



Pregunta **4**

Sin responder aún

Puntúa como 2,00

**Haga una lista de todos los requisitos funcionales con los que deberá cumplir su proyecto en la entrega final.**

En el contexto de sistemas embebidos, los requisitos funcionales se refieren a las especificaciones detalladas de las funciones y el comportamiento que un sistema embebido debe cumplir para satisfacer sus objetivos y requerimientos. Estos requisitos describen las acciones y operaciones específicas que el sistema debe realizar en respuesta a estímulos o entradas particulares. Los requisitos funcionales son esenciales para definir cómo un sistema embebido debe interactuar con su entorno y proporcionar las capacidades necesarias.

Algunos ejemplos de requisitos funcionales en sistemas embebidos podrían incluir:


- 1. Procesamiento de datos:** Especificaciones sobre cómo el sistema embebido debe procesar los datos de entrada, como la velocidad de procesamiento, los algoritmos a utilizar y los resultados esperados.
- 2. Comunicación:** Requisitos que definen cómo el sistema debe comunicarse con otros dispositivos o sistemas, incluyendo protocolos de comunicación, tasas de transferencia de datos y latencia.
- 3. Control:** Descripciones detalladas de cómo el sistema debe controlar dispositivos o actuadores, como motores, sensores o pantallas, incluyendo los valores de referencia, los umbrales y las respuestas esperadas.
- 4. Gestión de energía:** Especificaciones sobre cómo el sistema debe administrar el consumo de energía, como el modo de ahorro de energía, la gestión de baterías y la duración de la energía.
- 5. Interfaz de usuario:** Requisitos que describen la interacción del usuario con el sistema, incluyendo la apariencia de la interfaz, las acciones disponibles y las respuestas del sistema a las entradas del usuario.
- 6. Seguridad:** Especificaciones relacionadas con medidas de seguridad y protección de datos, como autenticación, cifrado y gestión de amenazas.
- 7. Tiempos de respuesta:** Requisitos que establecen los límites de tiempo dentro de los cuales el sistema debe responder a eventos o entradas específicas.
- 8. Cumplimiento de estándares:** Especificaciones que indican qué estándares o normativas debe cumplir el sistema embebido, como regulaciones de seguridad, estándares de comunicación, etc.

Estos son solo algunos ejemplos de requisitos funcionales en sistemas embebidos, y pueden variar significativamente según la aplicación y el contexto específico del sistema en cuestión. La definición precisa de estos requisitos es fundamental para el diseño, desarrollo y verificación exitosos de sistemas embebidos.

La redacción de requisitos funcionales en sistemas embebidos es una parte crucial del proceso de desarrollo, ya que establece la base para el diseño y la implementación. Aquí hay algunas recomendaciones para la redacción efectiva de requisitos funcionales en sistemas embebidos:

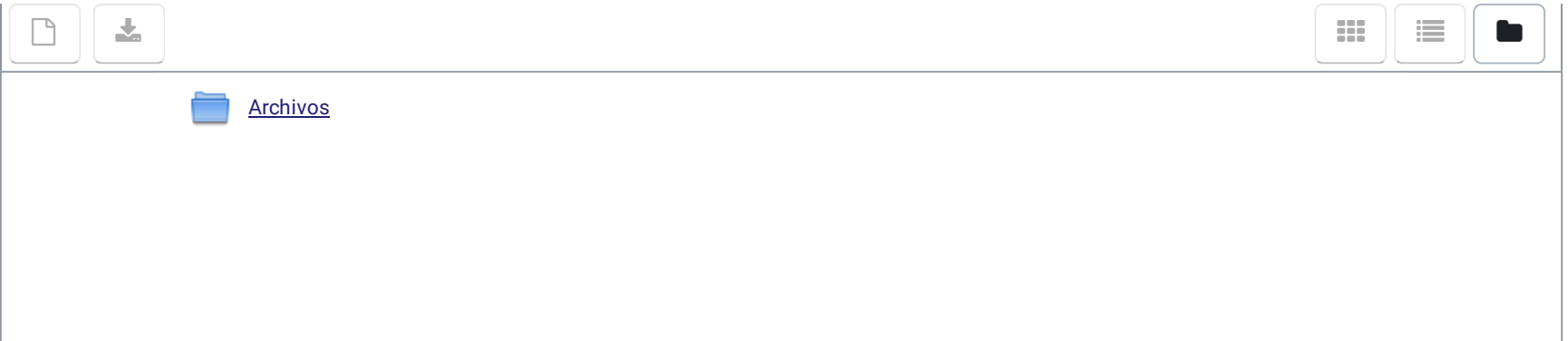
- 1. Sé claro y conciso:** Los requisitos deben ser redactados de manera clara y concisa. Utiliza un lenguaje simple y evita la ambigüedad. Deben ser comprensibles tanto para los desarrolladores como para los usuarios finales.
- 2. Usa verbos de acción:** Utiliza verbos de acción como "debe", "deberá", "debería" para indicar las acciones que el sistema debe realizar. Esto ayuda a clarificar las expectativas y las responsabilidades del sistema.
- 3. Especifica condiciones y criterios de aceptación:** Además de describir lo que el sistema debe hacer, incluye condiciones claras para determinar cuándo se considera que un requisito se ha cumplido. Esto facilita la verificación y validación posterior.
- 4. Evita requisitos redundantes:** Asegúrate de que cada requisito sea único y no se duplique en otros lugares de la documentación. La redundancia puede llevar a la confusión y a problemas de mantenimiento.
- 5. Mantén la trazabilidad:** Asocia cada requisito con el objetivo o la necesidad que está destinado a abordar. Esto facilita la trazabilidad y ayuda a garantizar que cada requisito sea realmente necesario.
- 5. Prioriza los requisitos:** Establece una jerarquía de prioridades para los requisitos. Esto ayuda a los equipos de desarrollo a centrarse en los aspectos más críticos del sistema cuando se enfrentan a restricciones de tiempo o recursos.

Al seguir estas recomendaciones, puedes mejorar significativamente la calidad y la eficacia de los requisitos funcionales en sistemas embebidos, lo que a su vez contribuirá a un desarrollo más suave y exitoso de los sistemas.



<Lista aquí los requisitos funcionales o sube un documento con dichos requisitos>

Tamaño máximo de archivo: 50 MB, número máximo de archivos: 1



### Tipos de archivo aceptados

Archivos de documentos .doc .docx .epub .gdoc .odt .oth .ott .pdf .rtf

Documento OpenDocument Text .odt

documento PDF .pdf

documento Word .doc

Documento Word 2007 .docx

Pregunta **5**

Sin responder aún

Puntúa como 1,00

**Haga una lista de todos los requisitos NO funcionales con los que deberá contar su proyecto en la entrega final.**

En el contexto de sistemas embebidos, los requisitos no funcionales se refieren a los aspectos y características del sistema que no están relacionados directamente con las funciones específicas que debe realizar, sino que se centran en cómo el sistema debe operar, su rendimiento, seguridad, escalabilidad y otros atributos de calidad. Estos requisitos son igualmente importantes que los requisitos funcionales, ya que afectan la experiencia del usuario y la capacidad del sistema para cumplir con sus objetivos. Aquí tienes algunos ejemplos de requisitos no funcionales en sistemas embebidos:

- 1. Rendimiento:** Establece la velocidad, la eficiencia y la capacidad de respuesta del sistema. Ejemplo: "El sistema embebido debe ser capaz de procesar un mínimo de 1000 transacciones por segundo".
- 2. Disponibilidad:** Define el tiempo durante el cual el sistema debe estar en funcionamiento y accesible. Ejemplo: "El sistema debe tener una disponibilidad del 99.9% durante las horas hábiles".
- 3. Seguridad:** Especifica las medidas de seguridad que deben implementarse para proteger los datos y el sistema en sí. Ejemplo: "El sistema debe cumplir con los estándares de seguridad ISO 27001".
- 4. Fiabilidad:** Indica la capacidad del sistema para funcionar de manera consistente y sin fallos. Ejemplo: "El sistema no debe fallar más de una vez al mes".
- 5. Escalabilidad:** Describe la capacidad del sistema para crecer y adaptarse a cambios en la carga de trabajo. Ejemplo: "El sistema debe ser escalable para admitir hasta 1000 usuarios concurrentes".
- 6. Mantenibilidad:** Define cómo se debe mantener y actualizar el sistema con el tiempo. Ejemplo: "El sistema debe permitir actualizaciones de software sin tiempo de inactividad".
- 7. Interoperabilidad:** Especifica cómo el sistema debe interactuar con otros sistemas o dispositivos. Ejemplo: "El sistema debe ser compatible con los protocolos de comunicación estándar de la industria".
- 8. Usabilidad:** Describe la facilidad de uso del sistema y la experiencia del usuario. Ejemplo: "La interfaz de usuario debe ser intuitiva y requerir una capacitación mínima para los usuarios".
- 9. Consumo de energía:** Indica los requisitos de eficiencia energética del sistema. Ejemplo: "El sistema debe operar con un consumo de energía no superior a 5 vatios".
- 10. Cumplimiento normativo:** Establece los estándares y regulaciones específicos que el sistema debe cumplir. Ejemplo: "El sistema debe cumplir con las regulaciones de seguridad de la FDA para dispositivos médicos".

**11. Tiempos de respuesta:** Define el tiempo máximo permitido para que el sistema responda a eventos específicos. Ejemplo: "El sistema debe proporcionar una respuesta en menos de 100 milisegundos a las solicitudes de usuario".

**12. Tolerancia a fallos:** Describe cómo el sistema debe manejar situaciones de fallo sin interrupciones graves. Ejemplo: "El sistema debe ser capaz de recuperarse automáticamente de fallos de hardware sin pérdida de datos".

Estos son solo ejemplos de requisitos no funcionales en sistemas embebidos, y la lista puede variar según las necesidades específicas del proyecto y la aplicación. Estos requisitos son fundamentales para garantizar que el sistema embebido cumpla con los estándares de calidad y rendimiento deseados.

↵


A ▾


B


I


T ▾


Ff ▾


 ▾


 ▾


















<Lista aquí los requisitos NO funcionales o sube un documento con dichos requisitos>


Tamaño máximo de archivo: 50 MB, número máximo de archivos: 1












 [Archivos](#)

Tipos de archivo aceptados

Archivos de documentos .doc .docx .epub .gdoc .odt .oth .ott .pdf .rtf  
Documento OpenDocument Text .odt  
documento PDF .pdf  
documento Word .doc  
Documento Word 2007 .docx

**Pregunta 6**

Sin responder aún

Puntúa como 1,00

Describa detalladamente el escenario de pruebas necesario para demostrar todas las características de su proyecto final. El escenario puede incluir maquetas, espacios de ejecución para robots, sujetos de prueba, etc.

