1.

El ciclo de eventos es el modelo en el que se basa javaScript. En este modelo hay un bucle el cúal se encarga de comprobar si hay alguna función pendiente de ejecutar en una cola en la que se apilan estas. Al crear un callback o función asíncrona está función se añade a la cola, lo que permite que la función que la invocó continue sin esperar a la respuesta de esta.

2.

Hay tres maneras de abodar la asincronía:

- **callbacks :** son funciones que se pasan como parámetros a otras y que se ejecutarán una vez termine la primera.
- **Promesas:** es un objeto que devuelven las funciones modernas y este respresenta la finalziación o el fallo de una función. Este objeto permite adjuntar callbacks para controlar los errores en caso de que los haya o para añadir la funcionalidad una vez la promesa finalize correctamente.
- **Async/ await :** una función async, define una función asíncrona y esta devolverá una promesa. Mediante la expresión await + funct dentro de una función async se pausa la ejecución hasta que se resuelva la promesa de la función llamada con la expresión. Una vez se resuelve la promesa se reanuna la ejecución de la función principal.

3.

La arquitectura propuesta constará de dos partes independientes

- Primera para tratado de datos
  - Sistema basado en Arquitectura brooker MQTT
  - El cliente en local enviará la información a través de los eventos necesarios
  - El servidor en local correspondiente a cada cliente, desarrollado en node con typescript, por la facilidad mediante librerias para trabajar con este tipo de broker. Además de ello trabaja muy bien con eventos y ofrece un mayo rendimiento frente a otros lenguajes de programación.
  - La bbdd sería una no relacional por la diversidad de datos que se podrían recibir y se encontraría en la nube.

- Segunda para la visualización de datos
  - Basado en una arquitectura cliente servidor.
  - El cliente podría estar hecho en Angular ya que su curva de aprendizaje no es muy elevado y que junto con frameworks como bootstrap pueden dar una mayor facilidad de programación.
  - El servidor en la nube por su parte sería un API Rest, el cuál se encargarría de tomar los datos de las bbdd que ha guardado previamente el servidor del broker MQTT.