



1. DATOS INFORMATIVOS

Nombre del estudiante:	Caetano Flores, Jordan Guaman, Anthony Morales y Leonardo Narváez
Docente:	Mgt. Jenny Alexandra Ruiz Robalino
Fecha:	20/10/2025
NRC:	27837

2. Introducción

El desarrollo de software constituye una disciplina esencial dentro de la Ingeniería de Software, cuyo propósito es producir sistemas de calidad que satisfagan las necesidades de los usuarios y las organizaciones. Este proceso no se limita a la programación, sino que abarca un conjunto de actividades estructuradas y metodológicas, conocidas como Proceso de Desarrollo de Software (PDS), que definen qué se hace, quién lo hace, cuándo y cómo se realiza.

En este contexto, el Análisis de Sistemas de Información (ASI) representa una etapa crítica, ya que permite comprender el problema que se desea resolver, identificar los requisitos del sistema y definir una base sólida para el diseño posterior. Mediante el análisis, los ingenieros de software descomponen el sistema en sus partes fundamentales, examinan los procesos de negocio, los flujos de información y las interacciones entre los actores involucrados, asegurando que el sistema resultante responda a las necesidades reales del usuario.

Asimismo, el modelado con casos de uso, propuesto por Ivar Jacobson, se presenta como una herramienta clave dentro del análisis funcional, al permitir representar las interacciones entre los actores y el sistema de forma gráfica y textual. Los casos de uso no solo comunican los requisitos funcionales de manera clara, sino que también facilitan la validación con los usuarios y la trazabilidad hacia las fases de diseño y prueba.

De manera complementaria, el Capítulo I del libro “Análisis y diseño de sistemas” de Kendall & Kendall (2011) profundiza en la importancia de los sistemas de información en las organizaciones modernas, destacando su papel estratégico en la toma de decisiones y en la automatización de procesos. Este marco teórico refuerza la necesidad de aplicar métodos estructurados y participativos para el desarrollo de sistemas eficaces, alineados con los objetivos del negocio.

3. OBJETIVO DEL TALLER

- **Comprender el concepto y la importancia del Proceso de Desarrollo de Software (PDS)**, reconociendo sus principales actividades, fases y modelos de referencia.
- **Identificar las etapas y objetivos del Análisis de Sistemas de Información**, comprendiendo su rol en la detección de necesidades y la definición de requisitos funcionales.
- **Aplicar la técnica de Casos de Uso** para modelar de forma clara y estructurada las interacciones entre los actores y el sistema, utilizando sus elementos principales (actores, casos de uso, relaciones y frontera del sistema).
- **Relacionar los fundamentos teóricos del análisis y diseño de sistemas** con las perspectivas planteadas por Kendall & Kendall, valorando su aplicación en contextos organizacionales reales.
- **Desarrollar una visión integral del ciclo de vida del software**, comprendiendo la relación entre análisis, diseño, construcción y evolución del sistema.
- **Fomentar el pensamiento crítico y analítico** para evaluar modelos de desarrollo y metodologías de análisis, priorizando la calidad, la trazabilidad y la comunicación efectiva entre los miembros del equipo de desarrollo.

4. DESARROLLO

Tabla de Preguntas U1T1

N.º	Pregunta (Kendall & Kendall, p. 48)	Respuesta del estudiante
1	¿Cuál es el papel principal del analista de sistemas dentro de una organización?	El papel principal del analista de sistema es actuar como el puente fundamental entre las necesidades del negocio y las soluciones tecnológicas. Su misión consiste en investigar y comprender los problemas y procesos de una organización, para luego traducir los requerimientos de los usuarios en especificaciones técnicas clara que el equipo de desarrollo pueda implementar. En esencia, el analista diseña un sistema informático eficiente que resuelve dichas necesidades, asegurando que la solución final esté alineada con los objetivos estratégicos y aporte un valor real a la empresa.
2	¿Qué habilidades personales y técnicas son necesarias para	Un analista de sistemas eficaz es, por encima de todo, un solucionador de problemas que combina habilidades personales y técnicas. Debe ser un comunicador capaz de forjar relaciones duraderas, poseer una sólida ética



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

	desempeñar eficazmente esta función?	personal y profesional, y ser individuo disciplinado y motiva para coordinar proyectos. Técnicamente, necesita suficiente experiencia con las computadoras para entender sus capacidades, poder programar, deducir los requerimientos de información de los usuarios y comunicar eficazmente esas necesidades a los programadores.
3	¿Por qué es importante la comunicación entre el analista y los usuarios finales?	Porque debe tener una comprensión precisa de los requerimientos, ya que su función principal es traducir las necesidades del negocio en especificaciones técnicas, es decir si no hay comunicación clara y constante, se dificulta capturar correctamente las necesidades de los usuarios. Los usuarios finales son los expertos en sus propios procesos, entonces el analista debe escucharlos activamente para entender no solo lo que dicen que quieren, sino lo que realmente necesitan para ser más eficientes.
4	¿Cómo contribuye el analista al proceso de toma de decisiones empresariales?	El analista de sistemas contribuye al proceso de toma de decisiones empresariales mediante la recopilación y análisis de información sobre los usuarios, la capacidad de uso y la eficiencia de los sistemas actuales. Presenta propuestas que incluyen alternativas técnicas y su respectivo análisis costo-beneficio. Además, puede emitir recomendaciones fundamentadas. Si la administración aprueba alguna de estas propuestas, el analista continúa desarrollando el proyecto en esa dirección, asegurando que las soluciones estén alineadas con los objetivos del negocio.
5	Explique con un ejemplo práctico cómo un analista identifica y soluciona un problema en un sistema existente.	<ul style="list-style-type: none"> • Escenario: Errores en la Nómina de una Empresa. • Sistema Existente: Un proceso semi-manual donde los supervisores de planta registran las horas extras en hojas de Excel, las envían por correo a RRHH, y un asistente de RRHH las re-digita en un sistema de nómina antiguo. • Problema: Los pagos de nómina son frecuentemente incorrectos, generando quejas de los empleados y trabajo extra para RRHH <p>Pasos:</p> <ol style="list-style-type: none"> 1. Identificación de los problemas, oportunidades y objetivos.



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

		<p>El analista es llamado por la administración de RRHH debido a las constantes quejas.</p> <ul style="list-style-type: none"> • Problema: El analista entrevista a la administración ("entrevistar a los encargados") y confirma que el problema es la alta tasa de errores en el pago de horas extras. • Oportunidad: Identifica que la doble digitación manual (Supervisor -> Excel, RRHH -> Nómina) es la fuente del error y puede mejorarse con un sistema de información ("oportunidades... mejorar mediante el uso de sistemas de información"). • Objetivo: Descubre que el objetivo de la empresa ("qué trata de hacer la empresa") es reducir los errores de nómina en un 95% y procesar la nómina dos días más rápido. • Resultado: El analista entrega un informe de viabilidad ("resultado de esta fase") que define el problema y los objetivos. La administración aprueba el proyecto. <p>2. Determinación de los requerimientos de información</p> <p>El analista busca entender las necesidades del "factor humano".</p> <ul style="list-style-type: none"> • Métodos: Realiza entrevistas ("métodos interactivos como entrevistas") con los supervisores de planta (el "quién") para entender cómo registran las horas (el "qué" y "cómo") al final de sus turnos (el "cuándo"). • HCI: Determina que los supervisores necesitan un sistema que sea fácil de usar en un entorno ruidoso ("fácil de usar, aprender"), legible en una tablet ("perceptible, legible") y que requiera pocos clics ("apoyar las tareas laborales individuales"). • Resultado: El analista comprende que los supervisores necesitan una forma de aprobar horas digitalmente y que RRHH necesita un reporte automático, no una hoja de Excel. <p>3. Análisis de las necesidades del sistema</p> <p>El analista traduce los requerimientos en un plan técnico.</p> <ul style="list-style-type: none"> • Herramientas: Crea un Diagrama de Flujo de Datos (DFD) ("herramientas como los diagramas de flujo de datos") que muestra el proceso actual (Supervisor -> Excel -> Email -> RRHH -> Sistema) identificando los cuellos de botella.
--	--	--



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

		<ul style="list-style-type: none"> • Análisis: Prepara una propuesta de sistemas ("prepara una propuesta de sistemas") que incluye un análisis de costo-beneficio ("análisis de costo-beneficio de las alternativas") de dos opciones: <ol style="list-style-type: none"> 1. Desarrollar una app interna. 2. Comprar un software de control de asistencia e integrarlo. • Decisión: La administración acepta la recomendación de comprar e integrar el software existente. <p>4. Diseño del sistema recomendado</p> <p>El analista usa la información recolectada para diseñar la solución.</p> <ul style="list-style-type: none"> • Diseño Lógico: Diseña cómo se conectará el nuevo software de asistencia con el sistema de nómina antiguo. • Diseño de HCI: Configura la interfaz gráfica de usuario (GUI) ("interfaces gráficas de usuario GUI") del nuevo software para que los supervisores solo vean los botones de "Aprobar" y "Rechazar" en sus tablets. • Diseño de Salida: Diseña un nuevo reporte automático ("diseñar una salida (ya sea en pantalla o impresa)") que el sistema de nómina leerá. • Resultado: Entrega los paquetes de especificación de programas ("paquetes de especificación") a los programadores para que construyan el <i>punte</i> (la integración) entre los dos sistemas. <p>5. Desarrollo y documentación del software</p> <p>El analista trabaja con los programadores para construir el puente.</p> <ul style="list-style-type: none"> • Desarrollo: Los programadores codifican la integración. • Documentación: El analista desarrolla la documentación ("desarrolla... una documentación efectiva"): <ul style="list-style-type: none"> ○ Un manual de procedimientos ("manuales de procedimientos") de 2 páginas para los supervisores. ○ Un sitio interno con preguntas frecuentes (FAQ) ("sitios Web con preguntas frecuentes") para RRHH sobre cómo leer el nuevo reporte. <p>6. Prueba y mantenimiento del sistema</p>
--	--	---



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

		<p>El analista y los programadores prueban la solución antes de lanzarla.</p> <ul style="list-style-type: none">• Pruebas: Primero, usan datos de muestra ("datos de muestra") (Ej. Supervisor A aprueba 5 horas extra).• Pruebas Reales: Luego, usan los datos reales ("datos reales del sistema actual") de la semana anterior para procesar una nómina paralela y asegurarse de que los cálculos coincidan.• Detección: Descubren un error en el cálculo de horas nocturnas (es "menos costoso detectar los problemas antes"). Los programadores lo corrigen. <p>7. Implementación y evaluación del sistema</p> <p>El analista lanza el nuevo sistema.</p> <ul style="list-style-type: none">• Capacitación: El analista capacita a los usuarios ("capacitar a los usuarios") (supervisores y personal de RRHH) sobre cómo usar el nuevo sistema.• Conversión: Se planifica la conversión ("planear una conversión"): el viernes se usa el Excel por última vez, y el lunes siguiente se enciende el nuevo sistema.• Instalación: Ayuda a instalar el equipo ("instalar equipo") (las tablets para los supervisores) y el sistema entra en producción.• Evaluación: Dos semanas después, el analista evalúa y confirma que los errores de nómina se han reducido en un 98%.
--	--	---

Tabla de Preguntas U1T2

N°	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) Especificar detalladamente el sistema
2	¿Qué actividad forma parte del análisis?	A) Compilar el código B) Definir casos de uso C) Instalar el software	B) Definir casos de uso

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) Las interacciones entre usuarios y procesos
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) Conectar los requisitos con el diseño y pruebas
5	¿Qué metodología es recomendada para el análisis en administración pública?	A) SCRUM B) Métrica Versión 3 C) CMMI	C) CMMI

Tabla de Preguntas U1T3

N°	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) El conjunto de actividades, métodos y prácticas
2	¿Cuál es la primera actividad del proceso de desarrollo?	A) Especificación del software B) Validación C) Evolución	A) Especificación del software
3	¿Qué caracteriza al modelo en cascada?	A) Desarrollo iterativo B) Etapas secuenciales C) Entregas parciales	B) Etapas secuenciales
4	¿Qué principio rige el Proceso Unificado?	A) Programación estructurada B) Dirigido por casos de uso C) Desarrollo sin documentación	B) Dirigido por casos de uso

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) Facilita el control de calidad
---	---	---	-----------------------------------

Tabla de Preguntas U1T4

N°	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. Identificar la funcionalidad desde el punto de vista del usuario.
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.	A. Un componente interno del sistema.
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. Código fuente
4	En el caso de uso 'Registrar artículo', ¿cuál es una precondición válida?	A. El usuario debe estar registrado como vendedor. B. El sistema debe estar apagado. C. El artículo debe estar adjudicado.	A. El usuario debe estar registrado como vendedor.
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.	A. El primero describe la forma de usar el sistema; el segundo, la información que contiene.

		<p>B. Ambos representan el mismo tipo de relaciones.</p> <p>C. Ninguno se usa en análisis.</p>	
--	--	--	--

5. CONCLUSIONES

- El analista de sistemas cumple un papel estratégico en las organizaciones, al fungir como enlace entre los requerimientos del negocio y las soluciones tecnológicas. Su labor no se limita a la interpretación técnica, sino que implica comprender los procesos organizacionales, identificar oportunidades de mejora y garantizar que las soluciones informáticas aporten valor real y sostenible a la empresa.
- El análisis de sistemas de información se consolida como una etapa fundamental dentro del ciclo de vida del desarrollo de software, pues permite definir con precisión los requisitos funcionales y no funcionales antes de iniciar el diseño o la implementación. Un análisis exhaustivo previene errores costosos y asegura que el sistema final responda a las necesidades reales de los usuarios.
- El uso de modelos de casos de uso, propuestos por Ivar Jacobson, facilita la comunicación entre los analistas, diseñadores y usuarios finales, al representar de manera gráfica y textual cómo los actores interactúan con el sistema. Esta herramienta promueve la trazabilidad entre los requisitos y los entregables, y fortalece la validación temprana de las funcionalidades.
- Se confirma que un proceso de desarrollo estructurado, como el Proceso Unificado o el modelo iterativo-incremental, favorece la planificación, el control de calidad y la colaboración entre los equipos. A través de sus fases bien definidas (inicio, elaboración, construcción y transición), se garantiza la entrega progresiva de valor y la adaptación a los cambios del entorno.
- En síntesis, el éxito de un proyecto de software depende tanto del dominio técnico como de la capacidad analítica, comunicativa y ética del equipo. La comprensión profunda del negocio, la correcta definición de requerimientos y la validación continua son pilares para alcanzar productos de software confiables, útiles y alineados con los objetivos institucionales.

6. RECOMENDACIONES

- Involucrar activamente a los usuarios finales en todas las fases del proceso de desarrollo, desde la recolección de requisitos hasta la validación del sistema. Esto

garantiza una comprensión más realista de las necesidades y facilita la aceptación del producto final.

- Aplicar metodologías estructuradas de análisis, como Métrica Versión 3 o enfoques basados en casos de uso, para asegurar la trazabilidad y coherencia entre los requisitos, el diseño y las pruebas. Estas metodologías ayudan a documentar de forma sistemática cada decisión tomada durante el proyecto.
- Fomentar la formación integral del analista de sistemas, fortaleciendo tanto sus habilidades técnicas (bases de datos, modelado, herramientas CASE) como sus competencias blandas (comunicación, liderazgo, resolución de conflictos). Este equilibrio mejora la gestión de equipos interdisciplinarios y la comprensión del dominio del negocio.
- Priorizar la fase de análisis antes de la implementación. Saltar prematuramente a la programación puede derivar en errores de diseño, sobrecostos y retrabajos. Una buena práctica es validar los requerimientos mediante prototipos o revisiones iterativas antes de iniciar la construcción del sistema.
- Promover la mejora continua del proceso de desarrollo de software (PDS), evaluando regularmente sus resultados, herramientas y metodologías. Adoptar enfoques ágiles o híbridos, cuando el contexto lo permita, puede optimizar la comunicación y la adaptabilidad a los cambios del entorno organizacional.

7. REFERENCIAS

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

Pressman, R. S., & Maxim, B. R. (2015). Ingeniería del software: Un enfoque práctico (8.ª ed.). McGraw-Hill Education

Dennis, A., Wixom, B. H., & Roth, R. M. (2021). Systems Analysis and Design (8.ª ed.). Wiley.

Whitten, J. L., Bentley, L. D., & Dittman, K. C. (2007). Systems Analysis and Design Methods (7.ª ed.). McGraw-Hill.

Larman, C. (2005). Aplicando UML y patrones: Introducción al análisis y diseño orientado a objetos y al desarrollo iterativo (3.ª ed.). Pearson Educación.