



**INSTITUTO TECNOLÓGICO
de Pabellón de Arteaga**

ITEC

Ingeniería de software

Maestro: Eduardo Flores Gallardo

Alumno: Leslie Guadalupe Esparza Caldera

Tics 4

Carrera: Ingeniería en tecnologías de la información y comunicación

Fecha: 04/02/2019

INDICE

OBJETIVO GENERAL DEL CURSO.....	3
COMPETENCIAS DEL CURSO.....	3
TEMARIO.....	3
TAREA 1	5
TAREA 2.....	7
TAREA 3.....	8
ACTIVIDAD 1.....	8
ACTIVIDAD 2.....	9
ACTIVIDAD 3.....	10

OBJETIVO GENERAL DEL CURSO

Aplicar técnicas de adquisición de datos (entrevistas, cuestionarios, sondeos, entre otros) para el desarrollo de su proyecto de software.

Aplicar los elementos y conceptos integrados en los procesos de desarrollo de software para la documentación adecuada de su proyecto de software.

COMPETENCIAS PREVIAS

Construir diagramas de tiempo para determinar ruta(s) crítica(s) en la administración de un proyecto.

Aplicar técnicas de modelado para la solución de problemas.

Aplicar la sintaxis de un lenguaje orientado a objetos.

Aplicar un lenguaje orientado a objetos para la solución de problemas.

Identificar los tipos de organización, así como sus estructuras y funciones correspondientes.

Identificar y analizar necesidades de información para su representación, tratamiento y automatización para la toma de decisiones.

Diseñar esquemas de bases de datos para generar soluciones al tratamiento de información.

TEMARIO

Unidad	Temas	Subtemas
1.	Introducción a la ingeniería del software y sistemas de información	<p>1.1. Conceptos de Ingeniería del Software: mitos, paradigma, ingeniería de software, calidad, proceso, método, herramienta, espectro de gestión.</p> <p>1.2. La importancia de la ingeniería del software.</p> <p>1.3. Historia de la Ingeniería del Software.</p> <p>1.4. Los sistemas de información: concepto, características, estructuras, procesos, clasificación, ERP's, CRM, SCM.</p>
2.	Modelos de la ingeniería del software	<p>2.1. Modelo de capacidad de madurez.</p> <p>2.2. Marco de trabajo para el proceso.</p> <p>2.3. Modelos de la ingeniería del software: modelo de cascada, modelo de prototipos, modelo de espiral, modelo de Proceso Unificado Racional (RUP).</p> <p>2.4. Tendencias modernas de modelos de la ingeniería del software.</p>

3.	Planificación del proyecto de software	<p>3.1. Aplicación de herramientas para estimación de tiempos y costos de desarrollo de software: GANTT, PERT/CPM, uso de software para la estimación de tiempos y costos.</p> <p>3.2. Ámbito del software: recursos humanos, recursos de software reutilizables, recursos del entorno.</p>
		3.3. Análisis y gestión del riesgo: estrategias, identificación, proyección, refinamiento, reducción, supervisión y gestión del riesgo.
4.	Análisis del proyecto de software	<p>4.1. Modelado: análisis, diseño, documentación.</p> <p>4.2. Construcción: codificación, pruebas y evaluación, manual del usuario, manual técnico.</p> <p>4.3. Medida, métrica e indicador.</p> <p>4.4. Tipos de métricas: métricas de proceso, métricas de proyecto, métricas orientadas a punto de función, métricas orientadas al tamaño, métricas para la calidad del software.</p> <p>4.5. Implementación y mantenimiento: entrega, retroalimentación del cliente.</p>
5.	Calidad del software	<p>5.1. Definición de calidad y calidad del software.</p> <p>5.2. Importancia de la calidad.</p> <p>5.3. La calidad y la globalización.</p> <p>5.4. Aseguramiento de la calidad del software (SQA): definición y propósito del SQA, problemas que resuelve el SQA, roles y responsabilidades de los equipos de desarrollo, habilidades y capacidades del personal del SQA, Actividades del SQA.</p> <p>5.5. Derecho informático aplicado al software: piratería y falsificación, autoría y creación, contratos y licencias.</p>

TAREA 1

11 razones por las que el desarrollo de software profesional nunca dejará de existir

- **#1 la tecnología siempre seguirá evolucionando**
Siempre habrá nuevos cambios en la tecnología, siempre buscará la manera de ser mejorada, para conseguir un mayor avance.
- **#2 las herramientas de programación serán más intuitivas y poderosas**
Cada día hay cambios en distintas aplicaciones de desarrollo con diferentes tipos de mejoras que hacen un bien ya que son con el fin de poder optimizar el trabajo, aunque cueste un poco comprender el manejo de ellas por lo mismo hay que aprender bien cómo usarlas.
- **#3 habrá herramientas de programación apropiadas para cada tipo de industria**
Cada industria contará con su propio sistema que se hará cargo de distintas tareas específicas para dar un mayor apoyo y avanzar más el trabajo.
- **#4 el mercado de trabajo seguirá adaptándose**
Cada que el mercado avance o la empresa el software o sistema también lo hará, para lo cual también tendrá mejoras de manera continua.
- **#5 cualquier profesión podrá requerir cierta alfabetización en código**
En este tiempo cualquier persona puede requerir del conocimiento de programas, aunque sea en lenguajes no tan complejos, puesto que el conocimiento serviría para poder obtener distintos trabajos. Aunque esto no le quita el puesto a un verdadero programador que puede manejar algunos lenguajes un poco más complejos.
- **#6 siempre habrá demanda de desarrollo de software expertos para tareas complejas**
Aunque creen un sistema autónomo tiene que depender del que la creo o mejor dicho del liderazgo humano, que en este caso serían desarrolladores de software o de sistemas que serían los más preparados para ayudar cuando este llegue a fallar o para hacer unas mejoras al mismo.
- **#7 el aprendizaje automático no desplazará a los programadores**
Por más autónomo que sea no puede realizar totalmente solo necesita ayuda de sus desarrolladores, esto se usa incluso en corporaciones muy grandes.
- **#8 la civilización está cambiando y los ordenadores están en el centro de nuestras vidas.**

Todo en nuestro entorno cambia entre más avanzamos más pronto te das cuenta que las tecnologías se están volviendo parte de él. Nunca hubiéramos imaginado las grandes situaciones en el que ocuparías la tecnología.

- **#9 desarrollar software no solo es picar código**

Para poder desarrollar un software primero tienes que ver el problema, analizarlo puesto que es más importante que solo la sintaxis del software, primero tienes que tener bien claro lo que quieres realizar.

- **#10 hay problemas que las maquinas no son capaces de solucionar bien**

Necesita la ayuda del propio desarrollador para resolverlo puesto que la maquina solo hace lo que le pides, mas no por eso resolverá el problema.

- **#11 existen proyectos de software antiguos que van a durar eternamente**

Siempre quedara la historia del que un día fue un gran sistema y a el cual se le pueden hacer ciertas mejoras para que siga funcionando aun siendo uno de los más antiguos.

Article title: 11 razones por las que el desarrollo de software profesional nunca dejará de existir | Velneo

Website title: Velneo

URL: <https://velneo.es/11-razones-las-desarrollo-software-profesional-nunca-dejara-existir/>

11 razones por las que el desarrollo de software profesional nunca dejará de existir |

Velneo

<https://velneo.es/11-razones-las-desarrollo-software-profesional-nunca-dejara-existir/>

TAREA 2

Definición de ingeniería de software

#1

Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software. [Bohem, 1976]

#2

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, 1972]

#3

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, 1993]

M.C. José Martín Olguín Espinoza. (2004). Análisis Orientado a Objetos Ingeniería del Software. 10/02/2019, de bibing Sitio web: <http://yaqui.mxl.uabc.mx/~molguin/as/IngSoft%201-4.pdf>

#4

“Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software” - B. Bohem

En esta definición del proceso de desarrollo Software, se introduce como parte inherente del producto a obtener, la perspectiva de las necesidades de usuario a las que debe dar respuesta: “Aquellos en los que las necesidades del usuario se traducen en requerimientos, estos se transforman en diseño y este a su vez se implementa en código que es probado, documentado y certificado para su uso” - G. Booch, I. Jacobson, y J. Rumbaugh

miguel Ángel moreno Martin. (2009). Filosofía Lean aplicada a la Ingeniería del Software 2. La Ingeniería del Software. 10/02/2019, de universidad de Sevilla Sitio web: <http://bibing.us.es/proyectos/abreproy/70201/fichero/02+-+Ingenieria+del+Software.pdf>

#5

La ingeniería de software ágil combina una filosofía con un conjunto de lineamientos de desarrollo. La filosofía pone el énfasis en: la satisfacción del cliente y en la entrega rápida de software incremental, los equipos pequeños y muy motivados para efectuar el proyecto, los métodos informales, los productos del trabajo con mínima ingeniería de software y la sencillez general en el desarrollo. Los lineamientos de desarrollo enfatizan la entrega sobre el análisis y el diseño (aunque estas actividades no se desalientan) y la comunicación activa y continua entre desarrolladores y clientes.

Roger S. Pressman. (2006). ingeniería del software. 10/02/2019, de Mc Graw Hill Sitio web: <https://s3.amazonaws.com/academia.edu.documents/45525376/Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1549833601&Signature=6y%2FiOQzcSMpGahIXg1MbccJTys%3D&response-content-disposition=inline%3B%20filename%3DIngenieria.de.software.enfoque.practico..pdf>

MI DEFINICION:

La ingeniería de software se encarga de darle un buen diseño al programa, pero antes de eso tiene que observar primero el problema para después aplicar la solución más óptima, e incluso también puede realizar solo mejoras a programas ya antes hechos esto es para que el software siga activo. También depende mucho del proceso de cómo realizarlo tiene que seguir ciertos pasos, aunque estos no sean exactamente con ese orden.

TAREA 3

Importancia del software

La calidad del software es muy importante puesto que los errores se hacen notar de muchas maneras ya sea bloqueando el sistema o con un pantallazo azul, por lo general es lo que pasa esto es porque se pasó algún error, pero esto no fue de manera intencional puesto que el desarrollador no toma en cuenta lo que el usuario puede hacer con el software, para corregir estos errores los desarrolladores tienen que ver de manera en lo que el usuario pueda hacer.

La calidad del producto no depende de darle una presentación bonita o algún color bonito si no que es a la calidad del software para poderle evitar problemas a los usuarios y posibles usuarios que al verlo les puede llamar la atención tener su propio software.

La ingeniería de software se centra más en herramientas y posibles soluciones al software ya que tiene que tener previstos los errores que el usuario pueda ocasionar esto a su vez también puede ocurrir por no haber hecho pruebas con anticipación.

El automatizar los proyectos significa hacerle cambios mejoras e incluso incluir mejores cosas al software para que siga presente esto es para que el software tenga actualizaciones más usabilidad puesto que siempre están en busca de algo mejor.

Existe la posibilidad de una brecha digital:

- La primera es porque no todos saben la manera correcta de usarla, ni los lados buenos que tiene, esto es porque ahí quienes se estancan en un solo punto que es en quedarse en el pasado.
- La segunda brecha es porque quieren abusar de las tecnologías de manera incorrecta ya que todo lo quieren realizar por medio de ella y ahí cosas que necesitan ser de contacto y no simplemente a través de un monitor o algún software

WordPress (28 abril, 2011). Importancia del Software. 17 febrero 2019, de Ingeniería de software Sitio web: <https://isofwareunesum.wordpress.com/2011/04/28/importancia-del-software/>

Actividad 1

¿Por qué se requiere tanto tiempo para terminar el software?

- Para completarlo de acuerdo a las necesidades de cada uno ya que tiene que ser entendible y para que todo mundo lo pueda usar sin complicaciones

¿Por qué son tan altos los costos de desarrollo?

- Por el tiempo invertido, mas todos los datos que tienen que tener en cuenta, para poder desarrollarlo

¿Por qué no podemos detectar todos los errores antes de entregar el software a nuestros clientes?

- Cuando lo entregas tú piensas que el usuario no va a realizar ciertas acciones y es lo contrario por lo cual causa errores

¿Por qué dedicamos tanto tiempo y esfuerzo a mantener los programas existentes?

- Para que sea conocido y también por el uso que le vayan a dar puede generar buenas ganancias. Aunque esto se base solo en ir modificando o mejor dicho hacerle mejoras al software

¿Por qué seguimos con dificultades para medir el avance mientras se desarrolla y mantiene el software?

- Porque cada que el software se desarrolla no se sabe con exactitud que errores va a tener puesto que no se prevén los errores realizaos por el usuario

Escriba algunos ejemplos (tanto positivos como negativos) que indiquen el efecto del software en nuestra sociedad

- Mas comunicación
- Menos interacción con personas en la vida

Escriba al menos 5 ejemplos de la forma en que se aplica la ley de las consecuencias imprevistas al software de cómputo

- Errores imprevistos
- Mal uso del software
- Mas uso de tecnología

ACTIVIDAD 2

¿Cuáles son las capas de la ingeniería de software?

- Herramientas
- Métodos
- Proceso
- Compromiso de calidad

¿Cuáles son los atributos del buen software?

- Calidad
- Funcionalidad

- Seguridad
- Usabilidad
- El buen software debe de entregar al usuario la funcionabilidad y el desempeño requeridos, y debe de ser sustentable, confiable y utilizable

¿Cuál es la diferencia entre ingeniería de software y ciencias de la computación?

- la ingeniería son los que hacen sistemas internos y las ciencias de la comunicación a lo externo de la computadora
- las ciencias de la computación se enfocan en teoría y fundamentos; mientras la ingeniería de software se enfoca en el sentido práctico del desarrollo y en la distribución de software
-

¿Cuáles son los elementos de un proceso de software?

- Actividad
- proceso
- acciones
- tareas

Actividad 3

Describa con sus propias palabras una estructura de proceso. Cuando se dice que las actividades estructurales son aplicables a todos los proyectos, ¿significa que se realizan las mismas tareas en todos los proyectos sin que importe su tamaño y complejidad?

- No porque depende de que se trata el proyecto y lo que tiene que llevar a cabo porque hay algunas que llevan mas proceso por la diferencia de programa a realizar y actividades que tendrá que realizar

Las actividades sombilla ocurren a través de todo el proceso de software. ¿piensa usted que son aplicables por igual a través del proceso, o que algunas concentran en una o más actividades estructurales?

- hay algunas que si completan e incluso llegan para ayudar a terminar por completo el proceso

A medida que el software gana ubicuidad, los riesgos para el público (debidos a programas defectuosos) se convierten en motivo de preocupación significativa. Desarrolle un escenario catastrófico pero realista en el que la falla de un programa de computo pudiera ocasionar un gran daño (económico o humano).

- Un ejemplo sería el de una tienda que tiene un sistema de pago que este a su vez va registrando todas las ventas que se hicieron en el día, el inventario e incluso la forma de pago que se tubo para la compra y tener la cantidad exacta de lo que se vendió, pero si este sistema falla va haber perdidas mayores ya que no se registrara nada y los productos no serían cobrados ni registrados