* **Disponibilidad**: Se refiere al porcentaje tiempo que el sistema debe de estar disponible para su uso y completamente operacional. Ejemplo: “El sistema debe estar al menos el 99.5% disponible durante la semana entre las 6:00 AM y la media noche, y al menos el 99.95% disponible entre las 4:00 PM y las 6:00 PM”.
* **Eficiencia**:Cantidad de código y de recursos informáticos (CPU, memoria) que precisa un programa para desempeñar su función. Un programa que suma dos números y necesita 2 MB de memoria para funcionar, o que tarda 2 horas en dar una respuesta, es poco eficiente.
* **Seguridad**: Hasta qué punto se controlan los accesos ilegales a programas o datos. Un programa que permite el acceso de personas no autorizadas a ciertos datos es poco íntegro.
* **Interoperabilidad**: El coste y esfuerzo necesario para hacer que el software pueda operar conjuntamente con otros sistemas o aplicaciones software externos.
* **Confiabilidad**: Hasta qué punto se puede confiar en el funcionamiento sin errores del programa. Por ejemplo, si el programa suma dos números, pero en un 25% de los casos el resultado que da no es correcto, es poco fiable.
* **Robustez**: Atributos del software que posibilitan la continuidad del funcionamiento bajo condiciones no usuales (ingreso de datos no válidos, defectos en el software o el hardware conectado, condiciones de operación no esperadas).
* **Escalabilidad**: Capacidad para permitir una demanda creciente de usuarios o procesos sin desmejorar el desempeño.
* **Portabilidad**: Capacidad de operar en diferentes plataformas tecnológicas.
* **Flexibilidad**: Capacidad de adaptar el código a nuevos requerimientos.
* **Usabilidad**: El coste y esfuerzo de aprender a manejar un producto, preparar la entrada de datos e interpretar la salida del mismo.
* **Mantenibilidad:** El coste de localizar y corregir defectos en un programa que aparecen durante su funcionamiento.
* **Reusabilidad**: Hasta qué punto se puede transferir un módulo o programa del presente sistema a otra aplicación, y con qué esfuerzo.
* **Probable**: El coste de probar un programa para comprobar que satisface sus requisitos. Por ejemplo, si un programa requiere desarrollar una simulación completa de un sistema para poder probar que funciona bien, es un programa difícil de probar.
* **Componentes de Terceros:** Se describen los componentes comprados que serán usados por el sistema, las licencias aplicables, restricciones de uso, y cualquier característica sobre compatibilidad e interoperabilidad.
* **Interfaz de Usuario:** Descripción de las interfaces del usuario (de forma general) que serán implementadas.
* **Interfaz de Hardware:** Se especifica cualquier interfaz de hardware que será soportada por el usuario, incluyendo la estructura lógica, direcciones físicas, comportamiento esperado, etc.
* **Interfaz de Software:** Se describe cualquier interfaz de software que interactúa con el sistema. Pueden ser componentes comprados, re usados de otra aplicación o que se están desarrollando pero que quedan fuera del alcance del sistema aunque debe interactuar con ellos.
* **Interfaz de Comunicación:** Se describe cualquier interfaz de comunicación a otros sistemas o dispositivos, tales como una red local, remota, dispositivos seriales, etc.
* **Requerimientos de Licencias:** Se define cualquier requerimiento de ejecución de licencias u otra restricción de uso que debe ser exhibida por el software.
* **Estándares aplicables:** Se nombra cualquier estándar que se vaya a aplicar y se hace la referencia específica al estándar, incluyendo las secciones específicas que serán aplicadas.