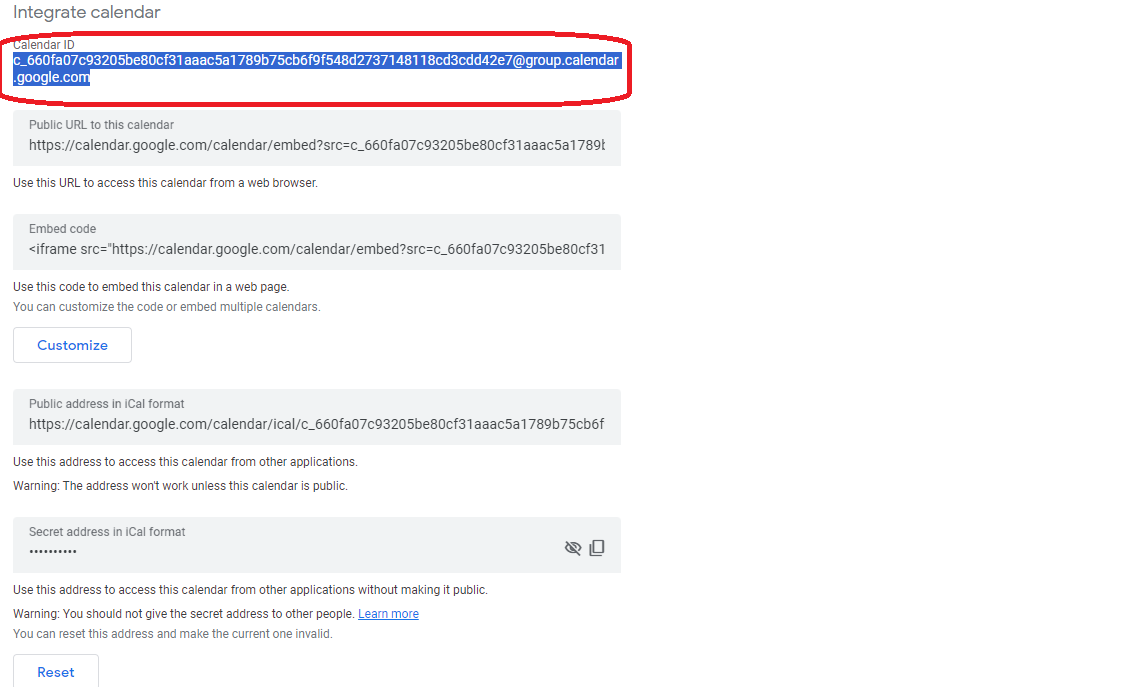
**Paso 68:** Ahora para que esté un poco más organizado vamos a dejarlo todo en una sola línea de texto, ten mucho cuidado de colocar todo bien.



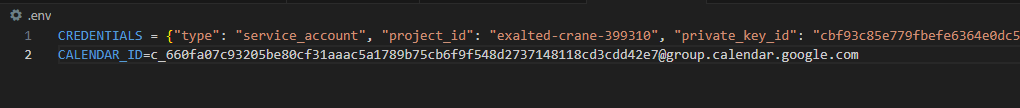
**Paso 69:** Ahora volveremos a nuestro calendario y vamos a settings, y vamos hacía abajo. (Paso 39 y 40 por si lo olvidaste.)



**Paso 70:** Buscaremos la opción de “Calendar ID” y copiaremos todo lo que dice ahí



**Paso 71:** Ahora volvemos a nuestro archivo “.env” creamos una variable llamada “CALENDAR\_ID” y la haremos igual al texto que acabamos de copiar.



**Paso 72:** Ahora vamos a ir a nuestro archivo js “google\_calendar.js” y vamos a importar las apis de google (línea 2) y también vamos a cargar las variables de entorno desde el archivo “.env” (línea 3)

Texto

Descripción generada automáticamente

**Paso 73:** Ahora hacemos que lea las credenciales y el id del calendario almacenándolo (línea 6 y 7)

Texto

Descripción generada automáticamente

**Paso 74:** Ahora vamos a configurar la API de Google Calendar con la (línea 10) vamos a solicitar acceso para la administración de calendarios del usuario mientras que en la (línea 11) vamos a crear un objeto “calendar” el cual va a interactuar con la API de Google Calendar usando la función de la biblioteca de “googleapis” llamada “calendar” y vamos a usar la versión 3 del api.

Texto

Descripción generada automáticamente

**Paso 75:** Y vamos a autenticar la aplicación usando las credenciales a ver si están correctas. (línea desde la 14 hasta la 19)

Texto

Descripción generada automáticamente

**Paso 76:** Vamos a definir nuestra zona horaria que en caso de Colombia es “-05:00”

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 77:** Ahora vamos a crear una función llamada “dateTimeForCalendar” y le haremos una función flecha (línea 25 a 27)

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente  
**Paso 78:** Ahora vamos a crear una variable que obtiene la fecha y hora actual (línea 28) además de eso vamos a extraer los componentes de la fecha como el año (línea 31) el mes (línea 32) como dato curioso los meses comienzan desde 0 en javascript así que hacemos que se agregue un +1 para que comience en 1 luego haremos una condicional con los meses de que en caso de que el número del mes sea menor a 0 agregue un 0 al comienzo (línea 33 a 35) también haremos los días (línea 36) y haremos la misma condicional con los días (línea 37 a 39) luego vamos a capturar las horas (línea 40) y la misma condicional (línea 41 a 43) y por último los minutos (línea 44) y la misma condicional (línea 45 a 47)

Captura de pantalla de computadora

Descripción generada automáticamente

**Paso 79:** Ahora dentro de la misma función vamos a construir una cadena de la fecha y la hora en un formato (ISO 8601) (Línea 50) también vamos a convertir esa cadena en un objeto date (línea 53) luego vamos a calcula la hora de inicio para agregar el evento al calendario (línea 56) y la hora de su finalización que será una hora después (línea 57) luego vamos a hacer que retorne las horas de cuando comenzó y cuando termina (líneas desde la 59 hasta la 62) , y dejamos ahí la función (línea 63)

Captura de pantalla de un teléfono celular

Descripción generada automáticamente

**Paso 80:** Ahora vamos a hacer que se imprima en la terminal para ver la hora para asegurarnos de que la función esta correcta (línea 66)

Texto

Descripción generada automáticamente

**Paso 81:** Ahora vamos a la terminal y vamos a colocar (node google\_calendar.js)

Texto

Descripción generada automáticamente

**Paso 82**: Y vemos como se agregó la hora correctamente, aunque no la comprendas si te sale algo parecido a fechas vamos bien.

Texto

Descripción generada automáticamente con confianza media

**Paso 83:** Ahora borramos donde imprimimos la hora (línea 66) y su comentario en caso de que lo hayas hecho (línea 65) y en su lugar vamos a crear una nueva función flecha llamada (insertEvent) y contendrá el argumento (event) que posteriormente vamos a crear.

Texto

Descripción generada automáticamente con confianza media

**Paso 84:** Realizaremos un try catch (líneas desde las 68 a la 72)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 85:** Dentro del try vamos a intentar insertar un evento en el calendario usando la api de Google Calendar (línea desde la 70 hasta la 74)

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Paso 86:** Dentro del try vamos a verificar que se haya insertado correctamente con una condicional (línea 77) en caso de que se haya insertado correctamente nos retorna 1 (línea 78) de lo contrario (línea 79) nos va a retornar 0 (línea 80)

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Paso 87:** Ahora en el catch (línea 82) vamos a capturar un error en caso de que lo haya no envié el error a la consola (línea 83) y además retorne 0 (línea 84) y vamos a terminar la función.

Texto

Descripción generada automáticamente

**Paso 88:** Ahora obtenemos la fecha y la hora para el evento (línea 89)

Texto

Descripción generada automáticamente

**Paso 89:** Ahora vamos a crear un evento para Google Calendar (línea 92) y lo llamaremos “event”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 90:** Dentro de este evento vamos a agregar las descripciones que nos pide crear un evento en google calendar, comenzaremos con por resumen (línea 93) es importante que mires las comillas ya que para seleccionar que agregaremos vamos a colocarla con comillas simple (‘ ’) y lo que se agregará con otro tipo de comillas conocidas como “plantillas de cadenas” o “template literals” las cuales son (` `), también agregaremos la descripción (líneas 94) y vamos a agregar la hora del comienzo (líneas desde la 95 hasta la 98) y cuando terminará el evento (líneas desde la 99 hasta la 102) y por último cerramos el evento (línea 103)

Texto

Descripción generada automáticamente

**Paso 91:** Ahora vamos a llamar la función “insertEvent” y le vamos a insertar el evento (línea 106) y luego en caso de que haya funcionado vamos a imprimir la respuesta (línea 107 a 109) y vamos a capturar el error en caso de que no haya funcionado (líneas desde la 110 a la 112)

Texto

Descripción generada automáticamente

**Paso 92:** Ahora vamos a la terminal de nuevo y vamos a colocar “node google\_calendar.js”

Texto

Descripción generada automáticamente con confianza baja

**Paso 93:** Si nos retorna “1” significa que funciono.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

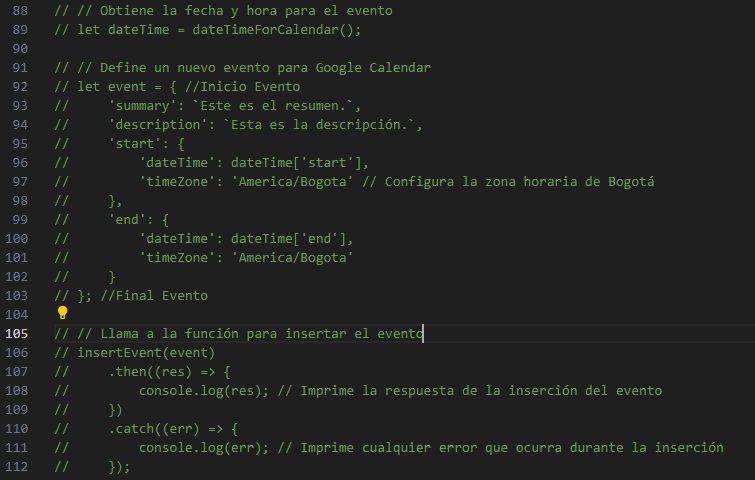
**Paso 94:** Vemos como efectivamente se agrego al calendario de google calendar.

Tabla

Descripción generada automáticamente con confianza media

**Paso 95:** Ahora que hemos agregado un evento al calendario vamos a comentar todo desde la (línea 88 a la 112), ya que no queremos que se sigan agregando cosas.

PD: Por si no sabías para hacerlo más sencillo puedes seleccionar todo el código y hacer “Ctrl + k” y luego “Ctrl + c” y si quieres descomentar mucho texto puedes hacer “Ctrl + k” y luego “Ctrl + u”



**Paso 96:** Ahora vamos a crear una función para obtener los eventos entre 2 fechas. Y la llamaremos “getEvent” (línea 115)

Interfaz de usuario gráfica, Texto

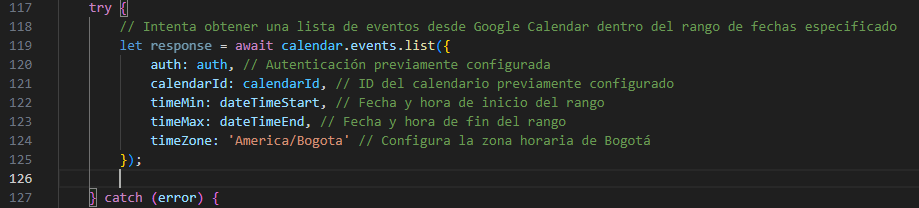
Descripción generada automáticamente

**Paso 97:** Realizaremos un “try” y “catch” (líneas desde la 117 a la 121)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 98:** Dentro del try vamos a obtener una lista de eventos desde google calendar (línea 119) y vamos a verificar la autenticación (línea 120) y el Id del calendario (línea 121) también la fecha y hora del inicio (línea 122) y la fecha y hora del final (línea 123) y por último vamos a configurar la zona horaria de Colombia (línea 124)

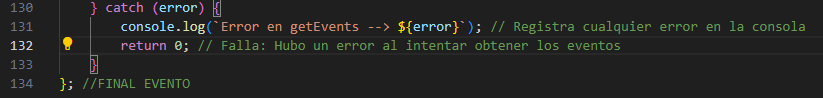


**Paso 99:** Ahora dentro del try vamos a obtener los eventos de la respuesta y lo vamos a almacenar (línea 128) y por último lo retornamos (línea 129)

Pantalla negra con letras blancas

Descripción generada automáticamente

**Paso 100:** Ahora por si sale mal el encontrar los eventos vamos a crear el catch donde se enviará un mensaje que contiene el error (línea 131) y también nos retornará 0 (línea 132)



**Paso 101:** Ahora vamos a definir la fecha de inicio (línea 137) y la fecha final (línea 138)

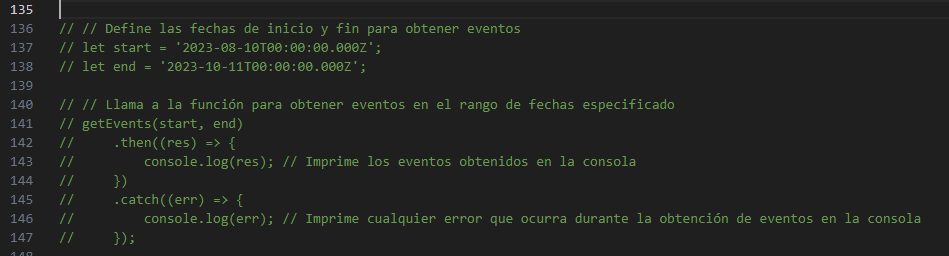
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 102:** Ahora vamos a llamar la función “getEvent” para obtener el rango de las fechas especificado (línea 141) y va a imprimir en caso de que salga bien (líneas desde la 142 hasta la 144) y en caso de que envíe error lo imprimirá (líneas desde la 145 hasta la 147)

Captura de pantalla de un celular

Descripción generada automáticamente **Paso 103:** Sin embargo, de momento vamos a comentar todo desde la (línea 136 hasta la 147)



**Paso 104:** Ahora vamos a crear una función para eliminar un evento por su id y la llamaremos (deleteEvent) (línea 150)

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

**Paso 105:** Dentro de la función vamos a crear un “try” “catch”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 106:** Dentro del try vamos a enviar la solicitud para eliminar un evento según su id (líneas desde la 153 hasta la 157) en la (línea 156) se recibe el id del evento el cual por el momento no conocemos, sin embargo, colócalo tal cual está que luego lo vamos a conseguir.

Texto

Descripción generada automáticamente

**Paso 107:** Ahora dentro del mismo try vamos a verificar si la eliminación fue exitosa (línea 160) y en caso de que si imprima 1 (línea 161) de lo contrario (línea 162) vamos a retornar un 0 (línea 163)

Texto

Descripción generada automáticamente

**Paso 108:** Ahora en el catch vamos (línea 165) vamos a capturar que hubo un error y lo va a imprimir (línea 166) y también va a retornar un 0 (línea 167)

Texto

Descripción generada automáticamente

**Paso 109:** Ahora vamos a quitar todo el comentario desde la (línea 136 hasta la 147) es decir deshacer el paso 103

Texto

Descripción generada automáticamente

**Paso 110:** Ahora en la terminar vamos a depurar el node.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 111:** De la información que nos envía vamos a buscar el “id”

Texto

Descripción generada automáticamente

**Paso 112:** Ahora que tenemos el id vamos de nuevo a comentar el código desde la (línea 136 hasta la 147)

Texto

Descripción generada automáticamente

**Paso 113:** Ahora vamos a capturar el id y lo vamos a definir (línea 172)

Texto

Descripción generada automáticamente

**Paso 114:** Ahora vamos a llamar a la función para eliminar el evento y le pasaremos el id (línea 175) y en caso de que se elimine correctamente imprimirá la respuesta en caso de que sea 1 se eliminó con éxito (línea 176 a 178) de lo contrario hará un catch que enviará un error (línea 179 a la línea 181)

Texto

Descripción generada automáticamente

**Paso 115:** Ahora vamos a nuestro terminar y de nuevo vamos a ejecutar el node

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 116:** Si la respuesta es 1 significa que se eliminó todo correctamente.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**Paso 117:** Podemos ver como se eliminó correctamente nuestro evento del calendario y con esto hemos finalizado nuestro proyecto donde aprendimos a crear eventos, ver la información de estos y eliminarlos usando una api.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente