



Universidad
Católica del
Uruguay

Introducción a la Ingeniería de software

Contenido

1	INTRODUCCIÓN	3
2	SOFTWARE	3
2.1	COMPONENTES DEL SOFTWARE	3
2.2	CARACTERÍSTICAS DEL SOFTWARE	4
2.3	ATRIBUTOS DE UN BUEN SOFTWARE	4
2.4	CLASIFICACIONES DEL SOFTWARE	5
2.4.1	<i>Utilización del software y su funcionalidad</i>	5
2.4.2	<i>El tratamiento comercial que tiene</i>	6
2.4.3	<i>Exigencias en eficiencia y los factores críticos que se le exigen</i>	6
3	INGENIERÍA DE SOFTWARE	6
3.1	DEFINICIÓN DE INGENIERÍA DEL SOFTWARE	7
3.2	HISTORIA	7
3.3	CAPAS DE LA INGENIERÍA DE SOFTWARE	8
3.3.1	<i>Enfoque de Calidad</i>	8
3.3.2	<i>Métodos</i>	9
3.3.3	<i>Procesos</i>	9
3.3.4	<i>Herramientas</i>	10
4	RESUMEN DE CONCEPTOS	11
5	BIBLIOGRAFÍA	12

1 Introducción

La ingeniería de software es una disciplina que provee métodos y técnicas para desarrollar y mantener software de calidad. Surge a partir de la necesidad de sistematizar la creación de software, proceso intrínsecamente creativo, proveyendo así un enfoque sistemático, cuantificado y disciplinado que permite monitorear el proyecto y disminuir la posibilidad de fracaso.

Entre los objetivos de la ingeniería de software se encuentran:

- Mejorar el diseño de aplicaciones o software de tal modo que se adapten de mejor manera a las necesidades de las organizaciones o finalidades para las cuales fueron creadas.
- Promover mayor calidad al desarrollar aplicaciones complejas.
- Brindar mayor exactitud en los costos de proyectos y tiempo de desarrollo de los mismos.
- Aumentar la eficiencia de los sistemas al introducir procesos que permitan medir, mediante normas específicas, la calidad del software desarrollado, buscando siempre la mejor calidad posible según las necesidades y resultados que se quieren generar.
- Generar una mejor organización de equipos de trabajo en el área de desarrollo y mantenimiento de software.
- Detectar, a través de pruebas, posibles mejoras para un mejor funcionamiento del software desarrollado.

2 Software

El software es aquella parte de un sistema computacional que consiste de datos o instrucciones, en contraste con el hardware, que son los componentes físicos en los cuales se construye el sistema.

Hoy en día, es una parte integral de la mayoría de los sistemas y es un importante diferenciador de negocio.

- **IEEE Std. 610** define el software como *“programas, procedimientos y documentación y datos asociados, relacionados con la operación de un sistema informático.”*
- Según el **Webster’s New Collegiate Dictionary**, *“software es un conjunto de programas, procedimientos y documentación relacionada asociados con un sistema, especialmente un sistema informático”*.

2.1 Componentes del software

Existe la percepción de que el software sólo consiste en programas. Sin embargo, esto no es así. Podemos definir al software como el conjunto de tres componentes:



Figura 1 Componentes del Software

1. **Programas:** los programas son conjuntos de instrucciones que proporcionan la funcionalidad deseada cuando son ejecutadas por el ordenador. Están escritos usando lenguajes específicos que los ordenadores pueden leer y ejecutar.
2. **Datos:** los programas proporcionan la funcionalidad requerida manipulando datos. Usan datos para ejercer el control apropiado en lo que hacen. El mantenimiento y las pruebas de los programas también necesitan datos. El diseño del programa asume la disponibilidad de las estructuras de datos tales como bases de datos y archivos que contienen datos.
3. **Documentación:** además de los programas y los datos, los usuarios necesitan también una explicación de cómo usar el programa. Documentos como manuales de usuario y de operación son necesarios para permitir a los usuarios operar con el sistema. Los documentos también son requeridos por las personas encargadas de mantener el software para entender el interior del software y modificarlo, en el caso en que sea necesario.

2.2 Características del software

Es importante examinar las características del software que lo diferencian de otras cosas que el hombre puede construir:

- El software es esencialmente un conjunto de instrucciones (programas) que proporcionan la funcionalidad requerida, los datos relacionados y documentos. Por lo tanto, **el software es un elemento lógico** y se diferencia del hardware, un elemento físico, en sus características.
- El **software se desarrolla, no se fabrica** en el sentido clásico. Aunque existen similitudes entre el desarrollo del software y la construcción del hardware, ambas actividades son fundamentalmente distintas. El proceso usado para construir software es diferente de la fabricación del hardware, donde las máquinas se usan para producir partes y cada trabajador sólo necesita realizar la tarea asignada o usar una máquina.
- Cada producto software es diferente porque se construye para cumplir los **requisitos únicos de un cliente**. Cada software necesita, por lo tanto, ser construido usando un enfoque de ingeniería.
- Construir un producto software implica **entender** qué es necesario, **diseñar** el producto para que cumpla los requisitos, **implementar** el diseño usando un lenguaje de programación y **comprobar** que el producto cumple con los requisitos. Todas estas actividades se llevan a cabo mediante la ejecución de un proyecto software y requiere un equipo trabajando de una forma coordinada.
- En el software, **el recurso principal son las personas**. No es siempre posible acelerar la construcción de software añadiendo personas porque la construcción de software requiere un esfuerzo en equipo. El equipo tiene que trabajar de forma coordinada y compartir un objetivo de proyecto común. Se necesita comunicación efectiva dentro del equipo. Un nuevo miembro del equipo no es inmediatamente productivo, se necesita la iniciación adecuada al equipo y la formación para realizar el trabajo. Esto requiere una inversión de tiempo y esfuerzo por parte de los miembros del equipo existentes lo que a su vez puede distraer en el cumplimiento de su propio trabajo.
- **El software no se estropea**. Los defectos no detectados harán que falle el programa durante las primeras etapas de su vida. Sin embargo, una vez que se corrigen (suponiendo que no se introducen nuevos errores) los fallos disminuyen.
- **Si bien el software no se estropea, se deteriora**. Durante su vida, el software sufre cambios (mantenimiento). Conforme se hacen los cambios, es bastante probable que se introduzcan nuevos defectos, lo que hace que el software se vaya deteriorando debido a los cambios.
- **El software es re-utilizable**, permitiendo al ingeniero concentrarse en elementos verdaderamente innovadores de un diseño. Como la industria del hardware, la del software está intentando adoptar el mecanismo de reutilizar para hacer más fácil y más rápida la construcción. Existen algunos elementos reutilizables a través de librerías de funciones y objetos reutilizables que combinan funciones y datos.

2.3 Atributos de un buen software

Todo buen software debe proporcionar la funcionalidad y el rendimiento requeridos a los usuarios de forma sostenible, fiable y aceptable.

1. **Mantenibilidad:** el software debe ser capaz de mantenerse y evolucionar para cumplir con las necesidades de cambio.
2. **Fiabilidad:** el software debe ser confiable al ser ejecutado.
3. **Eficiencia:** el software no debe hacer mal uso (derroche) de los recursos del sistema.
4. **Aceptabilidad:** el software debe ser aceptado por los usuarios para los que se diseñó. Esto significa que ha de ser entendible, usable y compatible con otros sistemas.

2.4 Clasificaciones del software

2.4.1 Utilización del software y su funcionalidad

A continuación se describen categorías genéricas para las aplicaciones del software:

Software de sistemas	<p>El software de sistemas es un conjunto de programas que han sido escritos para servir a otros programas. Algunos, como los compiladores y editores, procesan estructuras de información que son complejas pero determinadas. Otros procesan datos, en gran medida indeterminados, como ciertos componentes del sistema operativo, utilidades de manejo de periféricos y procesadores de telecomunicaciones.</p> <p>En cualquier caso, el área del software de sistemas se caracteriza por tener:</p> <ul style="list-style-type: none">• Fuerte interacción con el hardware de la computadora• Gran utilización por múltiples usuarios• Operación concurrente que requiere una planificación• Compartición de recursos y una sofisticada gestión de procesos• Estructuras de datos complejas y múltiples interfaces externas
Software de tiempo real	<p>Es el software que coordina/analiza/controla sucesos del mundo real conforme ocurren. Posee elementos como:</p> <ul style="list-style-type: none">• Componente de adquisición de datos que recolecta y da formato a la información recibida del entorno externo• Componente de análisis que transforma la información según lo requiera la aplicación• Componente de control/salida que responda al entorno externo• Componente de monitorización que coordina todos los demás componentes, de forma que pueda mantenerse el respuesta en tiempo real
Software de gestión	<p>El proceso de la información comercial constituye la mayor de las áreas de aplicación del software. Los sistemas discretos como nóminas, cuentas de haberes-débitos, inventarios, etc., han evolucionado hacia el software de sistemas de información de gestión (SIG) que accede a una o más bases de datos que contienen información comercial.</p> <p>Las aplicaciones en esta área reestructuran los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones. Además de las tareas convencionales de procesamiento de datos, las aplicaciones de software de gestión también realizan cálculo interactivo (por ejemplo: el procesamiento de transacciones en puntos de venta).</p>
Software de ingeniería y científico	<p>Este tipo de software está caracterizado por los algoritmos de manejo de números. Las aplicaciones van desde la astronomía a la vulcanología, desde el análisis de la presión de los automotores a la dinámica orbital de las lanzaderas espaciales y desde la biología molecular a la fabricación automática.</p> <p>Las nuevas aplicaciones del área de ingeniería/ciencia se han alejado de los algoritmos convencionales numéricos. El diseño asistido por computadora (CAD), la simulación de sistemas y otras aplicaciones</p>

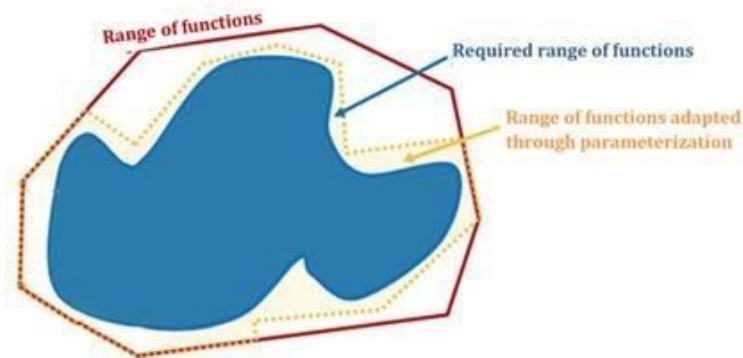
	interactivas han comenzado a coger características del software de tiempo real e incluso de software de sistemas.
Software empotrado	<p>Los productos inteligentes se han convertido en algo común en casi todos los mercados de consumo e industriales. El software empotrado reside en memoria de sólo lectura y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo.</p> <p>El software empotrado puede ejecutar funciones muy limitadas y curiosas (por ejemplo: el control de las teclas de un horno microondas) o suministrar una función significativa y con capacidad de control (por ejemplo: funciones digitales en un automóvil, tales como control de la gasolina, indicadores en el salpicadero, sistemas de frenado, etc.)</p>
Software de computadoras personales	<p>El mercado del software de computadoras personales ha germinado en las pasadas décadas.</p> <p>El procesamiento de textos, las hojas de cálculo, los gráficos por computadora, multimedia, entretenimiento, gestión de bases de datos, aplicaciones financieras, de negocios y personales y redes o acceso a bases de datos externas son algunas de los cientos de aplicaciones.</p>
Software basado en web	Las páginas web visitadas por un explorador son software que incorpora instrucciones ejecutables y datos.
Software de inteligencia artificial	<p>El software de inteligencia artificial hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo.</p> <p>Los sistemas expertos, también llamados sistemas basados en el conocimiento, reconocimiento de patrones (imágenes y voz), redes neuronales artificiales, prueba de teoremas y los juegos son representativos de las aplicaciones de esta categoría.</p>

Tabla 1 - Clasificación de software según su funcionalidad

2.4.2 El tratamiento comercial que tiene

Los productos software pueden desarrollarse para un cliente particular o para un mercado general. Por lo tanto, los productos software pueden ser:

- **Paquetes desarrollados para ser vendidos a un ámbito de clientes diferentes**
 - Constituyen a la mayoría de los sistemas consumidos por usuarios promedio. Son aquellos programas que se desarrollan para un público en general, sin importar el hardware o sistema operativo (aunque esto es relativo) que posea el equipo.
 - Es el tipo de software menos personalizable, ya que todos los usuarios poseen una misma versión del producto.
- **Paquetes parametrizables o paramétricos**
 - Hace referencia al tipo de software que puede adaptado al rango deseado de funciones mediante el establecimiento de parámetros, siempre dentro del rango provisto por el fabricante.



- o Por lo general, se posee un programa con un conjunto significativo de funcionalidades, de las cuales se requiere un subconjunto. Se pueden parametrizar aspectos como:
 - Definición de roles y autorizaciones
 - Creación de usuarios y asignación de roles
 - Definición de procesos y adaptación de flujos de trabajo.
 - Configuración de interfaces
 - Ajustes de cálculo de puntajes
 - Definición de advertencias y límites de alarma.
- o El término proviene del ámbito matemático y hace referencia al uso de parámetros o variables que permiten manipular o alterar el resultado final de una ecuación o sistema.
- o Este tipo de software es más personalizable que el anterior, ya que permite adaptar el producto a las necesidades de negocio.
- **Software a medida (personalizados)**
 - o Este tipo de software es desarrollado para un cliente individual de acuerdo a su especificación.
 - o Suele emplearse para tareas especializadas o de alta precisión, donde los requisitos de performance o uso son vitales para la tarea a realizar.
 - o Es el software más personalizable de todos, ya que literalmente está hecho a la medida del cliente.

2.4.3 Exigencias en eficiencia y los factores críticos que se le exigen

- **Software de tiempo real (STR)**



- o Se denomina software de tiempo real a todo tipo de sistema informático capaz de reaccionar a estímulos externos con una gran velocidad, al tal punto que la toma de decisión se realiza de forma casi inmediata.(micro o milisegundos)
- o En un sistema en tiempo real, el correcto comportamiento del sistema depende no solo de los resultados lógicos de los cálculos, sino del instante físico en el que estos se producen. Un sistema en tiempo real cambia su estado en función del tiempo físico, por ejemplo, una reacción química continúa cambiando su estado incluso después de que su sistema informático de control se haya detenido.
- o Este tipo de sistema debe reaccionar a los estímulos del objeto controlado (o del operador) dentro de los intervalos de tiempo dictados por su entorno.
- o Algunos ejemplos incluyen: software de control de tránsito aéreo, dispositivos de control de signos vitales (como los marcapasos), sistemas de transacciones en línea,

controladores de plantas nucleares o químicas, sistemas de frenos en automóviles modernos.

- **Transaccional**

- o Un sistema transaccional es un tipo de sistema de información diseñado para recolectar, almacenar, modificar y recuperar todo tipo de información que es generada por las transacciones en una organización. Los hay de dos tipos:

- En línea (online)



- Es un tipo de procesamiento que facilita y administra aplicaciones transaccionales, usualmente para entrada de datos y recuperación y procesamiento de transacciones (gestor transaccional).
- Este tipo de software suele poseer una arquitectura cliente-servidor debido a que generalmente las empresas lo utilizan en una red informática distribuida.
- Son sistemas de respuesta corta (2 o 3 segundos)
- Algunos ejemplos incluyen: sistemas de banca en línea, de compra por internet, software de atención al cliente de forma presencial, cajeros electrónicos (ATM), etc.

- Por lotes (Batch)



- Un software por lotes es aquel que realiza su ejecución sin la supervisión o control de un usuario. Por lo general se lo emplea para ejecutar tareas repetitivas sobre conjuntos grandes de datos.
- Casi cualquier programa puede ejecutar en modo batch, siempre y cuando pueda especificarse los distintos pasos de ejecución o las entradas de usuario a partir de un script.
- Algunos ejemplos incluyen: “renderizado” de una película, generación de extractos bancarios, cálculo de intereses u otros índices económicos, generación automática de archivos, entre otros.
- Tiempo ejemplo 20 minutos, u horas.

3 Ingeniería de software

La ingeniería del software es una disciplina de ingeniería preocupada por todos los aspectos de la producción y desarrollo de software rentable, desde las primeras etapas de especificación del sistema hasta su mantenimiento después de que haya sido puesto en uso. Provee teorías, métodos y herramientas para el desarrollo profesional de software.

La ingeniería del software debería adoptar un enfoque sistemático y organizado para su trabajo y usar las herramientas y técnicas apropiadas dependiendo del problema a solucionar, las restricciones de desarrollo y los recursos disponibles. Este enfoque es con frecuencia calificado de modelo de proceso de software (en el sentido general) o de proceso de desarrollo de software (en el sentido específico).

El proceso de desarrollo de software específico consiste en un conjunto particular de **prácticas de desarrollo de software** que son realizadas por el ingeniero de software en un orden predeterminado.

Los ingenieros han de adoptar un enfoque sistemático y organizacional para su trabajo. Cuando se habla de **prácticas de desarrollo de software** se hace referencia a un requisito empleado para recomendar un enfoque disciplinado y uniforme del proceso de desarrollo de software, es decir, una actividad bien definida que contribuye a la satisfacción de los objetivos del proyecto. Generalmente, la salida de una práctica se convierte en la entrada de otra.

Entre las prácticas de desarrollo de software se encuentran las siguientes (las prácticas pueden depender en base al proceso y la terminología asociada al mismo):

- **Elección de Ciclo de Vida**
- **Gestión de proyecto**
- **Desarrollo de software**
 - Ingeniería de requisitos
 - Análisis de sistemas
 - Diseño
 - Implementación e implantación
 - Mantenimiento
- **Procesos integrales**
 - Gestión de configuración
 - Verificación y validación
 - Documentación
 - Capacitación

3.1 Definición de Ingeniería del Software

La **ingeniería del software** es la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software, que es la aplicación de la ingeniería del software (IEEE, 1990).

El término ingeniería de software puede definirse como:

1. Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía. **DRAE** (Diccionario de la Real Academia Española).
2. Ingeniería del Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software. **(Zelkovitz)**.
3. Ingeniería del Software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software. **(Bohem)**.
4. Ingeniería del Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales. **(Bauer)**.
5. La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de ingeniería al software. **(IEEE)**.

3.2 Historia

El término ingeniería del software apareció por primera vez en la conferencia de ingeniería de software de la OTAN en 1968 y fue mencionado para provocar el pensamiento sobre la crisis de software del momento. Desde entonces, ha continuado como una profesión y campo de estudio dedicado a la creación de software de alta calidad, barato, con capacidad de mantenimiento y rápido de construir.

Debido a que el campo es todavía relativamente joven comparado con otros campos de la ingeniería, existe todavía mucho debate sobre qué es realmente la ingeniería del software y si se merece el título

de ingeniería. Ha crecido orgánicamente fuera de las limitaciones de ver el software sólo como programación.

3.3 Capas de la Ingeniería de Software

La ingeniería del software es una tecnología multicapa. Se puede ver como un conjunto de componentes estratificados, que reposan sobre ese enfoque de calidad.



Dichos componentes son:

- **Enfoque de calidad:** El enfoque de ingeniería del software cuenta con un compromiso organizacional con la calidad porque no es posible incorporar la ingeniería del software en una organización que no está centrada en conseguir calidad.
- **Métodos:** las actividades técnicas requeridas para la creación de productos de trabajo.
- **Procesos:** un marco de trabajo que ayuda al jefe de proyecto a controlar la gestión del proyecto y las actividades de ingeniería.
- **Herramientas:** la ayuda automatizada para los procesos y métodos.

3.3.1 Enfoque de Calidad

Dado que la competitividad entre organizaciones es cada día más fuerte se torna necesario que las empresas se preocupen por crear mejores productos. Para ello, han de ser capaces de desarrollar y entregar software confiable, en tiempo y forma, y conforme con los requisitos y presupuesto acordado con el cliente.

No sólo la competitividad es importante, también se ha de tomar conciencia y responsabilidad de las consecuencias que un defecto en el producto pueda ocasionar. Ha habido casos en que personas han muerto debido a un defecto de software. A su vez, se debe considerar que cada defecto detectado representa un costo adicional.

Las principales razones de contar con un enfoque de calidad son:

- Satisfacción del cliente
- Competencia
- Errores

En el curso se mencionarán certificaciones internacionales como ser ISO y CMMi, además de estándares de la IEEE.

3.3.1.1 Gestión de calidad de software de la ISO 9000

Se trata de un conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta por medios tales como:

- Planificación de la calidad.
- Control de la calidad.
- Aseguramiento (garantía) de la calidad.
- Mejora de la calidad.

3.3.1.2 CMMi

Los trabajos y aportaciones realizadas por el Instituto de Ingeniería del Software a la Ingeniería del software (SEI) de la Universidad Carnegie Mellon son referente mundial de primer orden, siendo la aportación más significativa los modelos de madurez de las capacidades: CMM y CMMi;

CMMi es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

3.3.1.3 Estándares de la IEEE

IEEE ha desarrollado estándares para todas las áreas de Ingeniería del Software.

Algunos de ellos, correspondientes a las principales áreas específicas de la Ingeniería del Software, por ejemplo El modelo de proceso (1074) entre otros.

3.3.2 Métodos

La construcción de software implica una amplia colección de actividades técnicas. La capa de métodos contiene los métodos definidos para realizar esas actividades de forma eficiente. Se centra en las **actividades técnicas** que se deben realizar para conseguir las tareas de ingeniería. Proporciona el “cómo” y cubre las actividades de ingeniería fundamentales. Las personas involucradas usan los métodos para realizar las actividades de ingeniería fundamentales necesarias para construir el software.

Los métodos dependen de un conjunto de principios básicos que gobiernan cada una de las áreas de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas. Abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento.

Para varias actividades de proceso, la capa de métodos contiene el correspondiente conjunto de métodos técnicos para usar. Esto abarca un conjunto de reglas, los modos de representación gráficos o basados en texto, y las guías relacionadas para la evaluación de la calidad de la información representada. Para definir la capa de métodos, es necesario seleccionar un método adecuado de un amplio rango de métodos disponibles.

3.3.3 Procesos

El fundamento de la ingeniería del software es la **capa de proceso**, proveyendo un **marco de trabajo** y definiendo los procesos que se emplearán para construir el software, las actividades y tareas que un jefe de proyecto tiene que gestionar, estableciendo así la entrega efectiva de la tecnología de la ingeniería del software.

Las áreas claves del proceso forman la base del control de gestión de proyectos del software y establecen el contexto en el que se aplican los métodos técnicos, se obtienen productos de trabajo (modelos, documentos, datos, informes, formularios, etc.), se establecen hitos, se asegura la calidad y el cambio se gestiona adecuadamente.

El proceso de la ingeniería del software es la unión que mantiene juntas las capas de tecnologías y que permite un desarrollo racional y oportuno de la ingeniería del software. Es el fundamento de la ingeniería del software y da soporte a las capas de métodos y herramientas.

Todas las actividades, incluyendo aquellas de índole técnica, pueden ser visualizadas como parte del proceso. Lo mismo es cierto para cualquier recurso, incluyendo las herramientas usadas para construir el software.

Todos los enfoques de la construcción de software tienen un proceso, pero en muchos casos, son ad hoc, invisibles y caóticos. Una buena ingeniería de software hace que el proceso de software sea más visible, predecible y más útil para aquellos que construyen software.

3.3.4 Herramientas

La capa de herramientas proporciona soporte a las capas de proceso y métodos centrándose en el significado de la automatización de algunas de las actividades manuales.

Pueden utilizarse para automatizar las siguientes actividades:

- Actividades de gestión de proyectos.
- Métodos técnicos usados en la ingeniería del software.
- Soporte de sistemas general.
- Marcos de trabajo para otras herramientas.

La automatización ayuda a eliminar el tedio del trabajo, reduce las posibilidades de errores, y hace más fácil usar buenas prácticas de ingeniería del software. Cuando se usan herramientas, la documentación se convierte en una parte integral del trabajo hecho, en vez de ser una actividad adicional.

Las herramientas se pueden utilizar para realizar actividades de gestión de proyecto así como para actividades técnicas.

Existen una gran variedad de herramientas para múltiples actividades. Entre ellas se pueden destacar las siguientes:

- Herramientas de gestión de proyectos.
- Herramientas de control de cambios.
- Herramientas de análisis y diseño.
- Herramientas de generación de código.
- Herramientas de pruebas.
- Herramientas de documentación.
- Herramientas de prototipos.

Estas herramientas soportan las capas de proceso y de métodos en varias actividades.

4 Resumen de conceptos

<i>¿Qué es software?</i>	<i>Programas de computador y la documentación asociada. Los productos de software se pueden desarrollar para un cliente particular o para un mercado en general.</i>
<i>¿Qué es la Ingeniería de Software?</i>	<i>Es una disciplina de ingeniería que comprende todos los aspectos de la producción de software.</i>
<i>¿Cuál es la diferencia entre ingeniería de software y ciencia de la computación?</i>	<i>La ciencia de la computación comprende la teoría y los fundamentos; la ingeniería de software comprende las formas prácticas para desarrollar y entregar un software útil.</i>
<i>¿Cuál es la diferencia entre ingeniería de software e ingeniería de sistemas?</i>	<i>La ingeniería de sistema se refiere a todos los aspectos del desarrollo de los sistemas informáticos, incluyendo hardware, software e ingeniería de procesos. La ingeniería de software es parte de este proceso.</i>
<i>¿Qué es un modelo de procesos del software?</i>	<i>Una representación simplificada de un proceso del software, presentada desde una perspectiva específica.</i>
<i>¿Cuáles son los costos de la ingeniería de software?</i>	<i>A grandes rasgos, el 60% de los costos son el desarrollo, el 40 % restante son de pruebas. En el caso de software personalizado, los costos de evolución a menudo exceden los del desarrollo.</i>
<i>¿Qué son los métodos de la Ingeniería de Software?</i>	<i>Enfoques estructurados para el desarrollo de software que incluyen modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos.</i>
<i>¿Qué es CASE (ingeniería de software asistida por un computador)?</i>	<i>Sistemas de software que intentan proporcionar ayuda automatizada a las actividades del proceso del software. Los sistemas CASE a menudo se utilizan como apoyo al método.</i>
<i>¿Cuáles son los atributos de un buen software?</i>	<i>El software debe tener la funcionalidad y el rendimiento requerido por el usuario, además de ser mantenible, confiable y fácil de usar.</i>
<i>¿Cuáles son los retos fundamentales a los que se enfrenta la ingeniería de software?</i>	<i>Enfrentarse con la creciente diversidad, las demandas para reducir los tiempos de entrega y el desarrollo de software fiable.</i>

5 Bibliografía

- Pressman, R.S., Ingeniería del Software. Un enfoque práctico.
- Sommerville, I., Ingeniería de Software.
- Inteco. INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA