INSTITUTO SUPERIOR TECNOLÓGICO QUITO

Formamos tu **PROPÓSITO DE VIDA**

GUÍA DE PREPARACIÓN PARA EXAMEN COMPLEXIVO

UNIDAD DE TITULACIÓN

www.itq.edu.ec

INSTITUTO SUPERIOR TECNOLÓGICO QUITO - DESDE 1984 - SABER HACER

INSTITUTO SUPERIOR TECNOLÓGICO QUITO
Excelencia en Educación Superior

GUÍA GENERAL DE PREPARACIÓN PARA EXAMEN COMPLEXIVO

CARRERA DE DESARROLLO DE SOFTWARE

AUTOR (ES):

ELVIS PACHACAMA

XIMENA MARCILLO

MONICA RAMIREZ

GABRIEL HOYOS

EDICIÓN: TERCERA

AÑO: 2025

TRABAJO EN EDICIÓN:

DOXA EDITION

EQUIPO EDITORIAL:

PATRICIO CELI

GINO CORDERO

Este material está protegido por derechos de autor. Queda estrictamente prohibida la reproducción total o parcial de esta obra en cualquier medio sin la autorización escrita de los autores y el equipo editorial. El incumplimiento de esta prohibición puede conllevar sanciones establecidas en las leyes de Ecuador.

Todos los derechos están reservados.

ISBN:

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

SOBRE LOS AUTORES

Ximena Marcillo es una ingeniera de sistemas graduada con distinción de la Universidad Politécnica Salesiana en el año 2017. Su carrera ha estado marcada por una destacada trayectoria en proyectos relevantes en el campo de la ingeniería de sistemas. Su experiencia y habilidades técnicas le han permitido destacarse en el ámbito de la informática y la tecnología.

Como profesional comprometida con el desarrollo académico, Ximena Marcillo ha dejado una huella significativa en el campo de la educación. Publicó un libro sobre análisis de sistemas de información, consolidando así su experiencia y conocimiento en un recurso valioso para estudiantes y profesionales en este campo.

Su contribución al ámbito educativo no se limita a la publicación. Desde el año 2018 hasta el 2022, se desempeñó como docente en el Instituto Tecnológico Quito, específicamente en la carrera de Desarrollo de Software. Durante este tiempo, compartió su experiencia práctica y conocimientos teóricos con sus estudiantes, contribuyendo al desarrollo de futuros profesionales en el campo de la tecnología.

En reconocimiento a su dedicación y habilidades de liderazgo, en el año 2023, Ximena Marcillo fue asignada como la coordinadora de la carrera de Desarrollo de Software en el mismo instituto. Además, asumió el rol de directora de la Escuela de Tics (Tecnologías de la Información y Comunicación), demostrando así su capacidad para liderar a nivel institucional y para influir positivamente en el crecimiento y la dirección de programas académicos.

Con una combinación de experiencia en el sector privado, contribuciones significativas a la educación y roles de liderazgo en el ámbito académico, Ximena Marcillo se ha establecido como una figura integral en el campo de la ingeniería de sistemas y el desarrollo de software. Su perfil diversificado, que abarca tanto la práctica como la enseñanza, refleja un compromiso continuo con la excelencia en su campo y un deseo de influir positivamente en las futuras generaciones de profesionales en tecnología.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

SOBRE LOS AUTORES

Elvis Pachacama Cabezas es un educador y profesional con un título de tercer nivel en Ingeniería en Tecnologías de la Información, obtenido en la Universidad Estatal de Sumy en Ucrania (SUMDU). Se destaca por su participación en la vida estudiantil, ya que fue representante de los estudiantes ecuatorianos en el Consejo Estudiantil de la universidad, lo que sugiere un compromiso activo con los asuntos estudiantiles.

Además, Elvis se graduó con honores y recibió reconocimiento por su destacado desempeño académico en su título universitario. Esta distinción refleja su dedicación y excelencia en sus estudios.

Actualmente, Elvis Pachacama Cabezas es docente en el Instituto Superior Tecnológico Quito, donde imparte cátedra en la Carrera de Desarrollo de Software. Su posición como educador indica una dedicación al campo de la enseñanza y sugiere que posee conocimientos prácticos en el desarrollo de software, que comparte con los estudiantes.

Adicionalmente, es relevante mencionar que Elvis está próximo a cursar la maestría en inteligencia artificial en la Universidad Internacional de Valencia. Este detalle indica su interés en profundizar sus conocimientos en un campo avanzado y en constante evolución, como es la inteligencia artificial.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

SOBRE LOS AUTORES

Mónica Ramírez es una profesional con más de 10 años de experiencia en el apasionante mundo de las Tecnologías de la Información (TI). A lo largo de su carrera, se ha destacado por liderar proyectos innovadores y fomentar una comunicación efectiva en equipos multidisciplinarios. Su amor por la educación y la innovación la llevó a obtener una Maestría en Gerencia y Liderazgo Educativo, y una Maestría en Inteligencia Artificial Aplicada, lo que refleja su compromiso con el aprendizaje continuo y la innovación.

Con más de 8 años de experiencia como docente universitaria, Mónica ha dejado una huella significativa en el desarrollo académico y personal de sus estudiantes. Su enfoque pedagógico busca formar profesionales íntegros, combinando conocimientos técnicos con habilidades interpersonales esenciales para el ámbito laboral.

La sólida trayectoria de Mónica en TI, junto con su interés en áreas emergentes como la Inteligencia Artificial, le permite ofrecer una educación actualizada y relevante. Ella cree firmemente que la educación tiene el poder de transformar vidas, preparándolos para enfrentar los desafíos del futuro con confianza, conocimiento y creatividad.

Su carrera comenzó como Ingeniera de Software, donde adquirió habilidades técnicas esenciales. Luego, se desempeñó como Analista de Software y Administradora de Base de Datos, así como Analista de Ciencia de Datos, aplicando técnicas de análisis para extraer información valiosa que respalda la toma de decisiones estratégicas. Su capacidad de liderazgo la llevó a convertirse en Gerente de Desarrollo de Software, donde impulsó proyectos innovadores y mejoró la eficiencia de los equipos. Además, ha realizado importantes contribuciones en el ámbito educativo, trabajando como Analista Técnica en evaluación educativa a nivel nacional y como Docente Investigador.

Gracias a su experiencia en el sector público y privado, así como a sus valiosas contribuciones a la educación y su liderazgo académico, Mónica se ha consolidado como una referente en el ámbito de las Tecnologías de la Información y la Educación. Su compromiso con la excelencia y su anhelo de influir positivamente en las futuras generaciones de profesionales en tecnología son palpables en su trayectoria y en los logros que ha conseguido a lo largo de su carrera

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

SOBRE EL AUTOR

Gabriel Hoyos Ingeniero de sistemas graduado en la mención desarrollo de aplicaciones para la gestión en la Universidad Politécnica Salesiana en el año 2018. Su carrera ha estado marcada por una basta trayectoria en proyectos relevantes en el manejo, supervisión, soporte e implementación de ERP's así como desarrollo y diseño de los mismos. Su experiencia y habilidades técnicas le han permitido destacarse en el ámbito de la informática, desarrollo y la tecnología gracias a una sólida comprensión de los sistemas de planificación de recursos empresariales (ERP), con experiencia en la implementación, configuración y personalización de varias plataformas ERP para satisfacer las necesidades específicas de las organizaciones.

He liderado y gestionado proyectos de implementación de ERP desde la fase de planificación hasta la ejecución y el soporte post-implementación, asegurando el cumplimiento de plazos y presupuesto soy experto en trabajar con stakeholders para identificar y documentar requisitos de negocio, lo que me permite adaptar el ERP para mejorar la eficiencia operativa y apoyar los objetivos estratégicos de la empresa utilizado mi conocimiento en ERP´s para analizar y optimizar procesos de negocio, reduciendo costos y aumentando la productividad mediante la automatización y la mejora continua.

Además de proporcionar soporte técnico experto y de capacitación a usuarios finales, ayudándolos a maximizar el uso del sistema ERP y resolver cualquier problema técnico que pueda surgir mediante un profundo conocimiento de las tecnologías y herramientas utilizadas en el desarrollo de interfaces de software, incluyendo lenguajes de programación, bibliotecas, y frameworks de front-end como HTML, CSS, JavaScript, React, Angular, entre otros.

Al ser capaz de descomponer conceptos técnicos complejos en términos más simples que permitan facilitar la comprensión de los estudiantes, utilizando ejemplos claros y relevantes desarrollando planes de estudio que aborden tanto los fundamentos como las tendencias actuales en el desarrollo de interfaces, asegurando que los estudiantes adquieran conocimientos actualizados y aplicables. Lo cual me permite ofrecer orientación y apoyo a los estudiantes, ayudándoles a superar desafíos y a desarrollar sus habilidades de resolución de problemas y pensamiento crítico.

Esto a su vez le ha permitido mantenerse al día con las últimas tendencias y avances en el desarrollo de interfaces de software, incorporando nuevos conocimientos y tecnologías en la enseñanza siendo esta la pauta para comprometerse con el campo del desarrollo y trabajar con otros departamentos y disciplinas que me han permitido integrar el diseño, desarrollo e implementación de interfaces en un contexto más amplio, mostrando cómo se relaciona con otros aspectos dentro del desarrollo de software y la tecnología.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

CONTENIDO

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

ÍNDICE DE FIGURAS

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

ÍNDICE DE TABLAS

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

PRESENTACIÓN DE LA GUÍA DE INTEGRACIÓN CURRICULAR

Queridos Estudiantes ¡Bienvenidos!

El núcleo análisis de sistemas e innovación se encarga de estudiar sistemas existentes con el objetivo de mejorarlos o crear nuevos sistemas. En este contexto, un sistema se refiere a cualquier conjunto de componentes relacionados que trabajan juntos para lograr un objetivo. El Análisis de Sistemas estudia todos los aspectos de los sistemas computacionales, tanto el hardware como el software, el personal y el ambiente en el que opera procurando su calidad y seguridad.

El Análisis de Sistemas implica entender y documentar los requisitos del sistema, diseñar y desarrollar una solución, implementarla y mantenerla. Esto se logra mediante el uso de técnicas de investigación, análisis, diseño, seguridad y calidad de software, así como herramientas computacionales.

El estudiante estará en la capacidad de generar soluciones directas, genéricas, simples y rápidas para cualquier requerimiento orientado a soluciones en la web, siempre y cuando sean requerimientos con un alcance especifico, el mismo que debe contener seguridad y calidad de software.

Los procedimientos de evaluación que permitirán demostrar el logro de los resultados de aprendizaje definidos para este núcleo estructurante corresponden a: examen teórico que evaluará los conocimientos adquiridos y el examen práctico el cual valorará el dominio del núcleo estructurante.

NÚCLEOS ESTRUCTURANTE DE LA CARRERA

Análisis de Sistemas E Innovación

Lenguajes de Programación

Base de Datos

Infraestructura y Comunicaciones

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

NÚCLEO ESTRUCTURANTE Y EJE TÉMATICO DE LA GUÍA

ANÁLISIS DE SISTEMAS E INNOVACIÓN

Análisis de sistemas de información: Metodologías de desarrollo de sistemas de información, requerimientos funcionales y no funcionales.

Calidad de software: Estándares y criterios de calidad de software en el desarrollo de un sistema de información.

Seguridad Informática: Análisis y herramientas de detección de vulnerabilidad y riesgos para aplicativos informáticos.

ORIENTACIONES GENERALES DEL ESTUDIANTE

Estas orientaciones se han diseñado especialmente para usted por ende es importante que considere lo siguiente:

- Es recomendable disponer de un cierto dominio del inglés escrito que permita al estudiante leer la bibliografía de consulta.
- Es importante la disposición del estudiante para la realización de los ejercicios que cada día le propondrá el docente. El estudiante deberá cotejar a diario su solución con la propuesta por el docente con el fin de ir mejorando paulatinamente su estilo de programación.
- No limite su tiempo de programación al aula de clases, busque cada espacio que pueda para practicar lo que vio, generar nuevas dudas, y resolverlas. No hay mejor forma de aprender programación que, de hecho, programado.
- Utilice al menos una hora al para revisar el contenido de la guía.
- Recuerda que hay una evaluación por parcial, donde se tomara en cuenta las actividades de esta guía.
- El medio de comunicación que se utilizara en la asignatura es el correo

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

institucional; así como la aplicación de Telegram en el horario establecido en el aula virtual.

- Finalmente refuerce los conocimientos por medio de consultas realizadas en la biblioteca.

EVALUACIÓN INICIAL

La evaluación inicial de una materia implica descubrir los conocimientos previos de los estudiantes, para luego establecer el nivel adecuado de aprendizaje para el núcleo estructurante. Esto incluye evaluar el nivel de comprensión de los estudiantes, sus intereses, necesidades y habilidades. Además, es importante evaluar el ambiente de aprendizaje, tanto físico como social, para asegurar que los estudiantes puedan trabajar juntos y tener acceso a la información necesaria para aprender. Finalmente, el docente debe identificar los recursos necesarios para ayudar a los estudiantes a alcanzar sus objetivos educativos.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

EJE TEMÁTICO I

GUIA GENERAL DE ANÁLISIS DE SISTEMAS DE INFORMACIÓN

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

GUÍA GENERAL DE SISTEMAS DE INFORMACIÓN

1. DESCRIPCIÓN DE LA ASIGNATURA

La asignatura de Análisis de Sistemas de Información proporciona a los estudiantes una comprensión profunda de los principios, técnicas y herramientas necesarias para analizar, diseñar y mejorar sistemas de información efectivos. A través de una combinación de teoría y práctica, los estudiantes exploran el ciclo de vida completo del desarrollo de sistemas, desde la identificación de requerimientos hasta la implementación y mantenimiento.

Durante el curso, los estudiantes aprenden a identificar las necesidades de los usuarios y las partes interesadas, y a traducirlas en requerimientos claros y concisos. Se introducen y aplican diversas técnicas de recolección y análisis de requerimientos, así como herramientas de modelado para representar visualmente los procesos, datos y comportamientos del sistema.

Además, se exploran diferentes metodologías de desarrollo de sistemas, desde enfoques tradicionales como el modelo en cascada, hasta metodologías ágiles como Scrum y Kanban. Los estudiantes comprenden los principios fundamentales de cada metodología y aplican técnicas específicas para la planificación, ejecución y seguimiento de proyectos de desarrollo de sistemas.

Otro aspecto importante de la asignatura es el énfasis en la comunicación efectiva y la colaboración con los stakeholders del proyecto. Los estudiantes aprenden a documentar y presentar sus análisis y diseños de manera clara y concisa, y a trabajar en equipo para alcanzar los objetivos del proyecto.

2. BIBLIOGRAFÍA

Esta bibliografía proporciona una combinación de recursos esenciales para abordar los contenidos principales de la asignatura, así como fuentes adicionales para ampliar y profundizar en el tema de estudio.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

2.1. BÁSICA

- Pérez,M. (2012). *Desarrollo de elementos software para gestión de sistemas:* (2 ed.). Editorial ICB..

Este texto es un recurso fundamental para comprender los conceptos relacionados con los sistemas de información. Además, se centra en el desarrollo de componentes de software para la gestión de dichos sistemas, el análisis de requisitos para el desarrollo, la creación de elementos y el diseño de procesos de los sistemas. Todos estos aspectos son cruciales al diseñar e implementar un sistema de información y gestión.

- Análisis y Diseño de Sistemas de Información", Autor: James A. Senn, Editorial: McGraw-Hill, Año de publicación: 2016

Este libro es seleccionado debido a su enfoque exhaustivo en los principios fundamentales del análisis y diseño de sistemas de información. Proporciona una base sólida para comprender los conceptos clave y las mejores prácticas en este campo, lo que lo convierte en un recurso esencial para los estudiantes que desean dominar las habilidades necesarias en el desarrollo de sistemas.

- Ingeniería de Requisitos: Técnicas y Herramientas", Autor: Karl Wiegers, Editorial: Pearson, Año de publicación: 2003

Este libro es fundamental para comprender el proceso de ingeniería de requisitos, que es crucial en el análisis de sistemas de información. Ofrece una cobertura completa de las técnicas y herramientas utilizadas en la identificación, documentación y gestión de los requisitos del sistema, lo que lo convierte en una lectura obligatoria para los estudiantes que desean adquirir competencias en esta área clave

2.2. COMPLEMENTARIA

- Desarrollo Ágil de Software: Principios, Patrones y Prácticas", Autor: Robert C. Martin y Martin Fowler, Editorial: Prentice Hall, Año de publicación: 2002

Este libro complementario ofrece una visión profunda del enfoque ágil en el desarrollo de software. Ayuda a los estudiantes a comprender los principios y prácticas ágiles, que son cada vez más relevantes en el campo de la ingeniería de software y el análisis de sistemas de información.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Análisis y Diseño Orientado a Objetos", Autor: Grady Booch, Ivar Jacobson y James Rumbaugh, Editorial: Addison-Wesley, Año de publicación: 2007

Este libro proporciona una introducción completa al análisis y diseño orientado a objetos, que es un enfoque popular en el desarrollo de sistemas de información. Ayuda a los estudiantes a comprender los conceptos fundamentales y las mejores prácticas en el diseño de sistemas basados en objetos, lo que enriquece su comprensión y habilidades en el análisis de sistemas.

- Lean Software Development: An Agile Toolkit", Autor: Mary Poppendieck y Tom Poppendieck, Editorial: Addison-Wesley, Año de publicación: 2003

Este libro ofrece una perspectiva valiosa sobre el desarrollo ágil de software desde una perspectiva Lean. Ayuda a los estudiantes a comprender cómo aplicar los principios de Lean en el contexto del desarrollo de software, lo que puede mejorar su eficiencia y calidad en el análisis y diseño de sistemas de información.

- User Stories Applied: For Agile Software Development", Autor: Mike Cohn, Editorial: Addison-Wesley, Año de publicación: 2004

Este libro es una lectura útil para comprender en detalle el concepto de historias de usuario, que es una técnica fundamental en el desarrollo ágil de software. Proporciona ejemplos prácticos y consejos para escribir y gestionar historias de usuario efectivas, lo que puede mejorar la calidad y la comunicación en el proceso de análisis de sistemas de información.

3. COMPETENCIA ESPECIFICAS Y GENERICAS

- Valores & habilidades blandas: cultura: creatividad
- valores & habilidades blandas: justicia: resolución de problemas
- valores & habilidades blandas: lealtad: liderazgo
- valores & habilidades blandas: optimismo: planificación y gestión del tiempo

4. OBJETIVO GENERAL

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Desarrollar competencias integrales en el análisis de sistemas de información, abarcando la comprensión teórica de los fundamentos, la aplicación de técnicas avanzadas de análisis y diseño, y la internalización de principios éticos y responsabilidad profesional, a través de la exploración de conceptos clave, la práctica activa de técnicas de recolección y documentación de requerimientos, y la participación en proyectos prácticos, con el propósito de formar profesionales capaces de contribuir al éxito organizacional y al avance social en un entorno tecnológico dinámico, fomentando una actitud ética y comprometida con la calidad y la responsabilidad en todas las etapas del proceso de análisis de sistemas

UNIDAD 1: ANÁLISIS DE SISTEMAS DE INFORMACIÓN

Temas y Subtemas

- **Fundamentos de análisis de sistemas de información**
- **Ingenieria de requerimientos**
- **Diagramación UML**
- **Metodologias de desarrollo**
- **Ciclo de vida del desarrollo de un software**

Fundamentos de análisis de sistemas de información

El análisis de sistemas de información se enfoca en estudiar, diseñar y mejorar los sistemas que permiten a las organizaciones manejar grandes cantidades de datos. Un sistema de información está compuesto por un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información, apoyando la toma de decisiones, la coordinación, el control y el análisis en una empresa.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

El análisis de sistemas de información tiene como objetivo entender cómo fluye la información en una organización y cómo puede optimizarse para mejorar la eficiencia y efectividad. Esto incluye la identificación de los requerimientos del sistema, el diseño de soluciones tecnológicas y la evaluación de la factibilidad técnica y económica del proyecto.

Importancia de los Sistemas de Información

Los sistemas de información son fundamentales para el éxito de cualquier organización moderna, ya que proporcionan la infraestructura necesaria para gestionar la información, que es un recurso clave en la era digital. Aquí algunos puntos clave sobre su importancia:

- Soporte en la toma de decisiones: Los sistemas de información permiten a los gerentes acceder a datos actualizados y relevantes, lo que facilita la toma de decisiones estratégicas.
- Mejora de la eficiencia operativa: Ayudan a automatizar procesos y tareas rutinarias, lo que permite a las organizaciones operar de manera más rápida y eficiente.
- Ventaja competitiva: Permiten a las empresas responder rápidamente a cambios en el mercado y aprovechar oportunidades de negocio al analizar tendencias y comportamientos de los clientes.
- Optimización de recursos: Facilitan la administración de recursos humanos, financieros y tecnológicos, lo que permite una mejor asignación de estos recursos.
- Mejora de la comunicación: Facilitan la comunicación interna y externa, proporcionando herramientas para la colaboración entre equipos y la interacción con los clientes.

Características de los sistemas de información

Los sistemas de información son herramientas fundamentales en el ámbito organizacional, ya que permiten la recopilación, procesamiento y distribución de datos de manera eficiente. Entre sus características más destacadas se encuentran la capacidad de integrar diferentes fuentes de información, la adaptabilidad a las necesidades cambiantes de la organización y la facilitación de la toma de decisiones informadas. Además, estos sistemas son esenciales para mejorar la comunicación interna y externa, optimizar procesos operativos y fomentar una cultura de innovación y agilidad en el entorno empresarial. En este contexto, comprender las características de los sistemas de información se vuelve crucial para aprovechar al máximo su

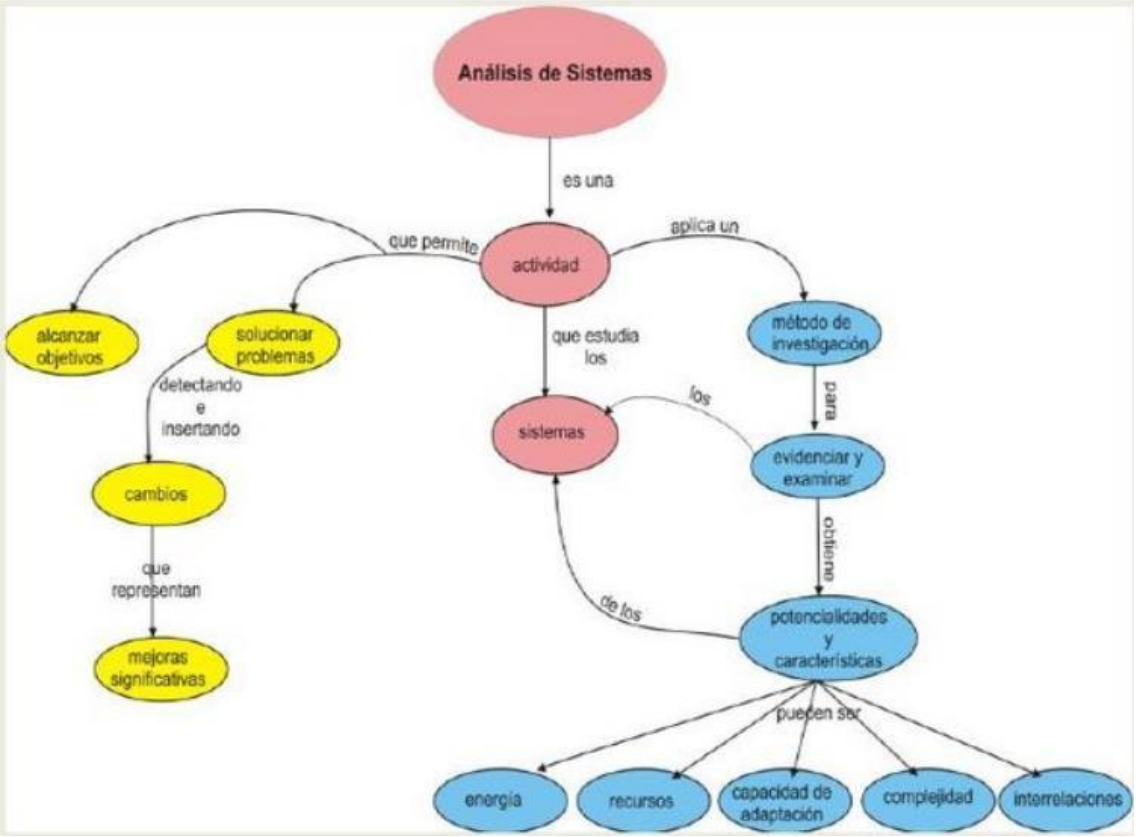
(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

potencial y contribuir al éxito de las organizaciones, entre las características que se puede mencionar son:

- Las partes que compone un sistema de información son elementos o subsistemas existentes.

- Los objetivos del sistema son la razón de ser de este.
- El medio ambiente de un sistema son todos los objetos que se encuentran fuera de las fronteras de los sistemas.
- Los elementos de control que está relacionado con la naturaleza de los sistemas, bien sean abiertos o cerrados y la manera como estos operan dentro de los niveles de desempeño considerados como aceptables (estándares).
- Los ajustes del sistema se realizan al comparar el desempeño del sistema actual con estos estándares.
- La información proporcionada al comparar los resultados con los estándares junto con el proceso de reportar las diferencias a los elementos de control se conoce como la retroalimentación.

Ilustración 1:
Análisis de sistemas



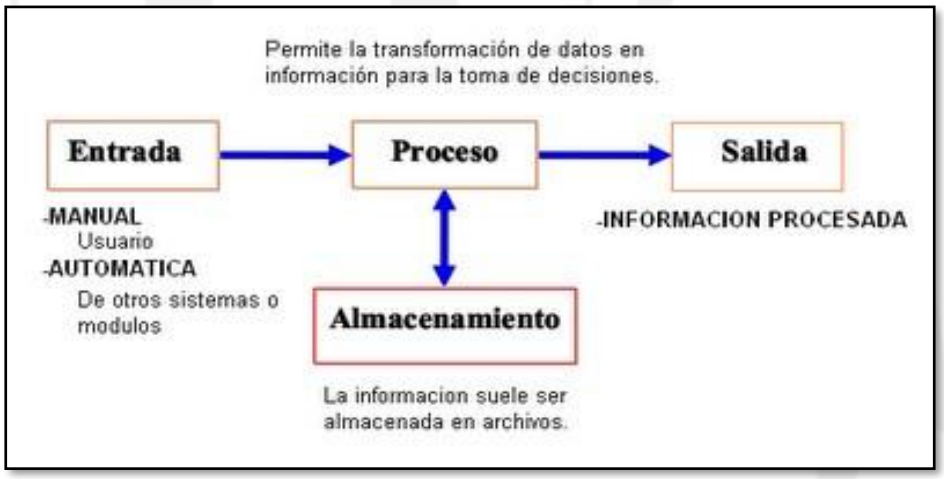
(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Nota: *en la ilustración se puede observar un mapa conceptual sobre el análisis de sistemas de información.*

Actividades del Sistema de Información: Entrada, Proceso y Salida

Un sistema de información es un conjunto de componentes interrelacionados que trabajan juntos para recolectar, procesar, almacenar y distribuir datos y convertirlos en información útil. Los sistemas de información facilitan la toma de decisiones, la coordinación, el control y la visualización de datos, y permiten a las organizaciones funcionar de manera más eficiente y competitiva. Para entender cómo funcionan estos sistemas, es esencial descomponer sus actividades principales en tres grandes etapas: entrada, proceso y salida.

Ilustración 2:
Actividades de un sistema de información



Nota: *en la ilustración se puede observar las actividades por las que pasa un sistema de información.*

Entrada:

La entrada en un sistema de información es la fase en la que se recopilan los datos necesarios para ser procesados. Estos datos pueden provenir de diversas fuentes, tanto internas como externas a la organización, y pueden estar en diferentes formatos, como texto, números, imágenes, sonido o video. La entrada de datos tiene dos aspectos clave:

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Captura de datos: La información se recoge de diversas fuentes, como formularios, sensores, escáneres, transacciones de ventas, encuestas, bases de datos existentes, entre otros.
- Verificación de datos: Asegura que los datos recopilados sean precisos, completos y estén en el formato correcto antes de ser procesados. Este proceso puede incluir la revisión de la calidad de los datos o la eliminación de duplicados.

Se puede decir que es el proceso mediante el cual se captura y prepara datos para su posterior procesamiento. Generalmente constituidas por:

- Datos, descripciones básicas de cosas, acontecimientos, actividades y transacciones que se registran, clasifican y almacenan pero que no se organizan de acuerdo con ningún significado especifico.
- Programas, aplicaciones y recursos que permiten desarrollar diferentes tareas en un computador y otro equipo tecnológico.
- Procedimientos, serie de pasos bien definidos que permitan y faciliten la realización de un trabajo de la manera más adecuada.
- Estándares, definen como interactúan los componentes informáticos que existen. Establecen los pasos a ser realizados para el uso de determinado sistema.
- Controles de acceso: aprobación de acceso, el sistema adopta la decisión de conceder o rechazar solicitudes de acceso a él.
- Hardware, conjunto de componentes que conforman la parte física de un computador.

Un ejemplo en un sistema de ventas, la entrada puede ser el registro de las compras realizadas por los clientes, ingresando información como el número de producto, cantidad comprada y detalles del cliente.

Proceso

La etapa de proceso implica la conversión de los datos crudos recopilados en información útil. Esta es la fase donde los datos son organizados, transformados, analizados o calculados para que puedan ser interpretados y usados para la toma de decisiones.

El procesamiento puede involucrar varias actividades, tales como:

- Clasificación de datos: Ordenar o agrupar los datos según ciertos criterios o parámetros.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Cálculos: Realizar operaciones aritméticas o lógicas sobre los datos, como sumar, promediar, comparar, etc.
- Almacenamiento: Guardar los datos procesados para usarlos en el futuro. En algunos sistemas, el almacenamiento es una parte crítica del proceso.
- Análisis: Descomponer los datos para identificar patrones, tendencias o anomalías que sean útiles para tomar decisiones.

Se puede decir que esta acción es la capacidad de efectuar operaciones con los datos guardados en las unidades de memoria. Capacidad del Sistema de Información (SI) para convertir la información procesada o datos de entrada en información para el exterior.

Un ejemplo de procesamiento en un sistema de información: En un sistema bancario, cuando un cliente realiza una transacción, el sistema no solo registra la entrada de datos, sino que actualiza el saldo de la cuenta y genera un registro de la transacción. También puede calcular intereses o tarifas relacionadas con el movimiento de fondos.

Salida

La salida es la última actividad del sistema de información y representa la presentación de los datos procesados en un formato que sea comprensible y útil para los usuarios finales. En esta etapa, la información transformada se entrega a las personas, máquinas o sistemas que la utilizarán para tomar decisiones o para otras actividades operativas.

Los tipos de salida pueden incluir:

- Informes: Documentos que presentan datos de manera estructurada, como informes financieros, de ventas o de rendimiento.
- Pantallas interactivas: Visualizaciones gráficas o tabulares de la información en tiempo real para usuarios dentro de un sistema informático.
- Gráficos y tablas: Representaciones visuales de los datos, como gráficos de barras, líneas o tablas de comparación que facilitan la interpretación.
- Alertas o notificaciones: Información de salida en forma de mensajes automáticos que pueden ser enviadas a los usuarios para indicar situaciones de alerta o recordatorios.

Se puede decir que es el resultado e información generados por el sistema. Para muchos usuarios finales la salida, es la única razón para el desarrollo del sistema y la base sobre la que ellos evalúan la utilidad de la aplicación. Para diseñar una salida se debe:

- Determinar qué información presentar.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Decidir si la información será presentada en forma visual o impresa y seleccionar el medio de salida.
- Disponer la presentación de la información en un formato aceptable.
- Decidir como distribuir la salida entre posibles destinatarios.

Un ejemplo de salida en un sistema de información en un sistema de inventarios, la salida puede ser un informe que detalla el stock disponible, las ventas recientes y las necesidades de reabastecimiento. Este informe puede ser utilizado por los gerentes para tomar decisiones de compra.

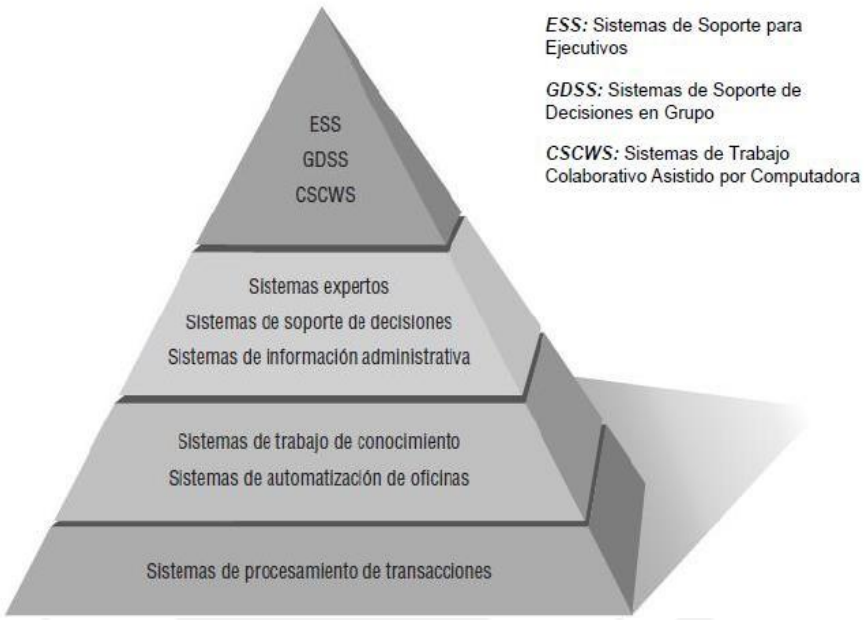
La comprensión de estas actividades es esencial para diseñar y mejorar los sistemas de información que soportan las operaciones diarias y la toma de decisiones estratégicas en las organizaciones.

Tipos de Sistemas de Información

Los sistemas de información son herramientas versátiles que se adaptan a las diversas necesidades de las organizaciones, y se clasifican en diferentes tipos según su función y el nivel de toma de decisiones que apoyan. Entre los principales tipos se encuentran los sistemas de procesamiento de transacciones, que manejan datos operativos diarios; los sistemas de información gerencial, que proporcionan informes para la toma de decisiones en niveles medios; y los sistemas de apoyo a la decisión, que asisten a la alta dirección en la formulación de estrategias. Cada tipo de sistema cumple un rol específico en la estructura organizacional, facilitando el flujo de información y mejorando la eficiencia operativa. Comprender estos tipos es esencial para implementar soluciones tecnológicas que se alineen con los objetivos empresariales.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ilustración 3:
Tipos de sistemas de información



***Nota**: en la ilustración se puede observar los tipos de sistemas de información*

A continuación, los diferentes tipos de sistemas:

Sistema de procesamiento de transacciones

Son sistemas de información computarizados utilizados para gestionar grandes cantidades de información referente a las transacciones rutinarias producidas en una empresa u organización. Se pueden identificar 5 categorías: ventas y marketing, producción, finanzas, contabilidad, recursos humanos.

Los sistemas de procesamiento de transacciones son sistemas que atraviesan límites y permiten que la organización interactúe con los entornos externos.

Sistema de conocimiento

Los Sistemas de Automatización de Oficinas, brindan apoyo a las personas que trabajan con datos no para crear conocimiento sino para analizar la información y transformar los datos o manipular la desierta forma antes de compartir los o diseminarlos.

Sistemas de Trabajo de Conocimiento brindan apoyo a profesionales como científicos, ingenieros y médicos, ayudándoles a crear conocimiento y a integrarlo a su organización o la sociedad.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Sistema de información administrativa

Estos sistemas incluyen el procesamiento de transacciones. Ayuda a los encargados de tomar decisiones (no rutinarias, para distinguir las de las del nivel operativo).

Para acceder a la información, los usuarios del sistema de información administrativa comparten una base de datos común; produciendo información que se utiliza en el proceso de toma de decisiones.

Sistema de soporte de decisiones

Este es el entorno en el que tiene lugar la toma de decisiones en una empresa, las decisiones se toman en todos los niveles y en todas las áreas. Pertenecen a una clase superior de sistemas de información computarizados, se ajustan más a la persona o el grupo usuario.

Sistema de Inteligencia artificial

La inteligencia artificial (IA) se refiere a la capacidad de las máquinas para realizar tareas que normalmente requieren inteligencia humana, como el aprendizaje, el razonamiento, la percepción y la toma de decisiones. Estos sistemas están diseñados para imitar funciones cognitivas humanas, permitiendo que las computadoras procesen y analicen datos de manera similar a como lo haría un ser humano. La IA puede mejorar su rendimiento a medida que recopila más información, lo que la hace cada vez más eficiente en la resolución de problemas complejos y en la automatización de procesos. En esencia, la inteligencia artificial busca crear tecnologías que no solo realicen tareas específicas, sino que también aprendan y se adapten a su entorno mediante procesos predictivos y generativos.

Sistemas expertos

Estos sistemas de información emulan el razonamiento y la toma de decisiones de un experto en un área específica del conocimiento. Estos sistemas son una aplicación de la inteligencia artificial y están diseñados para resolver problemas complejos al igual que lo haría un especialista humano. Se caracterizan por su capacidad para igualar o incluso superar las habilidades de un experto en un dominio particular, utilizando bases de conocimiento y reglas de inferencia para llegar a conclusiones o recomendaciones. Los sistemas expertos son ampliamente utilizados en diversas áreas, como la medicina, la ingeniería y la gestión empresarial, donde pueden desempeñar un papel crucial en el diagnóstico, la planificación y la toma de decisiones.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Integración de las tecnologías en los sistemas de información

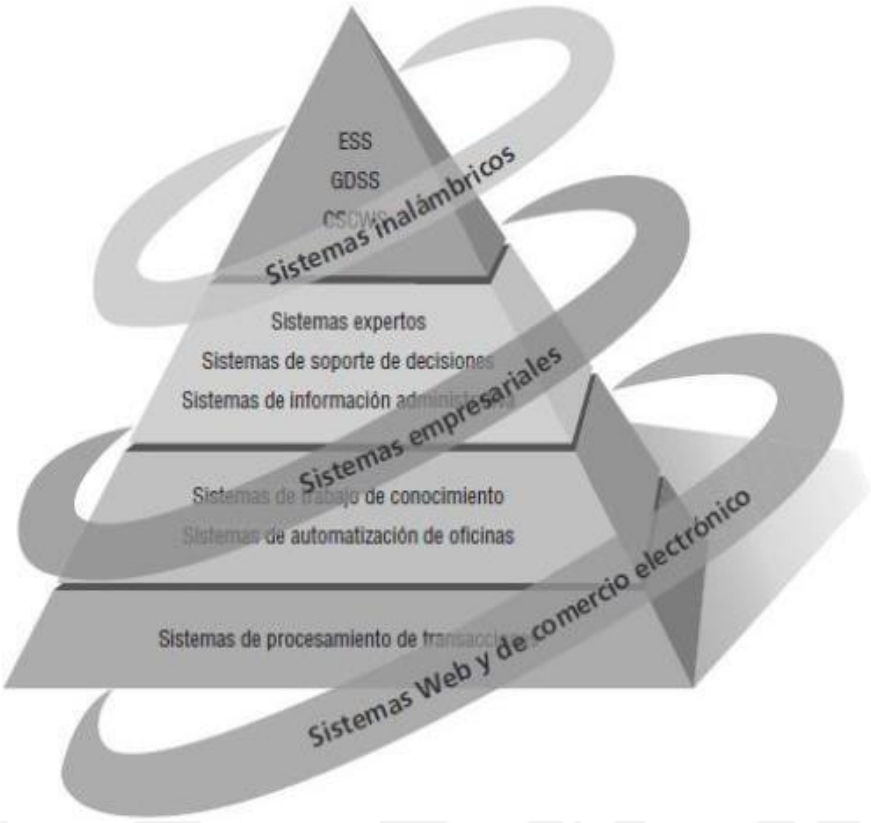
La integración de las tecnologías en los sistemas de información es un proceso fundamental que permite a las organizaciones optimizar sus operaciones y mejorar la toma de decisiones. A medida que el entorno empresarial se vuelve más dinámico y competitivo, la incorporación de herramientas tecnológicas, como el Big Data, la inteligencia artificial, el Internet de las Cosas (IoT) y la computación en la nube, se convierte en una necesidad imperante. Esta integración no solo facilita la recopilación y el análisis de grandes volúmenes de datos, sino que también permite una mejor comunicación y colaboración entre diferentes sistemas y departamentos. Al unificar estas tecnologías, las empresas pueden crear sistemas de información más eficientes y adaptables, capaces de responder rápidamente a las demandas del mercado y a las necesidades de los usuarios.

EL conjunto de productos y servicios que ofrecen las nuevas tecnologías ayudan a obtener soluciones reales para la organización. Algunos de estos procesos y productos son:

- Sistema de planeación de recursos empresariales, sistemas de planificación de recursos empresariales (ERP), (SAP, paquetes de Oracle).
- Sistema para dispositivos inalámbricos y portátiles
- Sistemas basados en la web
- Comercio electrónico
- Sistema de código abierto

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ilustración 4:
Integración de las tecnologías en los SI



Nota: *en la ilustración se puede observar la integración de las tecnologías en los diferentes tipos de sistemas de información.*

Ingeniería de requerimientos

La Ingeniería de Requisitos es esencial dentro del desarrollo de software y sistemas, ya que define y gestiona los requisitos que el sistema debe cumplir. Los requisitos representan las necesidades y expectativas de los usuarios, así como las restricciones que el sistema debe respetar. En términos simples, la ingeniería de requisitos busca responder a la pregunta: ¿Qué debe hacer el sistema y en qué condiciones?

Este proceso involucra la identificación, documentación, análisis, verificación y gestión de los requisitos durante todo el ciclo de vida del proyecto. Su propósito es asegurar que el sistema desarrollado cumpla con las expectativas del cliente o usuario final y que sea técnicamente viable.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Importancia de Ingeniería de requerimientos

La ingeniería de requisitos es crítica para el éxito de cualquier proyecto de desarrollo de software. Un mal manejo de los requisitos puede llevar a malentendidos, errores en el producto final, sobrecostos y demoras en el proyecto. Algunos de los puntos clave que destacan la importancia de la ingeniería de requisitos son:

- Alineación con las necesidades del negocio: A través de los requisitos, se asegura que el sistema esté alineado con los objetivos estratégicos y las necesidades operativas de la organización.
- Prevención de errores: Definir requisitos claros y completos desde el principio ayuda a reducir el riesgo de fallos o malentendidos, evitando retrabajos y costos adicionales.
- Optimización de recursos: Al establecer qué es esencial para el sistema, se puede gestionar mejor el tiempo, el presupuesto y los recursos humanos, priorizando los aspectos más importantes.
- Mejora de la calidad del software: Un enfoque adecuado en la recopilación y gestión de requisitos ayuda a garantizar que el producto final cumpla con los estándares de calidad y satisfaga a los usuarios.

Fases de Ingeniería de requerimientos

La ingeniería de requisitos se organiza en varias fases, cada una diseñada para cumplir un propósito específico dentro del proceso de desarrollo de software. Las principales fases son:

Elicitación de Requisitos:

Esta fase implica la obtención de los requisitos desde diversas fuentes, como los usuarios, los interesados (stakeholders), documentos y sistemas existentes. Se utilizan técnicas como entrevistas, cuestionarios, observación y análisis de documentos. La idea es captar las necesidades del usuario y las especificaciones técnicas desde el principio.

Análisis y Especificación de Requisitos

Una vez recopilados los requisitos, se analizan para asegurar que sean completos, coherentes y factibles. En esta fase, también se clasifican los requisitos como funcionales y no funcionales. Los requisitos funcionales definen qué debe hacer el sistema, mientras que los requisitos no funcionales especifican cómo debe comportarse (rendimiento, seguridad, etc.).

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Validación de Requisitos

En esta etapa, se verifica que los requisitos obtenidos y documentados sean correctos y representen fielmente las necesidades del usuario. Se trata de asegurarse de que no haya malentendidos ni errores en la interpretación de los requerimientos. Las revisiones y las pruebas tempranas juegan un papel importante en esta fase.

Gestión de Requisitos

A lo largo del ciclo de vida del proyecto, los requisitos pueden cambiar debido a nuevas necesidades del cliente, cambios en el entorno tecnológico o condiciones de mercado. La gestión de requisitos implica mantener un control sobre los cambios, asegurando que los requisitos estén actualizados y documentados correctamente, evitando así desalineaciones en el desarrollo del proyecto.

Tipos de requisitos

En la ingeniería de requisitos, los tipos de requisitos son fundamentales para el éxito de un proyecto, ya que determinan las expectativas y necesidades tanto del cliente como de los usuarios finales.

Comprender y clasificar adecuadamente estos tipos de requisitos permite a los equipos de desarrollo establecer un marco claro para la planificación, diseño y validación de sistemas, asegurando que se cumplan las expectativas y se minimicen los riesgos durante todo el ciclo de vida del proyecto. Existen dos tipos principales de requisitos en la ingeniería de requisitos:

Requisitos funcionales

Estos describen las funciones que el sistema debe realizar. Están directamente relacionados con las tareas y procesos que el sistema debe llevar a cabo. Por ejemplo, en un sistema de banca en línea, un requisito funcional sería la capacidad de realizar transferencias entre cuentas.

Requisitos no funcionales

Estos especifican criterios de calidad que el sistema debe cumplir, como el rendimiento, la usabilidad, la seguridad o la escalabilidad. Son esenciales para garantizar que el sistema funcione de manera eficiente y confiable bajo las condiciones previstas.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Herramientas y Técnicas en la Ingeniería de Requisitos

En la ingeniería de requisitos, se utilizan diversas herramientas y técnicas para capturar, gestionar y comunicar los requisitos de manera efectiva:

- Diagramas UML: Herramientas visuales como el diagrama de casos de uso y el diagrama de clases son útiles para representar cómo interactúan los usuarios con el sistema y cómo se estructura el sistema internamente.
- Historias de Usuario: Descripciones simples que detallan lo que un usuario final desea hacer con el sistema, facilitando la priorización de tareas.
- Revisión y validación de requisitos: Reuniones periódicas entre el equipo de desarrollo y los stakeholders para asegurar que los requisitos sigan siendo correctos y relevantes.
- Herramientas de gestión de requisitos: Herramientas como Jira o Confluence permiten rastrear los requisitos, hacer cambios y gestionar el flujo de trabajo a lo largo del ciclo de vida del proyecto.

Diagramación UML

Para poder visualizar la trazabilidad de los requisitos y procesos se usa el Lenguaje Unificado de Modelado (UML) que es una herramienta fundamental para el desarrollo y la documentación de sistemas de software. UML se utiliza para visualizar, especificar, construir y documentar los artefactos de un sistema. Este documento tiene como objetivo describir los conceptos, tipos, nomenclatura, simbología y la utilidad de UML en el diseño de sistemas de información.

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un estándar ampliamente utilizado para visualizar, especificar, construir y documentar los componentes de un sistema de software. Los diagramas UML proporcionan una representación visual clara y estructurada de cómo interactúan las diferentes partes del sistema, lo que facilita la comprensión y comunicación entre los desarrolladores, clientes y otros interesados.

En el desarrollo de software, los diagramas UML son esenciales porque permiten modelar las diversas fases del proceso de desarrollo, desde la definición de requisitos hasta la implementación y pruebas. A través de UML, los equipos pueden visualizar el diseño del sistema, identificar posibles problemas antes de la implementación y garantizar que el software desarrollado cumpla con los requisitos establecidos.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Tipos de Diagramas UML

UML incluye diferentes tipos de diagramas divididos en dos categorías principales: diagramas estructurales y diagramas de comportamiento.

Diagramas estructurales

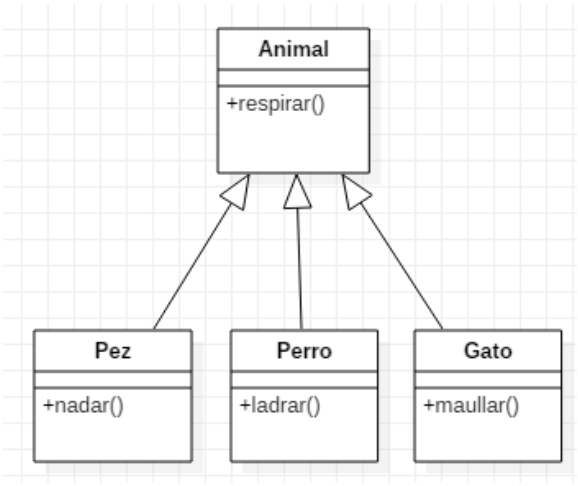
Los diagramas estructurales de UML son herramientas visuales fundamentales en el ámbito del desarrollo de software. Su principal función es representar de manera clara y comprensible la estructura estática de un sistema y sus componentes. Estos diagramas son especialmente valiosos para desarrolladores y partes interesadas, ya que les permiten visualizar de qué manera se organizan y se relacionan los diferentes elementos dentro del sistema, como clases, objetos y bases de datos.

Los diagramas estructurales se centran en cómo está organizada la información dentro de un sistema. Algunos de los más comunes incluyen:

Diagrama de clases

El diagrama de clases muestra las relaciones entre cada objeto en un sistema, representa la estructura estática del sistema mediante clases, sus atributos, métodos y las relaciones entre ellas (herencia, asociación, agregación, etc.). Es útil para mostrar la arquitectura del sistema y cómo se organizan los datos.

Ilustración 5:
Diagrama de clases



***Nota**: En la ilustración se puede observar un ejemplo del diagrama de clases de una clase animal que tiene varias derivadas*

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Los diagramas de clases en UML son herramientas esenciales para describir tanto la estructura como el comportamiento de un sistema. En esencia, las clases actúan como plantillas para crear objetos y se representan mediante rectángulos que se dividen en tres secciones: el nombre de la clase, sus atributos y sus métodos u operaciones. Si bien los atributos y métodos son opcionales, su inclusión permite detallar la funcionalidad de la clase de manera más precisa.

Las señales en estos diagramas indican las comunicaciones unidireccionales entre objetos, mientras que los tipos de datos definen los valores y tipos primitivos que pueden utilizarse. Por otro lado, los paquetes permiten organizar clasificadores relacionados dentro del diagrama y se simbolizan como rectángulos con pestañas, facilitando la visualización y la gestión de la información.

Las interfaces establecen conjuntos de comportamientos que deben ser implementados por una clase, mientras que las enumeraciones representan tipos de datos con valores predefinidos, lo que aporta claridad a la estructura del sistema. Los objetos, que son instancias específicas de clases, también pueden incluirse en el diagrama para mostrar ejemplos concretos de cómo funcionan en la práctica. Finalmente, los artefactos representan entidades físicas dentro del sistema, como bases de datos o componentes de software.

En conjunto, todos estos elementos permiten modelar de manera precisa la estructura y el funcionamiento de un sistema en UML, ofreciendo a los desarrolladores una visión clara y detallada que facilita el proceso de diseño y desarrollo.

Diagrama de objetos

Los diagramas de objetos en UML son representaciones que muestran instancias específicas de clases en un momento determinado, ilustrando cómo los objetos interactúan entre sí. A diferencia de los diagramas de clases, que ofrecen una visión general de la estructura del sistema, los diagramas de objetos son más detallados y concretos, ya que reflejan situaciones o escenarios específicos utilizando ejemplos reales de objetos y sus relaciones.

Cada objeto en un diagrama se representa mediante un rectángulo dividido en dos partes: el nombre del objeto y su clase, separados por dos puntos, junto con los valores de los atributos de ese objeto. Además, se pueden mostrar las conexiones o asociaciones entre objetos, que ilustran las relaciones que mantienen en un contexto particular.

Estos diagramas son muy útiles para visualizar el comportamiento de un sistema en un estado específico, proporcionando una instantánea de la situación de los objetos y sus interacciones. Se

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

suelen emplear en las fases de diseño detallado y en la documentación de instancias concretas del sistema, lo que ayuda a los desarrolladores a entender mejor la dinámica del sistema en situaciones reales.

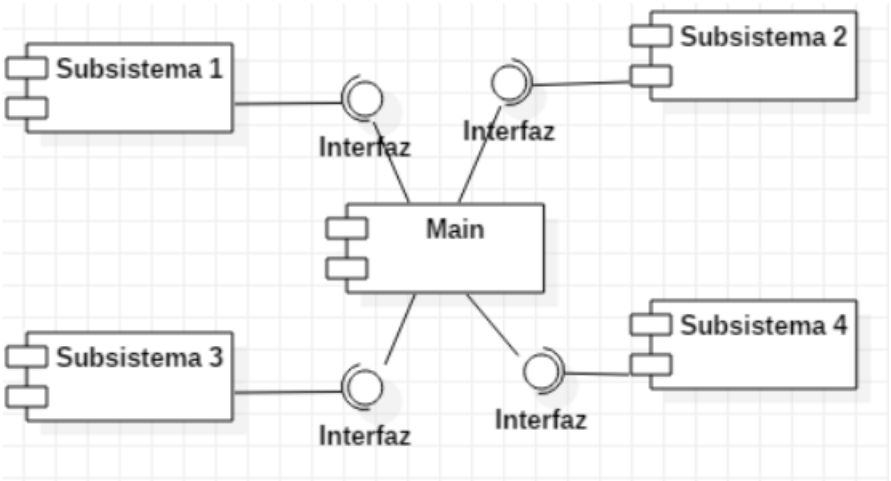
Este diagrama es una instancia del diagrama de clases, que muestra cómo los objetos (instancias de clases) interactúan en un momento particular. Ayuda a entender las relaciones dinámicas entre objetos.

Diagrama de componentes

Los diagramas de componentes en UML son representaciones visuales que muestran la arquitectura física de un sistema, destacando cómo se organizan e interactúan los distintos componentes de software. Un componente es una parte modular e independiente que encapsula una funcionalidad específica y puede ser desplegada por separado. Estos pueden incluir bibliotecas, bases de datos, archivos ejecutables y módulos de software. En estos diagramas, cada componente se representa como un rectángulo con dos pequeñas pestañas en la esquina superior izquierda. También se ilustran las interfaces que los componentes ofrecen o requieren para interactuar, representadas como círculos o líneas con conectores. Las dependencias entre componentes se indican mediante líneas con flechas, mostrando cómo un componente utiliza los servicios de otro. Estos diagramas son esenciales para comprender la organización y las relaciones entre los módulos de un sistema, facilitando el diseño de sistemas escalables y fáciles de mantener. Son especialmente útiles en la planificación de la implementación y la integración de sistemas complejos, permitiendo a los desarrolladores visualizar claramente la estructura del sistema y las conexiones entre sus partes. Representa los diferentes componentes de un sistema, como módulos o bibliotecas, y cómo interactúan entre sí. Se utiliza para visualizar la estructura física de un sistema, facilitando el diseño de sistemas complejos.

Ilustración 6:
Diagrama de componentes

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC



***Nota**: En la ilustración se puede observar un ejemplo del diagrama de componentes que incluye interfaces y subsistemas*

Diagrama de despliegue

Los diagramas de despliegue representan la arquitectura física de un sistema, mostrando cómo se distribuyen los componentes de software en el hardware, es decir, cómo se implementan y ejecutan las aplicaciones en servidores, dispositivos o nodos de red. Estos diagramas son útiles para visualizar la distribución de los sistemas en un entorno físico, ayudando a entender cómo interactúan las piezas de software con la infraestructura subyacente.

En un diagrama de despliegue, los nodos son los elementos clave, representados como cubos tridimensionales, que simbolizan los dispositivos de hardware o entornos de ejecución, como servidores o dispositivos móviles. Dentro de estos nodos, se despliegan los artefactos, que son componentes o archivos ejecutables, como bases de datos, aplicaciones o bibliotecas.

Las comunicaciones entre nodos se representan mediante líneas que indican las conexiones o redes, mostrando cómo fluyen los datos entre los diferentes nodos del sistema. Estos diagramas permiten identificar las interacciones entre los componentes distribuidos y los recursos necesarios para ejecutar el sistema de forma eficiente.

Son esenciales en el diseño de sistemas distribuidos, donde es crucial entender cómo las aplicaciones se despliegan y funcionan en distintos entornos, garantizando una correcta distribución de la carga de trabajo y un óptimo uso de la infraestructura disponible.

Este diagrama muestra la distribución física de los componentes en nodos de hardware. Es útil para entender cómo el software se despliega en la infraestructura de hardware.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Existen algunos diagramas más tales como diagrama de paquetes, estructuras, perfiles, entre otros.

Diagramas de comportamiento

Los diagramas de comportamiento se centran en cómo se comporta el sistema durante su ejecución. Estos diagramas son útiles para modelar las interacciones y los procesos dinámicos del sistema.

Estos diagramas son herramientas fundamentales para modelar y documentar los procesos dinámicos de un sistema. A diferencia de los diagramas estructurales, que se centran en la organización estática de los componentes, los diagramas de comportamiento capturan cómo interactúan los elementos del sistema a lo largo del tiempo. Estos diagramas permiten visualizar aspectos como las interacciones entre objetos, las secuencias de eventos y los flujos de trabajo, lo que resulta esencial para comprender el funcionamiento interno de una aplicación. Al proporcionar una representación clara de las acciones y reacciones dentro del sistema, los diagramas de comportamiento facilitan la identificación de requisitos y ayudan a los desarrolladores a planificar y diseñar soluciones efectivas antes de iniciar la programación. Entre los diagramas mas representativo están:

Diagrama de casos de uso

Los diagramas de casos de uso son herramientas visuales que representan las interacciones entre actores externos, como usuarios o sistemas, y un sistema específico. Estos diagramas muestran las diferentes funcionalidades que el sistema ofrece, describiendo acciones que los actores pueden realizar para alcanzar objetivos concretos.

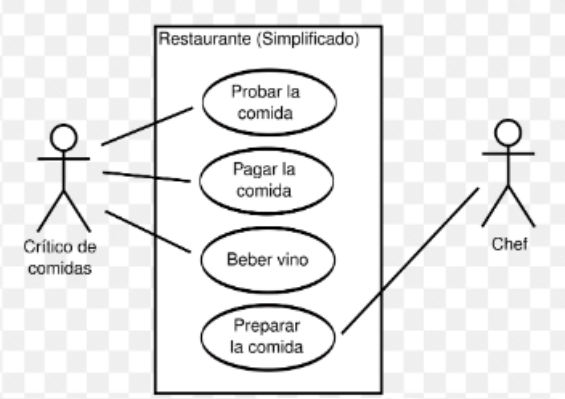
En estos diagramas, los actores se representan con figuras de personas o íconos, mientras que los casos de uso se dibujan como elipses que indican las funcionalidades del sistema, conectadas a los actores mediante líneas. El sistema en sí se representa como un rectángulo que engloba todos los casos de uso.

Los diagramas de casos de uso son esenciales para capturar los requisitos funcionales de un sistema, ayudando a diseñadores y desarrolladores a entender las funcionalidades necesarias y cómo los usuarios interactúan con el sistema. Además, sirven como un puente de comunicación entre equipos técnicos y no técnicos, facilitando la comprensión de las funcionalidades clave del sistema. Este diagrama representa las interacciones entre los usuarios (actores) y el sistema.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Define las funcionalidades que el sistema debe ofrecer desde la perspectiva del usuario final. Este diagrama es fundamental para capturar los requisitos funcionales.

Ilustración 7:
Diagrama de casos de usos



***Nota**: En la ilustración se puede observar un ejemplo del diagrama de casos de usi de un restaurante cuyos actores son el chef y el cliente*

Diagrama de secuencia

Los diagramas de secuencias son herramientas clave para modelar el flujo de mensajes y la interacción entre objetos en un sistema durante un proceso específico. Estos diagramas ayudan a comprender cómo los diferentes componentes colaboran a lo largo del tiempo para realizar una tarea.

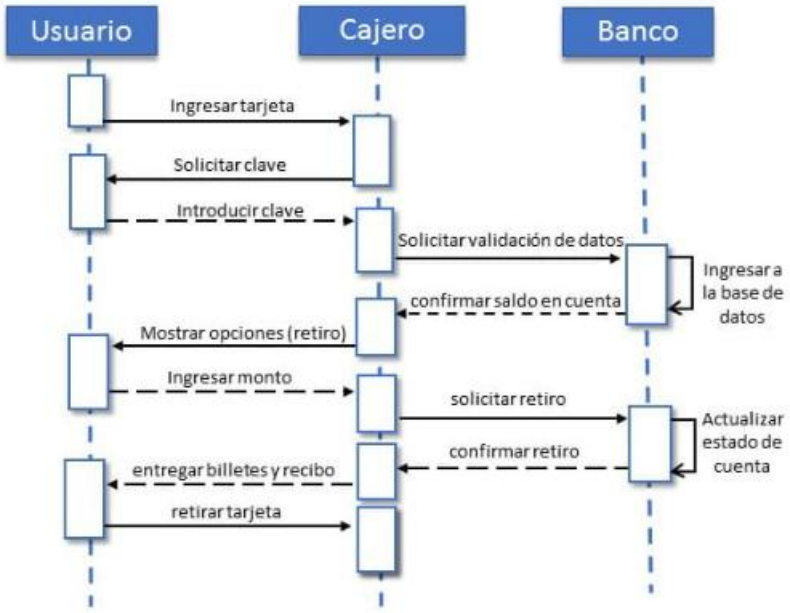
Cada diagrama de secuencia presenta objetos o actores alineados horizontalmente en la parte superior, mientras que el eje vertical representa el tiempo. A medida que se desciende en el diagrama, se muestra el orden en que se envían y reciben los mensajes entre los objetos. Las líneas de vida indican la duración de la existencia de un objeto durante la interacción, y los mensajes se representan como flechas horizontales que muestran la comunicación entre ellos.

Estos diagramas son fundamentales para visualizar la ejecución de un proceso dentro del sistema, proporcionando claridad sobre la secuencia de interacciones. Son especialmente útiles en la fase de diseño, ya que ayudan a definir la lógica detrás de funcionalidades específicas y a entender cómo se comunican los componentes del sistema.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Este diagrama describe el orden cronológico de los mensajes entre objetos o componentes del sistema. Es útil para modelar cómo los elementos del sistema interactúan a lo largo del tiempo, mostrando las llamadas a métodos o intercambios de datos.

Ilustración 8:
Diagrama de secuencia



***Nota**: En la ilustración se puede observar un ejemplo del diagrama de secuencia de un sistema de retiro de dinero que prestan los bancos a través de los cajeros automáticos*

Diagrama de actividades

Los diagramas de actividades son herramientas valiosas para modelar flujos de trabajo y procesos dentro de un sistema, ya sea en software o en entornos empresariales. Estos diagramas ilustran las acciones o tareas que se llevan a cabo y el orden en que se realizan, proporcionando una visión clara de cómo se desarrollan las actividades.

Un diagrama de actividades comienza con un nodo de inicio, representado por un círculo sólido, que indica el comienzo del flujo. A partir de ahí, se representan las acciones con rectángulos redondeados, que reflejan las tareas ejecutadas. Las transiciones entre estas acciones se muestran como flechas, marcando el paso de una actividad a la siguiente.

Estos diagramas también pueden incluir decisiones, representadas como rombos, que indican bifurcaciones en el flujo, permitiendo elegir entre diferentes caminos según ciertas condiciones.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Además, se utilizan nodos de combinación para fusionar varios flujos en uno solo, y nodos de fin, que son círculos con borde grueso, para señalar el final del proceso.

Los diagramas de actividades son especialmente útiles para describir procesos complejos, identificando puntos de decisión y la secuencia lógica de actividades. Son herramientas clave para el análisis y diseño de sistemas, facilitando la automatización de procesos y la documentación de procedimientos.

Este diagrama modela el flujo de trabajo o los procesos dentro de un sistema. Es similar a un diagrama de flujo, mostrando cómo se ejecutan las tareas y qué decisiones se toman durante el proceso.

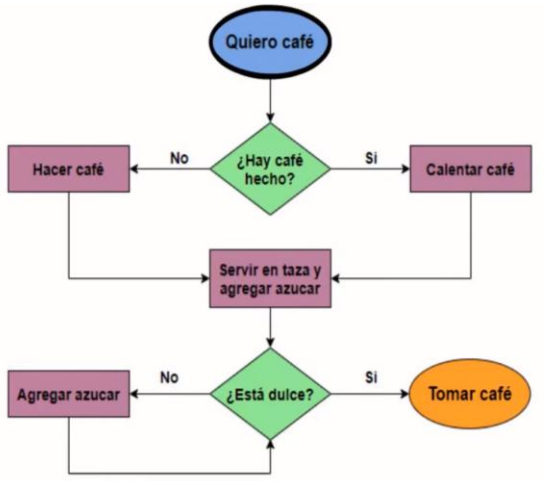
Diagrama de flujo

Los diagramas de flujo son herramientas visuales que representan el flujo de procesos o procedimientos en un sistema o negocio. Muestran los pasos secuenciales necesarios para completar una tarea, lo que ayuda a entender cómo se desarrollan las actividades y decisiones dentro de un sistema.

Estos diagramas utilizan varios símbolos estándar: los óvalos indican el inicio o el final de un proceso, los rectángulos representan actividades o tareas, los rombos señalan puntos de decisión y las flechas muestran la dirección del flujo entre las actividades.

Los diagramas de flujo son ideales para visualizar procesos de manera clara y lógica, permitiendo identificar pasos ineficientes o redundantes. Se utilizan en diversos campos, desde la ingeniería hasta la gestión empresarial, para mejorar la eficiencia y optimización de procesos.

Ilustración 9:
Diagrama de flujo



(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

***Nota**: En la ilustración se puede observar un ejemplo del diagrama de flujo sobre pasos para tomar café*

Existen algunos diagramas mas tales como diagrama de tiempo, estados, comunicación, entre otros.

Beneficios del Uso de UML en el Desarrollo de Software

El uso de UML (Lenguaje Unificado de Modelado) en el desarrollo de software presenta numerosos beneficios que mejoran la planificación y ejecución de proyectos. Al ofrecer un conjunto estándar de diagramas visuales, UML facilita una comprensión clara de la arquitectura, el comportamiento y la estructura del sistema para todos los involucrados, desde desarrolladores hasta personas no técnicas.

Esto favorece una comunicación eficiente, ayuda a detectar errores e inconsistencias de manera temprana y permite un diseño más robusto y escalable. Además, UML fomenta la reutilización de componentes y mejora la documentación del proyecto, optimizando así el mantenimiento y la evolución del software a lo largo del tiempo. Se puede citar los siguientes beneficios:

- Comunicación clara: UML proporciona una forma estándar de comunicar la estructura y el comportamiento del sistema a todos los involucrados, sin necesidad de un lenguaje de programación específico.
- Visualización y planificación: Los diagramas UML permiten a los equipos de desarrollo visualizar el sistema completo antes de que comience la codificación. Esto ayuda a identificar posibles problemas de diseño desde el inicio.
- Documentación completa: Los diagramas UML sirven como una excelente herramienta de documentación para mantener registros claros de cómo se diseñó e implementó el sistema. Esto facilita el mantenimiento y futuras mejoras del software.
- Facilita la colaboración: Al ser un lenguaje visual estándar, UML mejora la colaboración entre diferentes partes del equipo, como diseñadores, desarrolladores, gerentes de proyectos y clientes, permitiendo una mejor alineación de expectativas.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Metodologías de desarrollo

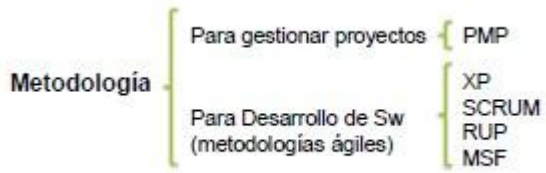
Las metodologías de desarrollo de software son enfoques estructurados que guían la planificación, diseño, implementación, y mantenimiento de proyectos de software. Estas metodologías establecen procesos y prácticas para asegurar que el desarrollo se lleve a cabo de manera organizada, eficiente y de acuerdo a los requisitos del cliente. Existen diversos enfoques, desde los tradicionales como el modelo en cascada, que sigue un enfoque secuencial, hasta los métodos ágiles, que priorizan la flexibilidad, la iteración continua y la colaboración cercana con el cliente. Elegir la metodología adecuada depende del contexto del proyecto, los requisitos del negocio y el equipo de desarrollo.

Se puede decir que la metodología es la forma de hacer las cosas, que se alimentan de buenas prácticas, para aumentar las posibilidades de éxito de un proyecto.

Características de las metodologías de desarrollo

Las metodologías de desarrollo de software se caracterizan por definir un conjunto de prácticas y principios que guían el ciclo de vida de un proyecto, desde su concepción hasta su entrega y mantenimiento. Cada metodología establece su propio enfoque para gestionar tareas, recursos, tiempos y calidad, permitiendo adaptar el proceso de desarrollo a las necesidades específicas de cada proyecto. Entre las características más comunes se encuentran la planificación estructurada, la asignación de roles y responsabilidades, la evaluación continua del progreso y la integración de retroalimentación. Algunas metodologías, como las ágiles, enfatizan la flexibilidad y la colaboración, mientras que otras, como el modelo en cascada, se centran en procesos más secuenciales y rígidos. Estas características permiten a los equipos de desarrollo optimizar sus procesos y garantizar la entrega de productos de software de alta calidad.

Ilustración 10:
Metodologías ágiles



Nota: *En la ilustración se puede observar las diferentes metodologías en la cual se puede trabajar en un sistema.*

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Modelos de procesos prescriptivos

La historia indica que estos modelos han dado cierta estructura útil al trabajo de ingeniería de software y que constituyen un mapa razonablemente eficaz para los equipos de software. Sin embargo, el trabajo de ingeniería de software y el producto que genera siguen al borde del caos. Dentro de estos tenemos:

- Modelos de cascada o secuencial
- Modelos de proceso incremental
- Modelos de proceso evolutivo

Modelo de cascada o secuencial

Este modelo es conocido también como ciclo de vida clásico, su enfoque es sistemático y secuencial, para el desarrollo de software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue, para concluir con el apoyo del software terminado. No se pueden cambiar los requerimientos, para desarrollo predictivo.

Ilustración 11:
Modelo de cascada

Nota: *en la ilustración se puede observar los procesos del modelo en cascada y como es su funcionamiento.*

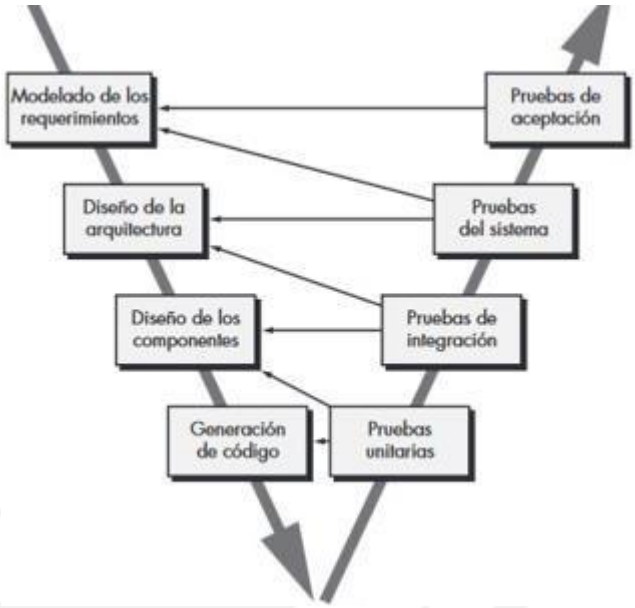
Una variante de la representación del modelo de la cascada se denomina modelo en V.

A medida que el equipo del software avanza hacia abajo desde el lado izquierdo de la V, los requerimientos básicos del problema mejoran hacia representaciones técnicas cada vez más detalladas del problema y de su solución.

A la izquierda serie de pruebas para que validen cada uno de los modelos creados.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ilustración 12:
Modelo en V



Nota: *en la ilustración se puede observar el modelo del proceso en V y su funcionamiento.*

Entre las desventajas de este modelo se puede citar que es raro que los proyectos reales sigan el flujo secuencial propuesto por el modelo. Aunque el modelo lineal acepta repeticiones, lo hace en forma indirecta.

A menudo, es difícil para el cliente enunciar en forma explícita todos los requerimientos. El modelo de la cascada necesita que se haga y tiene dificultades para aceptar la incertidumbre que existe al principio de muchos proyectos.

El cliente debe tener paciencia. No se dispondrá de una versión funcional del programa hasta que el proyecto este muy avanzado.

El modelo de cascada suele ser inapropiado para esas labores de desarrollo de software. No obstante, sirve como un modelo de proceso útil en situaciones en las que los requerimientos son fijos y el trabajo avanza en forma lineal hacia el final.

Modelo incremental

Este modelo combina elementos de los flujos de proceso lineal y paralelo, es decir, aplica secuencias lineales en forma escalonada a medida que avanzan las actividades.

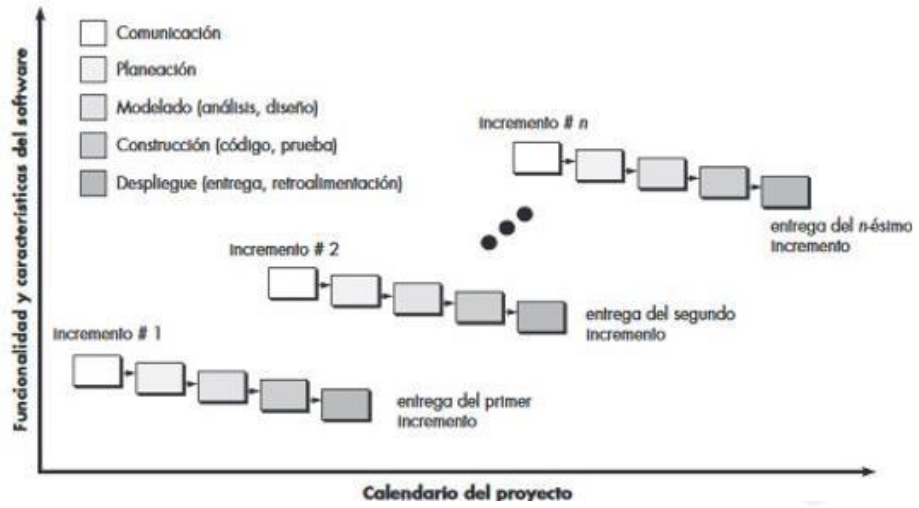
Se abordan los requerimientos básicos, pero no se proporcionan muchas características suplementarias. El cliente usa el producto sometiéndolo a una evaluación detallada y en función a esto se desarrolla un plan para el incremento que sigue.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Para cualquier incremento puede incorporar el paradigma del prototipo. Se centra en que en cada incremento se entrega un producto que ya opera. Trata de dividir al producto en partes al procedimiento global. Para requerimientos bastantes claros, permite dividir al sistema en módulos.

- Incremento → producto
- Iteraciones → proyectos pequeños funcionales (módulos)
- Es útil en particular cuando no se dispone de personal para la implementación completa del proyecto en el plazo establecido por el negocio.
- Los incrementos se planean para administrar riesgos técnicos. Por ejemplo, un sistema grande tal vez requiera que se disponga de hardware nuevo que se encuentre en desarrollo y cuya fecha de entrega sea incierta.

Ilustración 13:
Modelo del proceso incremental



Nota: *en la ilustración se puede observar el funcionamiento del proceso incremental al momento de utilizar en la realización de un sistema.*

Fases del proceso incremental

- *Requerimientos:* objetivos centrales y específicos que persigue el proyecto.
- *Definición de las tareas y las iteraciones:* lista de tareas agrupadas en iteraciones que tendrá el proyecto. Esta agrupación no puede ser aleatoria, cada una debe perseguir objetivos específicos.
- *Diseño de los incrementos:* establecidas las iteraciones, es preciso definir cuál será la evolución del producto en cada una de ellas.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- *Desarrollo del incremento:* posteriormente se realizan las tareas previstas y se desarrollan los incrementos establecidos en la etapa anterior.
- *Validación de incrementos:* al termino de cada iteración, los responsables de la gestión del proyecto deben dar por buenos los incrementos que cada una de ellas ha arrojado. Si no son los esperados o si ha habido algún retroceso, es necesario volver la vista atrás y buscar las causas de ello.
- *Integración de incrementos:* una vez son validados, los incrementos dan forma a lo que se denomina línea incremental o evolución del proyecto en su conjunto. Cada incremento ha contribuido al resultado final.
- *Entrega del producto:* cuando el producto en su conjunto ha sido validado y se confirma su correspondencia con los objetivos iniciales.

Entre las desventajas de esta metodología están:

- Es difícil evaluar el costo total.
- Difícil de aplicar a los sistemas transaccionales que atienden a ser integrados y a operar como un todo.
- Los errores en requisitos pueden detectarse tarde.
- No recomendable para sistemas en tiempo real, procesamiento distribuido o alto índice de riesgos.
- Requiere mucha planeación tanto administrativa como técnica.
- Requiere metas claras para conocer el estado del proyecto.

Modelos de proceso evolutivo

El modelo de proceso evolutivo fue diseñado explícitamente para adaptarse a un producto que evoluciona con el tiempo.

Los modelos evolutivos son iterativos y se caracterizan por la manera en la que permiten desarrollar versiones cada vez más completas del software.

Los modelos más comunes del proceso evolutivo son:

- Modelo de Prototipado
- Modelo Espiral

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

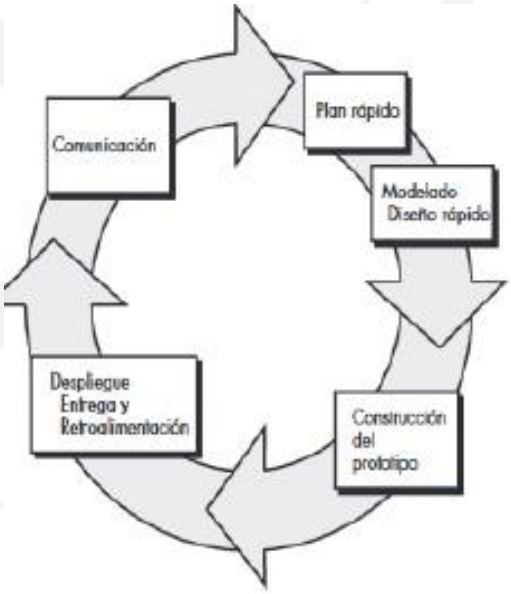
Metodología de prototipado

Esta metodología se realiza cuando el cliente tiene una necesidad legítima, pero ignora los detalles, como primer paso desarrolle un prototipo.

Para cuando el cliente defina un conjunto de objetivos generales para el software, pero que no identifique los requerimientos detallados para las funciones y características.

El paradigma de hacer prototipos ayuda a los participantes a mejorar la comprensión de lo que hay que elaborar cuando los requerimientos no están claros.

Ilustración 14:
Modelo de proceso evolutivo



Nota: *en la ilustración se puede observar el funcionamiento de un software al utilizar el modelo de proceso evolutivo al crear y diseñar un sistema.*

Entre las desventajas de este modelo se puede citar:

- El usuario quiere empezar a trabajar desde el primer momento con el prototipo para solucionar su problema particular, cuando el prototipo es solo un modelo de lo que será el producto.
- Los prototipos pueden generar otro tipo de problemas si su presentación y discusión con los usuarios no es controlada: puesto que son modelo sin conclusos, los usuarios

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

suelen enfocarse en aspectos “superficiales” del prototipo que los pueden dejar inconformes luego de verlos por primera vez.

- Es posible que se pierda mucho tiempo, innecesariamente, tratando de hacer entender al usuario la finalidad real de los prototipos.
- Requiere participación activa del usuario, al menos, para evaluar el prototipo. Y mucho más involucramiento si queremos que participe en su creación.
- Falta de experiencia que tienen muchos Analistas Funcionales en programación y en actividades de diseño de interfaces de usuario.

Modelo espiral

Este modelo se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada. Tiene el potencial para hacer un desarrollo rápido de versiones cada vez más completas.

En cada paso evolutivo se marcan puntos de referencia puntuales:

- combinación de productos del trabajo y condiciones que se encuentran a lo largo de la trayectoria de la espiral Tiene dos características distintivas principales:
- Enfoque cíclico para el crecimiento incremental del grado de definición de un sistema y su implementación, mientras que disminuye su grado de riesgo.
- Conjunto de puntos de referencia de anclaje puntual para asegurar el compromiso del participante con soluciones factibles y mutuamente satisfactorias.
- El primer circuito da como resultado el desarrollo de una especificación del producto; las vueltas sucesivas se usan para desarrollar un prototipo y versiones más sofisticadas del software y cada paso por la región de planeación da como resultado ajustes en el plan del proyecto.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ilustración 15:
Modelo espiral

Nota: *en la ilustración se puede observar el modelo espiral para el desarrollo de un sistema.*

A diferencia de otros modelos del proceso que finalizan cuando se entrega el software, el modelo espiral puede adaptarse para aplicarse a lo largo de toda la vida del software de cómputo.

El modelo espiral es un enfoque realista para el desarrollo de sistemas y de software a gran escala, como el software evoluciona a medida que el proceso avanza, el desarrollador y cliente comprenden y reaccionan mejor ante los riesgos en cada nivel de evolución.

Usa prototipos como mecanismo de reducción de riesgos en cualquier etapa de la evolución del producto.

Mantiene el enfoque de escalón sistemático sugerido por el ciclo de vida clásico, pero lo incorpora en una estructura iterativa que refleja al mundo real en una forma más realista.

Entre las desventajas de este modelo se puede citar:

- Es difícil convencer a los clientes de que el enfoque evolutivo es controlable
- Demanda mucha experiencia en la evaluación del riesgo y se basa en él para llegar al éxito.
- No hay duda de que habrá problemas si algún riesgo importante no se descubre y administra.

Metodologías agiles

Las metodologías ágiles son un conjunto de enfoques para el desarrollo de software que priorizan la flexibilidad, la colaboración continua con el cliente y la capacidad de adaptarse a cambios durante el ciclo de vida del proyecto.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Las metodologías ágiles permiten incorporar cambios con rapidez en el desarrollo de software, en circunstancias cambiantes del entorno, permitiendo mejorar la calidad del producto.

Con el uso de metodologías agiles es posible alertar de forma rápida los errores o problemas que puedan suceder sea lo largo del proyecto, estas metodologías de software moderno que se derivan de las metodologías espiral e incremental.

Ilustración 16:
Metodología ágil

Nota: *En la ilustración se puede observar el ciclo de la metodología ágil, así como su funcionamiento.*

El ciclo general de las metodologías ágiles se basa en un enfoque iterativo e incremental, dividido en fases cortas y repetitivas llamadas sprints o iteraciones, que generalmente duran entre una y cuatro semanas.

Este enfoque permite una retroalimentación continua, la adaptación a los cambios de requisitos y la entrega frecuente de incrementos funcionales del producto, lo que maximiza la satisfacción del cliente y la calidad del software.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ilustración 17:
Ciclo general de la metodología ágil

Nota: *En la ilustración se puede observar el ciclo general de la metodología ágil, dando como resultado la elaboración del sistema.*

Extrem Programming XP

Esta metodología se fundamenta en la comunicación, simplicidad, retroalimentación, disciplina y respeto. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas específicas.

La comunicación, entre ingenieros de software y otros participantes para establecer las características y funciones requeridas para el software.

La simplicidad en el diseño sólo para las necesidades inmediatas, que se implemente con facilidad en forma de código. El rediseño se realiza posteriormente.

La retroalimentación, se obtiene de tres fuentes, el software implementado, el cliente y otros miembros del equipo de software. Utiliza pruebas unitarias.

Entre las características de esta metodología están:

- Adecuada para proyectos con requisitos imprecisos y muy cambiantes.
- Se propone que exista una interacción constante entre cliente y equipo de desarrollo.
- La planificación no debe ser estrictas inflexible y abierta.
- Promoviendo el trabajo en equipo.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Iteración es cortas pero funcionales.
- Los integrantes del grupo colaboran al mismo nivel.
- Programación en parejas, siguiendo estándares que puedan documentarse.
- Cuando se entrega un incremento a un cliente se realizan historias del usuario o casos de uso

Scrum

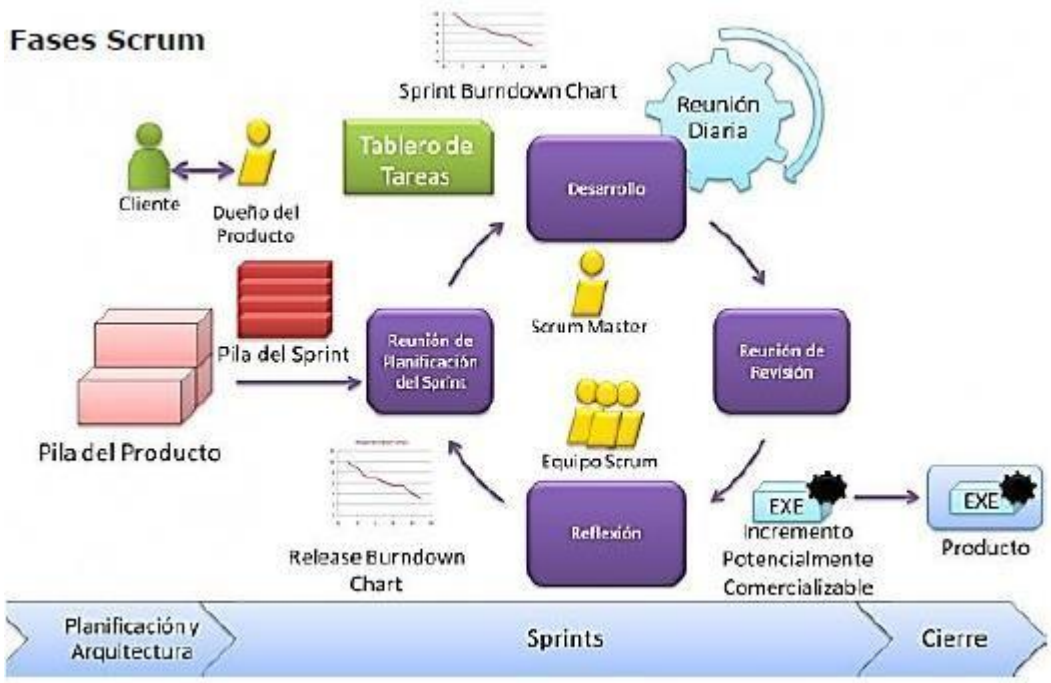
Es una de las metodologías ágiles más populares para el desarrollo de software y la gestión de proyectos. Se basa en un enfoque iterativo e incremental, dividiendo el trabajo en sprints cortos, que suelen durar entre dos y cuatro semanas. Durante cada sprint, el equipo se enfoca en un conjunto de tareas priorizadas del backlog, que es una lista de funcionalidades o requisitos del producto.

En Scrum, el equipo se organiza en tres roles clave: el Product Owner, que define las prioridades del backlog; el Scrum Master, que guía al equipo y asegura que se sigan los principios de Scrum; y el equipo de desarrollo, que se dedica a completar las tareas del sprint.

El proceso incluye reuniones regulares, como las daily standups, donde el equipo revisa su progreso, y una revisión de sprint al final de cada iteración para mostrar el trabajo completado al Product Owner y otros interesados. También se lleva a cabo una retrospectiva para identificar mejoras en el proceso.La flexibilidad de Scrum, su enfoque en la colaboración y su capacidad para adaptarse a cambios en los requisitos lo convierten en una metodología eficaz para entregar valor de manera continua y mejorar la eficiencia del equipo.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ilustración 18:
Fases de la metodología scrum



Nota: *En la ilustración se puede observar las fases de la metodología scrum por la cual pasa un proyecto de software.*

Existen algunas metodologías agiles adicionales entre las que se puede mencionar: Kanban, Lean Development, Crystal,Dynamic Systems Development Method (DSDM), Feature-Driven Development (FDD), Agile Unified Process (AUP), Adaptive Software Development (ASD), Disciplined Agile Delivery (DAD) y cada una tiene sus particularidades, pero comparten los principios ágiles como la iteración, la colaboración y la adaptabilidad, pero cada una tiene enfoques y prácticas específicas.

Ciclo de vida del desarrollo de un software

El Ciclo de Vida del Software es un proceso estructurado que define todas las etapas por las que pasa un proyecto de desarrollo de software, desde la concepción inicial hasta el retiro del producto. Comprender el ciclo de vida del software es esencial para asegurar que el sistema desarrollado cumpla con los requisitos, se mantenga dentro del presupuesto y sea entregado a tiempo.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Cada fase del ciclo de vida tiene un conjunto de actividades específicas que ayudan a gestionar el desarrollo de manera más eficiente y organizada. Vamos a explorar cada una de las fases y su importancia.

Fases del Ciclo de Vida del Software

Las fases del ciclo de vida del software son los pasos estructurados que se siguen para desarrollar, implementar y mantener un sistema o aplicación. Este ciclo abarca desde la concepción inicial del proyecto hasta su eventual retiro o actualización.

Análisis de Requisitos

Esta es la fase inicial del ciclo de vida del software, donde se recopilan y definen los requisitos del sistema. Es una etapa crucial, ya que los requisitos establecen lo que el sistema debe hacer y cómo debe comportarse. Los analistas de requisitos se encargan de recopilar información a través de entrevistas, reuniones y estudios de viabilidad.

Un análisis de requisitos bien ejecutado asegura que se construya el sistema adecuado, alineado con las necesidades del cliente. También previene problemas de ambigüedad o malentendidos en fases posteriores.

Diseño

En esta fase, se planifica la arquitectura del software. El diseño incluye tanto el diseño de alto nivel (definiendo la estructura general del sistema, sus módulos y la interacción entre ellos) como el diseño detallado (describiendo cómo cada componente individual funcionará).

El diseño actúa como un plano para los desarrolladores. Un diseño sólido facilita la implementación y garantiza que el sistema sea escalable, eficiente y fácil de mantener.

Desarrollo (Codificación)

En la fase de desarrollo, los programadores comienzan a escribir el código del software basándose en el diseño creado en la etapa anterior. Aquí es donde los desarrolladores crean el sistema real, utilizando lenguajes de programación adecuados para el proyecto.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

La fase de codificación es donde se materializa el proyecto. Es fundamental que los programadores sigan las buenas prácticas de desarrollo, como el uso de estándares de codificación, para garantizar que el software sea confiable y fácil de mantener.

Pruebas

Una vez que el código ha sido escrito, entra en la fase de pruebas. Durante esta etapa, se realiza una serie de pruebas para identificar errores, bugs y problemas de rendimiento. Las pruebas pueden incluir pruebas unitarias, pruebas de integración, pruebas de sistema y pruebas de aceptación del usuario.

Las pruebas son esenciales para asegurar que el software funcione correctamente y cumpla con los requisitos especificados. Detectar y corregir errores antes de la entrega evita problemas mayores en producción y mejora la calidad del producto final.

Implementación (Despliegue)

Después de que el software ha pasado las pruebas, llega la fase de implementación. En esta etapa, el sistema es instalado y puesto en funcionamiento en el entorno del cliente. El software puede ser desplegado gradualmente, con fases piloto o directamente a toda la organización, dependiendo del enfoque elegido.

Una implementación bien planificada asegura que el sistema se integre sin problemas en el entorno del cliente y que los usuarios puedan comenzar a utilizar el software sin interrupciones en sus operaciones diarias.

En esta fase es importante la elaboración y entrega del manual de usuario al cliente final, este manual es un documento que proporciona instrucciones detalladas sobre el uso correcto y eficiente del software o sistema. Está diseñado para guiar a los usuarios, tanto técnicos como no técnicos, en la operación de las distintas funciones y características del sistema. El manual suele incluir una descripción general del sistema, el acceso y la navegación por sus interfaces, el uso de cada módulo o funcionalidad, así como procedimientos para realizar tareas específicas. También aborda posibles problemas o errores comunes, ofreciendo soluciones o pasos para resolverlos. Un buen manual de usuario es clave para facilitar la adopción del sistema, mejorar la experiencia del usuario y reducir la necesidad de soporte técnico continuo.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Mantenimiento

Una vez que el sistema está en funcionamiento, comienza la fase de mantenimiento. Durante esta etapa, el software es monitoreado y se realizan correcciones de errores o ajustes basados en el feedback del usuario. También puede incluir la adición de nuevas funcionalidades para adaptarse a las necesidades cambiantes del negocio o del entorno.

El mantenimiento es crucial para la longevidad del software. A medida que las empresas evolucionan, el software debe adaptarse a nuevos requisitos, mejorar su seguridad y corregir errores. El mantenimiento garantiza que el sistema siga siendo útil y eficiente a lo largo del tiempo.

Retiro

Finalmente, cuando el software se vuelve obsoleto o es reemplazado por una solución más moderna, entra en la fase de retiro. Esto implica desactivar el sistema y, en algunos casos, migrar los datos a un nuevo sistema.

El retiro planificado del software asegura que no haya pérdida de datos importantes y que los sistemas nuevos puedan funcionar sin interferencias de la tecnología obsoleta.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Autoevaluación de Sistemas de Información

1. Cuáles son los tipos de sistemas de información que este integrados a los sistemas web y de comercio electrónico.
2. ¿Qué es un sistema de información y cuáles son sus componentes principales según el análisis de sistemas?
3. Escriba los pasos del ciclo de vida de desarrollo del software
4. Cual son los pasos para crear un proyecto de software
5. Cuál es el flujo de proceso que repite una o más veces las actividades antes de pasar a la siguiente
6. Escriba una metodología que permita el desarrollo de un sistema
7. Cuál es el ciclo general de las metodologías agiles
8. ¿Cuál son las actividades de un sistema de información?
9. Cual son los elementos del triángulo de calidad de un sistema
10. ¿Cuál son los procesos genéricos del software?
11. ¿Cuál es la esencia de la práctica de la ingeniería del software?
12. ¿Cómo los diagramas UML contribuyen a la documentación y visualización del diseño de sistemas de información?
13. ¿Cuáles son las fases principales de la ingeniería de requerimientos en el desarrollo de un sistema de información?
14. ¿Qué características diferencian a las metodologías ágiles de los modelos tradicionales como la cascada en el desarrollo de software?
15. ¿Cuáles son las fases del ciclo de vida del software y cuál es su importancia en el desarrollo de sistemas?
16. ¿Cómo la ingeniería de requisitos asegura el éxito de un proyecto de software durante el ciclo de vida del desarrollo?

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Resumen de la Unidad 1

En esta unidad se explora los fundamentos del análisis de sistemas de información, subrayando su importancia para las organizaciones en la era digital. Un sistema de información se compone de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen datos, transformándolos en información útil para la toma de decisiones, la coordinación y el control. Estos sistemas son esenciales para optimizar la eficiencia operativa, apoyar decisiones estratégicas, mejorar la comunicación interna y externa, y ofrecer una ventaja competitiva al permitir respuestas rápidas a los cambios del mercado. El análisis de sistemas de información se centra en entender cómo fluye la información dentro de una organización y cómo se puede mejorar su manejo mediante soluciones tecnológicas. Se destacan las etapas clave de estos sistemas: entrada, donde se recopilan los datos; proceso, donde se transforma la información; y salida, donde se presenta la información útil a los usuarios finales. Los sistemas de información pueden ser sistemas de procesamiento de transacciones, sistemas de apoyo a decisiones y sistemas de inteligencia artificial, que ayudan a gestionar grandes volúmenes de datos y resolver problemas complejos. Además se menciona diversas metodologías de desarrollo de software, como el modelo en cascada, el modelo V, espiral, el modelo incremental y las metodologías ágiles. Estas metodologías proporcionan un marco estructurado para planificar, diseñar, implementar y mantener proyectos de software, asegurando que los productos sean eficientes y cumplan con los requisitos del cliente. En particular, las metodologías ágiles como Scrum y Extreme Programming (XP) se destacan por su flexibilidad y capacidad de adaptarse a cambios en los requisitos, promoviendo la entrega continua de valor y la colaboración cercana con el cliente.

También se aborda la ingeniería de requerimientos, enfatizando la necesidad de identificar, analizar y validar los requisitos del sistema para garantizar que el software cumpla con las expectativas y necesidades del cliente. Finalmente, el ciclo de vida del software incluye fases como el análisis de requisitos, diseño, desarrollo, pruebas, implementación y mantenimiento, todas fundamentales para asegurar el éxito y la sostenibilidad de los sistemas de información.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

EJE TEMÁTICO II

GUIA GENERAL DE CALIDAD DE SOFTWARE

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

GUÍA GENERAL DE CALIDAD DE SOFTWARE

1. DESCRIPCIÓN DE LA ASIGNATURA

La materia de Calidad de Software ofrece a los estudiantes una comprensión integral de los principios fundamentales para asegurar la calidad de los productos de software desarrollados. Se cubren conceptos básicos de calidad de software, modelos y estándares para su evaluación, así como métricas y técnicas de medición. Se profundiza en la importancia de las pruebas de software y se presentan diversas técnicas y tipos de pruebas. Al finalizar la asignatura, los estudiantes habrán adquirido los conocimientos y herramientas necesarios para garantizar la calidad de los productos de software que desarrollen, preparándolos para aplicar estos conceptos de manera efectiva en su carrera profesional.

2. BIBLIOGRAFÍA

La bibliografía básica debe contener los textos que son necesarios y obligatorios para el desarrollo de la asignatura, y que cubren los contenidos y competencias principales. La bibliografía complementaria debe contener los textos que son de apoyo y consulta recomendada, pero no obligatoria, y que aportan diversidad, profundidad y actualización al tema de estudio. Se recomienda al menos 2 títulos para la bibliografía básica y al menos 4 títulos para la bibliografía complementaria.

2.1. Básica

- Pressman, R. S. (2014). Ingeniería del software: Un enfoque práctico. McGraw-Hill Interamericana.

  Este libro es considerado un clásico en el campo de la ingeniería del software. Proporciona una introducción completa y práctica a los principios, técnicas y herramientas fundamentales para el desarrollo de software de alta calidad. Aborda temas esenciales como la gestión de proyectos, el diseño de software, la ingeniería de requisitos y las pruebas de software, todos ellos cruciales para comprender y aplicar los conceptos de calidad de software de manera efectiva. La presentación clara y estructurada del contenido facilita la comprensión y el aprendizaje autónomo.

- Sommerville, I. (2016). Ingeniería de software (10ma edición). Pearson Educación.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Esta obra es otra referencia ampliamente utilizada en la enseñanza de la ingeniería de software. Ofrece una visión integral de los procesos, métodos y técnicas involucrados en el desarrollo de software de calidad. Destaca por su enfoque práctico, casos de estudio realistas y ejemplos aplicados que ayudan a los estudiantes a entender los conceptos teóricos y su aplicación en situaciones reales. Cubre temas relevantes como la gestión de la calidad del software, la mejora continua y las prácticas ágiles, proporcionando una base sólida para comprender los aspectos clave de la calidad en el desarrollo de software.

2.2. Complementaria

- Pfleeger, S. L., & Atlee, J. M. (2015). Ingeniería de software: Teoría y práctica. Pearson Educación.

  Este libro complementario ofrece una perspectiva más teórica y profunda sobre la ingeniería de software. Explora los fundamentos teóricos detrás de los procesos de desarrollo de software y examina en detalle conceptos como la calidad del producto, la gestión de la configuración y la seguridad del software. Su enfoque equilibrado entre la teoría y la práctica lo hace útil para estudiantes que desean profundizar en los principios subyacentes de la calidad de software.

- Sommerville, I., & Calvo-Manzano, J. A. (2011). Calidad del producto software. Pearson Educación.

  Esta obra se centra específicamente en el aspecto de la calidad del producto software. Analiza los criterios de calidad, los modelos de calidad y las técnicas de evaluación que son esenciales para garantizar la entrega de software de alta calidad. A través de ejemplos concretos y estudios de casos, ayuda a los estudiantes a comprender cómo aplicar métodos y herramientas para medir, evaluar y mejorar la calidad del software durante todo el ciclo de vida del desarrollo.

- Beizer, B. (1990). Software Testing Techniques (2da edición). International Thomson Computer Press.

  Este libro se enfoca en las técnicas de prueba de software, un aspecto crucial para asegurar la calidad del producto final. Proporciona una amplia cobertura de diferentes técnicas de prueba, desde pruebas unitarias hasta pruebas de sistema, y discute estrategias para diseñar y ejecutar pruebas efectivas. Es una lectura fundamental para

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

estudiantes que desean comprender en profundidad cómo realizar pruebas exhaustivas y garantizar la calidad del software mediante la detección y corrección temprana de errores.

- IEEE Computer Society. (2014). IEEE Standard for Software Quality Assurance Processes. IEEE Press.

  Este estándar de calidad de software establece las prácticas recomendadas para el aseguramiento de la calidad del software en proyectos de desarrollo. Proporciona directrices detalladas sobre la planificación, implementación y seguimiento de procesos de aseguramiento de calidad, ayudando a los estudiantes a entender cómo establecer un marco de trabajo sólido para garantizar la calidad del software en cualquier contexto de desarrollo. Su adopción facilita la estandarización y mejora continua de los procesos de calidad en la industria del software.

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS

4. Valores & habilidades blandas: cultura: creatividad
5. valores & habilidades blandas: justicia: resolución de problemas
6. valores & habilidades blandas: lealtad: liderazgo
7. valores & habilidades blandas: optimismo: planificación y gestión del tiempo

8. OBJETIVO GENERAL

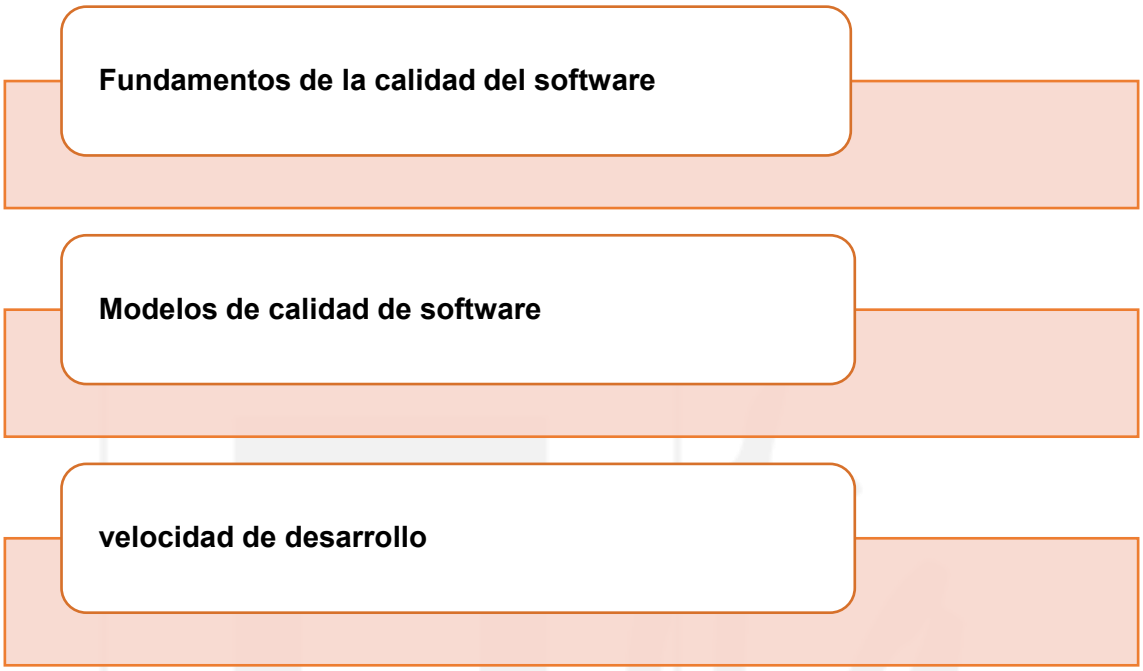
El objetivo debe incluir los ámbitos cognitivo, procedimental (habilidades de pensamiento y destrezas sensoriales y motoras) y actitudinal.

Desarrollar competencias cognitivas, procedimentales y actitudinales que permitan a los estudiantes comprender, aplicar y evaluar los principios, metodologías y técnicas relacionadas con la calidad de software, con el fin de mejorar la eficacia y eficiencia del proceso de desarrollo de software, promover la entrega de productos software de alta calidad y fomentar una cultura de mejora continua en el ámbito de la ingeniería de software.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

UNIDAD 1: FUNDAMENTOS DE CALIDAD DE SOFTWARE

Temas y Subtemas



Fundamentos de calidad de software

Conjunto de propiedades y de características de un producto o servicio que le confieren aptitud para satisfacer necesidades explicitas o implícitas

SOFTWARE

El término software es un vocablo ingles que fue tomado por otros idiomas y designa a todo componente intangible (no físico) que forma parte de dispositivos como computadoras, teléfonos móviles o tabletas y que permiten su funcionamiento.

El software este compuesto por un conjunto de aplicaciones y programas diseñados para cumplir diversas funciones dentro de un sistema. Además, está formado por la información del usuario y los datos procesados.

Los programas que forman parte de software le indican al hardware (parte física de un dispositivo), por medio de instrucciones, los pasos a seguir.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Calidad de Software

Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares del desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario.

Factores que determinan la calidad de Software

Según MacCall los factores que determinan la calidad del software se dividen en tres aspectos importantes:

- Características Operativas
- Capacidad de soportar los cambios
- Adaptabilidad a nuevos entornos

CARACTERÍSTICAS OPERATIVAS

- Corrección. ¿Hace lo que quiere?
- Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?
- Eficiencia. ¿Se ejecutará en mi hardware lo mejor que pueda?
- Seguridad (Integridad). ¿Es seguro?
- Facilidad de uso. ¿Está diseñado para ser usado?

CAPACIDAD DE SOPORTAR LOS CAMBIOS

- Facilidad de mantenimiento. ¿Puedo corregirlo?
- Flexibilidad. ¿Puedo cambiarlo?
- Facilidad de prueba. ¿Puedo probarlo?

ADAPTABILIDAD A NUEVOS ENTORNOS

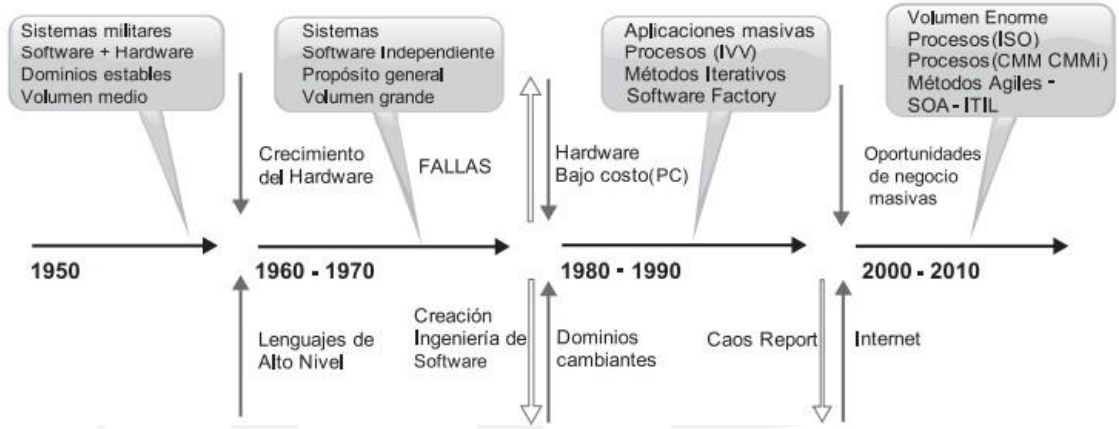
- Portabilidad. ¿Poder usarlo en otra maquina?
- Reusabilidad. ¿Poder reutilizar alguna parte del software?
- Interoperabilidad. ¿Poder hacerlo interactuar con otro sistema?

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Evolución histórica

En esta presentación se utilizará un modelo simplificado como se muestra en la ilustración. Se hablará de los eventos e hitos más importantes que constituyeron cambios en el proceso evolutivo que nos ocupa en esta sección.

Ilustración 19:
Evolución histórica de la calidad de software



Nota: *en la ilustración se puede observar una línea de tiempo sobre la calidad de software.*

Calidad vs velocidad de desarrollo

ESTÁNDARES

Para que los proyectos de desarrollo fueran predecibles se necesitó establecer algunos estándares para diferenciar tareas que se realizaban a lo largo del ciclo de vida del software. Por lo tanto, cualquier cosa que pudiera reutilizarse era predecible y se tomaba como un factor de valor para reducir costos. De esta manera se generaron activos y procedimientos para su construcción en disciplinas como el trabajo con requerimientos, diseño y código. Estos estándares fueron propuestos como una manera de garantizar la calidad de procesos y productos.

CREATIVIDAD

Con los procedimientos y activos estándares agrupados en nuevas metodologías y modelos de referencia se comenzó a transitar por la etapa ya descrita que se dio a partir de la instalación de Internet como plataforma de comunicación y oferta de consumo de servicios. Por ese motivo los tiempos de salida al mercado y de estabilidad de los requerimientos se acortaron de una manera impensada. Por esta razón se comenzó a cuestionar los estándares, ya que estos proponían una forma de trabajo cuya estabilidad contrastaba con la dinámica de la nueva etapa. Así fue que

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

nacieron las metodologías ágiles (Mary and Tom Poppendiek), a partir de que la comunidad comenzó a ponderar la creatividad y velocidad de adaptación a los nuevos escenarios por sobre el control y la predictibilidad en los proyectos. Este fenómeno de estabilidad de procesos versus creatividad ya se había percibido décadas atrás entre el software factory de Japón y la industria del software norteamericano, cuando se analizó la calidad de excelencia de los productos orientales a expensas de la ausencia de nuevos productos, los cuales eran ofrecidos por las empresas de EE. UU, que trabajan en un ambiente mucho menos controlado (Michael Cusumano).

MADUREZ

Como muchas veces a lo largo de la historia de esta industria. Hace algunos años se instaló una idea de moda. Esta vez consiste en pensar que cuanto menos estructuradas sean las empresas, mejor estarán posicionadas para enfrentar cambios y ganar nuevos negocios. Sin embargo, como Cosummano muestra en su libro, también están en condiciones mucho más vulnerables para perder todo lo realizado en solo unos meses. La razón de esto la aplica W. Demming en su frase acerca de cuál es el objetivo de las empresas. Actualmente, el desafío con respecto a garantizar la calidad de las organizaciones dedicadas al desarrollo de software es organizarse para que sus proyectos sean ordenados y predecibles: aunque deben a su vez tener la capacidad de dejar de lado estos procesos rápidamente para reorganizarse adaptándose a los cambios. Para esto es muy importante lograr madurez en estos procesos de manera de seguirlos cómodamente para que no se constituyan en una carga que aumente los costos e impida contar con agilidad para cambiar.

Modelos de calidad del software

SURGIMIENTO Y EVOLUCIÓN

Como mencionamos en la sección de historia, con el objetivo de establecer formas estándares en las prácticas y activos de desarrollo aparecieron a lo largo de los años distintas normas y modelos de referencia. Los llamamos así porque constituyen la referencia al momento de implementar una mejora de procesos en una organización con el objetivo de aumentar la calidad de procesos y productos generados. Un fin para el cual se ha utilizado estos modelos ha sido la calificación, clasificación y comparación de empresas por parte de los compradores de productos de software. El ejemplo más notorio, ya citado, es el del DoD y el modelo CMM, y después CMMi. Básicamente, con ellos se buscó establecer estándares de organizaciones. La utilización inteligente de estos modelos contribuyó a mejorar los productos de software, mientras que el mal uso generó frustraciones y despilfarro de recursos. En la década de 1990 el modelo CMM (Capability Maturity

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Model) se convirtió en el estándar de hecho a nivel global. Más tarde, en la década de 2000, fue reemplazado por su versión mejorada CMMi (Capability Maturity Model Integration).

MODELOS

A continuación, presentamos un listado de normas y modelos citando en forma resumida las características salientes. Como el modelo CMMI que esta focalizado al desarrollo de software. Los otros modelos mencionados si bien son más abarcativas, porque incluyen procesos relacionados a aspectos administrativos y de formación de recursos humanos. El modelo ITIL (Information Technology Infrastructure Library), presenta entre sus componentes importantes el software como parte integrante de los servicios de IT (Information Technology).

CMM (CAPABILITY MATURIRY MODEL)

Organización: SEI (Software Engineering Institute)

- Estructura o Niveles
- Inicial
- Repetible
- Definido
- Gestionado
- Optimizado
- Áreas de proceso

Organización: SEI (Software Engineering Institute)

- Estructura o Vistas por niveles y capacidades o Áreas de proceso o Grupos de áreas de procesos
- Ingeniería (7 áreas de procesos)
- Proyectos (5 áreas de procesos)
- Organizativas (6 áreas de procesos)

ISO/IEC 12207 Information Technology / Software Life Cycle Processes, 1995-2008

Organización: ISO (International Standard Organization)

- Estructura
- Procesos principales

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Adquisición
- Suministro
- Desarrollo
- Operación
- Mantenimiento o Procesos de soporte
- Documentación
- Gestión de la configuración
- Aseguramiento de calidad
- Verificación
- Validación
- Revisión conjunta
- Auditoría
- Revisión conjunta
- Auditoría
- Resolución de problemas
- Procesos de la organización
- Gestión
- Infraestructura
- Mejora
- Recursos humanos

ÁREAS DE APLICACIÓN O ADMINISTRACIÓN DE SERVICIOS

- Service Support
- Service Delivery o Guías Operacionales
- ICT Infrastructure Management
- Security Management
- The Business Perspective

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Application Management
- Software Asset Management
- Guías de implementación
- Planning to Implement Service Management
- ITIL Small-Scale Implementation

Con el tiempo y por diferentes razones surgieron otros modelos inspirados en los anteriores que buscaron cubrir necesidades puntuales. Como por ejemplo el MOPROSOFT (Modelo de Procesos para la Industria del Software), concebido en México, con el objetivo de ser aplicado a las pequeñas y medianas empresas dedicadas al desarrollo de software. Está inspirado en los niveles 2 y 3 del modelo CMM y en las normas ISO/IEC 15504.

Algunos de los criterios con los que fue elaborado este modelo son los siguientes:

- Procesos estratificados según el organigrama típico de este tipo de organizaciones (gerencia alta, gerencia media y operación).
- La gerencia alta tiene como actividades centrales la de definir las estrategias de la organización y promover las mejoras continuas.
- La gerencia media es la encargada de proveer los recursos para los proyectos y monitorear el cumplimiento de las planificaciones orientadas a cumplir con los objetivos estratégicos.
- La operación es la encargada de llevar adelante los proyectos.
- Los procesos deben ser definidos de tal forma que mantengan entre ellos una relación integradora.
- El proceso de administración de proyectos es atómico.
- La ingeniería de productos es desarrollada con el soporte de otros procesos orientados a garantizar y controlar la calidad de los mismos (verificación, validación, documentación y control de la documentación).

Se le asigna especial atención a la administración de los recursos que aportan al conocimiento de la organización: productos generados por proyectos, mediciones, documentación de procesos y datos relevados de su implementación en los proyectos, así como lecciones aprendidas

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Que son Normas ISO

ISO es la "Organización Internacional para la Normalizacion" (International Standard Organization) cuyas oficinas generales se encuentran en Ginebra Suiza. Dicha Organización se funda en 1946.

¿QUÉ SIGNIFICA ISO?

ISO se toma del vocablo griego "ISOS" y significa "igual". ISOS es raíz del prefijo ISOS y aparece en palabras como:

ISOMETRICO = Dimensiones iguales

ISOSELES = Lados iguales

¿CUÁL ES SU PROPÓSITO?

Su propósito es desarrollar y promover Normas de uso común entre países a nivel mundial. Este trabajo se realiza a través de ciertos comités técnicos y miles de subcomités y equipos de trabajo

¿CÓMO SE CREÓ LA SERIE ISO 9000?

En 1980 se formó el Comité Técnico ISO-TC – 176 para asuntos de sistemas de calidad (65 delegados de 25 países) y fue el autor de la serie ISO 9000.

La primera publicación de la norma fue en el año de 1987

La edición con revisión 1 fue en septiembre de 1994

La edición con revisión 2 fue en diciembre de 2000

¿SON ESTAS NORMAS INAMOVIBLES?

"NO", estas normas son revisadas cada 5 años. En 1992 se inicia la primera revisión y con la finalidad de contar con una mejor opinión de los diferentes países que participan en el desarrollo en implantación de esta Norma, se prolonga el tiempo y fue hasta septiembre de 1994 cuando se publicó.

De la misma manera, con la versión 2000 pasan 6 años para su revisión y publicación. La última revisión fue 8 años mas tarde.

¿QUE ES LA SERIE ISO 9000?

Es una familia de normas desarrolladas por el Comité Técnico en Calidad de ISO, para normalizar a nivel internacional todos los aspectos relacionados con la gestión y el aseguramiento de la calidad. Esta serie esta compuesta por tres normas:

ISO 9000: SGC. Fundamentos y Vocabulario

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

ISO 9001: SGC. Requerimientos

ISO 9004: SGC. Directrices para la mejora del desempeño

ISO 9126: Evaluación y mejora de la calidad del software en el desarrollo y mantenimiento

ISO 25000: Evaluación y mejora de la calidad del software en el ciclo de vida del desarrollo de software.

ISO 27001: Implementación de sistemas de gestión de la seguridad de la información en organizaciones de cualquier tipo y tamaño.

IEEE 730

Estándar para la Gestión de la Calidad del Software

Características:

- Proporciona un marco para la gestión de la calidad del software.
- Incluye actividades de planificación, aseguramiento de la calidad, evaluación y mejora.

Aplicación:

Gestión de la calidad del software en proyectos de desarrollo y mantenimiento.

¿Qué es una métrica en normas ISO?

En el contexto de las normas ISO, una métrica es una medida cuantitativa o cualitativa que se utiliza para evaluar el desempeño, la calidad o el cumplimiento de un proceso, producto o servicio. Las métricas proporcionan datos objetivos que permiten:

- Monitorear el progreso: Las métricas permiten realizar un seguimiento del desempeño a lo largo del tiempo y detectar tendencias.
- Evaluar la eficacia: Las métricas ayudan a determinar si un proceso o sistema está logrando sus objetivos.
- Identificar áreas de mejora: Las métricas señalan áreas donde se pueden realizar mejoras para optimizar el desempeño.
- Tomar decisiones informadas: Las métricas proporcionan datos objetivos que respaldan la toma de decisiones.

Ejemplo de una métrica en normas ISO:

En la norma ISO 9001 (Sistemas de gestión de la calidad), se podría utilizar la métrica del "porcentaje de satisfacción del cliente" para evaluar la calidad del servicio al cliente. Esta métrica

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

se calcularía mediante encuestas o retroalimentación de los clientes, y permitiría a la organización monitorear y mejorar continuamente la satisfacción del cliente.

Normas ISO relevantes:

ISO 9126 (Reemplazada por ISO 25000):

Esta norma definía un modelo de calidad para el software, estableciendo características y métricas para evaluar la calidad del software.

Fue reemplazada por la serie ISO 25000, que proporciona un marco más completo para la evaluación de la calidad del software.

ISO 25000 (Calidad del producto software y sistemas):

Esta serie de normas proporciona un marco para la evaluación de la calidad del software y los sistemas. Incluye modelos de calidad, métricas y procesos de evaluación. Su objetivo es estandarizar la evaluación de la calidad del software, facilitando la comunicación y la comparación entre diferentes productos y organizaciones.

ISO 27000 (Sistemas de gestión de la seguridad de la información): Esta familia de normas se centra en la seguridad de la información. Proporciona un marco para establecer, implementar, mantener y mejorar un sistema de gestión de la seguridad de la información (SGSI).

Resumen de la Unidad 2

En esta unidad, se exploraron los fundamentos esenciales que sustentan la calidad en el desarrollo de software. Se abordaron los principios básicos que guían la producción de software de alta calidad, incluyendo la identificación de requisitos, el diseño adecuado, la implementación robusta y las pruebas exhaustivas. Además, se examinaron diversos modelos de calidad de software, tales como el modelo de calidad ISO/IEC 25010, el modelo de madurez CMMI (Capability Maturity Model Integration) y el modelo de calidad de McCall. Estos modelos proporcionan un marco de referencia para evaluar y mejorar la calidad del software a lo largo de su ciclo de vida. También se analizó la importancia de la velocidad de desarrollo en el contexto actual de la ingeniería de software, considerando las demandas del mercado y las expectativas de los clientes. Se discutieron estrategias y prácticas ágiles que permiten acelerar el proceso de desarrollo sin comprometer la calidad del producto final. En resumen, esta unidad proporcionó una comprensión profunda de los principios, modelos y prácticas clave que influyen en la calidad y la velocidad del desarrollo de software.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

UNIDAD 2: CALIDAD DE SOFTWARE

Temas y Subtemas

- Calidad de software
- dominio del negocio
- ingenieria de software

Causas que deterioran la calidad en el software

A pesar de que las causas que afectan la calidad de software siempre están presentes, la mala calidad no es un atributo inevitable de todo software.

El software de mala calidad siempre representa riesgos.

Las causas que afectan la calidad de software son resultado de malas prácticas que aparecen dese la concepción del sistema.

El no cotar con sistemas de software con factores de calidad como alta disponibilidad, desempeño y la facilidad de adaptarse a cambios derivada en un sin número de problemas.

Un problema principal del software de mala calidad son los costos que deriva después de su implementación, muchas veces estos costos son subestimados y se desconoce el impacto que pueden llegar a generar.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

La analogía es un iceberg en donde en la superficie aparecen los costos visibles a corto plazo, sin embargo, los que terminan causando más daño son los ocultos.

Ilustración 20:
Factores que deterioran la calidad de software

Nota: *en la ilustración se puede observar la mala calidad del software y los factores que deterioran la calidad de software*

Decimos que las actividades, recursos y personas requeridas para mantener la operación del software una vez implementado, son designadas al proceso conocido como mantenimiento de Software.

Las 5 principales causas que conllevan a una mala calidad de software y que serán recurrentes año tras año son:

FALTA DE DOMINIO DEL NEGOCIO

En la mayoría de os proyectos los desarrolladores en un principio no son expertos en los conceptos y temas propios del negocio, para el cual se está desarrollando el software.

Con el tiempo ellos logran conocer mucho sobre el negocio y se llegan a convertir en unos verdaderos expertos.

Sin embargo, muchos de estos desconocimientos al inicio se traducen en un buen número de defectos introducidos al sistema por reglas requerimientos funcionales malentendidos.

Una solución es introducir a expertos del negocio al inicio del proyecto, que den orientación a analistas y desarrolladores.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Los analistas deberán trabajar en documentar el entendimiento a través de metodologías, diagramas y notaciones estándares que faciliten su validación con los usuarios de negocios.

Ejemplos de estos diagramas y notación son UML y BPMN.

Posteriormente se recomienda conducir revisiones con los expertos de negocios para verificar que la documentación es correcta.

DESCONOCIMIENTO DE LA TECNOLOGÍA

La mayoría de los desarrolladores son conocedores de varios lenguajes y tecnologías informáticas.

Sin embargo, las aplicaciones empresariales actuales de múltiples capas son un enredo complejo de micos lenguajes y plataformas de software.

Estos niveles incluyen la interfaz de usuario, la lógica empresarial y la gestión de datos, y pueden interactuar a través de middleware con sistemas de recursos empresariales y aplicaciones heredadas escritas en lenguajes arcaicos.

Pocos desarrolladores conocen todos estos lenguajes y tecnologías, y tiene suposiciones incorrectas sobre cómo funcionan otras tecnologías.

Esto llega a ser la fuente principal de los defectos no funcionales que causan interrupciones dañinas, corrupción de datos y fallas de seguridad durante la operación.

La mejor manera de mitigar esta causa es entrenar a los desarrolladores en diferentes tecnologías, realizando revisiones entre pares con otros desarrolladores que trabajen en diferentes aspectos de la aplicación.

CALENDARIOS POCOS REALISTAS

Cuando los desarrolladores se ven obligados a sacrificar buenas prácticas de desarrollo de software por planes y calendarios mal elaborados y extremadamente cortos, los resultados no son buenos.

Los pocos resultados exitosos se basan en actos heroicos que rara vez se repiten.

Al trabajar a un ritmo vertiginoso, los desarrolladores más estresados comenten más errores y tienen menos tiempo para encontrarlos.

La única manera de mitigar esto es a través de la aplicación de fuerte prácticas de gestión de proyectos.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Controlar los compromisos a través de la planificación y el seguimiento para identificar problemas, así como el control de los cambios en los requerimientos son prácticas críticas para proporcionar un entorno profesional para el desarrollo de software.

NO IMPLEMENTAR INGENIERÍA DE SOFTWARE

La mayoría de las actividades de desarrollo de software implican el cambio o la mejora del código.

Estudios demuestran que la mitad del tiempo dedicado a modificar software se gasta comprendiendo la lógica del código fuente.

El código complejo frecuentemente es difícil de entender y la modificación conduce a numerosos errores y efectos secundarios negativos imprevistos.

Estos defectos recién inyectados causan retrabajos costosos y liberaciones retardades.

La manera de mitigar esta causa es volver a partes críticas del código guiado por información de análisis de código arquitectónico y estático.

UTILIZAR MALAS O NULAS PRÁCTICAS DE DESARROLLO DE SOFTWARE

La mayoría de las aplicaciones multi-nivel grandes son construidas y mantenidas por equipos distribuidos, algunos o todos los cuales pueden ser, subcontratados de otras compañías.

En consecuencia, la organización adquiriente a menudo tiene poca visibilidad o control sobre la calidad del software que está recibiendo.

Por varias razones, los niveles del modelo CMMI no siempre han garantizado entregas de software de alta calidad.

Para mitigar los riesgos de problemas de calidad en el software suministrado externamente, los administradores deben implementar objetivos, los administradores deben implementar objetivos de calidad en sus contratos y una sólida garantía de calidad para el software entregado.

Resumen de la Unidad 2

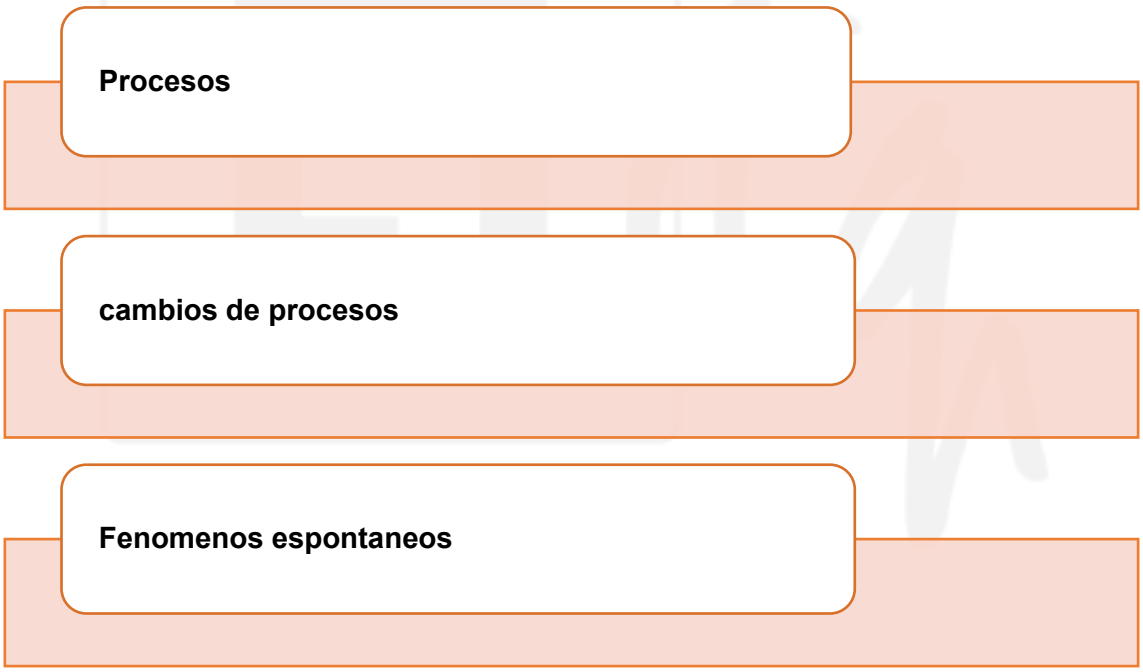
En esta unidad, se exploraron los conceptos fundamentales relacionados con la calidad de software, el dominio del negocio y la ingeniería de software. Se analizó en detalle la importancia de la calidad del software en el desarrollo de sistemas efectivos y confiables, destacando la necesidad de seguir procesos y estándares rigurosos para garantizar productos finales libres de errores. Además, se examinó la importancia de comprender el dominio del negocio en el contexto

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

del desarrollo de software, enfatizando la necesidad de colaboración estrecha entre los desarrolladores de software y los expertos en el dominio para garantizar que las soluciones propuestas satisfagan las necesidades y expectativas del cliente. Por último, se profundizó en los principios y prácticas de la ingeniería de software, abordando aspectos como la gestión de requisitos, el diseño de sistemas, la implementación de código y la realización de pruebas. En resumen, esta unidad proporcionó una visión integral de cómo la calidad de software, el dominio del negocio y la ingeniería de software se entrelazan para producir soluciones tecnológicas exitosas y satisfactorias para las organizaciones y los clientes.

UNIDAD 3: PROCESOS

Temas y Subtemas



Trabajo con la organización – mejora de procesos

VISIÓN DEL CAMBIO

En la sección anterior hemos analizado cuáles son las causas del fracaso de os proyectos de desarrollo de software. En esta sección analizaremos y haremos algunas propuestas acerca de cómo cambiar distintos aspectos dentro de una organización, con el objetivo de mejorar la

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

calidad de los productos que en ella se generan. Cualquier cambio, obviamente, será en la dirección de una mejora por lo que llamaremos a este tipo de proyectos Mejoras de Procesos. Cualquier cambio generará otros laterales que deberán gestionarse para evitar consecuencias no deseadas. Este capítulo nos explicará qué podemos esperar de los cambios realizados y cómo debemos administrar estos sucesos; cuáles son las cuestiones a tener en cuenta a la hora de planificar los cambios, quiénes podrán ayudarnos, cómo debemos comunicar todas las novedades; cómo preparar una organización para el proceso de mejoras. Tratamos estos temas primero ya que deseamos crear el contexto apropiado en el cual instalar los cambios asociados a la mejora de procesos conociendo los efectos que estos generarán y habiendo desarrollado un criterio acerca de cómo administrarlos.

PRIMEROS PASOS EN UN PROCESO DE MEJORA

Seguiremos a la teoría de las restricciones en la formulación de los pasos a dar en un proceso de mejoras. Sin embargo, antes de trabajar en dicho proceso debe tomarse la decisión de hacerlo. Para que esto suceda son variadas las causas por las cuales los responsables de las organizaciones deciden mejorar y acudir a alguien que los ayude en este proyecto. ¿Cuáles son estas causas?

- Trabajar de una mejor manera para generar un ambiente de trabajo más valorado por sus miembros y captar así a los mejores profesionales al convenir a la empresa en un lugar deseado.
- Mejorar los procesos de desarrollo para mejorar la calidad de los productos generados y hacer a la empresa más competitiva, y como consecuencia de esto ganar una Proción mayor del mercado.
- Ser beneficiado por incentivo legales instituidos por política públicas y lograr beneficios impositivos que favorezcan las finanzas de la organización

Estas son algunas de las motivaciones por las cuales una organización decidiría dejar de ser una empresa con proyectos tipo 2 o 3 y convertirse en una con proyectos de tipo 1.

- Se trata de entender al negocio del desarrollo de software como una actividad en la cual la categorización de los participantes depende de la forma de organización y trabajo.
- Los negocios a los cuales una empresa puede acceder, al momento de ser escrito este libro, están fuertemente condicionados por la calidad de sus procesos, evaluada por algún organismo y según algún modelo de referencia.
- Un mismo proyecto podrá ser llevado adelante con un menor costo cuanto mejores y más maduros sean los procesos con los que cuente la organización

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Los recursos que no se invierten en mejoras se gastan en trabajos de reparación de errores para disminuir las fallas de los sistemas

Cualquiera sea la causa del evento que dispare la realización de la mejora de procesos empezaremos trabajando de la forma que se muestra en la tabla 3. En ella aparecen las preguntas que nos haremos y el orden en que nos las haremos. Si no nos respondemos cualquiera de ellas no podremos avanzar a la próxima.

Tabla 1:
Primeros pasos del proceso de mejoras

Preguntas	Objetivos	Nivel de resistencia
¿Qué hay que cambiar?	• Evaluación de la situación, descripción de la realidad actual, identificación de los problemas principales y supuestos que lo sustentan. • Diagnóstico, análisis, asistemático	1. Falta de consenso del problema a resolver
¿Hacia dónde debe conducir el cambio? •	Bosquejar visión y solución, describir estrategia. Tácticas y buenas prácticas	1. Falta de consenso sobre la dirección que se le debe dar a la solución. 2. Falta de consenso en cuanto a que la solución planteada a. produce los b. resultados esperados. 3. Preocupación acerca de que la

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

		solución generará efectos no deseados
¿Cómo implementar el cambio?	Desarrollo de tácticas y planes de detallados. Sincronización de esfuerzos	5. Falta de una clara visión del camino y los obstáculos en el logro de la solución.
	• Planear, asignar responsabilidades y liderazgo.	6. Falta de confianza, temores y dudas acerca del plan elaborado para la solución.

Es importante saber dónde buscar las respuestas y cómo responder a estas preguntas. Para respondernos la primera debemos descubrir y listar los problemas de la organización, de los cuales surgirán los aspectos a cambiar. Luego será posible acordar estos cambios con los miembros. Los problemas no son difíciles de detectar ya que su manifestación es visible a los que los padecen y sus consecuencias afectan directamente al negocio. Un síntoma generalizado es mezclar estos con sus causas. Pero en esta primera instancia solo nos interesa determinar los problemas sin causas ni responsables. Por lo tanto, acordaremos un listado de inconvenientes que nos permitan direccionar los cambios que a continuación implementaremos. Es importante manifestar explícitamente que no se buscan responsables, para que sea posible un análisis objetivo y despersonalizado. Cuando se logre contar con el listado de problemas, analizaremos las causas y avanzaremos en un listado de alternativas orientadas a cambiar diferentes aspectos en la empresa considerados causas de estos. Es importante generar un espacio en el cual podamos tener en cuenta la opinión de los diferentes miembros de la organización. Las opiniones que por distintas razones no aporten a nuestro listado, en el futuro se constituirán en obstáculos y resistencia a la mejora de procesos. Es importante remarcar que durante este proceso los

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

mentores tendrán un comportamiento de facilitadores del trabajo de los miembros. Coordinarán las tareas interpretando los acontecimientos. Este rol será muy importante en esta etapa ya que determinar los cambios necesarios es una tarea ardua y difícil. Todo el tiempo se deben confrontar propuestas distintas y contrapuestas. Este análisis y su balance en todos los casos deben ser claramente fundamentados a partir de sus ventajas y desventajas. Una vez que se acuerden los problemas y los cambios a generar, habremos pasado las dos etapas de mayor dificultad. Esto es así porque su logro depende, como dijimos, de acuerdos entre miembros de una organización entre los cuales siempre existen intereses opuestos. A diferencia de la próxima etapa, que depende de la planificación técnica y de los recursos con que se cuenta o no para el desarrollo del proyecto asociado a la mejora de procesos. Si estas preguntas no son resueltas de una manera franca y efectiva tanto por las personas que implementarán el cambio como por las que se verán directamente implicadas, el cambio propuesto será muy difícil de lograr.

Trabajando en los cambios

FORMA DE TRABAJO

A la hora de trabajar en las mejoras, un estándar de hecho es el modelo IDEAL. Hemos participado en numerosos proyectos en los cuales se desarrollaban las tareas asociadas a las mejoras aplicando este modelo a las diferentes áreas de procesos del modelo CMMi, por ejemplo, como veremos en el capítulo siete. Con esta presentación queremos revisar y cambiar la actitud y forma de trabajo en la utilización de este modelo.

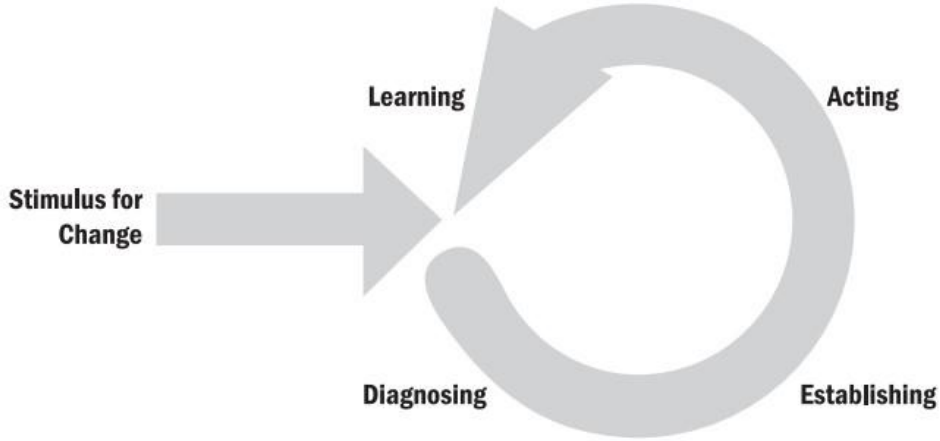
MODELO IDEAL

IDEAL: Initiating, Diagnosing, Establishing, Acting, Learning.

En la ilustración se muestra el ciclo de vida de este modelo utilizando con frecuencia en las mejoras de procesos (imagen adaptada del sitio del SEISoftware Enginnering Institute).

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ilustración 21:
Ciclo de vida del modelo IDEAL



Nota: *en la ilustración se puede observar el ciclo de vida del modelo IDEAL.*

Se trata de trabajar en forma iterativa sobre cada aspecto a mejorar, como lo indica este ciclo. Luego del gap análisis y de la planificación del proyecto se acostumbra utilizarlo como forma de organización de los planes pilotos en las áreas a cambiar. El desarrollo de estos últimos nos permite probar el funcionamiento de los cambios propuestos antes de institucionalizarlos y, por lo general, se llevan adelante en uno de los proyectos de desarrollo en marcha y es la oportunidad de validar las definiciones realizadas en la fase de definiciones y diseño de los cambios. Sin embargo, muchas veces este ciclo finaliza como se muestra en la siguiente secuencia de pasos:

1. Inicio
2. Análisis y presentación de resultados
3. Planificación
4. Diseño
5. Implementación
6. Búsqueda de los culpables del fracaso del proceso de mejoras

Con frecuencia este modelo se utiliza como un patrón en los procesos de mejora sin la participación en las fases de planificación y de diseño de todos los involucrados. Es un error que muchas organizaciones cometen. Con el tiempo la fase de Aprendizaje (Learning) se convierte en una búsqueda de los culpables del fracaso en la implementación de los cambios. La no participación de todos los involucrados se debe generalmente a que se tiene la sensación de que no es posible la participación de todos los miembros, cuando en realidad no se sabe cómo hacer

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

para que todos participen y encausar esa participación. Esta es la causa más frecuente de fracaso en la implementación de cambios y lo que causa la posterior búsqueda de los culpables.

MODELO EOALG

EOALG: Escuchar y Observar A La Gente El modelo IDEAL bien aplicado da muy buenos resultados, sin embargo, debe ser tenido en cuenta como una referencia, un marco de trabajo, pero sabiendo que dentro de este marco se debe trabajar con flexibilidad, creatividad y sin miedo a equivocarse10. Por la razón ya expuesta proponemos dar un nuevo sentido a las etapas del modelo utilizando la secuencia de pasos que sigue:

1. Presentar
2. Escuchar
3. Compartir
4. Escuchar y observar
5. Inducir
6. Escuchar y observar
7. Guiar
8. Escuchar y observar
9. Evaluar
10. Escuchar y observar
11. Acordar y cambiar
12. Comunicar

En base a esto y reforzando el primer paso a dar en un proceso de mejoras, según la primera sección, es fundamental la participación de todos los involucrados y la comunicación de todo lo que suceda en el proceso de mejoras.

DOS FENÓMENOS ESPONTÁNEOS

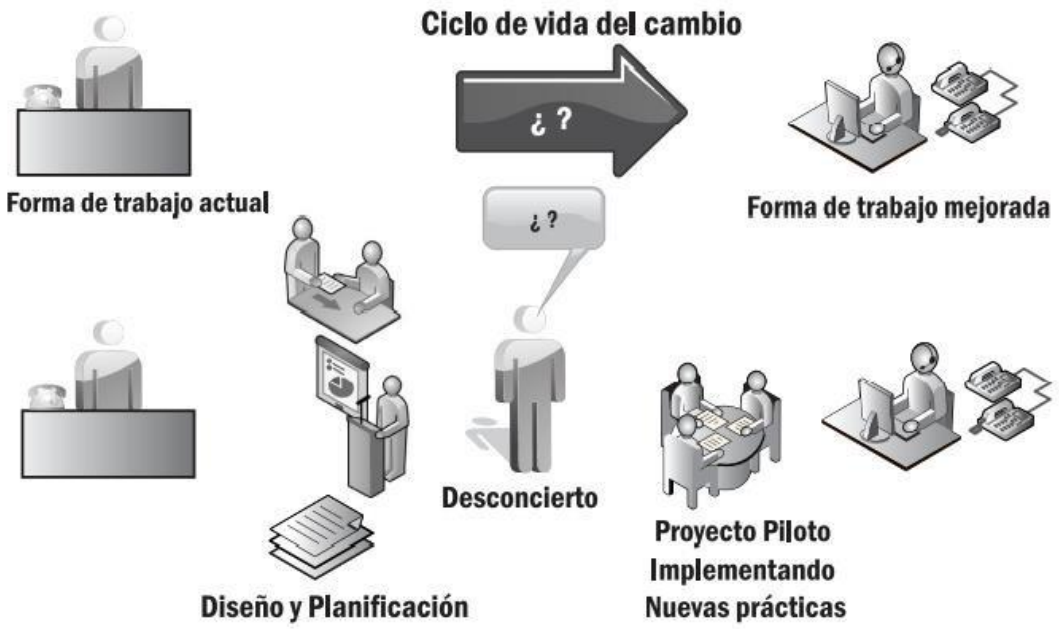
A medida que el proceso de mejoras se institucionaliza y avanza en su desarrollo se generan dos fenómenos que si no son bien administrados pueden atentar contra los cambios planificados. A continuación, analizamos ambos.

DESCONCIERTO

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

¿Qué ocurre en el tiempo que va desde la decisión de cambiar hasta que el cambio se hace efectivo? Esto se muestra en la parte superior de la figura 3.2. Todo proceso de cambios conlleva una capacitación acerca de las buenas prácticas a incorporar en las actividades básicas que, como ya expresamos, siguiendo a Steve McConnell, son la base de las mejoras. En este proceso, que va desde la forma original de trabajo a la forma mejorada, ocurre uno de los fenómenos que llamamos Desconcierto. Este fenómeno es analizado genéricamente por Tom DeMarco y Timothy Lister, en Peopleware — Productive Projects and Teams, 1987. Los miembros fueron capacitados en la nueva forma, pero no tienen experiencia con ella, no tienen madurez, se sienten inseguros, necesitan convencerse de que lo que se les propone es superior a lo anterior. Esta etapa de desconcierto es el punto de inflexión en el cambio. Si los miembros de la organización encuentran en los mentores el apoyo necesario, terminan reemplazando la forma de trabajo. En cambio, si la duda y la inseguridad superan al apoyo y acompañamiento, entonces gana el desconcierto y se vuelve a la forma original, mala pero segura. En la parte inferior de la figura 3.2 se muestra, a modo de ejemplo, el fenómeno mencionado. También se muestra que una forma efectiva de acompañamiento es la realización de proyectos pilotos, que luego de analizar los resultados obtenidos y los ajustes necesarios permiten institucionalizar los cambios a partir de esta primera experiencia.

Ilustración 22:
Proceso de cambio de tareas



Nota: *en la ilustración se puede observar el ciclo de vida del cambio.*

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Autoevaluación de Calidad de Software

1. ¿Qué significa la sigla ISO en relación a la calidad del software?
2. ¿Qué es la verificación de software?
3. ¿Qué es la validación de software?
4. ¿Qué es la metodología Agile en relación a la calidad del software?
5. ¿Qué es el diseño orientado a objetos?
6. ¿Qué es la técnica de revisión por pares en la calidad del software?
7. ¿Cuál de las siguientes opciones NO es un objetivo del control de calidad del software?
8. ¿Cuál de las siguientes actividades NO es parte de la gestión de la calidad del software?
9. ¿Cuál de las siguientes opciones describe mejor la prueba de integración?
10. ¿Cuál de las siguientes opciones NO es una técnica de prueba de software?
11. ¿Qué es el análisis estático de código?
12. ¿Cuál de las siguientes opciones NO es un tipo de mantenimiento de software?
13. ¿Las pruebas de regresión solo se realizan durante el desarrollo inicial del software?? Verdadero o falso
14. ¿La calidad del software se refiere únicamente a la ausencia de errores en el código? Verdadero o falso
15. ¿La prueba de aceptación es el último paso en el ciclo de vida del software? Verdadero o falso
16. ¿La documentación del software es importante solo para los desarrolladores? Verdadero o falso
17. ¿La calidad del software se puede medir objetivamente mediante la cantidad de características incluidas? Verdadero o falso
18. ¿El uso de patrones de diseño es importante para garantizar la calidad del software? Verdadero o falso

Resumen de la Unidad 3

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

En esta unidad, se profundizó en el estudio de los procesos en el contexto del aseguramiento de la calidad de software. Se examinaron los diferentes procesos involucrados en el ciclo de vida del desarrollo de software, desde la planificación inicial hasta el mantenimiento y la mejora continua. Se analizó la importancia de establecer procesos bien definidos y documentados para garantizar la consistencia y la eficacia en la producción de software de alta calidad. Además, se exploraron estrategias para gestionar y gestionar cambios en los procesos existentes, reconociendo que la adaptabilidad y la flexibilidad son esenciales para mantener la relevancia y la efectividad en entornos dinámicos. Por último, se discutieron los fenómenos espontáneos que pueden surgir durante el desarrollo de software, como los errores imprevistos o las oportunidades inesperadas de mejora, y se exploraron enfoques para identificar, gestionar y aprovechar estos fenómenos de manera efectiva. En resumen, esta unidad proporcionó una comprensión integral de cómo los procesos, el cambio de procesos y los fenómenos espontáneos influyen en la calidad de software y cómo pueden ser gestionados para mejorar continuamente los resultados del desarrollo de software.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

EJE TEMÁTICO III

GUÍA GENERAL DE SEGURIDAD INFORMÁTICA

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

GUÍA GENERAL DE SEGURIDAD INFORMÁTICA

1. DESCRIPCIÓN DE LA ASIGNATURA

La asignatura de Seguridad Informática se centra en explorar los principios, técnicas y herramientas esenciales para salvaguardar la información y los sistemas informáticos de posibles amenazas y ataques cibernéticos. A lo largo de este curso, los estudiantes se embarcan en un viaje para adquirir habilidades prácticas en la detección de vulnerabilidades, la evaluación de riesgos y la implementación de estrategias de seguridad efectivas. Además de aprender los aspectos técnicos, se profundiza en el ámbito ético y de responsabilidad en relación con las pruebas de penetración, asegurando que se realicen de manera ética y legal. Estas competencias son vitales para aquellos que desean incursionar en el campo de la seguridad informática, especialmente en roles enfocados en pruebas de penetración y evaluación de vulnerabilidades. A lo largo del curso, los estudiantes se involucran en ejercicios prácticos, laboratorios y estudios de caso, lo que les permite aplicar los conceptos teóricos aprendidos y desarrollar habilidades para identificar y mitigar riesgos de seguridad informática de manera efectiva.

2. BIBLIOGRAFÍA

Estos libros proporcionan una base sólida para el estudio de la seguridad informática, abordando temas clave como criptografía, principios de seguridad de la información, ingeniería de seguridad, y análisis de malware. La bibliografía complementaria ofrece recursos adicionales que profundizan en aspectos específicos de la seguridad informática y proporcionan una perspectiva más amplia sobre el tema.

2.1. Básica

- Criptografía y seguridad de las comunicaciones: principios y prácticas (Stallings, W., 2017):

Este texto ha sido seleccionado por su exhaustividad en los fundamentos de la criptografía y la seguridad de las comunicaciones, proporcionando una comprensión sólida de los principios y prácticas necesarios para proteger la información en entornos digitales. Su enfoque claro y didáctico lo convierte en una herramienta valiosa para los estudiantes al desarrollar su proceso de aprendizaje de manera autónoma. Además, su estructura facilita la asimilación de los conceptos a través de secciones claramente diferenciadas y actividades de autoevaluación al final de cada tema.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Principios de seguridad de la información (Whitman, M. E., & Mattord, H. J., 2018):

Este libro es esencial para comprender los principios básicos de la seguridad de la información. Su enfoque práctico y su cobertura exhaustiva de temas como la gestión de riesgos, la seguridad de la red y la protección de la información lo convierten en una lectura fundamental para los estudiantes de seguridad informática. Además, su enfoque en la ética y la responsabilidad en el manejo de la información sensibiliza a los estudiantes sobre la importancia de abordar los desafíos éticos en el campo de la seguridad informática.

2.2. Complementaria

- Secrets and Lies: Digital Security in a Networked World (Schneier, B., 2015):

Este libro ofrece una perspectiva única sobre la seguridad digital en un mundo interconectado. Bruce Schneier, un experto reconocido en seguridad informática, proporciona insights profundos sobre los desafíos y amenazas en el mundo digital actual. Su enfoque práctico y su estilo de escritura accesible hacen que sea una lectura valiosa para aquellos que desean comprender mejor el panorama de seguridad actual.

- Security Engineering: A Guide to Building Dependable Distributed Systems (Anderson, R., 2008):

Seleccionado por su enfoque en la ingeniería de seguridad, este libro ofrece una guía completa para construir sistemas distribuidos confiables. Richard Anderson aborda una amplia gama de temas, desde la gestión de riesgos hasta la implementación de controles de seguridad, proporcionando a los estudiantes una comprensión profunda de cómo diseñar sistemas seguros desde cero.

- NIST Special Publication 800-53: Security and Privacy Controls for Federal Information Systems and Organizations (Ross, S. M., Wright, J., & Shirey, R., 2018):

Esta publicación del Instituto Nacional de Normas y Tecnología (NIST) establece estándares de seguridad y privacidad para sistemas de información federales y organizaciones. Su inclusión en la bibliografía complementaria proporciona a los

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

estudiantes una referencia autorizada sobre los controles de seguridad que deben implementarse en entornos gubernamentales y organizacionales.

- Malware Analyst's Cookbook: Tools and Techniques for Fighting Malicious Code (Northcutt, S., & Zeltser, L., 2014):

Este libro es una guía práctica para analizar y combatir el malware. Ofrece una variedad de herramientas y técnicas para identificar, analizar y mitigar el malware, lo que lo convierte en una lectura valiosa para aquellos interesados en el campo de la seguridad informática y la ciberseguridad.

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS

- Valores & habilidades blandas: cultura: creatividad
- valores & habilidades blandas: justicia: resolución de problemas
- valores & habilidades blandas: lealtad: liderazgo
- valores & habilidades blandas: optimismo: planificación y gestión del tiempo

4. OBJETIVO GENERAL

El objetivo de la materia de Seguridad Informática es desarrollar competencias cognitivas, procedimentales y actitudinales en los estudiantes, dotándolos de los conocimientos, habilidades y herramientas necesarias para comprender, analizar y aplicar medidas de seguridad eficientes en los sistemas y redes de información. A través de un enfoque teórico y práctico, la asignatura aborda los principios, conceptos y técnicas fundamentales de seguridad informática. Los estudiantes aprenderán a identificar y evaluar riesgos, implementar medidas de seguridad, utilizar herramientas y técnicas de protección, y gestionar incidentes de seguridad de manera ética y legal. Esto se realiza con el propósito de garantizar la confidencialidad, integridad y disponibilidad de la información, así como preservar la privacidad y asegurar el correcto funcionamiento de los sistemas. Además, la asignatura tiene como objetivo preparar a los estudiantes para enfrentar los desafíos y demandas del entorno digital actual, donde los ataques cibernéticos son cada vez más frecuentes y sofisticados, promoviendo una cultura de responsabilidad y conciencia en seguridad informática.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

5. UNIDADES

UNIDAD 1: INTRODUCCIÓN A LA SEGURIDAD INFORMÁTICA

Temas y Subtemas

- Seguridad informática
- Virus informácos
- Mecanimos preventivos en seguridad informático

La seguridad en términos generales

Al hablar de términos de seguridad informática se debe entender a las bases que conforman los cimientos de esta ciencia, para las partes más complejas de esta disciplina, una de estas bases es el concepto de seguridad, la cual consiste en un estado de bienestar, es la ausencia de riesgo por la confianza que existe en alguien o algo, si la seguridad se aborda desde el tema disciplinario el concepto se puede definir como una ciencia interdisciplinaria para evaluar y gestionar los riesgos a los que se encuentra una persona, un animal, el ambiente o un bien. Existen países en donde la seguridad es un tema nacional, aunque depende del tipo de seguridad, existen muchos tipos de ésta, por ejemplo, la seguridad ambiental, la seguridad económica, la seguridad sanitaria y en casi la mayoría de los países cuando se hace un análisis de la palabra seguridad, se hace referencia a la seguridad de las personas, por ejemplo, evitar el estado de riesgo de un robo, de un daño físico o de un bien material. La seguridad siempre busca la gestión de riesgos, esto quiere decir que se tenga siempre una forma de evitarlo o prevenirlo y que se pueda realizar ciertas acciones para

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

evitar esas situaciones de la mejor forma. Se definió que la seguridad podría ser catalogada como la ausencia de riesgo, la definición de este término involucra cuatro acciones que siempre están inmersas en cualquier asunto de seguridad como son:

- Prevención del riesgo
- Transferir el riesgo
- Mitigar el riesgo
- Aceptar el riesgo

Así que, cuando se está buscando hacer algo más seguro, estas acciones son algo que se debe de considerar sin importar el área, se aplica a cualquier intento de tener mejor o mayor seguridad en cualquier tema que se requiera.

Conceptos de seguridad informática

Lo primero que se debe mencionar es que en muchos casos se suelen confundir dos conceptos la seguridad informática y la seguridad de la información, aunque suenen muy parecidos tienen puntos clave que hacen una diferencia. La seguridad informática se encarga de la seguridad del medio informático, según varios autores la informática es la ciencia encargada de los procesos, técnicas y métodos que buscan procesar almacenar y transmitir la información, mientras tanto la seguridad de la información no se preocupa sólo por el medio informático, se preocupa por todo aquello que pueda contener información, en resumen, esto quiere decir que se preocupa por casi todo, lo que conlleva a afirmar que existen varias diferencias, pero lo más relevante es el universo que manejan cada uno de los conceptos en el medio informático. Según Aguilera (2011), se puede definir a la seguridad informática como la disciplina encargada de plantear y diseñar las normas, procedimientos, métodos y técnicas con el fin de obtener que un sistema de información sea seguro, confiable y sobre todo que tenga disponibilidad. Actualmente la informática está siendo inundada por toda la información posible, pero la información por sí sola sigue siendo un universo más grande y en muchos casos más compleja de manejar, ya que los procesos en muchos casos no son tan visibles para los involucrados. La principal tarea de la seguridad informática es la de minimizar los riesgos, en este caso provienen de muchas partes, puede ser de la entrada de datos, del medio que transporta la información, del hardware que es usado para transmitir y recibir, los mismos usuarios y hasta por los mismos protocolos que se están implementando, pero siempre la tarea principal es minimizar los riesgos para obtener mejor y mayor seguridad. Lo que debe contemplar la seguridad se puede clasificar en tres partes como son los siguientes:

- Los usuarios

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- La información
- La infraestructura

Los usuarios son considerados como el eslabón más débil de la cadena, ya que a las personas es imposible de controlar, un usuario puede un día cometer un error y olvidar algo o tener un accidente y este suceso puede echar a perder el trabajo de mucho tiempo, en muchos casos el sistema y la información deben de protegerse del mismo usuario. La información se considera como el oro de la seguridad informática ya que es lo que se desea proteger y lo que tiene que estar a salvo, en otras palabras, se le dice que es el principal activo. Por último, está la infraestructura esté puede ser uno de los medios más controlados, pero eso no implica que sea el que corre menos riesgos, siempre dependerá de los procesos que se manejan. Se deben de considerar problemas complejos, como los de un acceso no permitido, robo de identidad, hasta los daños más comunes, por ejemplo, robo del equipo, inundaciones, incendios o cualquier otro desastre natural que puede tener el material físico del sistema de la organización.

Aguirre (2006), también afirma que la seguridad informática puede definirse como el conjunto de métodos y de varias herramientas para proteger el principal activo de una organización como lo es la información o los sistemas ante una eventual amenaza que se pueda suscitar.

Los virus Informáticos

Unos de los primeros conceptos cuando se habla de seguridad informática, es el de virus informático. Las computadoras solo entienden código binario como ceros y unos, en el mundo de las computadoras y de la informática existen muchos conceptos como el de programas, videojuegos, sistemas operativos y cualquier clase de software. El software es uno de los conceptos más abstractos, se lo define como todo lo intangible de la computadora, son instrucciones que el ordenador espera que se realicen, las cuales pueden ser instrucciones complejas o instrucciones sencillas. Según Beynon-Davies (2015), el término software o programa es utilizado para describir una secuencia de varias instrucciones que es leído por un computador, los cuales son escritos en un determinado lenguaje de programación que pueden ser clasificados de la siguiente manera:

- Lenguaje de máquina
- Lenguaje ensamblador
- Lenguaje de alto nivel

Analizado el tema clave sobre el software, un virus informático es un programa que tiene como objetivo dañar o cambiar el funcionamiento de la computadora. Esta es una definición bastante

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

clara, pero el virus informático no siempre tiene que ser un programa completo, puede ser hasta cierto punto fragmentos de un programa. Según Vieites (2013), se define al virus informático, como un programa desarrollado en un determinado lenguaje de programación (C++, C, ensamblador, etc.) con el objetivo de infectar uno o varios sistemas informáticos, utilizando varios mecanismos de propagación o auto replicación, el cual trata de reproducirse de forma acelerada para extender su alcance. Un virus informático puede hacer muchas cosas, por ejemplo, eliminar archivos, evitar accesos a las computadoras, robo de información, bloqueo de funciones de un sistema operativo o de programas dentro de una computadora. También Vieites (2013), indica que existen varios tipos de virus que se los puede definir de la siguiente manera:

- Virus de sector de arranque (BOOT)
- Virus de archivos ejecutables
- Virus de macros
- Virus de lenguaje de Script
- Malware
- Gusanos
- Troyanos
- Spyware
- Keyloggers
- Adwares
- Dialers
- Backdoors
- Otros
- Rootkits
- Bacteria
- Bombas de tiempo
- Pishing

Se mencionó algunos, ya que la lista es bastante grande pero la mayoría son programados para causar daños relacionados con la red y tener la capacidad de auto propagación, esto quiere decir que se multiplica el mismo muchas veces y se posiciona en partes automatizadas del sistema

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

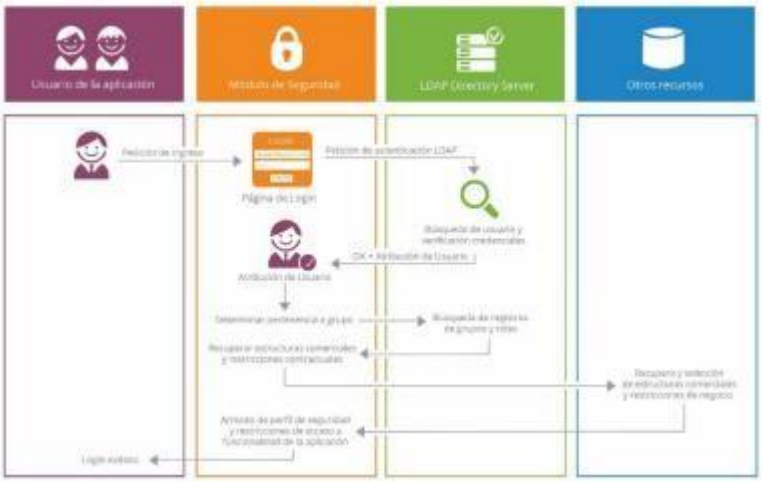
operativo infectado. Las bombas de tiempo, son virus que se activan al pasar un determinado tiempo o al producir un evento, el que puede ser, por ejemplo, abrir el navegador, pero los eventos suelen ir relacionados con ciertos cálculos matemáticos y registros de memoria, aunque también existen los que se activan con tareas sencillas, estos son solamente algunos de los tipos que se podrían mencionar. También existe el denominado software malicioso que no es considerado como virus como tal, pero que también genera daños a la computadora, algo muy importante que se debe tener claro es que, el software malicioso debe de tener ciertas características para ser considerados como virus informático, una de las características elementales es que debe de poder reproducirse y generar copias, ya que es la forma en la que se propagan teniendo un comportamiento biológico similar al de los virus que se pueden encontrar en la naturaleza y atacan a los animales y personas.

Conceptos de autentificación

La autentificación se puede definir como un proceso en el que se busca confirmar algo como verdadero, no se busca verificar un usuario, ya que la autenticación no siempre está relacionada con estos, en muchos casos se quiere saber si un cambio o un dato es correcto, no se debe cometer el error en pensar que solamente las personas necesitan este proceso, este puede ser para cualquiera, un sistema, un dispositivo o una persona. La autenticación es bastante usada en el mundo de la computación, sólo que actualmente la contraseña del correo o de una red social ha hecho olvidar que este método de validación era ya muy común, por ejemplo, todas las credenciales que expiden para realizar una votación en determinado país es un método de autenticación, otro ejemplo es cuando se ingresa a un país y solicitan un documento como la visa o pasaporte, también es un método de autenticación, otro caso es cuando se asigna un número de cuenta o ID de identificación en el trabajo para acceder a ciertas áreas o también para llevar un registro de los movimientos y en caso de ser necesario poder validar esos movimientos. La Figura muestra un ejemplo de autentificación de usuarios.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ilustración 23:
Autentificación de usuario



Nota: *en la ilustración se puede observar los pasos para la autentificación de un usuario en un sistema.*

Existen diversos tipos de autenticación, se va a conocer algunos de ellos los más implementados ya que todos los días se trabaja en encontrar más y mejores métodos. Se tiene los tipos de autenticación en los que se tiene algo conocido, en teoría únicamente por el usuario, por ejemplo, una contraseña, eso es lo más común, pero en teoría, ya que, si se proporciona el usuario y la contraseña del correo electrónico, también puede entrar otro usuario y no significa que sea la persona dueña de la cuenta. Otro tipo de autenticación es la que se basa en algo de propiedad del usuario, por ejemplo, la tarjeta de crédito, pasaportes o también son los Tokens que generan números aleatorios o palabras claves. También existen las tarjetas conocidas como inteligentes o que contienen cierta información, se pueden parecer a una tarjeta de crédito, pero el comportamiento o información puede variar. Se tiene también los tipos de autenticación basados en una característica física, este tipo en comparación con lo que ya se mencionó se puede decir que son los más nuevos. Cuando se habla de características físicas se puede mencionar a:

- La voz
- Las huellas dactilares
- El ojo
- La escritura

La autenticación se puede considerar como parte de un método de control de acceso, la mayoría de las ocasiones esto se complementa con otras partes de un sistema, ya que hoy en día debido al manejo de la información y la personalización de los gadgets que se tiene disponibles, se vuelve

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

una labor compleja la de tener control y manejo dentro del sistema. Los tipos de autenticación no son excluyentes, así que, si se usa un método, no es una barrera para usar otro, de hecho, en sistemas complejos el usuario se puede encontrar con sistemas que utilizan tres tipos de autenticación, obviamente se tiene que pensar en el usuario, a veces es muy molesto siempre y cuando analizando el costo vs el beneficio.

Mecanismos preventivos en seguridad informática

Los mecanismos preventivos en la seguridad informática son los más olvidados, los cuales son vistos como una pérdida de tiempo, la parte administrativa en la mayoría de los casos lo ve como un costo extra, es algo parecido como por ejemplo, con los seguros médicos o seguros de vehículos, se puede pagar 10 años el seguro de un carro y nunca tener un accidente, en primera instancia se podrá analizar que es algo muy bueno, pero después en algún momento se podrá pensar que es un desperdicio haber pagado una cantidad 10 años y sin usarla. La definición de los mecanismos preventivos, consiste en una serie de revisiones periódicas, algunos cambios o mejoras de diferentes aspectos que pueden ser de hardware, software o de cualquier elemento involucrado en los sistemas y procesos, por eso es que las revisiones dependen de los procesos de la empresa y cada una tiene sus propios procesos. Los mecanismos preventivos en realidad son a largo plazo y por esta razón son considerados por la mayoría como una pérdida de tiempo y dinero. La mayoría de los ataques informáticos se pueden evitar o por lo menos disminuir el impacto, si se hiciera utilizando mecanismos preventivos, deficiencia de sistemas y otros problemas podrían encontrarse, evitarse y resolverse gracias a un buen trabajo durante esta etapa. La Barrera más fuerte a la que se enfrenta una empresa al querer aplicar los mecanismos preventivos, es la aceptación y el compromiso de todos los involucrados, hacer entender que no es una carga, es parte de los procesos y de lo que se debe hacer bien en la organización. Entre los elementos que se pueden aplicar en los mecanismos preventivos se puede mencionar a:

- *El respaldo de información:* Es uno de los procesos más comunes que se pueden realizar en las compañías y que gozan de cierta aceptación general, las empresas entienden que los problemas con información son muy costosos, parece muy fácil pero seleccionar los mecanismos de respaldo no es tan sencillo como se analizar, se tiene que considerar los siguientes factores: Qué formatos de archivo se tienen, por ejemplo, MP3, archivos de texto, bases de datos y otros, las imágenes y videos por ejemplo, son archivos que normalmente necesitan atención especial.
- *Horario de respaldo:* Otro reto es a qué hora se puede hacer el respaldo, es común seleccionar las horas de menos tráfico.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- *Control de los medios:* El tener acceso a respaldos es algo de alto riesgo, se puede robar la información, manipular, perder, así que, el respaldo es una solución, pero también es otro problema que se debe resolver.
- *La comprensión de la información:* No toda la información se puede comprimir, pero existe alguna que, sí lo necesita, así que se deben hacer las valoraciones respectivas.

Estos son sólo algunos de los puntos que se deben considerar, solamente para el mantenimiento y respaldo de la información. Otros ejemplos de proceso que se tienen en el mecanismo preventivo son:

- Actualización de sistemas.
- Antivirus
- Firewall
- Navegación por internet
- Contraseñas
- Accesos remotos.

Estos son sólo algunos de los procesos, pero la organización puede personalizar lo que quiere considerar en los mecanismos preventivos.

Mecanismos correctivos en seguridad informática

Los mecanismos correctivos tienen una gran diferencia en tiempo con los mecanismos preventivos, estos se aplican cuando, después de que algo sucedió y la función principal es corregir las consecuencias. Entre las características que tienen los mecanismos correctivos normalmente son muy caros, esto se debe a que el problema ya se lo tiene encima y no se puede tenerlo durante mucho tiempo, así que, contratar expertos para resolver el problema o el tiempo que le dedicara a el equipo de trabajo siempre va a costar mucho, en un porcentaje muy alto se acaban pagando servicios de solución a otras empresas, adquiriendo soluciones o comprando software y parches de actualización que logran resolver el problema. Otra característica de los mecanismos correctivos es que el tiempo es limitado, así que el tiempo se vuelve algo muy apreciado en estos casos, pero también es muy escaso. Probablemente la empresa o la persona puede poder obtener dinero, pero tiempo es casi imposible. Dentro de los mecanismos de corrección se tienen diferentes pasos de ejecución para enfrentar este problema serio en los que se puede mencionar:

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- *Catalogación y asignación de problemas:* En este paso se hace un catálogo de los problemas a los que se pueden enfrentar, detectar y clasificar es algo muy recurrente en todo lo relacionado con la seguridad informática, ya que es una forma para poder saber cómo abordar las situaciones y buscar alguna respuesta o solución a lo que se presenta.
- *Análisis del problema:* En este paso es muy evidente que la actividad que se hace es analizar el problema que se ha presentado, en muchos casos esta parte se realiza por los expertos, ya no, por las personas involucradas en el problema.
- *Análisis de la solución:* Antes de intentar solucionar el problema se debe de analizar la propuesta de la solución, se ha cometido un error, puede ser que no de forma directa, pero es un error, el impacto no va a ser más o menos, si es culpa del usuario o de un tercero, así que la solución tiene que estar bien planteada y ejecutada. Antes de empezar a realizar los cambios, actualizaciones y movimientos se debe tratar de analizar y de predecir qué es lo que va a suceder.
- *La documentación:* Este componente es vital, ya que los cambios que se hacen probablemente son algo que se hizo con un tiempo limitado, rápido y que involucraron muchos recursos, así que la documentación es muy importante, ya que puede ser que por las velocidades no se recuerden todos los pasos y cambios que se han realizado. En caso de encontrar algún problema se puede consultar la documentación para detectar si la solución era correcta.

Mecanismos de detección en seguridad informática

Los mecanismos de detección son los más complejos y son en los que se necesita tener alto grado de conocimientos técnicos dependiendo de la materia que se aborde, por ejemplo, seguridad de plataformas en línea, en específico de un tipo de bases de datos o tecnología como Wordpress, esto depende del sistema, aplicación o el ecosistema que tenga funcionando. Los mecanismos de detección parten de que se tiene la idea de que un atacante es capaz de violar la seguridad y puede haber realizado una intrusión total o parcial a un determinado recurso. Siempre que se trabaja en los mecanismos de detección se tiene la premisa en mente, se debe de trabajar como si lo que se fuera a encontrar es lo peor y se debe estar preparados para la peor de las situaciones posibles. Estos mecanismos de detección tienen dos objetivos:

Poder detectar el punto exacto del ataque para poder llegar a una solución y recuperarse del mismo, pero no siempre es posible esto, depende de los problemas que se afrontan.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Detectar la actividad que se considera sospechosa y conocer lo sucedido, ya que si no se encuentra donde fue el ataque, lo mínimo que se necesita es saber qué fue lo que sucedió y partir de esa parte.

Lo que es ideal es que se cumpla el objetivo primero, pero no siempre sucede lo ideal, así que se tiene que adaptar al problema, a la situación y todo lo que va saliendo en cada uno de los casos. Uno de los conceptos que están inmersos en este tipo de mecanismos es la intrusión, la cual se la define como una secuencia de acciones realizadas de forma deshonesta, en donde la mayoría de las ocasiones se quiere lograr acceso no autorizado. Dentro de los mecanismos de detección el término más famoso de seguridad informática es el de detección de intrusiones, la cual se define como el proceso de identificación y respuesta ante las actividades ilícitas observadas contra algunos recursos de la red, sistema, plataforma o empresa. Los mecanismos de detección de intrusión tienen unos pasos que se ejecutan como manera básica de detección que se menciona a continuación:

Revisión de patrones de acceso: En este caso lo que se hace es ver los patrones de acceso, esto quiere decir que se va a analizar los accesos y tratar de encontrar si se está manejando un patrón, por ejemplo, acceso a determinadas horas o el mismo usuario haciendo accesos a la misma sección o módulo. Los patrones siempre van a indicar algo, pueden ser muchos o Las mayorías falsas alarmas, pero es seguro, que si se hizo un ataque se puede encontrar patrones que llamen la atención para después encontrar el problema.

Revisión de transacción: En la mayoría de los casos se obtienen ciertos archivos o se intenta descargar o subir algo de información, así que la transacción es un método muy rápido para lograr esto, la mayoría de los intentos van a ir acompañados de al menos una transacción, esto no es una garantía, pero es algo muy probable, siempre durante la detección si se logra encontrar una transacción es como encontrar el objetivo del atacante lo cual es muy valioso.

Bloqueo automático: Algunas aplicaciones no tienen un sistema de bloqueo, así que, aunque en algunos casos se encuentre el problema y ya se tenga las razones, Si no se cuenta con un mecanismo de bloqueo de emergencia, el atacante podrá seguir haciendo lo que quería. Algunos de los mecanismos de bloqueo comunes son los de paro absoluto, es decir el bloqueo del sistema completo, es algo un poco drástico, pero en muchas ocasiones no se quiere otro riesgo y se considera la mejor opción a la mano.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Fundamentos de seguridad informática

LOS TRES PILARES DE LA SEGURIDAD

Los datos son valores, números, medidas, textos, documentos en bruto, la información es el valor de esos datos, es lo que aporta conocimiento. Los manuales de procedimientos, los datos de los empleados, de los proveedores y clientes de la empresa, la base de datos de facturación son datos estructurados de tal forma que se convierten en información, que aportan valor como compañía. Los pilares de la seguridad de la información se fundamentan en esa necesidad que todos tienen de obtener la información, de su importancia, integridad y disponibilidad de la información para sacarle el máximo rendimiento con el mínimo riesgo. La ilustración muestra los principales pilares de la seguridad de la información.

Ilustración 24:
Pilares de la seguridad

Nota: *en la ilustración se puede observar los pilares de la seguridad en un ciclo.*

Según la Figura, la seguridad está fundamentada por 3 pilares, pero puede haber más que puedan fundamentar a la seguridad, en este caso, si alguno de los lados es débil se perderá seguridad o usabilidad, si falta alguno de los lados la organización queda expuesta a ataques, para esto se debe conocer en detalle cuál es la función de cada lado en el gráfico.

Ahora que se compre la importancia de la información se puede deducir que si aquella, que es vital para la organización cayera en manos inapropiadas puede perder su valor, se perderá intimidad o capacidad de maniobra y además la reputación puede verse dañada sin contar con que la información puede ser accedida por cibercriminales y cualquier otra potencial fuente de riesgos para un determinado proyecto.

Confidencialidad: La confidencialidad consiste en asegurar que sólo el personal autorizado accede a la información que le corresponde, de este modo cada sistema automático o individuo solo

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

podrá usar los recursos que necesita para ejercer sus tareas, para garantizar la confidencialidad se recurre principalmente a tres recursos:

Autenticación de usuarios: Sirve para identificar qué quién accede a la información es quien dice ser.

Gestión de privilegios: Para los usuarios que acceden a un sistema puedan operar sólo con la información para la que se les ha autorizada

y sólo en la forma que se les autorice, por ejemplo, gestionando permisos de lectura o escritura en función del usuario.

Cifrado de información: Según Costas Santos (2011), el cifrado también denominado encriptación, evita que ésta sea accesible a quién no está autorizado, para ello se transforma la información de forma inteligible a una no legible y es aplicable tanto a la información que esté autorizado para ello como para la que no lo está, sólo mediante un sistema de contraseñas puede extraerse la información de forma inteligible y es aplicable tanto a la información que está siendo transmitida como a la almacenada.

Los principios de confidencialidad no solo deben aplicarse para proteger la información sino todos aquellos datos e información de los que sea responsables. La información puede tener carácter confidencial no solo por ser de alto valor para la organización, sino por ejemplo porque puede estar amparada por legislación de protección de datos de carácter personal, un ejemplo de violación de la confidencialidad son las filtraciones sufridas por entidades bancarias, grandes empresas y gobiernos para exponer públicamente algunas de sus actividades.

La integridad: Es el segundo pilar de la seguridad, consiste en asegurarse de que la información no se pierde ni se ve comprometida voluntaria e involuntariamente, el hecho de trabajar con información errónea puede ser tan nocivo para las actividades como perder la información, de hecho, si la manipulación de la información es lo suficientemente sutil puede causar que se arrastre una cadena de errores acumulativos y que sucesivamente se tome decisiones equivocadas. Para garantizar la integridad de la información se debe considerar lo siguiente:

Monitorear el tráfico de red para descubrir posibles intrusiones.

Auditar los sistemas para implementar políticas de auditorías que registre quien hace que, cuando y con qué información.

Implementar sistemas de control de cambios, algo tan sencillo como por ejemplo comprobar los resúmenes de los archivos de información almacenados en sistema para comprobar si cambian o no.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Como otro recurso se tiene las copias de seguridad, que en caso de no conseguir impedir que se manipule o pierda la información permitan recuperarla en su estado anterior.

Disponibilidad: Para poder considerar que se dispone de una seguridad mínima en lo que a la información respecta, se tiene a la disponibilidad, de nada sirve que solo el usuario acceda a la información y que sea incorruptible, si el acceso a la misma es tedioso o imposible, la información para resultar útil y valiosa debe estar disponible para quien la necesita, se debe implementar las medidas necesarias para que tanto la información como los servicios estén disponibles, por ejemplo un ataque distribuido de denegación de servicio o DDoS puede dejar inutilizada una tienda online impidiendo que los clientes accedan a la misma y puedan comprar. Ogro ejemplo de perdida de disponibilidad sería que la dirección de correo electrónico sea utilizada para lanzar campañas de spam y en consecuencia añadida a listas negras, impidiendo que ninguno de los destinarios de los emails legítimos los reciba. Para este propósito se implementan políticas de control como:

- El acuerdo de nivel de servicio o (SLA).
- Balanceadores de carga de tráfico para minimizar el impacto de DDos.
- Copias de seguridad para restauración de información perdida
- Disponer de recursos alternativos a los primarios.

La información y sistemas son seguros si sólo accede a la información y recursos quién debe, sí se puede detectar y recuperar de manipulaciones voluntarias o accidentales de la información y si se puede garantizar un nivel de servicio y acceso a la información aceptable según las necesidades. Carpentier (2016), indica que el uso de sistemas de información implica establecer normas y procedimientos aplicados al uso y sistemas de información ante posibles amenazas como:

- Elaborar varias normas y procedimientos.
- Definición de acciones que deben emprender las personas.
- Definición del perímetro que se va a afectar.

EVALUACIÓN DE RIESGOS, AMENAZAS Y VULNERABILIDADES

Cuando se plantea mejorar la seguridad de una empresa se debe tener en cuenta varios factores que se muestra a continuación:

- Recursos

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Amenazas
- Vulnerabilidades
- Riesgos

Se entiende a los **recursos** como los bienes tangibles e intangibles con los que se cuenta para realizar las tareas, la información de que se dispone es un bien intangible, ya sean las bases de datos de clientes, proveedores, los manuales de producción, las investigaciones y las patentes. Por otro lado, se tiene a los bienes tangibles, qué son los recursos físicos de que se dispone en la empresa, servidores, equipos de red, computadoras, teléfonos inteligentes, vehículos, bienes inmuebles, etc., la ilustración muestra un ejemplo de bienes tangibles e intangibles.

Ilustración 25:
Ejemplos de bienes tangibles o intangibles

Nota: *en la ilustración se puede observar los bienes tangibles e intangibles de una empresa.*

El riesgo es la probabilidad de que algo negativo suceda dañando los recursos tangibles o intangibles y por tanto impidiendo desarrollar la labor profesional. Las amenazas son esos sucesos que pueden dañar los procedimientos o recursos, mientras que las vulnerabilidades son los fallos de los sistemas de seguridad o en los propios que el usuario utiliza para desarrollar las actividades que permitirían que una amenaza tuviese éxito a la hora de generar un problema. El principal trabajo de un responsable de la seguridad es la evaluación de los riesgos identificando las vulnerabilidades, amenazas y en base a esta información evaluar los riesgos a los que están sujetos las actividades y recursos. Se debe considerar el riesgo como la probabilidad de que una amenaza concreta aproveche una determinada vulnerabilidad se puede aplicar la representación clásica que indica lo siguiente, mostrado en la ilustración.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ilustración 26 *Formula para medir el riesgo*

Nota: *en la ilustración se puede observar la fórmula para medir el riesgo.*

La ilustración anterior indica que el riesgo es igual al resultado de sumar el impacto producido por la amenaza por la probabilidad de que una vulnerabilidad permita que dicha amenaza tenga éxito. En un sistema de evaluación de riesgo sencillo se podría asignar un valor numérico a la importancia de una vulnerabilidad y otro valor a la importancia de una amenaza, la tabla muestra un ejemplo de evaluación de riesgos.

Tabla 2:
Evaluación de riesgos

Amenaza	Impacto	Probabilidad	Riesgo
Robo de credenciales en un sistema de control biométrico	3	0	0
Infección por Spyware	3	2	6
Perdida de suministros eléctricos	1	11	1

Nota: *en la tabla se puede observar las amenazas, el impacto, la probabilidad y el posible riesgo.*

En la tabla anterior las amenazas que no causan daño tendrían un impacto 0, mientras que las que causan un gran daño tendrían un valor de impacto igual a 3, del mismo modo la probabilidad puede ser nula, baja, media o alta, con lo que se podría dar valores de probabilidad de 0 a 3, multiplicando ambos valores se obtendría el valor de riesgo, de esta forma se podría clasificar los distintos riesgos a los que se está expuesto y actuar en consecuencia, empezando por los de mayor gravedad, por ejemplo, si existe una amenaza de ataques mediante robo de credenciales para acceder a un recurso cifrado, el impacto de perder dicha información seria alta, pero ya que se tiene un sistema de autenticación biométrico, las probabilidades de que exista una probabilidad aprovechable son prácticamente nulas, por lo tanto el riesgo es cero para esa amenaza.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Resumen de la Unidad 1

En esta unidad, se exploraron los conceptos fundamentales de la seguridad informática, centrándose en la protección de sistemas y datos contra amenazas cibernéticas. Se examinaron los diversos tipos de ataques y amenazas, incluidos los virus informáticos, malware, phishing y ataques de denegación de servicio (DDoS), entre otros. Se analizó en detalle el funcionamiento y los efectos de los virus informáticos, así como las estrategias de prevención y mitigación disponibles. Además, se estudiaron los mecanismos preventivos en seguridad informática, que incluyen el uso de software antivirus, firewalls, actualizaciones de software y políticas de seguridad de la información. A lo largo de la unidad, se enfatizó la importancia de la conciencia y la educación en seguridad informática, así como la implementación de medidas proactivas para proteger la integridad, confidencialidad y disponibilidad de los sistemas y datos.

UNIDAD 2: HACKING ÉTICO

Temas y Subtemas

- **Hacking Ético**
- **Vulnerabilidades**
- **Encriptación**

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Las vulnerabilidades

Definiendo a muy grandes rasgos que es una vulnerabilidad, una vulnerabilidad de una manera muy general es un fallo en un sistema que puede ser explotada por un atacante generando un riesgo para la organización o para el mismo sistema. Existen dos tipos de vulnerabilidades que se mencionan a continuación:

- Las lógicas
- Las físicas

VULNERABILIDADES FÍSICAS

Las vulnerabilidades físicas son las que van a afectar a la infraestructura de la organización de manera física y se pueden mencionar en este tipo de clasificación a los desastres naturales, como ejemplo se podría mencionar una vulnerabilidad alta de este tipo si se vive en una zona de alto riesgo de sismos, ya que puede presentarse una negación en el servicio, una afectación en la disponibilidad y a partir de ahí se podría empezar con problemas. Si la organización está en una zona que generalmente se inunda, se tiene también otro tipo de vulnerabilidad. Otra de las opciones físicas son los controles de acceso, en muchas ocasiones se tiene los accesos a la infraestructura crítica y no se tiene los accesos pertinentes, cualquier persona podría abrir una puerta, podría entrar y constituye un gran riesgo para la organización porque cualquier usuario podría ingresar con una USB y copiar información, podría infectar la misma infraestructura.

VULNERABILIDADES LÓGICAS

Las vulnerabilidades lógicas son las que van a afectar directamente la infraestructura y el desarrollo de la operación de estos, estas pueden ser de:

- Configuración
- Actualización
- Desarrollo

Las de configuración en el sistema operativo, pueden ser las configuraciones por defecto del sistema o incluso de algunas aplicaciones del servidor que se tenga expuesta, puede ser también la configuración de algunos firewalls que no está gestionado de una manera correcta y también de infraestructura perimetral. Las vulnerabilidades de actualización, en muchas ocasiones hay empresas que no actualizan sus sistemas, van saliendo las vulnerabilidades y es un punto que se debe tomar en cuenta.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Actualmente, en los equipos XP de Windows no se les está dando soporte y muchas empresas tienen estos sistemas, cuando se realiza un escaneo en una determinada red al no tener soporte estos equipos ya son vulnerables. Las vulnerabilidades de desarrollo, aquí se puede mencionar las inyecciones de código en SQL, Cross Site Scripting, esto puede variar dependiendo del tipo de aplicación, la validación de los datos. Cada escáner de vulnerabilidades utiliza distintas escalas, en estas escalas se va a poder auditar en base a una metodología de pruebas de penetración, de cumplimiento, si se va a auditar una red interna o una aplicación web, es muy distinto el escáner que se va a utilizar.

Encriptación

La encriptación o también conocido como cifrado, es un procedimiento en el que se busca que la información sea ilegible, ya aplicado este procedimiento la información es inservible para cualquier persona que no sea la autorizada, aunque el mensaje sea interceptado, como en muchos casos la información simplemente no significa nada para el interceptor, ya que no cuenta con los elementos involucrados en la encriptación, así que la información simplemente no sirve, la ilustración muestra un ejemplo de encriptación.

Ilustración 27:
Encriptación

Nota: *en la ilustración se puede observar los pasos de encriptación de un archivo*

Se puede decir también, que la encriptación busca la seguridad y la persistencia de los datos mediante un proceso en el cual se involucran algunas partes claves dependiendo del método, por ejemplo, en algunos métodos se utilizan contraseñas o llaves para autentificar la encriptación y la desencriptación de la información, siempre se debe de recordar los objetivos principales de la encriptación y cifrado de datos que se nombran a continuación:

- Confidencialidad

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Autenticación
- Integridad de los datos.

La confidencialidad consiste en que la información sólo puede ser accedida por su legítimo dueño o destinatario, la autentificación quiere decir que el emisor y el receptor son los que pueden confirmar la identidad, finalmente la integridad de la información significa que no debe ser posible que sea alterada en caso de que sea interceptada la información. Según Marrero Travieso (2003), existen muchas amenazas de varias fuentes principalmente de internet que pueden ser:

- Correos electrónicos infectados por virus
- Firewalls mal Configurados
- Suplantación de contraseñas
- Contraseñas débiles
- Robo y destrucción de información, etc.

MÉTODOS DE ENCRIPTACIÓN

Algunos de los métodos de encriptación disponibles actualmente y que son bastantes conocidos se puede mencionar a:

- Encriptación simétrica
- Encriptación asimétrica de clave pública y privada.
- Encriptación WPA
- Encriptación WEP
- Firma digital

Estos métodos mencionados anteriormente son la mayoría que se va a encontrar en el mundo de la seguridad informática. Estos métodos de encriptación son bastantes buenos para almacenar y transferir la información.

ENCRIPTACIÓN SIMÉTRICA

Según (Santos, 2014) este tipo de criptografía está basado en métodos criptográficos que usan una misma clave para cifrar y descifrar el mensaje, estos extremos cuando establecen la comunicación deben establecer un acuerdo sobre la clave que tienen que usar, para posteriormente los dos tener acceso a la misma clave, en donde la remitente cifra el contenido de la misma y el destinatario la descifra con el mismo mecanismo. Se puede indicar varios ejemplos de cifrado simétrico.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Algoritmo de cifrado DES, usa claves basados en 56 bits
- Algoritmos de cifrado 3DES, Blowfish, e IDEA, usan claves de 128 bits
- Algoritmos de cifrado RC5 y AES

ENCRIPTACIÓN ASIMÉTRICA

También (Santos, 2014) indica que este tipo de encriptación se basa en que si el emisor cifra la información el receptor lo puede descifrar o viceversa, en este caso cada usuario del sistema debe poseer una pareja de claves y se tiene dos tipos.

- *Clave privada*: Custodiada por el propietario, por lo tanto, solo él tiene acceso a ella sin darla a conocer a nadie.
- *Clave pública:* conocida por uno o todos los usuarios Como ejemplo de este tipo de algoritmos usados por este tipo de cifrado se tiene a MD5 y SHA

FIRMA DIGITAL

La Firma digital, es algo habitual en el uso de documentos oficiales, es decir documentos que involucran a una institución gubernamental. El objetivo de la firma es autentificar la identidad de quién envía el mensaje y quién firma el documento, las firmas digitales acostumbran manejar diferentes datos, además de información que se envía, por ejemplo, la hora y la fecha en que se hizo. La firma digital es una forma matemática de adjuntar la identidad de una persona a un mensaje, está basada en la criptografía de clave pública, esto quiere decir que estos sistemas están utilizando dos claves, la primera sería la clave pública que es la que se conoce y la otra clave sería una clave privada qué es la que solamente el emisor del mensaje conoce.

ENCRIPTACIÓN WEP Y WPA

La encriptación WEP y WPA tienen algo en común, las dos son aplicadas a las señales inalámbricas y están basados en protocolos de conexión Wifi la primera y la segunda se basa en servidores de autentificación. En el caso de WEP se tiene tres opciones, de 64 bits, de 128 bits y 256 bits, en donde la más utilizada es la de 128 bits ya que ofrece un buen nivel de seguridad sin tener que ser tan grande y sin aumentar lo complicado del tema. Actualmente la encriptación de 256 bits aún no es soportada por todos los dispositivos. Existen siempre diferentes opiniones de cómo es que se puede considerar a un método de cifrado, como un buen método de cifrado o un método confiable, pero se puede llegar a una conclusión, un sistema de cifrado se puede considerar como bueno cuando la seguridad de cifrado consiste en la clave y no en el algoritmo. Aunque se conozca

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

el algoritmo, no se puede llegar a un descifrado de la información gracias a la clave. La mayoría de las aplicaciones que se dan a la encriptación hoy en día son:

- Mensajes de autenticidad
- Facturas electrónicas
- Banca electrónica
- Votos electrónicos
- Notificaciones
- Mensajería instantánea
- Correos electrónicos
- Almacenamiento de información

Autoevaluación de Seguridad Informática

1. ¿Qué es una inyección SQL y como se puede prevenir?
2. ¿Qué es la gestión identidades y accesos (IAM, por sus siglas en inglés)?
3. ¿Qué es un ataque de ransomware y cómo se puede prevenir?
4. ¿Qué es el escaneo de puertos y cómo se puede prevenir?
5. ¿Qué es un cifrado?
6. ¿Qué es una red privada virtual (VPN) y como se utiliza en la seguridad informática?
7. ¿Qué es el cifrado homomórfico y cómo se utiliza en la seguridad informática?
8. ¿Cuál de las siguientes es una técnica de ataque de fuerza bruta?
9. ¿Cuál de las siguientes opciones no es una buena práctica de seguridad informática?
10. ¿Qué es un ataque de denegación de servicio (DoS)?
11. ¿Qué es la autentificación de dos factores?
12. ¿Qué es el pentesting?
13. ¿Cuál de las siguientes opciones NO es una fase típica del proceso del pentesting?
14. ¿Qué es un informe de pentesting?

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

15. ¿Los firewalls son programas de software que utilizan para proteger una red de posibles ataques externos? Verdadero o falso
16. ¿La ingeniería social es un método comúnmente utilizado por los hackers para obtener información confidencial? Verdadero o falso
17. ¿El cifrado de archivos es un método eficaz para proteger la información confidencial de posibles ataques? Verdadero o falso
18. ¿El pishing es una técnica de ataque que se utiliza para engañar a los usuarios para que instalen software malicioso en su sistema? Verdadero o falso
19. ¿Es seguro compartir contraseñas entre varias cuentas en línea, siempre y cuando se utilicen servicios de cifrado? Verdadero o falso
20. ¿La seguridad informática y la seguridad de la información tiene el mismo concepto? Verdadero o falso
21. ¿Los certificados SSL se utilizan para proteger la información de la transmisión no autorizada a través de internet y deshabilita la conexión cifrada? Verdadero o falso

Resumen de la Unidad 2

En esta unidad, se exploraron conceptos avanzados en el ámbito de la seguridad informática, centrándose en el hacking ético, la identificación de vulnerabilidades y el uso de la encriptación para proteger la información. Se examinaron los principios y prácticas del hacking ético, que incluyen el análisis de sistemas y redes con el fin de identificar y remediar vulnerabilidades de seguridad. Se discutió la importancia de este enfoque para fortalecer la seguridad de los sistemas informáticos y prevenir ataques cibernéticos maliciosos. Además, se estudiaron diferentes tipos de vulnerabilidades comunes, como fallos de seguridad en software y protocolos de red, y se exploraron técnicas de explotación utilizadas por los atacantes. Por último, se abordó el tema de la encriptación como una medida fundamental para proteger la confidencialidad de la información, examinando algoritmos y protocolos de encriptación, así como su aplicación en diferentes contextos, como la comunicación segura y el almacenamiento de datos sensibles. A lo largo de la unidad, se enfatizó la importancia de adoptar prácticas seguras y proactivas para mitigar riesgos y proteger la integridad, confidencialidad y disponibilidad de la información en entornos digitales.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC