



FACI

FACULTAD DE CIENCIAS E INGENIERIA

CARRERA:

INGENIERIA SOFTWARE

TEMA:

TAREAS TEÓRICAS

AUTOR:

➤ Danny Quito

ASIGNATURA:

INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE

DOCENTE:

DENIS DARIO MENDOZA CABRERA

CURSO:

B2-Primer semestre

PERIODO:

Agosto 2025 a Noviembre 2025

MILAGRO-ECUADOR

DEFINICIÓN DEL SOFTWARE

El hardware de un sistema informático se puede identificar fácilmente, ya que son los dispositivos físicos. El software, no obstante, es un poco más complicado de definir; en ocasiones se lo define por exclusión: Todo aquello que no es hardware, se considera software. El software comprende, claro está, los programas que dirigen el funcionamiento del sistema; pero también incorpora otros componentes como documentación, bases de datos o procesos de operación y mantenimiento regular, que son inmateriales.

TIPOS DE SOFTWARE

La clasificación más exhaustiva del software se puede hallar en la décima edición de "Ingeniería de Software" de Roger S. Pressman, en la que el software se divide en estas categorías principales:

Software de Sistemas

Lo componen todos los programas que se requieren para respaldar a otros, como los compiladores, los sistemas operativos o las aplicaciones de administración de redes. Su rasgo más característico es su elevado nivel de interacción con el hardware, porque en numerosas ocasiones tienen que administrar de manera eficaz cómo otros programas o usuarios acceden al hardware. Serían el sistema Windows, Linux, Android entre otros.

Software de Aplicación

Son aplicaciones creadas con el fin de solucionar dificultades particulares del comercio. En esta categoría, incluiríamos el software de administración de bancos o de

grandes empresas en general, como los ERP (Enterprise Resource Planning), que sería Planificación de Recursos Empresariales.

Software de Ingeniería y Ciencias

La meta es desarrollar sofisticados algoritmos matemáticos con el propósito de representar y simular sistemas o procesos complejos, como lo son, por ejemplo, las reacciones nucleares, los modelos meteorológicos, la red eléctrica de una nación o el diseño de un avión.

Software Incrustado

Se encuentra dentro de un producto o sistema y su propósito es regularlo y determinar su conducta. Se caracteriza por ser muy específico y de tamaño reducido, con la necesidad de funcionar en tiempo real. Desde el regulador de temperatura en una casa hasta el sistema de frenos de un automóvil, todos están controlados por este tipo de software.

Software de Línea de Producto

Su propósito es proporcionar una funcionalidad específica al consumir. Las aplicaciones de contabilidad para pequeñas empresas, las hojas de cálculo o los procesadores de texto son ejemplos de esta categoría.

Software de Aplicaciones de WEB (“WEBAPPS”)

Su uso se ha ampliado en años recientes gracias a la proliferación de dispositivos móviles que tienen acceso a redes. Al principio, solo se trataban de archivos de hipertexto para presentar información; no obstante, en la actualidad tienen la capacidad de computación y están integradas con bases de datos y aplicaciones corporativas. Por medio de ellas se puede manejar una cuenta bancaria, efectuar cualquier tipo de compras, jugar juegos muy sofisticados o averiguar el clima en cualquier lugar del planeta. Para que tengan éxito, la rapidez, la comodidad y la vistosidad son factores decisivos.

Software de Inteligencia Artificial (I.A)

Las aplicaciones de robótica, la visión artificial con inteligencia artificial y las redes neuronales son parte del software de IA. Para resolver problemas, emplean algoritmos que no son numéricos, como los árboles lógicos de búsqueda.

Software en la Nube

El software en la nube posibilita brindar servicios de informática por medio de internet, como el alojamiento de información, los servidores o la ejecución remota del software. El software en la nube ha representado un nuevo enfoque para utilizar los recursos tecnológicos, con el objetivo de conseguir una mayor eficiencia y unos gastos más reducidos. Hay ambientes de desarrollo que posibilitan la programación en la nube directamente.

EJEMPLOS COTIDIANOS DE SOFTWARE

Software de Televisión(TV)

Por ejemplo, el de Smart TV emplea un sistema operativo integrado en la pantalla, como WebOS o Tizen (sistemas operativos), que posibilitan conectarse a internet, instalar aplicaciones y enlazarse con dispositivos, entre otras capacidades adicionales.

Software de Autos Modernos

La mayor parte y prácticamente todos los automóviles actuales utilizan software para gestionar sistemas de infoentretenimiento, navegación GPS, sensores de seguridad y conexión con teléfonos celulares. Aplicaciones como Tesla OS y Android Auto proporcionan características sofisticadas, que van desde la asistencia en la conducción hasta la reproducción de música.

Software de ATM

Los cajeros automáticos son capaces de realizar su funcionamiento completo a través del software; este gestiona las operaciones bancarias, asegura la seguridad, conecta con los servidores y hace más fáciles las transacciones financieras.

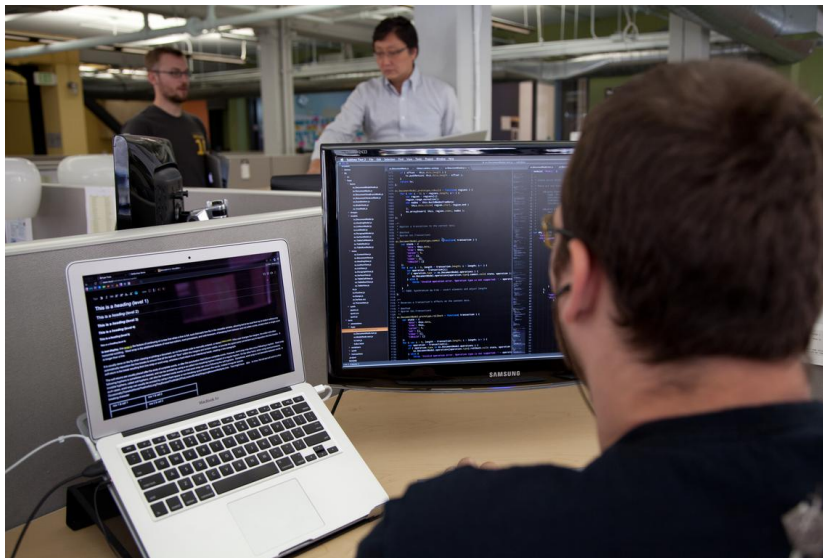
INGENIERIA DE SOFTWARE

La ingeniería de software es una rama de la ingeniería que tiene como objetivo el desarrollo de sistemas informáticos a un costo asequible. Este es inmaterial y abstracto. No está limitado por materiales, ni regido por procesos de manufactura o

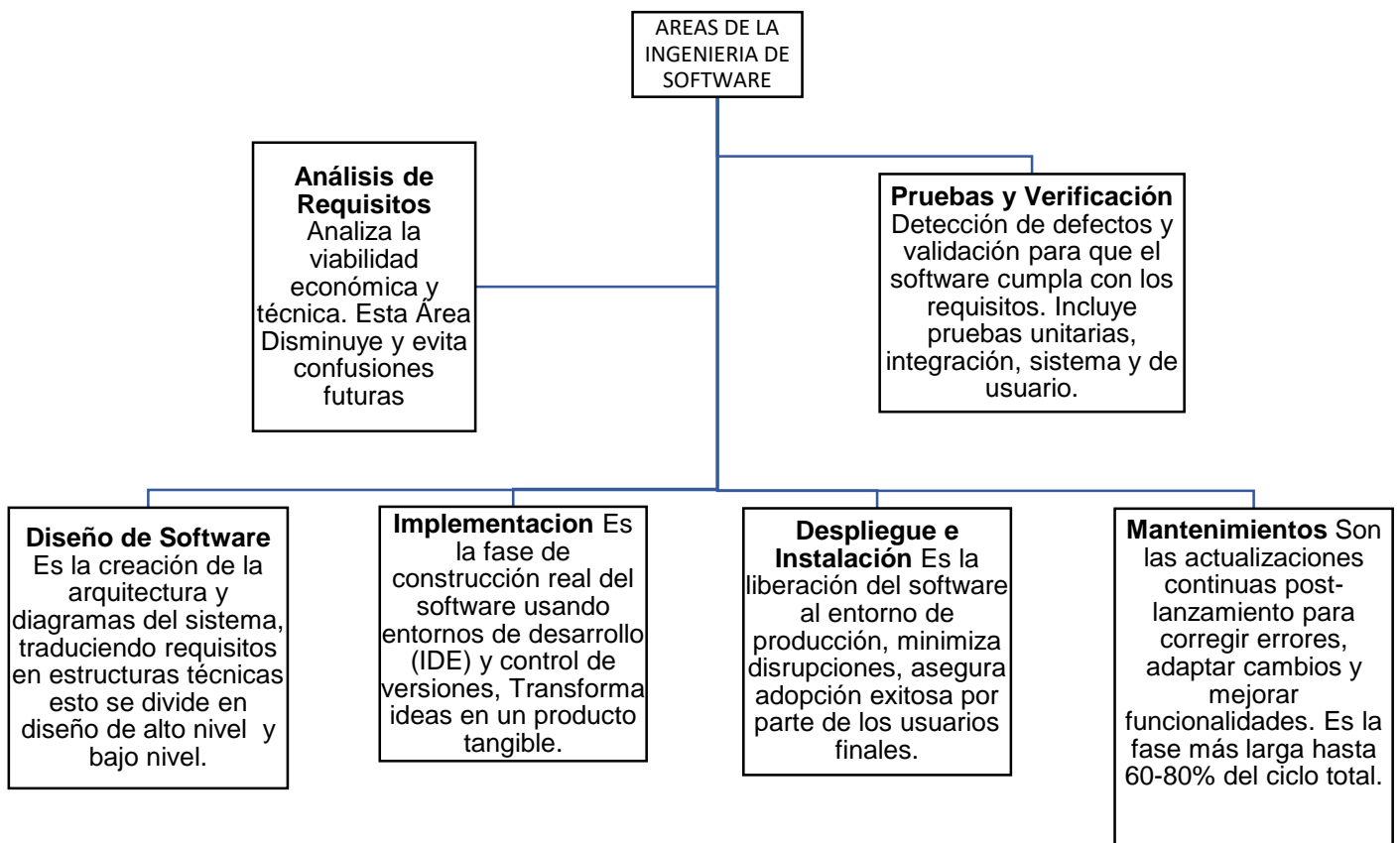
leyes físicas. Sin embargo, la ausencia de limitaciones naturales puede resultar en un software muy complejo y, por ende, difícil de comprender.

TAREAS QUE REALIZA UN INGENIERO DE SOFTWARE

Al examinar la abundante información contenida en libros e incluso las muestras diarias de nuestra vida cotidiana, podemos comprender que un ingeniero de software se encarga de diseñar, desarrollar y mantener los programas y sistemas de información. Sus funciones o tareas abarcan examinar las necesidades del cliente, diseñar arquitecturas de software, programar principalmente, ejecutar pruebas y, sobre todo, corregir fallos y documentar procedimientos. Además de trabajar con grupos de colegas y equipos multidisciplinarios, todo con el fin de mejorar la eficacia de las aplicaciones, garantizar la calidad del producto y usar metodologías ágiles para ofrecer a los clientes soluciones tecnológicas eficaces y seguras.



A continuación, presento un **Mapa Mental** con las Áreas de la Ingeniería de Software:



DILEMAS ETICOS QUE PUEDEN ENFRENTAR LOS INGENIEROS SOFTWARE

Privacidad y Recolección de Datos

Muchas aplicaciones recopilan datos personales de los usuarios, a menudo sin su consentimiento explícito o conocimiento claro sobre cómo se usarán. Esto plantea preguntas sobre el derecho a la privacidad frente a los intereses comerciales o de seguridad. Preguntémonos es ético que una aplicación rastree la ubicación de un usuario sin notificarle claramente, pensándolo incluso si esos datos se usan para el usuario mismo o para mejorar el servicio del mismo es correcto almacenar los datos en un espacio donde el usuario no podrá eliminar cuando quiera.

Uso de software para fines maliciosos

Uno de los dilemas más comunes que enfrenta un desarrollador de software, es crear una APP de reconocimiento facial sabiendo que podría utilizarse para vigilar a las poblaciones vulnerables. Por esa misma razón los desarrolladores y empresas deben evaluar el impacto potencial de sus creaciones y establecer límites éticos claros sobre su uso cuando se trata sobre todo con usuarios.

ENSAYO BREVE CON ANALISIS DE LOS DILEMAS

¿Cómo debería actuar un Profesional según el código de Ética?

Primeramente, debemos saber que es la Ética en Software, hay que entender que son los principios y normas morales, una vez entendiendo la Ética nos planteamos en los dilemas que analizaremos brevemente.

Un ingeniero de Software debe ser Ético y profesional en todo momento eso incluye en la vida cotidiana, ya que la “Privacidad y datos” de nuestros usuarios, clientes, incluso sin que haya un contrato establecido personalmente, deben mantenerse en total confidencialidad, no recopilar datos innecesarios y asegurar que estén resguardados, sin abusar de la funcionalidad de las aplicaciones que se crean para fines lucrativos o públicos. Y así mismo respecto al segundo dilema no Ético, cuando hablamos del uso de Software malicioso como habíamos mencionado anteriormente, sobre los principios Éticos que un Ingeniero debe tener en cuenta, no se debe abusar de las vulnerabilidades ni desventajas de las aplicaciones para ningún fin que involucre robar o dañar a nuestros usuarios, esto incluye tomar medidas preventivas para resguardar la integridad de nuestros clientes que podrían ser nuestras mismas familias. En conclusión, tanto en dilemas de Privacidad de datos o los Software maliciosos, debemos tener conciencia, es decir pedir solo los datos necesarios y tomar medidas de seguridad para que no haya vulnerabilidad en las aplicaciones.

MODELOS DE DESARROLLO DE SOFTWARE

Modelo en Cascada

Se trata de un proceso de desarrollo secuencial, en el cual el desarrollo fluye hacia abajo sobre las etapas que forman parte del ciclo vital. Es un enfoque metodológico que organiza de manera estricta las fases del ciclo de vida del software, de modo que el comienzo de cada etapa, debe esperar hasta que termine la justa anterior.

Modelo Iterativo

Implica la repetición de múltiples ciclos de vida en cascada. Al concluir cada iteración, el cliente recibe una versión mejorada o con más funciones. El cliente es la persona que, luego de cada iteración, examina el producto y lo ajusta o sugiere mejoras. Estas iteraciones continuarán hasta que se logre un producto que cumpla con lo que necesita el cliente. Intenta disminuir el riesgo que se presenta entre las necesidades del cliente y el producto final debido a malentendidos en la fase de recolección de requisitos.

Modelo Ágil

Se trata de un método que se basa en la cooperación y la adaptabilidad, el cual es incremental e iterativo. Segmenta el desarrollo en iteraciones, breves saltos, que normalmente duran de una a cuatro semanas, y al final de cada una entrega un software funcional. Se guía por el Modelo Ágil, dándole prioridad a la colaboración con los clientes, a las personas y sus interacciones, al software que funciona y a la capacidad de adaptarse al cambio.

TABLA COMPARATIVA DE LOS TRES MODELOS

	MODELO EN CASCADA	MODELO ITERATIVO	MODELO AGIL
Descripción	Es un proceso secuencial en el cual el desarrollo fluye hacia abajo sobre las etapas que forman parte del ciclo vital	Implica la repetición de múltiples ciclos de vida en cascada	trata de un método que se basa en la cooperación y la adaptabilidad, el cual es incremental.
Ventajas	<ul style="list-style-type: none"> Modelo en el que todo marcha bien ordenado. Las etapas no se combinan. Sencillo y fácil de implementar. 	<ul style="list-style-type: none"> Desarrollo en ciclos breves. Es más sencillo manejar los riesgos. 	<ul style="list-style-type: none"> Flexibilidad amplia para incluir modificaciones en los requerimientos. Optimiza la cooperación entre el equipo y el cliente.
Desventajas	<ul style="list-style-type: none"> Los proyectos, con poca frecuencia, siguen una secuencia lineal y resulta complicado determinar todos los requisitos iniciales. 	<ul style="list-style-type: none"> Los requisitos no existen establecidos desde el inicio tampoco es posible visualizarlo como un inconveniente porque tienen la capacidad de aparecer dificultades con la arquitectura. 	<ul style="list-style-type: none"> Precisa de una gran participación del cliente, lo cual no siempre es factible. Si no se gestiona de manera adecuada, puede producir documentación restringida.

Uso de Modelos	Es factible en Hospitales, en la gestión de citas y un cronograma estricto.	Una universidad requiere una gestión de estudiantes.	Un negocio quiere vender productos personalizados.
----------------	---	--	--

CASO HIPOTETICO USANDO UN MODELO

En un negocio que requiere vender productos personalizados como la vestimenta hecha a medida se usa un “Método de desarrollo Ágil” ya que los requerimientos iniciales son imprecisos porque la empresa desea ajustarse pronto a las tendencias del mercado y los gustos de los usuarios, esto es perfecto para aplicar el Método Ágil ya que es flexible para incluir modificaciones. El equipo tiene que lanzar una versión mínima viable en unos pocos meses y actualizarla con regularidad según la retroalimentación y esto es posible con este método.

Método Ágil	Breve Explicación	Aplicación en el Caso
Planificación	Se define una lista priorizada de funcionalidades con el cliente, identificando las más valiosas para el MVP. Se establece la visión general del proyecto.	El negocio y el equipo de desarrollo acuerdan una lista de pendientes con funciones clave: catálogo de productos, carrito de compras y sistema de pagos. Se prioriza el catálogo para el MVP.

Bibliografía

SEBATIAN RUBEN GOMEZ PALOMO

APROXIMACION A LA INGENIERIA DEL SOFTWARE 2-EDI. EDITORIAL

UNIVERSITARIA RAMON ARECES. GOOGLE ACADEMICO. SACADO

DEL:https://books.google.com.ec/books?hl=es&lr=&id=8wnUDwAAQBAJ&oi=fnd&pg=PA5&dq=related:nrI50OmMaZ0J:scholar.google.com/&ots=D6xAVraZNg&sig=VtZNSaJnDWgdi68Jeu2p8xtSMoQ&redir_esc=y#v=onepage&q&f=false

IAN SOMMERVILLE

INGENIERIA DEL SOFTWARE. 7MA- EDI. GOOGLE ACADEMICO. SACADO

DEL:https://books.google.com.ec/books?hl=es&lr=&id=gQWd49zSut4C&oi=fnd&pg=PA1&dq=Software+&ots=s948wsAutf&sig=y4KkWLu9zL76F6-ehC3tI_gPLqM&redir_esc=y#v=onepage&q&f=false