

Análisis de Sistemas

Materia:
Ingeniería de
Requerimientos

Docente contenidista: PEPE, Jonathan Leonel

Revisión: Coordinación



Contenido

Concepto de Sistema	4
Sistema de información.....	5
Análisis de Sistema	5
Proceso de desarrollo de software.....	6
Modelo de ciclo de vida	6
Modelo en cascada	7
Requerimiento	8
Analista de requerimiento.....	9
Participantes.....	10
Stakeholder	10
Clases de usuarios.....	12
Necesidad del usuario	14
Ingeniería de requerimientos	15
Bibliografía	17

Clase 1



iTe damos la bienvenida a la materia
Ingeniería de Requerimientos!

En esta clase vamos a ver los siguientes temas:

- Definición y componentes de un Sistema.
- Características de un Sistema de información.
- Las 6 actividades básicas en el desarrollo de software.
- Actividades de un Analista de requerimientos.
- Clases de usuarios y sus necesidades.

¿Listos para empezar?

Hoy conoceremos los fundamentos de un Sistema, como así también el proceso y las actividades en el desarrollo de software.

Exploraremos el rol del Analista de requerimientos y clasificaremos las necesidades del usuario.

¡Seguimos adelante!

Concepto de Sistema

Existen múltiples definiciones de **Sistema**, pero todas ellas coinciden en lo siguiente:

"Un Sistema es una colección de componentes relacionados entre sí que trabajan conjuntamente para cumplir con un objetivo"

A su vez, un **Subsistema** es lo que en la definición anterior se especifica como *componente* de ese sistema.

La siguiente imagen pone lo anterior de manifiesto:

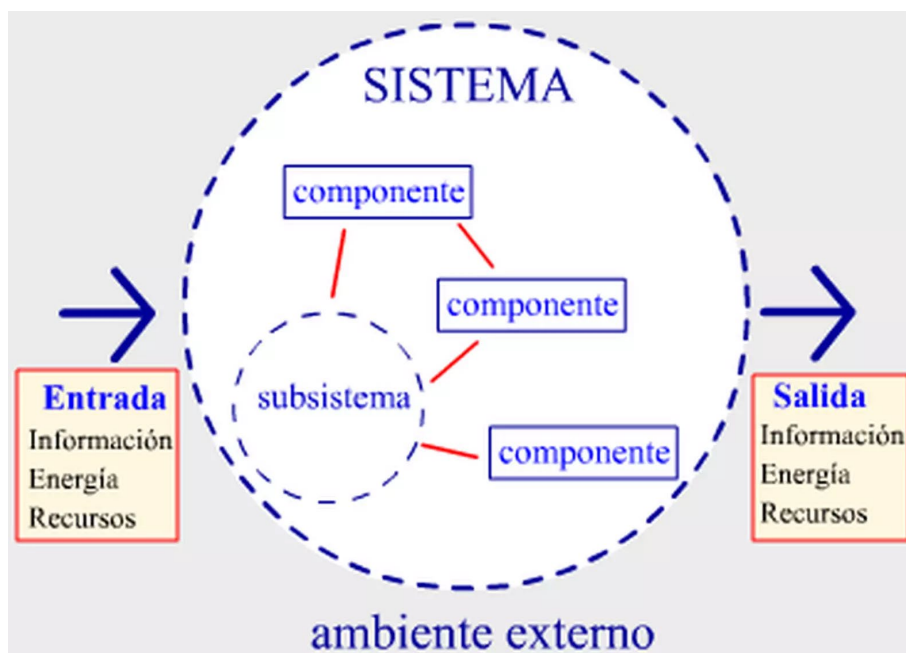


Imagen 1. Diagrama de un Sistema.

Sistema de información

Para construir una definición de Sistema de Información, resulta necesario definir en primera instancia qué consideramos por información.

En este sentido diremos que la **información** es un dato o un conjunto de datos que, en un contexto determinado, tienen un significado para alguien y transmiten un mensaje útil.

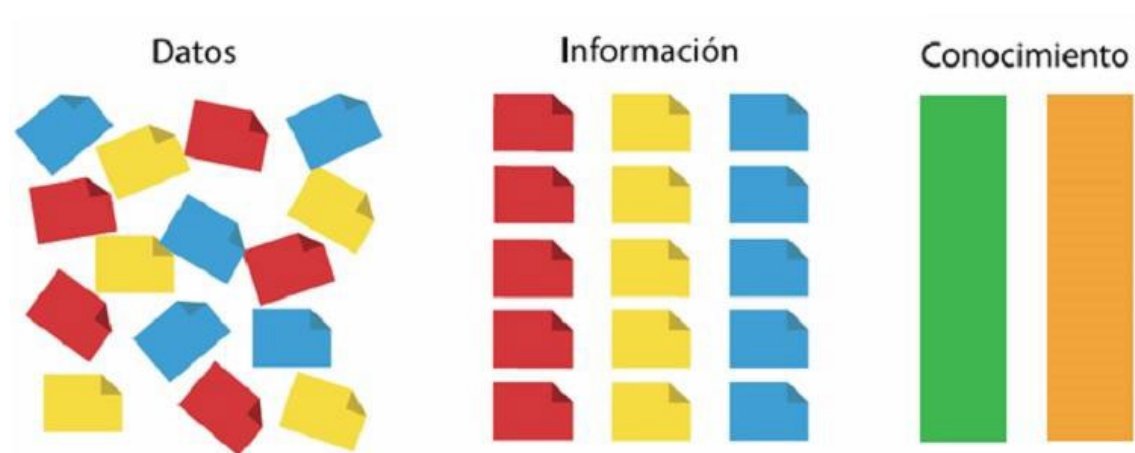


Imagen 2. Datos, información y conocimiento.

Por lo tanto, se entiende por **Sistema de Información** al conjunto de Subsistemas para recolectar, almacenar, procesar y distribuir información para la planificación, decisión y señalamiento de un sistema objeto (e.g. organización) del cual forma parte.

Análisis de Sistema

Ahora nos preguntamos, ¿qué significa **analizar un Sistema**?

Es el proceso que sirve para recopilar e interpretar los hechos, diagnosticar problemas y utilizar dichos hechos a fin de mejorar el Sistema.



El análisis especifica qué es lo que el Sistema debe hacer y cómo alcanzar los objetivos. Su responsabilidad radica en conducir el estudio del Sistema para conocer los hechos importantes en relación con la actividad del negocio.

Proceso de desarrollo de software

Cuando se trabaja en la construcción de un producto o Sistema es importante ejecutar una serie de pasos predecibles que ayuda a obtener a tiempo un resultado de alta calidad.

El mapa que se sigue se llama **proceso de desarrollo de software**, entendiendo al "proceso" como un *diálogo* en el que el conocimiento que debe convertirse en software se reúne e incorpora en éste, generando una interacción constante entre usuarios, diseñadores y tecnología.

Modelo de ciclo de vida

Un **ciclo de vida** describe las actividades que se realizan en cada etapa de un proyecto de desarrollo de software y cómo las actividades se relacionan entre sí de forma lógica y cronológica.

Existen varios modelos, con sus ventajas e inconvenientes:

- Modelo en cascada
- Modelo en V
- Modelo iterativo
- Modelo incremental
- Modelo en espiral
- Modelo de prototipos



A continuación, desarrollaremos el Modelo en cascada, pero te propongo investigar sobre los demás para debatir la próxima clase.

Modelo en cascada



El desarrollo en cascada (***waterfall model***) es un procedimiento secuencial y lineal que divide al desarrollo de software en sucesivas fases que se suceden entre sí y se ejecutan una sola vez.

De allí su analogía con una cascada de agua.

Las fases en las que se divide el proyecto son, básicamente, las siguientes:

- **Planificación:** La planificación tiene por objetivo proporcionar un marco de trabajo que permita hacer estimaciones razonables de recursos, costos y planificación temporal.
Esta actividad incluye la recopilación de requerimientos.
- **Implementación:** La implementación es la parte del proceso en la cual el equipo de desarrollo modela, diseña y programa el código del proyecto.
- **Pruebas:** Las pruebas de software son parte esencial del proceso de desarrollo del software y tiene la función de detectar los errores de software lo antes posible.
- **Documentación:** La documentación del diseño interno del software tiene por objetivo facilitar su mejora y su mantenimiento a lo largo del proyecto.
- **Despliegue:** El despliegue comienza cuando el código ha sido suficientemente probado, aprobado para su liberación y distribuido en el entorno de producción.
- **Mantenimiento:** El mantenimiento es la modificación de un producto de software después de la entrega, para corregir errores no detectados oportunamente, mejorar su rendimiento u otros atributos.

Sin embargo, la utilización del modelo en la práctica es controvertida. Por una parte, en el desarrollo de software las fases de proyecto no suelen estar claramente diferenciadas entre sí. Por otra parte, la secuencia lineal del *waterfall model* no suele coincidir con la realidad. Es por eso, que más adelante, estudiaremos modelos iterativos e incrementales que se utilizan en la actualidad.

Requerimiento

Si bien abordaremos este concepto en profundidad en la próxima clase, nada impide que podamos realizar una primera aproximación diciendo que un **requerimiento** es una necesidad o solicitud cuyo objetivo es resolver un problema.

Dicho de otro modo, es una característica que un sistema debe tener para cubrir alguna de las necesidades que lo motivan.

La definición más formal la podemos tomar de la IEEE (*Institute of Electrical and Electronics Engineers*):

"Un requerimiento es una condición o capacidad que debe satisfacer o poseer un Sistema o un componente de un Sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto."

Analista de requerimiento

Es la persona (rol) responsable de:

- Definir los objetivos del proyecto y los beneficios al negocio.
- Identificar el problema a resolver y obtener los requerimientos.
- Identificar a los **involucrados** en el desarrollo del proyecto, así como a **las clases de clientes y usuarios**.
- Identificar el ambiente del dominio a desarrollar y estar preparado para desarrollar el sistema requerido.
- Administrar los requerimientos utilizando un proceso y un plan de requerimientos.
- Modelar los requerimientos.
- Realizar control de cambios en los requerimientos.

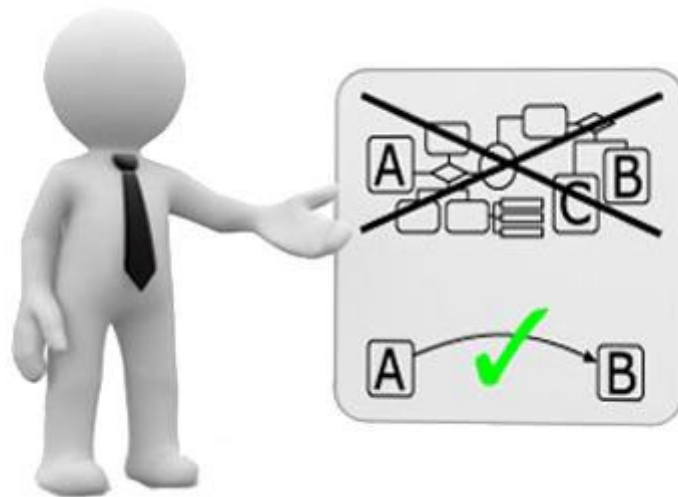


Imagen 3. Analista de requerimiento.

Resaltamos en negrita algunos conceptos clave que necesitamos definir y añadir a nuestra jerga profesional, veamos.

Participantes

Un **participante** es cualquier persona que se beneficie en forma directa o indirecta del sistema en desarrollo.

La pregunta que se debe responder para conocer a los participantes es la siguiente:

¿Quiénes tienen que ver con la solución del problema?

Dicho de otro modo, un participante es cualquier persona que tenga algo que ver en el resultado exitoso del proyecto.

Stakeholder

Stakeholder es una palabra del inglés que, en el ámbito empresarial, significa interesado o parte interesada.

Se refiere a todas aquellas personas u organizaciones afectadas por las actividades y las decisiones de una empresa.

En el siguiente diagrama se muestra quiénes pueden ser los Stakeholder internos y externos de una empresa.



Imagen 4. Stakeholder.

Dentro de las actividades y responsabilidades de los interesados se pueden mencionar:

- Educar al **Analista de requerimientos** acerca del negocio y sus objetivos.
- Ser claro y preciso acerca del problema que se quiere resolver.
- Colaborar con el Analista en la definición de los requerimientos.
- Revisar los documentos de requerimientos y el avance del proyecto.
- Comunicar a los Analistas sobre cambios en los requerimientos.
- Plantear costos y tiempos esperados de desarrollo y estar abierto a discutir cambios en los costos y tiempos de entrega.
- Estar dispuesto a reunirse con los desarrolladores para discutir distintos aspectos del proyecto.

Clases de usuarios

Los **usuarios** son aquellas personas que realizarán sus tareas usando directa o indirectamente el producto de software.

Pueden ser usuarios directos, indirectos o administradores:

- Los **usuarios directos** interactúan con el sistema, lo alimentan con datos.
- Los **usuarios indirectos** reciben las salidas/reportes del sistema, pero no interactúan personalmente.
- Los **administradores** supervisan la intervención en el desarrollo o uso del sistema (*ABM de usuarios, niveles de acceso, aplicar actualizaciones*).

En este sentido, el Analista de requerimientos debe identificar las diferentes clases de usuarios, seleccionar un representante de usuarios para cada clase de usuarios y trabajar con individuos representativos de su clase.

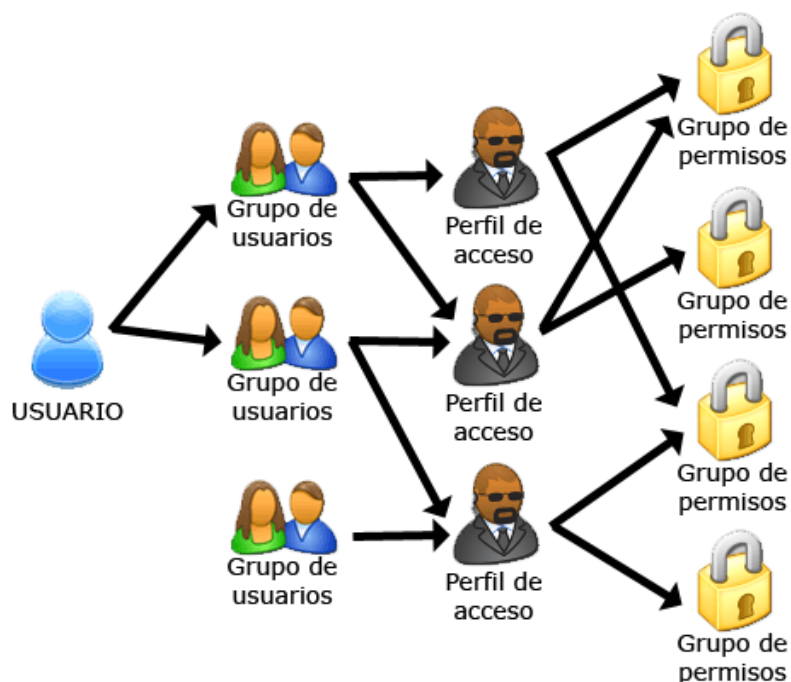


Imagen 5. Clases de usuarios.

Para clasificar a los usuarios es una buena práctica tener en cuenta:

- Las tareas que desempeñan en soporte de los procesos de la organización.
- La frecuencia con la que usan el Sistema.
- Las funciones que usan del Sistema.
- La experiencia en el dominio de la aplicación.
- Sus privilegios de acceso o niveles de seguridad.
- El tipo de uso que le dan al Sistema (*operación, administración, mantenimiento, supervisión*).

Necesidad del usuario

Una **necesidad** consiste en aquello que motiva a un cliente a comprar un producto de software.

Puede ser conocida (*el cliente puede expresarla con palabras*) o desconocida (*debemos averiguar cuál es*).



Los usuarios generalmente transmiten sus necesidades de manera general, buscando que el Analista de requerimientos sea un apoyo para desglosar la necesidad a su mínima expresión con el fin de identificar el conjunto de funcionalidades que se deben construir.

Existen diferentes técnicas para entender las necesidades del usuario que ahora simplemente mencionaremos, pero estudiaremos en detalle las próximas clases:



Imagen 6. Técnicas para entender las necesidades del usuario.

Con estos nuevos conceptos podemos concluir que los requisitos son condiciones o capacidades que debe tener un sistema (**requerimientos del sistema**) para satisfacer las **necesidades del usuario**.

Es el Analista quien debe transformar las necesidades del usuario en requerimientos del sistema, por lo tanto, debe entender la necesidad antes de pensar qué tecnología usar.

Ingeniería de requerimientos

Finalizaremos esta primera clase con una primera definición, que luego ampliaremos y profundizaremos, del nombre que lleva esta asignatura:

"La Ingeniería de requerimientos es un forma disciplinada y sistemática de llegar desde las necesidades de los usuarios a la especificación del Sistema."

De este modo, el objetivo principal de la Ingeniería de requerimientos es generar las especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento que debe tener el Sistema.

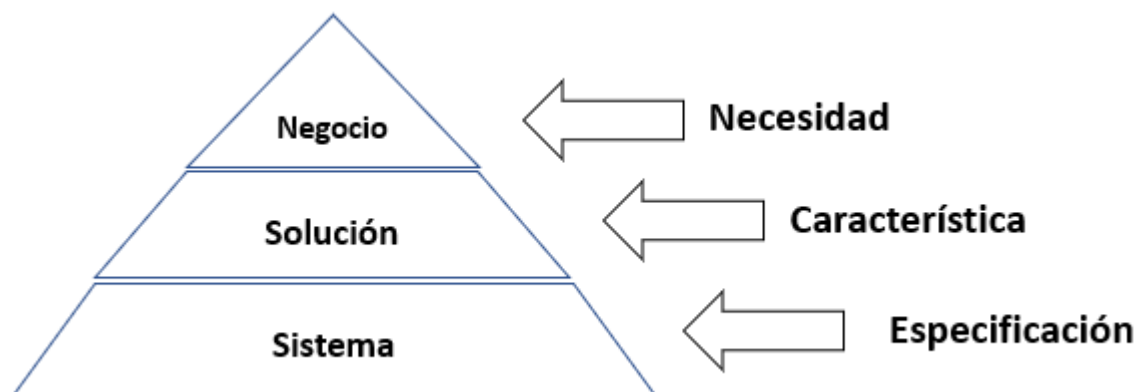
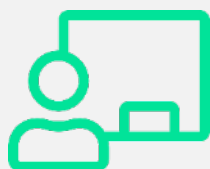


Imagen 7. Especificaciones del sistema.



Hemos llegado así al final de esta clase en la que vimos:

- Definición y componentes de un Sistema.
- Características de un Sistema de información.
- Las 6 actividades básicas en el desarrollo de software.
- Actividades de un Analista de requerimientos.
- Clases de usuarios y sus necesidades.



Te esperamos en la **clase en vivo** de esta semana.
No olvides realizar el **desafío semanal**.

¡Hasta la próxima clase!

Bibliografía

Del Águila Cano, I. M. (2019). Ingeniería de requisitos: Material didáctico. Cuaderno de teoría (Vol. 35). Universidad Almería.

Ramos, D., Noriega, R., Laínez, J. R., & Durango, A. (2017). Curso de Ingeniería de Software: 2ª Edición. IT Campus Academy.

Vazquez, C. E., & Simoes, G. S. (2016). Ingeniería de Requisitos: Software orientado al negocio. Brasport.