Proyecto Certificador de Desarrollo de SW 1

Tema Nº2:

Sistemas de Información (SI)

Introducción al Proyecto Certificador de Desarrollo de Software 1.

Indicador de logro Nº2:Identifica los sistemas de información en las organizaciones y las fases de desarrollo del software en proyectos informáticos.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº2:**

SISTEMAS DE INFORMACIÓN (SI)

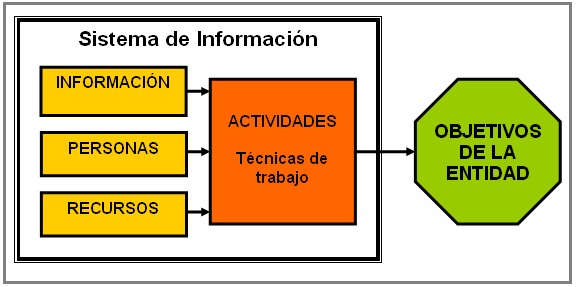
**OBJETO DE LA EXPERIENCIA**

Conoce de sistemas de información, su composición, clasificación, tipos de SI, principio de desarrollo y el ciclo de vida de un sistema de información, a su vez identifica a ingeniería de software, como una diciplina formada por un conjunto de métodos y técnicas que se utilizan en el desarrollo de software. Finalmente utiliza herramientas case adecuadas para la gestión del proyecto de software.

**MARCO TEÓRICO**

* 1. Sistemas de Información – SI

Un sistema de información es un conjunto de datos que interactúan entre sí con un fin común. En informática, los sistemas de información ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización.



* 1. Componentes de un SI

Un sistema de información debe cumplir con los siguientes componentes básicos interactuando entre sí:

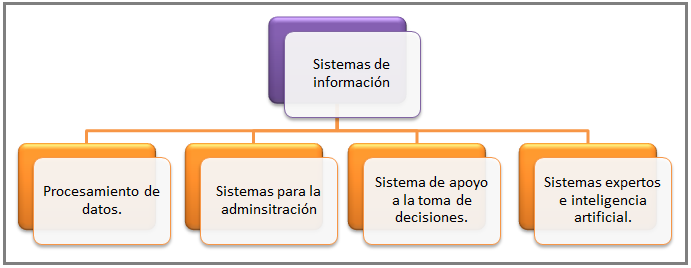
* El [hardware](https://es.wikipedia.org/wiki/Hardware), equipo físico utilizado para procesar y almacenar datos,
* El [software](https://es.wikipedia.org/wiki/Software) y los procedimientos utilizados para transformar y extraer información,
* Los [datos](https://es.wikipedia.org/wiki/Dato) que representan las actividades de la empresa,
* La red que permite compartir recursos entre [computadoras](https://es.wikipedia.org/wiki/Computadora) y dispositivos,
* Las personas que desarrollan, mantienen y utilizan el sistema.

Los sistemas de información son una combinación de tres partes principales: las personas, los procesos del negocio y los equipos de tecnologías de la información

****

* 1. Clasificación de los SI

Los sistemas de información, de manera general se pueden clasificar:



* 1. Tipos de Sistema de Información

Tendremos en cuenta los siguientes tipos:

* ERP**.** Los sistemas [Enterprise Resource Planning](https://rockcontent.com/es/blog/erp-para-alcanzar-ventas/) (o Planeamiento de Recursos de la Empresa) son softwares que integran diferentes procesos y datos de la empresa, reuniéndolos en un solo lugar.

De esta manera, los datos de todos los departamentos de la organización son integrados y almacenados.

Los datos brindados por los softwares ERP ayudan a traer más agilidad a los procesos y permiten cumplir la producción por demanda.

**Ejemplo**

De funcionamiento de software ERP es en el momento de la venta de una mercancía.

Mientras la venta es realizada, los departamentos de producción y de compras son automáticamente alertados.

Así, es posible verificar si hay todos los productos o si será necesario adquirir algo. Además, es posible identificar la necesidad de reponer los estoques.

* CRM Los softwares [Customer Relationship Management](https://rockcontent.com/es/blog/crm-en-ventas/" \t "_blank) (o Gestión de Relación con el Cliente) automatizan todas las funciones relativas al contacto con los clientes, permitiendo que las organizaciones recolecten y almacenen los datos de contacto, las preferencias de los clientes, el histórico de compras de ellos, entre otros.

Así, la empresa puede contactar los [clientes](https://rockcontent.com/es/blog/tipos-de-cliente/) para estrategias específicas, con el objetivo principal de atender a las necesidades de los consumidores de manera anticipada.

* SCM Ya los sistemas Supply Chain Management (o Administración de la Cadena de Suministro) integran los diferentes procesos relativos a los proveedores de servicios, productos e informaciones.

La finalidad es crear valor para el [consumidor](https://rockcontent.com/es/blog/comportamiento-del-consumidor/), satisfaciéndolo cuando él adquiere un producto o servicio.

Así, ese tipo de software integra los datos relativos a fabricantes, proveedores y puntos de venta, garantizando que los productos sean entregues en las cantidades necesarias y en el plazo correcto, evitando la falta de mercancía o el exceso de stock.

Así, se alcanza un buen nivel de servicio al mismo tiempo que los gastos son reducidos.

Es importante resaltar que este software es compuesto por los sistemas de gestión de suministros y componentes, de la [cadena de suministros](https://rockcontent.com/es/blog/cadena-de-suministro/), de la estructura de producto, del rastreo de origen y uso y de control de la cadena de suministros.

De esta manera, se consigue hacer desde la previsión de ventas, inventario y clasificación de productos hasta [reducir el costo](https://rockcontent.com/es/blog/reducir-costos-para-saas/) de manipulación y creación de piezas.

* SIGLos Sistemas de Información Gerenciales son dirigidos hacia el apoyo a la toma de decisiones y actúan en los niveles estratégico, operacional y táctico.

Las informaciones pueden ser reportadas por medio de gráficos, hojas de cálculo o, los habituales informes.

* 1. Principios en el desarrollo de un SI
* Implicar a los Usuarios del Sistema.
* Utilizar una estrategia de resolución problemas.
* Establecer fases y actividades.
* Documentar durante desarrollo de sistema.
* Establecer estándares.
* Gestionar los procesos y el proceso.
* Justificar el sistema como una inversión de capital.
* No tener miedo de revisar o cancelar algún objetivo.
* Dividir los problemas y resolver uno a uno.
* Diseñar Sistemascon previsión de crecimiento y cambio.
  1. El ciclo de vida de un SI

Comprende lo siguiente:

Identificación del problema

En la primera fase del ciclo de vida del desarrollo de sistemas el analista tiene que ver con la identificación de problemas, oportunidades y objetivos. La primera fase requiere que el analista observe lo que está sucediendo en el negocio. Luego, junto con los demás miembros de la organización, el analista hace resaltar los problemas. Las personas involucradas en la primera fase son los usuarios, analistas y administradores de sistemas que coordinan el proyecto. Las actividades de esta fase consisten en entrevistas a los administradores, los usuarios, estimación del alcance del proyecto y documentación de los resultados.

Determinación de los requerimientos de información

Entre las herramientas utilizadas para definir los requerimientos de información en el negocio se encuentran: muestreo e investigación de los datos relevantes, entrevistas, cuestionarios, el comportamiento de los tomadores de decisiones y su ambiente de oficina y hasta la elaboración de prototipos. En esta fase el analista está esforzándose por comprender qué información necesitan los usuarios para realizar su trabajo.

Análisis de las necesidades del sistema

Nuevamente, herramientas y técnicas especiales ayudan para que el analista haga las determinaciones de los requerimientos. Una herramienta de éstas es el uso de diagramas de flujo de datos para diagramar la entrada, proceso y salida de las funciones del negocio en forma gráfica estructurado.

Diseño del sistema recomendado

En esta fase del ciclo de vida del desarrollo de sistemas, el analista usa la información recolectada anteriormente para realizar el diseño lógico del sistema de información. El analista diseña procedimientos precisos para la captura de datos, a fin de que los datos que van a entrar al sistema de información sean correctos. Además, el analista también proporciona entrada efectiva para el sistema de información mediante el uso de técnicas para el buen diseño de formas y pantallas.

Desarrollo y documentación del software

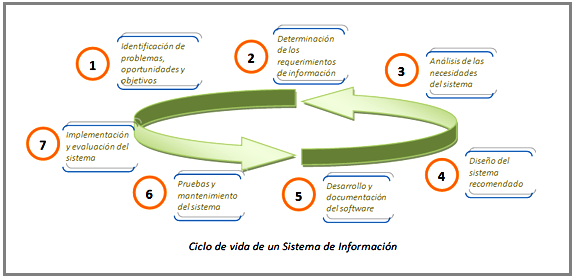
En la quinta fase del ciclo de vida del desarrollo de sistemas el analista trabaja con los programadores para desarrollar cualquier software original que se necesite. Durante esta fase, el analista también trabaja con los usuarios para desarrollar documentación efectiva para el software, incluyendo manuales de procedimientos. La documentación le dice al usuario la manera de usar el software y también qué hacer si se suceden problemas con el software.

Pruebas y mantenimiento del sistema

Antes de que pueda ser usado, el sistema de información debe ser probado. Es mucho menos costoso encontrar problemas antes de que el sistema sea entregado a los usuarios. Algunas de las pruebas son realizadas por los programadores solos, y otras por los analistas de sistemas junto con los programadores. El mantenimiento del sistema y de su documentación comienza en esta fase y es efectuado rutinariamente a lo largo de la vida del sistema de información.

Implementación y evaluación del sistema

En esta fase del desarrollo del sistema el analista ayuda a implementar el sistema de información. Esto incluye el entrenamiento de los usuarios para que manejen el sistema. Algún entrenamiento es hecho por los proveedores, pero la supervisión del entrenamiento es responsabilidad del analista de sistemas. La evaluación se muestra como parte de esta fase final de ciclo de vida del desarrollo del sistema, aunque se realiza durante cada fase.

****

* 1. Ingeniería de Software

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (software).

Esta disciplina trasciende la actividad de programación, que es la actividad principal a la hora de crear un software. El ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto.

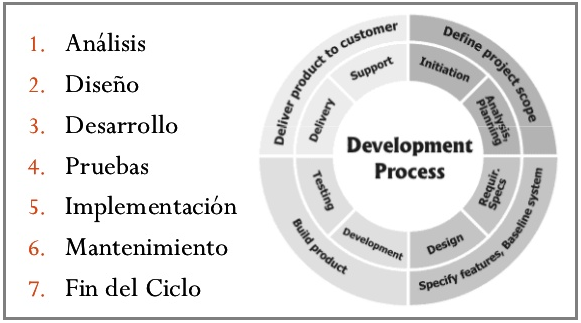
La ingeniería de software, por lo tanto, incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema.

Cabe destacar que el proceso de desarrollo de software implica lo que se conoce como ciclo de vida del software, que está formado por cuatro fases iniciales: concepción, elaboración, construcción y transición.

La concepción fija el alcance del proyecto y desarrolla el modelo de negocio; la elaboración define el plan del proyecto, detalla las características y fundamenta la arquitectura; la construcción es el desarrollo del producto; y la transición es la transferencia del producto terminado a los usuarios.

Una vez que se completa este ciclo, entra en juego el mantenimiento del software. Se trata de una fase de esta ingeniería donde se solucionan los errores descubiertos (muchas veces advertidos por los propios usuarios) y se incorporan actualizaciones para hacer frente a los nuevos requisitos. El proceso de mantenimiento incorpora además nuevos desarrollos, para permitir que el software pueda cumplir con una mayor cantidad de tareas.

* 1. Etapas de desarrollo de software
* **Etapa de análisis:** Es el proceso de investigar un problema que se quiere resolver. Definir claramente el Problema que se desea resolver o el sistema que se desea crear. Identificar los componentes principales que integrarán el producto.
* **Etapa de Diseño:** Es el proceso de utilizar la información recolectada en la etapa de análisis al diseño del producto. La principal tarea de la etapa de diseño es desarrollar un modelo o las especificaciones para el producto o Componentes del Sistema.
* **Etapa de Desarrollo:** Consiste en utilizar los modelos creados durante la etapa de diseño para crear los componentes del sistema.
* **Etapa de Pruebas o Verificación Prueba:** Consiste en asegurar que los componentes individuales que integran al sistema o producto, cumplen con los requerimientos de la especificación creada durante la etapa de diseño.
* **Etapa de Implementación o Entrega Implantación:** Consiste en poner a disposición del cliente el producto.
* **Etapa de Mantenimiento**: Consiste en corregir problemas del producto y re- liberar el producto como una nueva versión o revisión (producto mejorado).
* **Etapa final** EOL (End-of-Life) El fin del ciclo del producto consiste en realizar todas las tareas necesarias para asegurar que los clientes y los empleados están conscientes de que el producto ya no será vendido ni soportado.



* 1. Pruebas de Software

Las pruebas de software (testing) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada o stakeholder

Los probadores de software (testers) planifican y llevan a cabo pruebas de software de los ordenadores para comprobar si funcionan correctamente.

* 1. Tipos de Pruebas
* **Pruebas unitarias**

Objetivo: comprobar que un módulo de código funciona correctamente

* **Pruebas funcionales**

Objetivo: comprobar que el software desarrollado realiza de manera funcionalmente correcta aquello para lo que fue desarrollado

* **Pruebas integración**

Objetivo: comprobar que los módulos que componen el código desarrollado funcionan correctamente una vez que estos están integrados entre si

* **Pruebas de validación o aceptación**

Objetivo: comprobar que el software desarrollado cumple con las expectativas del cliente, tanto desde el punto de vista de la funcionalidad como de la satisfacción del cliente

* **Pruebas de caja blanca**

Objetivo: probar el funcionamiento de la estructura de control definida en la lógica (o configuración) del programa. Se ejecutan por lo menos una vez todos los flujos del programa.

* **Pruebas de caja negra**

Objetivo: comprobar que la funcionalidad del programa o del sistema está operativa.

Permite identificar:

* + Funciones incorrectas o ausentes
  + Errores de interface
  + Errores de estructura de datos o acceso a BD externas
  + Errores de rendimiento
  + Errores de inicialización o de terminación
  1. Herramientas CASE

Es un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software.

CASE se define también como: Conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

Historia de las Herramientas CASE:

Las Herramientas CASE tienen su inicio con el simple procesador de palabras que fue usado para crear y manipular documentación. Los setentas vieron la introducción de técnicas gráficas y diagramas de flujo de estructuras de datos. Sobre este punto, el diseño y especificaciones en forma pictórica han sido extremadamente complejos y consumían mucho tiempo para realizar cambios.

La introducción de las herramientas CASE para ayudar en este proceso ha permitido que los diagramas puedan ser fácilmente creados y modificados, mejorando la calidad de los diseños de software. Los diccionarios de datos, un documento muy usado que mantiene los detalles de cada tipo de dato y los procesos dentro de un sistema, son el resultado directo de la llegada del diseño de flujo de datos y análisis estructural, hecho posible a través de las mejoras en las Herramientas CASE.

Pronto se reemplazaron los paquetes gráficos por paquetes especializados que habilitan la edición, actualización e impresión en múltiples versiones de diseño. Eventualmente, las herramientas gráficas integradas con diccionarios de base de datos para producir poderosos diseños y desarrollar herramientas, podrían sostener ciclos completos de diseño de documentos.

Como un paso final, la verificación de errores y generadores de casos de pruebas fueron incluidos para validar el diseño del software. Todos estos procesos pueden saberse integrados en una simple herramienta CASE que soporta todo el ciclo de desarrollo.

* + 1. Clasificación de las Herramientas CASE

No existe una única clasificación de herramientas CASE y en ocasiones, es difícil incluirlas en una clase determinada. Podrían clasificarse atendiendo a:

* + Las plataformas que soportan.
  + Las fases del ciclo de vida del desarrollo de sistemas que cubren.
  + La arquitectura de las aplicaciones que producen.
  + Su funcionalidad.

Las herramientas CASE, en función de las fases del ciclo de vida abarcadas, se pueden agrupar de la forma siguiente:

1. **Herramientas integradas, I-CASE** (Integrated CASE, CASE integrado):

abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Son

llamadas también CASE workbench.

* ASADAL - Herramienta CASE especializada en Sistemas de Tiempo Real.
* CASE GENEXUS Tool.
* System Architect, herramientas CASE para Análisis y Diseño, incluye técnicas estructuradas y orientadas a objetos.
* Win A&D, herramientas CASE para Análisis y Diseño, incluye técnicas estructuradas y orientadas a objetos

2. **Herramientas de alto nivel, U-CASE** (Upper CASE - CASE superior) ofront-end, orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.

* Erwin.
* EasyCASE.
* Oracle Designer.
* PowerDesigner.
* System Architect.

.

3. **Herramientas de bajo nivel, L-CASE** (Lower CASE - CASE inferior) o back-end, dirigidas a las últimas fases del desarrollo: construcción e implantación.

* Microsoft Project.
* Racional Rose.
* JDeveloper.
* MagicDraw
* Visual Paradigm.
* Microsoft Visio.
* Enterprise Architect

4. **Juegos de herramientas o Tools-Case,** son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

**Actividad:**

Ingresa a la plataforma virtual, luego desarrolla la siguiente actividad propuesta:

1. **CUESTIONARIO TÉCNICO**
2. Conceptúe que es sistemas de información vs ingeniería de software.
3. Enumere los componentes de un sistema de información.
4. Describa 2 tipos de sistemas de información.
5. Determine las etapas de desarrollo de software.
6. Proponga un cuadro comparativo de diferencia entre sistemas de información vs ingeniería de software.

1. **CONCLUSIONES DE LA EXPERIENCIA**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_