

FORMATO DE TALLER N°

1. DATOS INFORMATIVOS

Nombre del estudiante:	Dennison Chalacan, Jeffrey Manobanda, Jhordy Marcillo
Docente:	Mgt. Jenny Alexandra Ruiz Robalino
Fecha:	20 de octubre de 2025
NRC:	27837

2. OBJETIVO DEL TALLER

Analizar y desarrollar las preguntas propuestas por Kendall & Kendall (Cap. II, p. 48) para fortalecer la comprensión del rol del analista de sistemas, sus habilidades clave y la interacción con los usuarios y el entorno organizacional, aplicando pensamiento crítico y técnico

3. DESARROLLO

Instrucciones: Lea el Capítulo II “El analista de sistemas” y desarrolle las preguntas planteadas en la página 48 del texto guía. Responda con fundamento teórico, ejemplos prácticos y argumentación personal

N.º	Pregunta (Kendall & Kendall, p. 48)	Respuesta del estudiante
1	¿Cuál es el papel principal del analista de sistemas dentro de una organización?	<p>El papel principal del analista de sistemas es actuar como enlace entre los usuarios y el equipo técnico de desarrollo con el fin de diseñar, implementar y mantener sistemas de información que satisfagan las necesidades del negocio. Teóricamente (según Kendall & Kendall), el analista identifica problemas, analiza los requerimientos del negocio y diseña sistemas de información que mejoren la eficiencia y eficacia de la organización. Por lo tanto, su función es traducir las necesidades operativas y estratégicas en especificaciones técnicas claras.</p> <p>Universidad de las Fuerzas Armadas – ESPE 100 años de excelencia académica</p> <p>Por ejemplo, si el departamento de marketing necesita un sistema para rastrear la efectividad de sus campañas en redes sociales, el analista no solo debe entender qué es "efectividad" para marketing (datos, KPIs), sino también cómo diseñar una base de datos y una interfaz que capture y presente esa información de manera útil. Por lo que el analista garantiza que la tecnología apoye efectivamente los</p>

		objetivos estratégicos de la empresa
2	¿Qué habilidades personales y técnicas son necesarias para desempeñar eficazmente esta función?	<p>Para ser eficaz, el analista necesita combinar habilidades técnicas, como el conocimiento en programación, bases de datos, modelado de sistemas, y metodologías de desarrollo, con habilidades blandas, como comunicación, liderazgo, trabajo en equipo, pensamiento analítico y empatía. Por lo que un analista técnicamente brillante pero incapaz de comunicarse con el gerente de finanzas fracasará. De hecho, la mayoría de los proyectos de sistemas fallan por una mala definición de requisitos (una falla de comunicación), no por una falla técnica de programación. Por ejemplo, un analista debe ser capaz de entender tanto el lenguaje del usuario (procesos y necesidades) como el del programador (código y arquitectura del sistema),</p> <p>Universidad de las Fuerzas Armadas – ESPE 100 años de excelencia académica actuando como traductor entre ambos mundos.</p>
3	¿Por qué es importante la comunicación entre el analista y los usuarios finales?	<p>La comunicación efectiva garantiza que el sistema desarrollado realmente responda a las necesidades del usuario, sin una comunicación fluida, se crea una brecha que lleva al fracaso del sistema. Dicho de otro modo, si el analista no entiende exactamente lo que el usuario necesita, cómo trabaja y qué problemas enfrenta, construirá un sistema que, aunque técnicamente funcione, no será utilizado. Un ejemplo práctico: Un analista diseña un nuevo sistema de facturación basándose en lo que él cree que el departamento de contabilidad necesita. Al entregarlo, los usuarios lo rechazan porque el sistema no permite aplicar notas de crédito de la forma específica que la ley del país exige, un detalle que el analista nunca preguntó y el usuario asumió que era obvio. Por ello, la comunicación constante (mediante entrevistas, prototipos, validaciones) es vital para garantizar la aceptación y la usabilidad del sistema, alineando las expectativas con el producto final.</p>
4	¿Cómo contribuye el analista al proceso de toma de decisiones empresariales?	<p>El analista contribuye de dos maneras principales: (1) proporcionando las herramientas para la toma de decisiones y (2) asesorando sobre la viabilidad de las soluciones. En primer lugar, Universidad de las Fuerzas Armadas – ESPE 100 años de</p>

		<p>excelencia académica los sistemas que el analista diseña (como los Sistemas de Soporte a la Decisión - DSS, o los Cuadros de Mando - Dashboards) transforman datos brutos en información útil. Por ejemplo, un gerente de ventas no toma decisiones mirando miles de registros de ventas; las toma mirando un reporte que el analista diseñó, el cual muestra "los 5 productos menos vendidos" o "la región con mayor crecimiento". Así pues, el analista define qué información es relevante y cómo debe presentarse para que la gerencia pueda decidir estratégicamente. Además, durante la fase de análisis, el analista evalúa la viabilidad (económica, técnica y operativa) de un proyecto. Al presentar un estudio de costo-beneficio, el analista está asesorando directamente a la alta dirección sobre si una inversión en tecnología es rentable o no.</p>
5	<p>Explique con un ejemplo práctico cómo un analista identifica y soluciona un problema en un sistema existente.</p>	<p>Situación (Problema): Una cadena de librerías utiliza un sistema de inventario obsoleto. Frecuentemente, la página web muestra libros "en stock" que ya se han vendido en la tienda física, generando quejas de clientes en línea. 1.</p> <p>Identificación del Problema (El Analista en acción): Observación y Entrevistas: El analista primero entrevista a los cajeros, quienes explican que el sistema tarda hasta 24 horas en sincronizar el inventario de la tienda física con el servidor de la tienda en línea.</p> <p>Análisis del Sistema (DFD): El analista mapea el flujo de datos actual y descubre que la actualización se hace mediante un proceso batch (por lotes) que corre solo una vez por noche.</p> <p>Diagnóstico: El problema no es el registro de la venta, sino la latencia (retraso) en la sincronización de las bases de datos (la de la tienda y la de la web).</p> <p>2. Diseño y Solución: Propuesta: El analista determina que un proceso batch ya no es viable para un negocio de e-commerce. Propone rediseñar la arquitectura para que funcione mediante servicios web (APIs).</p> <p>Nuevo Diseño: En el nuevo sistema, cuando un cajero vende un libro en la tienda (Sistema POS), este sistema hace una llamada inmediata (en tiempo real) a</p>

		<p>la API del servidor web, actualizando el stock de la base de datos central en segundos.</p> <p>Resultado: Finalmente, al implementar la solución, el inventario se sincroniza en tiempo real. De esta manera, el analista no solo solucionó las quejas de los clientes, sino que mejoró la integridad de los datos de la empresa y optimizó un proceso crítico de negocio.</p>
--	--	--

4. CONCLUSIONES

- El rol del analista de sistemas trasciende lo puramente técnico; es, en esencia, un rol de traductor y mediador estratégico dentro de la organización, asegurando que la tecnología sirva directamente a los objetivos del negocio. - Las habilidades blandas, especialmente la comunicación y la empatía, son tan críticas, o incluso más, que el dominio técnico. La capacidad de entender el "mundo" del usuario y gestionar sus expectativas es lo que define el éxito o fracaso de un sistema. - El analista no solo reacciona a problemas, sino que contribuye proactivamente a la toma de decisiones, diseñando las herramientas que convierten los datos en información gerencial y evaluando la viabilidad de las inversiones tecnológicas.

5. RECOMENDACIONES

- Fomentar un enfoque de "aprendizaje dual": mantenerse constantemente actualizado en las nuevas tecnologías (IA, cloud, ciberseguridad), pero, simultáneamente, dedicar tiempo a comprender a fondo el negocio o industria en la que trabajan (finanzas, logística, salud).
- Priorizar el uso de prototipos y metodologías ágiles durante el diseño. Permitir que el usuario "juegue" con un modelo temprano del sistema facilita una retroalimentación mucho más rica y rápida, evitando costosos rediseños después de la fase de programación.
- Se recomienda fomentar canales de comunicación claros entre analistas y usuarios para asegurar la correcta definición y seguimiento de los requerimientos del sistema.

REFERENCIAS

Kendall, K. E., & Kendall, J. E. (2011). Análisis y diseño de sistemas (8.ª ed.). Pearson Educación.

UNIDAD 1: Análisis de Sistemas de Información

1. Introducción al Análisis de Sistemas Software

El análisis es el proceso de descomponer un sistema o problema en sus partes para entender su estructura y funcionamiento. Según la Real Academia Española:

1. Distinción y separación de las partes de un todo para conocer sus principios o elementos.
2. Examen intelectual de una obra o realidad.
3. En informática: estudio mediante técnicas informáticas de los límites, características y posibles soluciones de un problema tratable por ordenador.

2. Objetivo del Análisis de Sistemas de Información (ASI)

El objetivo principal del análisis es obtener una especificación detallada del sistema de información que:

- Satisfaga las necesidades de información de los usuarios.
- Sirva como base para el diseño posterior del sistema.
- Reduzca los riesgos y errores antes de la construcción del software.

3. Actividades del Análisis de Sistemas de Información

1. Recolección de requisitos: identificación de lo que los usuarios necesitan.
2. Modelado del sistema: representación de procesos, actores y flujos de información.
3. Definición de casos de uso: descripción de cómo interactúan los usuarios con el sistema.
4. Identificación de subsistemas: división lógica de las funciones del sistema.
5. Análisis de clases y componentes: definición de los objetos y sus relaciones.
6. Trazabilidad: conexión entre los requisitos, el diseño y las pruebas.

4. Herramientas y Metodologías

- Métrica Versión 3 (MV3): metodología oficial española para el análisis y diseño de sistemas de información.

Enlace: <https://manuel.cillero.es/doc/metodologia/metrica-3/>

- OCW UC3M – Ingeniería del Software III: material abierto sobre análisis y diseño de sistemas.

Enlace: <http://ocw.uc3m.es/cursos-archivados/ingneieria-del-software-iii/material-de-clase>

5. Aplicabilidad del Análisis

El análisis se aplica para:

- Detectar problemas o ineficiencias en los procesos actuales.
- Identificar las necesidades de información reales.
- Establecer una base sólida para el diseño del software.
- Asegurar la alineación entre negocio y tecnología.

TALLER DE APLICACIÓN

Duración: 10 minutos

Objetivo: Verificar la comprensión de los conceptos clave del análisis de sistemas.

Nº	Pregunta	Opciones	Respuesta Correcta
1	¿Cuál es el propósito principal del análisis de sistemas de información?	A) Diseñar el sistema B) Especificar detalladamente el sistema C) Programar el sistema	B
2	¿Qué actividad forma parte del análisis?	A) Compilar el código B) Definir casos de uso C) Instalar el software	B
3	¿Qué representa el modelado del sistema?	A) Las interacciones entre usuarios y procesos B) La codificación del sistema C) La instalación del hardware	A
4	¿Cuál es la función de la trazabilidad?	A) Conectar los requisitos con el diseño y pruebas B) Controlar versiones del código C) Crear diagramas UML	A
5	¿Qué metodología es recomendada para el análisis en administración pública?	A) SCRUM B) Métrica Versión 3 C) CMMI	B

Conclusión

El análisis de sistemas de información permite identificar y documentar detalladamente los requisitos y necesidades del usuario, estableciendo una base sólida para el diseño y desarrollo del software. Gracias a esta fase, se minimizan riesgos de errores, se asegura la coherencia entre los diferentes componentes del sistema y se mejora la trazabilidad entre requisitos, diseño y pruebas, lo que contribuye a la calidad y efectividad del producto final.

Realizar un análisis facilita comprender a profundidad los procesos, actores y flujos de información de la organización, permitiendo detectar ineficiencias y problemas que podrían pasar desapercibidos. Además, ayuda a establecer prioridades, organizar subsistemas y garantizar que el software se alinee con los objetivos estratégicos del negocio.

El análisis de sistemas también sirve como una herramienta de mejora continua, ya que proporciona datos y modelos que pueden reutilizarse o adaptarse para futuras actualizaciones. Permite que los equipos de desarrollo adopten metodologías ágiles o tradicionales de manera informada, fomentando la innovación tecnológica y la optimización de los procesos organizacionales.

UNIDAD 1: Proceso de Desarrollo de Software

1. Introducción al Proceso de Desarrollo de Software

El proceso de desarrollo de software (PDS) define el conjunto de actividades, métodos, prácticas y transformaciones que permiten construir un producto de software de calidad. Este proceso establece el 'qué, quién, cuándo y cómo' del desarrollo.

2. Importancia del Proceso de Desarrollo

El desarrollo de software es un proceso complejo que requiere organización y planificación. Su correcta gestión permite reducir errores, optimizar recursos y mejorar la comunicación entre usuarios, analistas, diseñadores y programadores.

3. Actividades Fundamentales del PDS

1. Especificación del software: definir qué debe hacer el sistema.
2. Desarrollo: implementación de las funcionalidades definidas.
3. Validación: comprobación de que el sistema cumple los requisitos.
4. Evolución: adaptación del software a cambios y mejoras futuras.

4. Modelos de Proceso de Software

Existen diversos modelos que estructuran el proceso de desarrollo, entre los más destacados:

- Modelo en Cascada: desarrollo secuencial desde los requisitos hasta la entrega. Ideal para proyectos pequeños o con requisitos estables.
- Modelo Iterativo e Incremental: permite construir el software en versiones sucesivas, incorporando retroalimentación en cada iteración.
- Modelo en Espiral: combina iteración y gestión de riesgos; cada ciclo implica planificación, análisis de riesgos, desarrollo y evaluación.
- Proceso Unificado (USDP): orientado a objetos, dirigido por casos de uso, centrado en la arquitectura y basado en iteraciones.

5. Fases del Proceso Unificado

1. Inicio: definición de los objetivos del proyecto.
2. Elaboración: diseño de la arquitectura y gestión de riesgos.
3. Construcción: desarrollo e integración de componentes funcionales.
4. Transición: entrega y despliegue del producto final.

6. Beneficios de un Proceso Definido

- Favorece la comunicación entre equipos.
- Mejora la trazabilidad entre requisitos y entregables.
- Facilita el control de calidad.
- Permite una mejor gestión del tiempo y recursos.

TALLER DE APLICACIÓN

Duración: 8 minutos

Objetivo: Evaluar la comprensión de los conceptos clave del proceso de desarrollo de software.

Nº	Pregunta	Opciones	Respuesta Correcta
1	¿Qué define el proceso de desarrollo de software?	A) El diseño visual B) El conjunto de actividades, métodos y prácticas C) Solo la codificación	B
2	¿Cuál es la primera actividad del proceso de desarrollo?	A) Especificación del software B) Validación C) Evolución	A
3	¿Qué caracteriza al modelo en cascada?	A) Desarrollo iterativo B) Etapas secuenciales C) Entregas parciales	B
4	¿Qué principio rige el Proceso Unificado?	A) Programación estructurada B) Dirigido por casos de uso C) Desarrollo sin documentación	B
5	¿Cuál es un beneficio de tener un proceso definido?	A) Aumenta el trabajo manual B) Facilita el control de calidad C) Elimina la necesidad de pruebas	B

Conclusión

El proceso de desarrollo de software (PDS) permite organizar las actividades, métodos y roles necesarios para construir un sistema de manera ordenada y eficiente. La planificación adecuada del PDS contribuye a reducir errores, optimizar recursos y mejorar la comunicación entre los distintos actores involucrados, desde usuarios hasta desarrolladores, asegurando que cada fase cumpla con su propósito.

Elegir un modelo de desarrollo adecuado como Cascada, Iterativo, Espiral o Proceso Unificado permite adaptar el enfoque a las características del proyecto, el tamaño del equipo y la estabilidad de los requisitos. Esto asegura que el software evolucione de manera controlada, incorporando retroalimentación y gestionando riesgos de forma efectiva, lo que se traduce en productos más robustos y alineados con las necesidades del usuario.

El PDS no solo guía el desarrollo, sino que, combinado con un análisis detallado de requisitos, garantiza que los objetivos del proyecto se cumplan con calidad. Mientras el análisis determina qué debe hacer el sistema, el proceso define cómo y cuándo realizar cada actividad, incluyendo validación y evolución. Esto permite una trazabilidad clara entre requisitos, diseño, pruebas y entregables, promoviendo un producto final confiable y funcional.

Clase: Modelado Básico con Casos de Uso

Duración total: 25 minutos

Docente: Ing. Jenny Ruiz R.

Universidad de las Fuerzas Armadas ESPE

Objetivo de aprendizaje

Comprender la función de los casos de uso como herramienta para identificar, especificar y comunicar los requisitos funcionales de un sistema de software.

Estructura de la clase (25 minutos)

Fase	Tiempo	Actividad
Introducción	5 min	Motivación y presentación del tema con ejemplo práctico.
Desarrollo	15 min	Explicación del modelo de casos de uso y sus elementos.
Taller	5 min	Aplicación mediante cuestionario con retroalimentación.

Resumen del contenido principal

1. ¿Qué es un Caso de Uso?

Técnica creada por Ivar Jacobson para identificar funcionalidades del sistema. Describe cómo un actor interactúa con el sistema para alcanzar un objetivo y refleja los requisitos funcionales desde el punto de vista del usuario.

2. Elementos del modelo de casos de uso:

- Actores: entidades externas que interactúan con el sistema.
- Casos de uso: funcionalidad o servicio que ofrece el sistema a un actor.
- Relaciones: asociaciones actor–caso de uso (include/extend).
- Frontera del sistema: delimita lo que está dentro o fuera del sistema.

3. Ejemplo – Feria de Subastas:

Actores: Comprador, Vendedor, Observador.

Casos de uso: Registrar usuario, Registrar artículo, Pujar, Consultar lista de artículos, Modificar artículo.

4. Estructura textual de un caso de uso:

Nombre, Actores, Objetivo, Precondiciones, Postcondiciones, Escenario básico.

Taller de Aplicación (5 preguntas)

Nº	Pregunta	Opciones	Respuesta Correcta
1	¿Cuál es el objetivo principal del modelo de casos de uso?	A) Representar la estructura interna del sistema B) Identificar la funcionalidad desde el punto de vista del usuario C) Mostrar el código fuente del sistema	B
2	¿Qué representa un actor en UML?	A) Un componente interno del sistema B) Un rol que interactúa con el sistema C) Un archivo de configuración	B
3	¿Cuál de los siguientes elementos NO forma parte de la especificación textual de un caso de uso?	A) Escenario básico B) Postcondiciones C) Código fuente	C
4	En el caso de uso 'Registrar artículo', ¿cuál es una pre-condición válida?	A) El usuario debe estar registrado como vendedor B) El sistema debe estar apagado C) El artículo debe estar adjudicado	A
5	¿Qué diferencia principal hay entre el modelo de casos de uso y el modelo lógico de datos?	A) El primero describe la forma de usar el sistema; el segundo, la información que contiene B) Ambos representan el mismo tipo de relaciones C) Ninguno se usa en análisis	A

Conclusión

Los casos de uso constituyen una herramienta esencial para capturar y comunicar los requisitos funcionales del sistema desde la perspectiva del usuario. Permiten representar cómo interactúan los actores con el sistema para lograr objetivos específicos, asegurando que las necesidades reales de los usuarios sean correctamente identificadas y comprendidas por los desarrolladores.

La correcta identificación de actores, casos de uso, relaciones y fronteras del sistema facilita la organización de la información y la planificación del desarrollo. Además, la especificación textual, con sus elementos como precondiciones, postcondiciones y escenarios básicos, garantiza que cada funcionalidad esté claramente definida y pueda ser utilizada como referencia durante el diseño, la

implementación y las pruebas.

El uso de casos de uso permite que todos los integrantes del proyecto analistas, diseñadores, desarrolladores y usuarios tengan un entendimiento común de las funcionalidades del sistema. Esto no solo reduce la ambigüedad y posibles errores, sino que también sirve como base para elaborar diagramas UML, planificar pruebas y coordinar tareas en equipo, mejorando la eficiencia del proyecto.