

Listas de chequeo para validación de documentación de diseño

Danny Julián Perilla Mikán
Yesica Paola Garzón Plazas
David Jesús Garcés Díaz

Marzo, 2025

Índice

1. Introducción	2
2. Listas de chequeo	2
2.1. Mapa de procesos	2
2.2. Herramientas de recolección de datos	3
2.3. Requerimientos funcionales	3
2.4. Requerimientos no funcionales	4
2.5. Casos de uso	4
2.6. Historias de usuario	5
2.7. Diagrama de clases	5
2.8. Diagrama de despliegue	6
3. Conclusiones	6

1. Introducción

La evaluación de artefactos en el ámbito del desarrollo de software es el proceso de revisar, analizar y validar los diferentes productos tangibles generados durante el ciclo de vida del desarrollo del proyecto. El objetivo de esta evaluación es asegurar que los artefactos cumplen con los requisitos establecidos, siguen las mejores prácticas de desarrollo y diseño, y son de alta calidad. La evaluación puede incluir varios aspectos como la corrección, la completitud, la consistencia, la claridad y la usabilidad de los artefactos.

2. Listas de chequeo

Una lista de chequeo en el ámbito del desarrollo de software es una herramienta utilizada para evaluar de manera sistemática los artefactos generados durante el ciclo de vida del proyecto. Consiste en un conjunto de criterios o preguntas que permiten verificar aspectos clave como la corrección, completitud, consistencia, claridad y usabilidad de los productos desarrollados. Su propósito es garantizar que los artefactos cumplen con los requisitos establecidos y las mejores prácticas, facilitando la identificación de errores o mejoras antes de su implementación final.

2.1. Mapa de procesos

Ítem	Descripción	Sí/No	Observaciones
1	¿Se identifican los procesos clave, estratégicos y de apoyo?	Sí	
2	¿Las interacciones entre procesos están claras?	No	Falta detallar algunas interacciones.
3	¿Se identifican los responsables de cada proceso?	No	Algunos procesos no tienen responsable.
4	¿Los procesos están organizados lógicamente?	Sí	
5	¿El mapa es claro y fácil de entender?	Sí	
6	¿Se usan símbolos o notaciones estándar?	Sí	
7	¿Se identifican los flujos de entrada y salida?	Sí	Los flujos están claramente definidos.
8	¿El mapa refleja los objetivos de la organización?	Sí	
9	¿Los procesos estratégicos y operativos están conectados?	Sí	
10	¿El nivel de detalle es adecuado?	Sí	Se comprende en detalle.

Cuadro 1: Lista de chequeo para mapa de procesos

2.2. Herramientas de recolección de datos

Ítem	Descripción	Sí/No	Observaciones
1.1	¿Se han definido los objetivos de la recolección de datos?	Sí	
1.2	¿La herramienta es adecuada para el tipo de datos?	Sí	
1.3	¿Se garantiza la precisión y confiabilidad?	Sí	
1.4	¿Permite validación para evitar inconsistencias?	No	Faltan controles.
1.5	¿Facilita exportación y análisis de datos?	Sí	
1.6	¿Cumple con estándares de privacidad y seguridad?	No	Revisar políticas.
1.7	¿Hay alternativas en caso de fallos?	Sí	
1.8	¿Permite automatizar la recolección?	No	Parcialmente.
1.9	¿Cuenta con documentación o capacitación?	Sí	
2.0	¿Es escalable para mayores volúmenes de datos?	No	Se ralentiza.

Cuadro 2: Lista de chequeo para herramientas de recolección de datos

2.3. Requerimientos funcionales

Ítem	Descripción	Sí/No	Observaciones
1	¿Cada requerimiento está definido y sin ambigüedades?	Sí	
2	¿Los requerimientos cumplen los objetivos del sistema?	Sí	Están alineados con los objetivos.
3	¿Se identificó correctamente el usuario o actor?	Sí	
4	¿Son medibles y verificables?	Sí	
5	¿Cada requerimiento describe solo una funcionalidad?	Sí	
6	¿No hay ambigüedades ni términos subjetivos?	Sí	
7	¿Están organizados de forma lógica?	Sí	Están ordenados por prioridad.
8	¿Se identificaron sus dependencias?	Sí	Revisar casos de uso.
9	¿Tienen criterios claros de aceptación?	Sí	
10	¿Fueron validados con los stakeholders?	Sí	

Cuadro 3: Lista de chequeo para requerimientos funcionales

2.4. Requerimientos no funcionales

Ítem	Descripción	Sí/No	Observaciones
1.1	¿Se identificaron los requerimientos de rendimiento?	Sí	
1.2	¿El sistema cumple los estándares de seguridad?	Sí	
1.3	¿Se definió un nivel adecuado de disponibilidad?	Sí	Revisar SLA.
1.4	¿El sistema es escalable?	Sí	
1.5	¿Los requerimientos de usabilidad están bien definidos?	No	Falta retroalimentación de usuarios.
1.6	¿Existe compatibilidad con otros sistemas?	Sí	
1.7	¿El sistema es fácil de mantener y actualizar?	Sí	Falta plan de mantenimiento.
1.8	¿Se consideraron restricciones legales o normativas?	Sí	
1.9	¿El sistema es confiable y tolerante a fallos?	No	Falta probar recuperación ante fallos.
2.0	¿Los requerimientos están bien documentados?	Sí	

Cuadro 4: Lista de chequeo para requerimientos no funcionales

2.5. Casos de uso

Ítem	Descripción	Sí/No	Observaciones
1	¿El diagrama es claro y comprensible?	Sí	
2	¿Los actores están bien identificados?	Sí	
3	¿Los casos de uso están correctamente definidos?	No	Falta detallar algunos casos de uso.
4	¿Se incluyen relaciones incluye y extiende ?	Sí	
5	¿La asociación entre actores y casos de uso es correcta?	Sí	
6	¿Los nombres de los casos de uso son claros?	Sí	Pero se puede ser más específico.
7	¿El diagrama refleja los requisitos funcionales?	Sí	
8	¿Se siguen las notaciones UML?	Sí	
9	¿Incluye comentarios si es necesario?	Sí	
10	¿Es coherente con otros modelos del sistema?	Sí	Especialmente con el diagrama de clases.

Cuadro 5: Lista de chequeo para diagrama de casos de uso

2.6. Historias de usuario

Ítem	Descripción	Sí/No	Observaciones
1.1	¿Sigue el formato estándar (como-quiero-para)?	Sí	
1.2	¿Es clara y comprensible?	Sí	
1.3	¿Define bien el beneficio o valor para el usuario?	Sí	
1.4	¿Tiene criterios de aceptación especificados?	Sí	
1.5	¿Cada historia es independiente de otras historias?	No	Hay dependencia.
1.6	¿Es negociable y permite mejoras?	Sí	
1.7	¿Es pequeña y estimable en esfuerzo?	Sí	Algunas pueden subdividirse.
1.8	¿Es verificable mediante pruebas?	Sí	
1.9	¿Está alineada con los objetivos del negocio?	Sí	
2.0	¿Tiene prioridad en el backlog del producto?	No	Falta consenso sobre su prioridad.

Cuadro 6: Lista de chequeo para historias de usuario

2.7. Diagrama de clases

Ítem	Descripción	Sí/No	Observaciones
1	¿El diagrama es claro y fácil de entender?	Sí	
2	¿Los nombres son descriptivos y comprensibles?	Sí	
3	¿Todas las clases están identificadas?	Sí	
4	¿Los atributos están bien definidos?	Sí	
5	¿Están todas las asociaciones representadas?	Sí	
6	¿Las relaciones de herencia están claras?	Sí	
7	¿Se respeta el encapsulamiento?	Sí	
8	¿Se utilizan métodos polimórficos adecuadamente?	Sí	
9	¿Se incluyen comentarios explicativos?	No	Se recomienda agregar descripciones.
10	¿El diagrama refleja los requisitos funcionales?	Sí	

Cuadro 7: Lista de chequeo para la evaluación del diagrama

2.8. Diagrama de despliegue

Ítem	Descripción	Sí/No	Observaciones
1.1	¿Se representan correctamente los nodos?	Sí	
1.2	¿Los artefactos en cada nodo están identificados?	Sí	
1.3	¿Las conexiones reflejan la comunicación real?	Sí	
1.4	¿Se usan bien los estereotipos y notaciones UML?	Sí	
1.5	¿El diagrama es claro y comprensible?	Sí	
1.6	¿Las relaciones entre nodos están etiquetadas?	Sí	
1.7	¿Se incluyen protocolos de comunicación?	No	Especificar los protocolos de conexión.
1.8	¿Servidores, clientes y dispositivos están diferenciados?	Sí	
1.9	¿Los componentes en cada nodo están especificados?	Sí	
2.0	¿El diagrama refleja bien la arquitectura?	Sí	

Cuadro 8: Lista de chequeo para diagrama de despliegue

3. Conclusiones

Las listas de chequeo elaboradas proporcionan una herramienta estructurada y eficaz para evaluar distintos aspectos del desarrollo de software. A través de estos criterios, es posible identificar fortalezas y áreas de mejora en cada etapa del proyecto, garantizando claridad, cumplimiento de estándares y alineación con los objetivos del negocio. Además, la inclusión de observaciones permite detectar y corregir posibles deficiencias, asegurando que los diagramas y especificaciones reflejen con precisión la arquitectura y funcionalidad del sistema.