**UNIDAD 1**

*Introducción:*

**Narración:** José es un ingeniero que software que acaba de

lograr entrar en un proyecto de software,

sin embargo... ya ha pasado mucho tiempo

desde estuvo en alguno.

**Personaje principal (José):** mmmmm...debería repasar antes de

empezar con el proyecto.

**Compañera de trabajo (Gabriela):** ¡Hola José! ¿necesitas ayuda

para practicar?

**Personaje principal (José):** si, por favor me sería de mucha

ayuda.

**Mapas (Unidad 1):**

**Biblioteca:** Lugar en el juego en el cual se accederá para obtener información acerca de los diferentes temas de las 5 unidades.

**Casa de José:** Lugar en el que el jugador podrá realizar el cuestionario de la unidad 1.

**Resumen (Unidad 1):**

***1.- EL SOFTWARE***

El software es: 1) instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados; 2) estructuras de datos que permiten que los programas manipulen en forma adecuada la información, y 3) información descriptiva tanto en papel como en formas virtuales que describen la operación y uso de los programas; se construye con la aplicación de un proceso ágil y adaptable para obtener un resultado de mucha calidad, que satisfaga las necesidades de las personas que usarán el producto.

En la actualidad equipos de especialistas en software han reemplazado al programador solitario de los primeros tiempos.

El software heredado son sistemas que fueron desarrollados hace varias décadas y han sido modificados de manera continua para que satisfagan los cambios en los requerimientos de los negocios y plataformas de computación. La proliferación de tales sistemas es causa de dolores de cabeza para las organizaciones grandes, a las que resulta costoso mantenerlos y riesgoso hacerlos evolucionar, sin embargo, fueron esenciales para los nuevos sistemas de software, pues se desarrollaron a partir de estos.

***2.- LAS WEBAPPS***

En los primeros días de la Red Mundial (entre 1990 y 1995), los sitios web consistían en poco más que un conjunto de archivos de hipertexto vinculados que presentaban la información con el empleo de texto y gráficas limitadas; con el paso del tiempo nacieron los sistemas y aplicaciones basados en la web (webapps) y en la actualidad se han convertido en herramientas sofisticadas de cómputo que no sólo proporcionan funciones aisladas al usuario final, sino que también se han integrado con bases de datos corporativas y aplicaciones de negocios.

***Atributos de las webapps:***

* + **Uso intensivo de redes:** Una webapp reside en una red permitiendo acceso y comunicación mundial.
  + **Concurrencia:**A la webapp puede acceder un gran número de usuarios a la vez.
  + **Carga impredecible:** El número de usuarios de la webapp cambia, un día pueden ingresar solo 100 personas y al día siguiente quizás 10 000 usen el sistema.
  + **Rendimiento:** La webapp no debe hacer esperar demasiado al usuario o quizá decidan irse a otra parte.
  + **Disponibilidad:** Es frecuente que los usuarios de webapps populares demanden acceso las 24 horas de los 365 días del año.
  + **Orientadas a los datos:** Las webapps se utilizan en forma común para acceder a información que existe en bases de datos que no son parte integral del ambiente basado en web.
  + **Contenido sensible:** La calidad y naturaleza estética del contenido constituye un rasgo importante de la calidad de una webapp.
  + **Evolución continua:** Las aplicaciones web evolucionan en forma continua. No es raro que ciertas webapp (específicamente su contenido) se actualicen minuto a minuto o que su contenido se calcule en cada solicitud.
  + **Inmediatez:** Que el software llegue con rapidez al mercado.
  + **Seguridad:** Deben implementarse medidas estrictas de seguridad a través de la infraestructura de apoyo de una webapp y dentro de la aplicación misma.
  + **Estética:** uno de los principales atractivos de una webapp es su apariencia y percepción.

***3.- INGENIERÍA DE SOFTWARE***

La ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales, una definición más completa desarrollada por la IEEE es la siguiente:

*La ingeniería de software es: La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software.*

la ingeniería de software incluye un proceso, métodos y herramientas para administrar y hacer ingeniería con el software; se basan en un conjunto de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas. El proceso de ingeniería de software es el aglutinante que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de cómputo, los métodos incluyen un conjunto amplio de tareas y las herramientas proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos.

***4.-*** ***EL PROCESO DEL SOFTWARE***

Un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. Una actividad busca lograr un objetivo amplio, una acción es un conjunto de tareas que producen un producto importante del trabajo y una tarea se centra en un objetivo pequeño, pero bien definido.

la estructura del proceso incluye un conjunto de actividades sombrilla que son aplicables a través de todo el proceso del software. Una estructura de proceso general para la ingeniería de software consta de cinco actividades:

* + **Comunicación:** Tienen importancia crítica comunicarse y entender los objetivos de los participantes con respecto al proyecto.
  + **Planeación:** La actividad de planeación crea un “mapa” que guía al equipo.
  + **Modelado:** Crea un bosquejo con el fin de entender el panorama general.
  + **Construcción:** Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste.
  + **Despliegue:** El software se entrega al consumidor que lo evalúa y que le da retroalimentación.

las actividades estructurales se aplican en forma iterativa a medida que avanza el proyecto. Es decir, la comunicación, la planeación, el modelado, la construcción y el despliegue se ejecutan a través de cierto número de repeticiones del proyecto.

***5.- ACTIVIDADES SOMBRILLA***

Las actividades sombrilla ayudan al equipo que lo lleva a cabo a administrar y controlar el avance, la calidad, el cambio y el riesgo, Es común que las actividades sombrilla sean las siguientes:

* + **Seguimiento y control del proyecto de software:** Permite que el equipo de software evalúe el progreso comparándolo con el plan del proyecto y tome cualquier acción necesaria para apegarse a la programación de actividades.
  + **Administración del riesgo:** Evalúa los riesgos que puedan afectar el resultado del proyecto o la calidad del producto.
  + **Aseguramiento de la calidad del software:** Define y ejecuta las actividades requeridas para garantizar la calidad del software.
  + **Revisiones técnicas:** Evalúa los productos del trabajo de la ingeniería de software a fin de descubrir y eliminar errores antes de que se propaguen a la siguiente actividad.
  + **Medición:** Define y reúne mediciones del proceso, proyecto y producto para ayudar al equipo a entregar el software que satisfaga las necesidades de los participantes.
  + **Administración de la configuración del software:** administra los efectos del cambio a lo largo del proceso del software.
  + **Administración de la reutilización:** define criterios para volver a usar el producto. del trabajo y establece mecanismos para obtener componentes reutilizables.
  + **Preparación y producción del producto del trabajo:** agrupa las actividades requeridas para crear productos del trabajo, tales como modelos, documentos, registros, formatos y listas.

***6.- LA PRÁCTICA DE LA INGENIERÍA EN SOFTWARE***

*La esencia de la práctica*

* + Entender el problema (comunicación y análisis): Es conveniente dedicar tiempo al entendimiento del problema.
  + Planear la solución (modelado y diseño del software): Ahora que se entiende el problema crear u pequeño diseño.
  + Ejecutar el plan (generación del código): El diseño que creó sirve como un mapa para el sistema que quiere construir.
  + Examinar la exactitud del resultado (probar y asegurar la calidad): No se puede estar seguro de que la solución sea perfecta, pero sí de que se ha diseñado un número suficiente de pruebas para descubrir tantos errores como sea posible

*Principios generales*

los principios lo ayudarán a establecer un conjunto de herramientas mentales para una práctica sólida de la ingeniería de software.

* + *Primer principio: La razón de que exista todo:* Un sistema de software existe por una razón: dar valor a sus usuarios. Todas las decisiones deben tomarse teniendo esto en mente.
  + *Segundo principio: MSE (Mantenlo sencillo, estúpido…):* Todo diseño debe ser tan simple como sea posible, pero no más; esto facilita conseguir un sistema que sea comprendido y sea susceptible de recibir mantenimiento.
  + *Tercer principio: Mantener la visión:* Comprometer la visión de la arquitectura de un sistema de software debilita y, finalmente hará que colapsen incluso los sistemas bien diseñados.
  + *Cuarto principio: Otros consumirán lo que usted produce:* Siempre establezca especificaciones, diseñe e implemente con la seguridad de que alguien más tendrá que entender lo que usted haga.
  + *Quinto principio: Ábrase al futuro:* Los sistemas de software con verdadera “fortaleza industrial” deben durar mucho más tiempo. Para tener éxito en esto, los sistemas deben ser fáciles de adaptar a ésos y otros cambios.
  + *Sexto principio: Planee por anticipado la reutilización:* La reutilización del código y de los diseños son uno de los mayores beneficios de usar tecnologías orientadas a objetos. La planeación anticipada en busca de la reutilización disminuye el costo e incrementa el valor tanto de los componentes reutilizables como de los sistemas en los que se incorpora.

**Preguntas Unidad 1**

***El software***

**¿Qué es el software?**

* instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados.
* parte física de un ordenador o sistema informático. Está formado por los componentes eléctricos, electrónicos, electromecánicos y mecánicos, tales como circuitos de cables y luz, placas, memorias, discos duros, dispositivos periféricos y cualquier otro material en estado físico que sea necesario para hacer que el equipo funcione.
* Son instrucciones que nos ayudan a especificar lo que se quiere que haga un componente.
* Es el sistema con el cual manejo mi computadora.

**¿Qué es el software heredado?**

* Es aquel software que se tiene en alguna empresa realizadas por un ingeniero y requieren modificaciones; estas modificaciones las hace otro ingeniero de software basándose en los requerimientos que se hayan establecido.
* son sistemas que fueron desarrollados hace varias décadas y han sido modificados de manera continua para que satisfagan los cambios en los requerimientos de los negocios y plataformas de computación.
* Es un software muy antiguo que aún es funcional y óptimo para usarse en las compañías tan solo aplicando ligeras modificaciones en su código.
* es un sistema informático que ha quedado anticuado por sus pocos funcionamientos y dejo de usarse.

***Las Webapps***

**¿Cómo eran los sitios web en sus inicios?**

* un conjunto de archivos de hipertexto vinculados que presentaban la información con el empleo de texto y gráficas limitadas.
* Eran servicios que proporcionaban una comunicación global y traslado de información.
* conjunto de páginas web que son accesibles para todo el mundo

**¿Cuáles son los atributos de las Webapps?**

* Uso intensivo de redes, concurrencia, carga impredecible, rendimiento, disponibilidad, orientadas a los datos, contenido sensible, evolución continua, inmediatez, seguridad y estética.
* Funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad, precisión, adaptabilidad, atractividad.
* Implementación, pruebas y revisión, introducción y servicio operativo, mantenimiento y cuidado, documentación.

**¿A qué se refiere rendimiento en una webapp?**

* A que debe de tener una rápida respuesta y no hacer esperar mucho tiempo al usuario
* Es la capacidad de la página web de soportar un ataque de algún hacker
* La relación entre lo que invertimos a la webapp y lo que nos entregan
* hace referencia al resultado deseado efectivamente obtenido por cada unidad que realiza la actividad económica

**¿A qué se refiere concurrencia en las Webapps?**

* Que a la webapp pueda acceder un gran número de usuarios a la vez.
* cuando mantiene su buen funcionamiento a pesar de las variantes de los supuestos.
* Define y reúne mediciones del proceso, proyecto y producto para ayudar al equipo a entregar el software que satisfaga las necesidades de los participantes.
* habilidad de distintas partes de un programa, algoritmo, o problema de ser ejecutado en desorden o en orden parcial, sin afectar el resultado final.

***Ingeniería De Software***

**¿Qué es la ingeniería de software según la IEEE?**

* es La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software.
* es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales.
* es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema.
* Es la ciencia dedicada al estudio y la aplicación de las diferentes ramas que se ocupa en el desarrollo de implantación y evolución de sistemas integrados de la tecnología.

**¿Cuáles son los pasos para la ingeniería de software?**

* proceso, métodos y herramientas para administrar y hacer ingeniería con el software
* Comunicación, planeación, modelado, construcción y despliegue.
* Planteamiento, planeación, elaboración, pruebas.
* Seleccionar, registrar, examinar, idear definir, implantar y mantenimiento.

**¿Qué es el proceso en ingeniería de software?**

* Es el que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de cómputo
* Es un Conjunto de operaciones que permiten la integración de componentes al software
* Son actividades que nos ayudan a lograr el objetivo del proyecto.
* flujo continuo de las actividades de planeación, organización, dirección y control

***Proceso de software***

**¿cuáles son los pasos para el proceso de la ingeniería de software?**

* Comunicación, planeación, modelado, construcción y despliegue.
* Funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad, precisión, adaptabilidad, atractividad.
* Antecedentes del problema, planteamiento del problema, Marco teórico, elaboración y despliegue.
* Planificar, organizar, dirigir y controlar

**¿Cuál es la diferencia entre una actividad y una tarea?**

* una acción es un conjunto de tareas que producen un producto importante del trabajo y una tarea se centra en un objetivo pequeño, pero bien definido.
* Una actividad lo realiza todo un conjunto de personas y la tarea solo lo realiza una persona determinada.
* Las tareas son realizadas y elaboradas por los empleados de manera individual y las actividades son las que el líder del proyecto otorga a un equipo.
* No hay diferencia, actividad solo es un sinónimo de tarea y viceversa.

**¿Para qué sirve la planeación en un proceso de software?**

* Sirve para crear un “mapa” que guía a todo el equipo.
* En realidad, es una actividad que no tiene mucha importancia.
* Porque nos define una meta a la cual debe llegar el proyecto de software.
* Nos proporciona los antecedentes del problema que tratamos de solucionar.

***Actividades sombrilla***

**¿Cuál es la importancia de las actividades sombrilla?**

* ayudan al equipo a administrar y controlar el avance, la calidad, el cambio y el riesgo del proyecto.
* Nos ayudan a definir bien los requerimientos del software.
* Nos ayuda a crear un bosquejo sobre lo que se está trabajando en ese momento
* Nos dicen la forma en la cual realizar las actividades.

**¿Para qué sirve tener un seguimiento y control del proyecto de software?**

* Permite que el equipo evalúe el progreso comparándolo con el plan del proyecto y tome cualquier acción necesaria para apegarse a la programación de actividades
* Para definir y ejecutar las actividades requeridas para garantizar la calidad del software
* Solo para elaborar la documentación del sistema y posteriormente entregárselo al cliente
* administra los efectos del cambio a lo largo del proceso del software.

***La práctica de la ingeniería de software***

**¿Cuántos son los principios generales para la práctica de la ingeniería de software?**

* 6
* 3
* 12
* 5

**¿A qué se refiere examinar la exactitud del resultado?**

* Hace referencia a que no se puede estar seguro que la solución sea perfecta, pero es importante realizar un número suficiente de pruebas para descubrir tantos errores como sea posible.
* Que los resultados sean exactamente lo que el cliente esperaba del programa.
* Que los materiales y recursos que utilicemos en el proyecto están bajo el presupuesto del cliente.
* A que hay que tener siempre definidas las cantidades de tareas, actividades, procesos que se realizarán en el proyecto.