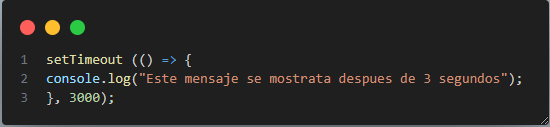
• Desarrollo de la actividad tipo investigación en la cual el aprendiz debe consultar en internet los siguientes interrogantes y dar un ejemplo de cada uno:

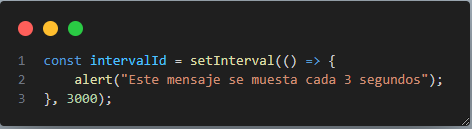
• Programación asíncrona:

o Que son los temporizadores setTimeout y setinterval (de un ejemplo)

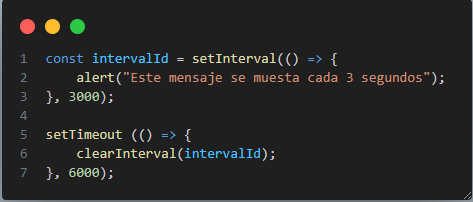
setTimeout: ejecuta una funcion una unica vez después de un retraso especificado



Setinterrval: Ejecuta una función repetidamente, en intervalos de tiempo especificados, hasta que se detiene esplicitamente.



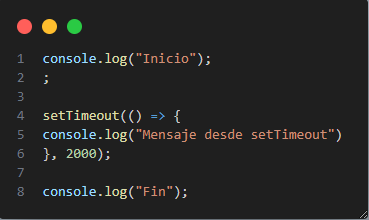
Para detener el intervalo



o Que es la asincronía y el evento loop (de un ejemplo)

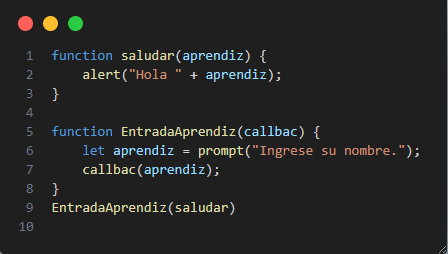
Asincronia permite que ciertas operaciones se realicen en segundo plano sin detener el flujo principal del programa. Esto es espcialmente útil en tareas que pueden tomar tiempo, como solicitudes de red, lectura de archivos o temporizadores, la asincronía permite que el código continue ejecutándose mientras espera el resultado de esas tareas.

Evento Loop; permite que JavaScript maneje tareas asincrónicas sin bloquear el flujo prinicpal



o Que son los callbacks (de un ejemplo)

Una función de callback es una función que se pasa a otra función como un argumento, que luego se invoca dentro de la funcion externa para completar algún tipo de rutina o acción



o Que son las promesas (de un ejemplo)

es un objeto que representa el eventual resultado(éxito o error) de una operación asincrónica.

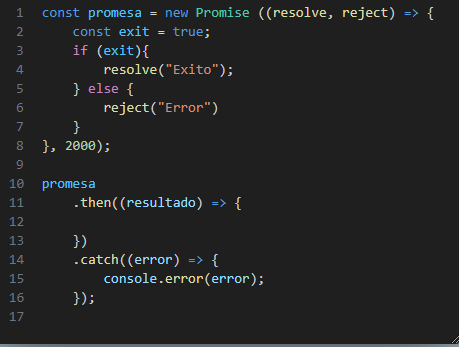
Las promesas facilitan el manejo de operaciones asincornicas, como solicitudes a un servidor, ya que nos permiten trabajar con el resultado cuando este disponible, si bloquear el código.

ESTADO DE UNA PROMESA

Pendiente: cuando la operación no ha terminado

Cumplida: cuando la operación se completa exitosamente.

Rechazada: cando ocurre un error en la operación.



o Que son las funciones asíncronas (async / await) (de un ejemplo)

async: declara una función como asincrónica, lo que permite usar la palabra clave await dentro de ella. Una función async siempre devuelve una promesa.

Await: se utiliza dentro de funciones async para esperar a que una promesa se resuelva o se rechace. Suspende la ejecución de la función hasta que la promesa se complete

