**-API**: Interfaz de programación de aplicaciones (Application Programming Interface). Es un conjunto de rutinas que provee acceso a funciones de un determinado software.

**-Stack**: Va apilando de forma organizada las diferentes instrucciones que se llaman. Lleva así un rastro de dónde está el programa, en qué punto de ejecución nos encontramos.

**-Memory Heap**: De forma desorganizada se guarda información de las variables y del scope. El espacio donde se guardan las funciones y variables.

-Schedule Tasks: Aquí se agregan a la cola, las tareas programadas para su ejecución.  
-Task Queue: Aquí se agregan las tares que ya están listas para pasar al stack y ser ejecutadas. El stack debe estar vacío para que esto suceda.  
-MicroTask Queue: Aquí se agregan las promesas. Esta Queue es la que tiene mayor prioridad.  
-El Event Loop: Es un loop que está ejecutando todo el tiempo y pasa periódicamente revisando las queues y el stack moviendo tareas entre estas dos estructuras.

**-Concurrencia**: Cuando dos o más tareas progresan simultáneamente.  
**-Paralelismo**: Cuando dos o más tareas se ejecutan, literalmente, a la vez, en el mismo instante de tiempo.  
**-Bloqueante**: Una llamada u operación bloqueante no devuelve el control a nuestra aplicación hasta que se ha completado. Por tanto el thread queda bloqueado en estado de espera.  
**-Síncrono**: Es frecuente emplear ‘bloqueante’ y ‘síncrono’ como sinónimos, dando a entender que toda la operación de entrada/salida se ejecuta de forma secuencial y, por tanto, debemos esperar a que  
se complete para procesar el resultado.  
**-Asíncrono**: La finalización de la operación I/O se señaliza más tarde, mediante un mecanismo específico como por ejemplo un callback, una promesa o un evento, lo que hace posible que la respuesta  
sea procesada en diferido.  
**-Call Stack**: La pila de llamadas, se encarga de albergar las instrucciones que deben ejecutarse. Nos indica en qué punto del programa estamos, por donde vamos.  
**-Heap**: Región de memoria libre, normalmente de gran tamaño, dedicada al alojamiento dinámico de objetos. Es compartida por todo el programa y controlada por un recolector de basura que se encarga de liberar aquello que no se necesita.  
**-Cola o Queue**: Cada vez que nuestro programa recibe una notificación del exterior o de otro contexto distinto al de la aplicación, el mensaje se inserta en una cola de mensajes pendientes y se registra su callback correspondiente. -**Eventloop o Loop de eventos**: Cuando la pila de llamadas (call stack) se vacía, es decir, no hay nada más que ejecutar, se procesan los mensajes de la cola. Con cada ‘tick’ del bucle de eventos, se procesa un nuevo mensaje.  
**-Hoisting**: Sugiere que las declaraciones de variables y funciones son físicamente movidas al comienzo del código en tiempo de compilación.  
**-DOM**: DOM permite acceder y manipular las páginas XHTML como si fueran documentos XML. De hecho, DOM se diseñó originalmente para manipular de forma sencilla los documentos XML.  
**-XML**: Lenguaje de marcado creado para la transferencia de información, legible tanto para seres humanos como para aplicaciones informáticas, y basado en una sencillez extrema y una rígida  
sintaxis. Así como el HTML estaba basado y era un subconjunto de SGML, la reformulación del primero bajo la sintaxis de XML dio lugar al XHTML; XHTML es, por tanto, un subconjunto de XML.  
**-Events**: Comportamientos del usuario que interactúa con una página que pueden detectarse para lanzar una acción, como por ejemplo que el usuario haga click en un elemento (onclick), que elija una  
opción de un desplegable (onselect), que pase el ratón sobre un objeto (onmouseover), etc.  
**-Compilar**: Compilar es generar código ejecutable por una máquina, que puede ser física o abstracta como la máquina virtual de Java.  
**-Transpilar**:Transpilar es generar a partir de código en un lenguaje código en otro lenguaje. Es decir, un programa produce otro programa en otro lenguaje cuyo comportamiento es el mismo que el original.

**🌮-call stack: el taquero (órdenes rápidas).  
👨‍🍳 -web APIs: la cocina.  
🌯 -callback queue: las órdenes preparadas.  
💁‍-event loop: el mesero.**