**Punto 1: Preguntas a responder**

**5.1 - Métricas para la calidad del software y su integración al proceso de ingeniería de software**

**Mediciones**

1. ¿Qué es una medición?
2. ¿Por qué son importantes?
3. ¿Cuáles son los pasos?
4. ¿Qué producto se obtiene?
5. ¿Cuáles son las cuatro razones para medir los procesos de software, los productos y los recursos?
6. ¿Cuál es el objetivo de todo proceso de medición?

**Factores de calidad**

1. Haga una lista de factores y métricas de calidad (McCall y Furps)

**Medidas, métricas e indicadores**

1. ¿Cuál es la definición de: medida, indicador y métricas?
2. ¿Qué permiten los indicadores?
3. ¿Cómo se puede medir la efectividad de un proceso de software?

**Establecimiento de un programa de métricas de software**

1. ¿Qué tendría en cuenta al instituir un programa de métricas de proceso?

**5.2- Mediciones del software**

**Métricas técnicas**

1. ¿Cuáles son las actividades de un proceso de medición?
2. ¿Cuales son os principios que se pueden asociar con las métricas técnicas?
3. ¿Cómo podemos valorar la calidad una métrica de sw?

**Métricas manuales y automatizadas**

1. ¿A qué nivel se realizan métodos manuales de estimación de sw?
2. ¿A qué nivel se realizan métodos automatizado de estimación de sw?

**Métricas orientadas al tamaño y a la función**

1. ¿Cuál es la diferencia entre medidas directas e indirectas?
2. ¿De donde provienen las métricas orientadas al tamaño? Ejemplos
3. ¿En qué consisten las métricas orientadas a la función?
4. ¿De donde surgen los puntos función? ¿Y cómo se calculan los puntos de función?
5. ¿Se puede usar siempre puntos de función?

**Métricas de producto, proceso y proyecto**

1. ¿Cuál es la relación entre las métricas de proceso, proyecto y producto? (ver gráfica)
2. ¿Qué implica las mediciones del proceso y de un proyecto?
3. ¿Para qué le sirven los indicadores a un líder de proyectos?
4. ¿Cuáles son las principales métricas a medir según Putnam y Myers y qué representan? (five core metrics)
5. ¿En qué se centra la medición del producto? ¿Cuáles son sus salidas?

**Métricas de calidad**

1. Arme un cuadro con las principales medidas de calidad (corrección, facilidad de mantenimiento, integridad y facilidad de uso) y sus medidas.

**Integración de métricas y líneas base**

1. Explique cómo aplicar las líneas base para poder integrar las métricas

**Aplicación de las métricas en Administración de Proyectos**

1. ¿Cuáles son los métodos prácticos genéricos para un líder de proyecto?

Los métodos prácticos genéricos para gerentes de proyectos de software, entre clientes de SPR, asemejan los siguientes:

* Un gerente de proyecto por cada ocho miembros del personal técnico
* Un gerente de proyecto de tiempo completo, por cada 1500 puntos función.
* Un gerente de proyecto por aproximadamente cada 150000 instrucciones de código fuente
* La administración de un proyecto comienza antes de los requisitos y se extiende hasta después de terminar el proyecto
* Trabajo de administración del proyecto = 35% de tiempo de administración entregable
* Trabajo del personal = 30% de tiempo de administración entregable
* Reuniones con otros gerentes o clientes = 22% de tiempo de administración entregable
* Trabajo departamental = 8% de tiempo de administración entregable
* Trabajo misceláneo: 5% de tiempo de administración entregable

1. ¿Cuáles son las tres ventajas de adquirir y usar herramientas de estimación de costos?

Las tres ventajas de adquirir y usar herramientas de estimación de costos son:

* + - 1. La curva de aprendizaje para manejar las herramientas de manera efectiva no es trivial
      2. Las herramientas necesitan calibrarse para ajustarlas a condiciones locales
      3. Algunas de las mejores herramientas de estimación de costos son bastantes costosas

**Métricas del modelo de análisis, diseño, desarrollo, pruebas y mantenimiento. Métricas orientadas a objetos**

1. ¿Cómo funciona la métrica Bang?

La métrica Bang puede emplearse para desarrollar una indicación del tamaño del software a implementar como consecuencia del modelo de análisis. Es “una indicación, independiente de la implementación, del tamaño del sistema”. Para calcular la métrica Bang, el desarrollador de software debe evaluar primero un conjunto de primitivas. Éstas se determinan evaluando el modelo de análisis y desarrollando cuentas para los siguientes elementos:

- Primitivas funcionales: Transformaciones que aparecen en el nivel inferior de un diagrama de flujo de datos.

- Elementos de datos: Los atributos de un objeto de datos, los elementos de datos no compuestos y aparecen en el diccionario de datos.

- Objetos: Objetos de datos.

- Relaciones: Las conexiones entre objetos de datos.

- Transiciones: El número de transiciones de estado en el diagrama de transición de estado.

Además de las seis primitivas nombradas arriba, se determinan medidas adicionales para:

- Primitivas modificadas de función manual: Funciones que caen fuera del límite del sistema y que deben modificarse para acomodarse al nuevo sistema.

- Elementos de datos de entrada: Aquellos elementos de datos que se introducen en el sistema.

- Elementos de datos de salid: Aquellos elementos de datos que se sacan en el sistema.

- Elementos de datos retenidos: Aquellos elementos de datos que son retenidos (almacenados) por el sistema.

- Muestras de datos: Las muestras de datos que existen en el límite de la i-ésima primitiva funcional.

- Conexiones de relación: Las relaciones que conectan el i-ésimo objeto en el modelo de datos con otros objetos.

1. ¿Qué métrica aplicarías en la calidad de las especificaciones?

* Especificidad (ausencia de ambigüedad)
* Compleción
* Corrección
* Comprensión
* Capacidad de verificación
* Consistencia interna y externa
* Capacidad de logro
* Concisión
* Trazabilidad
* Capacidad de modificación
* Exactitud
* Capacidad de reutilización

Además, las especificaciones de alta calidad deben estar almacenadas electrónicamente, ser ejecutables o, al menos, interpretables, anotadas por importancia y estabilidad relativas, con su versión correspondiente, organizadas, con referencias cruzadas y especificadas al nivel correcto de detalle.

1. ¿Cuáles son las medidas de complejidad del diseño? ¿Qué miden?

Las medidas de complejidad del diseño son:

* Complejidad estructural.
* Complejidad de datos: Proporciona una indicación de la complejidad en la interfaz interna de un módulo i.
* Complejidad del sistema: Se define como la suma de las complejidades estructural y de datos.

Miden la complejidad arquitectónica y global del sistema. Un crecimiento en los valores de complejidad lleva a una mayor probabilidad de que aumente el esfuerzo necesario para la integración y las pruebas.

1. ¿Cuáles son las métricas de diseño a nivel de componentes? ¿En qué consisten?

* Métricas de cohesión: Consiste en una colección de métricas que proporcionan una indicación de la cohesión de un módulo. Éstas son para cohesiones funcionales fuertes, cohesiones funcionales débiles y pegajosidad.
* Métricas de acoplamiento: Es una métrica para el acoplamiento del módulo que combina el acoplamiento de flujo de datos y de control, acoplamiento global y acoplamiento de entorno.
* Métricas de complejidad: Puede emplearse para predecir la información crítica sobre la fiabilidad y mantenimiento de sistemas software de análisis automáticos de código fuente. También realimentan la información durante el proyecto de software para ayudar a controlar la actividad del diseño.

1. ¿Qué se mediría en una IGU?

Se mediría la frecuencia de transición y el coste de transición de una entidad de representación a la siguiente cuando se realiza una tarea específica.

1. ¿Que sugiere Halstead que se mida?

Halstead sugiere que se mida:

* Longitud global del programa.
* Volumen mínimo potencial para un algoritmo.
* Volumen real.
* Nivel del programa.
* Nivel del lenguaje.
* Esfuerzo de desarrollo.
* Tiempo de desarrollo.
* Número esperado de fallos en el software.

1. ¿Cómo aplicarías Bang a los casos de prueba?

Se puede aplicar Bang, por ejemplo para estimar el número de pruebas asociadas con la interfaz hombre-máquina, examinando:

1. El número de transiciones contenidas en la representación estado-transición de IHM y evaluando las pruebas requeridas para ejecutar cada transición.
2. El número de objetos de datos que se mueven a través de la interfaz.
3. El número de elementos de datos que se introducen o salen.
4. ¿Qué es el IMS?

El IMS es un índice de madurez del software, sugerido por el estándar EEE 982.1-1988, que proporciona una indicación de la estabilidad de un producto software (basada en los cambios que ocurren con cada versión del producto).

1. ¿Qué sugieren medir en OO Lorenz y Kidd?

Lorenz y Kidd sugieren un conjunto de métricas que pueden ayudar durante la planificación del proyecto:

* Número de iteraciones principales.
* Número de contratos completos.

**Factores de ajuste**

1. ¿Qué factores influyen en la productividad del desarrollo?

Los factores más influyentes son: materiales reutilizables de alta calidad y niveles de experiencia de gerentes y personal técnico.

1. ¿Cuáles son artefactos reutilizables?

* Requisitos.
* Diseños.
* Código fuente.
* Materiales de prueba.
* Documentación del usuario.

1. ¿Cuáles son los factores principales de ajuste en el mantenimiento?

* Especialistas en mantenimiento de tiempo completo.
* Experiencia en la aplicación que se actualiza.
* Uso de tablas para alojar variables y constantes en vez de incorporarlos en el código fuente mismo.
* Módulos propensos a errores.

**5.3- Introducción a la estimación de costos de software**

**Funcionamiento de las herramientas de estimación de costos de software**

1. ¿Cómo funcionan las herramientas de estimación de costos de sw?

Las herramientas de estimación de costos de software funcionan tomando como base:

* + - 1. Contienen bases de conocimiento de cientos o miles de proyectos de software
      2. Realizan predicciones de tamaño, que las herramientas genéricas son incapaces de realizar
      3. Ajustan automáticamente estimaciones basadas en herramientas, lenguajes y procesos
      4. Predicen calidad y confiabilidad
      5. Predicen costos de mantenimiento y soporte antes de que estos ocurran en realidad

Lo que se realiza primordialmente es determinar el tamaño del proyecto basándose en las bases de conocimiento (principalmente en aplicaciones comerciales) o en entradas que se le dan a la herramienta por parte del gerente del proyecto o encargado de realizar estos cálculos. Luego se ingresan otros atributos que pueden afectar a estas estimaciones.

1. ¿Cuáles son los principios básicos de las herramientas comerciales de estimaciones de costos de sw? (fórmula)

Las herramientas comerciales se basan en enormes bases de conocimientos de proyectos anteriores, por lo que permiten heredar los atributos que afectan a las estimaciones de proyectos anteriores u otros proyectos similares. La fórmula siguiente ilustra los principios básicos de las herramientas comerciales modernas de estimación de costos de software:

Tamaño del proyecto X atributos del proyecto = Tiempo límite, esfuerzo, costos y entregables (estimados)

1. ¿Cuáles son los atributos que pueden afectar el resultado de una estimación?

Luego de determinado el tamaño se ingresan otros atributos que pueden afectar los resultados de la estimación, estos suelen ser:

* + - 1. Velocidad a que pueden cambiar los requisitos del proyecto
      2. Experiencia del equipo de desarrollo con este tipo de proyecto
      3. Proceso o métodos empleados para desarrollar el proyecto
      4. Actividades específicas a realizarse durante el desarrollo
      5. Número de incrementos, iteraciones o “carreras” que se utilizarán
      6. El o los lenguajes de programación a ser manejados
      7. Presencia o ausencia de artefactos reutilizables
      8. Suites de herramientas de desarrollo que se usarán para desarrollar el proyecto
      9. Entorno o ergonomía del espacio de trabajo en la oficina
      10. Separación geográfica del equipo en múltiples lugares
      11. Presión de los tiempos límites ejercida en el equipo por clientes o ejecutivos
      12. Obligaciones contractuales en términos de costos, fechas, defectos o características

1. ¿Cuáles son los tipos clave de atributos heredados que están asociados a la estimación? (Gráfica)



1. Nombre los pasos para las estimaciones manuales

**Omisiones accidentales**

1. ¿Qué actividades se incluye al estimar?

Las actividades que se incluyen al estimar son:

* Las estimaciones de costos precisas deben iniciar con predicciones de calidad, pues los costos de eliminación de defectos suelen ser más elevados que cualquier otro.
* En segundo lugar, como elementos de costo importantes, se encuentran gastos y esfuerzos dedicados a la producción de documentos en papel: planes, especificaciones, manuales de usuario, etc.
* Muchos proyectos grandes sitúan costos y fechas límite en el manejo de “requisitos progresivos” o nuevas características agregadas al proyecto, tras la fase de requisitos. Todos los desarrollos de software crecerán por requisitos progresivos y, por tanto, este factor debe ser parte integral de las estimaciones de todos los proyectos importantes.
* Una omisión frecuente en las estimaciones de costos es la exclusión accidental de viáticos (líneas aéreas, hoteles, etc.), para reuniones entre equipos de desarrollo en diferentes ciudades y países. Los viáticos pueden exceder el costo de la redacción del código de forma significativa y el tema no puede omitirse de manera accidental.
* Muchas estimaciones de costos de software, cubren sólo las actividades centrales del desarrollo de software y omiten temas tales como administración y soporte a proyectos (secretarias, administración, etc.). Estas actividades auxiliares son parte del proyecto y pueden ascender, en algunos casos, a 20% de los costos totales.
* Es muy común omitir las contribuciones de especialistas si sus habilidades sólo se necesitan sólo en una etapa del ciclo de desarrollo del software. Algunos grupos de especialistas que tienden a ser omitidos accidentalmente de las estimaciones de los costos de software, integrarían las áreas de control de calidad, redacción técnica, puntos de función, administración de bases de datos, optimización de rendimiento, redes y administración de sistemas. Las aportaciones combinadas de estos y otros especialistas pueden totalizar más de 20% del costo total de para el desarrollo de software y no deben omitirse de forma accidental.
* La omisión más común de las estimaciones internas de los costos de software para sistemas de información, son costos en que incurren los usuarios durante la definición de requisitos, creación de requisitos, revisiones de estado, revisiones de fases, documentación, inspecciones, pruebas de aceptación y otras actividades en que los desarrolladores tienen un rol clave. Como los representantes de usuarios no suelen considerarse parte del equipo del proyecto, sus contribuciones al proyecto rara vez se incluyen en las estimaciones del costo del software y estudios de medición. El esfuerzo real que aportan los usuarios a proyectos de desarrollo de software importantes puede acercarse al 20% del trabajo total en algunos casos.
* En el caso de muchos proyectos, el mantenimiento después de la entrega cuesta más que el desarrollo de la aplicación misma. No es recomendable detener la estimación en el punto de entrega del software, sin incluir cuando menos 5 años de estimaciones de mantenimiento y mejoras.

**Estimación de costos**

1. ¿Qué tipo de actividad es la estimación de costos? ¿De donde derivan?

La estimación de costos de software no es una actividad “autónoma”. Las estimaciones se derivan en gran medida de requisitos del proyecto y otros atributos asociados con el proyecto. Una estimación de costos es precursora de presupuestos departamentales y sirve también como documento base para comparar costos acumulados contra proyectados.

1. ¿Qué estimaciones se realizan durante el proceso de desarrollo?

Para cualquier proyecto más grande que uno trivial, se elaborarán múltiples estimaciones de costos durante el proceso de desarrollo, incluyendo, pero limitado, a las siguientes:

* Una estimación aproximada de requisitos del proyecto.
* Una estimación formal inicial derivada de requisitos de proyecto.
* Una o más estimaciones a la mitad del ciclo de vida, reflejando cambios en los requisitos.
* Una acumulación final de costos, empleando datos históricos del proyecto.

1. ¿Cuáles son los consejos para realizar estimaciones?

Sea preciso.

* Sea conservador.
* Base la estimación en datos históricos sólidos.
* Procure calidad, pues las características del software afectarán fechas límite y costos.
* Procure no incluir documentos en papel, ya que pueden valer más que el código fuente.
* Incluya administración de proyectos.
* Incluya los efectos de requisitos progresivos.
* No exagere el efecto de las herramientas, lenguajes y métodos.
* Lleve las fases anteriores a estimaciones de costos, a nivel de actividades.
* Esté preparado para defender las suposiciones de su estimación.

**Goal Question Metrics**

1. ¿Cuál es el objetivo de GQM?

El objetivo de estos marcos de trabajo es proporcionar las referencias necesarias para poder llevar a cabo el proceso de medición de una forma efectiva y sistemática, en base a una serie de objetivos.

1. ¿Cuáles son sus fases?

Sus faces son:

* Planificación: durante la cual se selecciona, define, caracteriza y planifica un proyecto para la aplicación de la medición, obteniéndose como resultado un plan de proyecto.
* Definición: durante la cual se define y documenta el programa de la medición (objetivos, preguntas, métricas e hipótesis).
* Recopilación de Datos: en la que se reúnen los datos reales de la medición.
* Interpretación: en la que se procesan los datos recopilados respecto a las métricas definidas en forma de resultados de medición, que proporcionan respuestas a las preguntas definidas, a partir de las cuales se puede evaluar el logro del objetivo planteado.

1. ¿Qué se planifica en el proceso de GQM?

Se planifica la documentación, procedimientos, calendarios y objetivos del programa de medición, así como un plan de formación de los desarrolladores implicados en el programa.

**Seis formas de estimar costos de software**

1. Arme un cuadro sinóptico explicando las seis formas de estimar costos (cap. 3)

**5.4- Estimaciones de costos de software**

**Herramientas de medición de la calidad de un producto**

1. ¿Cuáles son las principales tareas que debe realizar una herramienta de métricas?

* Adquisición de datos: conjunto de métodos y técnicas que permiten la obtención de datos necesarios para realizar la medición. Esta tarea presenta las siguientes posibilidades:
  + Manual
  + Semiautomática
  + Automática
  + Programable
* Análisis de las mediciones: incluye la habilidad para almacenar, recuperar, manipular y llevar a cabo el análisis de los datos. Esta tarea supone realizar las siguientes actividades:
  + Almacenamiento de los datos de la medición
  + Recuperación de los datos de la medición
  + Análisis Aritmético de resultados
  + Análisis estadístico de resultados
* Presentación de los datos: formatos que facilita la herramienta de medición para generar la documentación obtenida. Destacan las siguientes posibilidades de representación:
  + Tablas
  + Gráficos
  + Exportación de archivos a otras aplicaciones

1. ¿Cómo se clasifican las herramientas?

* Herramientas universales de métricas: aquellas que están diseñadas para soportar el trabajo con métricas y enfocadas a su obtención, análisis y presentación. Además estas herramientas deben proveer al usuario de:
  + Una interfaz de usuario flexible que permita realizar una obtención de datos exacta y efectiva.
  + Una amplia variedad de algoritmos o estándares para el análisis de un gran conjunto de métricas
  + La capacidad de generar un conjunto de informes flexibles y orientado a la representación grafica que además permita personalizar su aspecto en función de los requisitos del usuario.
* Herramientas especializadas de métricas: aquellas que están diseñadas para soportar:
  + Otro tipo de actividad o funciones aparte de la medición, pero que aportan ciertas métricas especificas acerca del ciclo de vida del software, por ejemplo las herramientas de gestión del proyecto.
  + Una métrica especifica o conjunto concreto de métricas, por ejemplo las herramientas de medición de la complejidad del código

1. ¿Cuáles son los tipos de herramientas especializadas?

* Herramientas de estimación software: aquellas herramientas que permiten estimar esfuerzo y duración del proyecto en función de parámetros de tamaño y del entorno de desarrollo software
* Herramientas de gestión de proyecto: aquellas herramientas que soportan la planificación, seguimiento y gestión de un proyecto basándose en los recursos (tiempo, personal, etc.)
* Herramientas de contabilidad temporal: aquellas herramientas que aportan información temporal sobre la ejecución carga del código
* Herramientas de análisis de código fuente: aquellas herramientas que recorren el código fuente escrito en un determinado lenguaje de programación y cuentan líneas de código, puntos función, etc. y calculan ciertas métricas de complejidad.
* Herramientas de seguimiento de cambios y defectos: las herramientas de seguimientos de cambios permiten almacenar y analizar los cambios realizados en código como resultado de la corrección de defectos o actividades de mejora. Las herramientas de seguimiento de defectos permiten detectar los defectos que presenta el código aportando información sobre la severidad del defecto, el tipo de defecto o la localización dentro del código.

**Punto 2: Cuadro comparativo de estimaciones costos de software**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Definición** | **Aplicación** | **Comparar** |
| **Estimación de requisitos** | Los requisitos de software son el punto de partida de todo proyecto nuevo y también un coadyuvante clave en proyectos de mejoras. Son también muy ambiguos, a menudo repletos de conjeturas erradas y errores severos, en general resultan difíciles de esclarecer en forma concisa y comprensible. | Etapa de captura de requisitos y análisis. |  |
| **Estimación de prototipos** |  |  |  |
| **Estimación de especificaciones y diseño de software** | Las especificaciones y diseño del software son una respuesta técnica a requisitos del usuario y sirven para describir en qué forma la aplicación de software que se construye, manejará los requisitos del usuario de manera automatizada. | Etapa de análisis y diseño. |  |
| **Estimaciones de inspecciones de diseño** |  |  |  |
| **Estimaciones de codificación** |  |  |  |
| **Estimación del control de configuración de software** |  |  |  |
| **Estimación de pruebas de software** |  |  |  |
| **Estimación de documentación del usuario y del proyecto** |  |  |  |
| **Estimación de la administración de proyectos** |  |  |  |
| **Estimación de costos de mantenimiento y mejoras** |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Diferencias** | **Quiénes las realizan** | **Herramientas** |
| **Estimación de requisitos** |  | Clientes, personal de mercadotecnia, personal de ventas, personal de ingeniería, analistas de sistemas, programadores, personal de control de calidad y gerentes de proyectos de software. | Requisite de Rational, Bachman Analyst Workbench, Texas Instruments Information Engineering Facility. |
| **Estimación de prototipos** |  |  |  |
| **Estimación de especificaciones y diseño de software** |  | Analistas de sistemas, personal de desarrollo y arquitectos. | Plantillas para métodos de diseño comunes. |
| **Estimaciones de inspecciones de diseño** |  |  |  |
| **Estimaciones de codificación** |  |  |  |
| **Estimación del control de configuración de software** |  |  |  |
| **Estimación de pruebas de software** |  |  |  |
| **Estimación de documentación del usuario y del proyecto** |  |  |  |
| **Estimación de la administración de proyectos** |  |  |  |
| **Estimación de costos de mantenimiento y mejoras** |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tipos** | **Impacto** | **Mediciones** | **Cambios** | **Cosas comunes** |
| **Estimación de requisitos** | De requisitos. |  | Puntos función.  Puntos de característica relacionada. |  |  |
| **Estimación de prototipos** | De prototipos. |  |  |  |  |
| **Estimación de especificaciones y diseño de software** | De especificaciones y diseño. |  | Puntos función. |  |  |
| **Estimaciones de inspecciones de diseño** | De inspecciones de diseño. |  |  |  |  |
| **Estimaciones de codificación** | De codificación. |  |  |  |  |
| **Estimación del control de configuración de software** | De control de configuración. |  |  |  |  |
| **Estimación de pruebas de software** |  |  |  |  |  |
| **Estimación de documentación del usuario y del proyecto** | De documentación del usuario y del proyecto. |  |  |  |  |
| **Estimación de la administración de proyectos** | De administración de proyectos. |  |  |  |  |
| **Estimación de costos de mantenimiento y mejoras** | De mantenimiento y mejoras. |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ANEXO**  **UNIDAD 5: Medición y estimación de costos** | **MEDICION Y ESTIMACION DEL SW** | **ESTIMACION DE COSTOS Y ADMINISTRACION DE PROYECTOS DE SW** | **ING SOFTWARE (Pressman 7ma.)** | **ING SOFTWARE (Pressman 5ta.)** | **ING. SOFTWARE (Sommerville 9na Ed.)** |
| **5.1 - Métricas para la calidad del software y su integración al proceso de ingeniería de software** | **2.2 hasta 2.2.4 , 4 hasta 4.4.1, 5.4.1 y 5.4.2, 5.5, 6, 9** | **1,2, 3, 4, 12, 13, 14, 15, 16, 17,19,20, 21, 22, 23** | **23, 25** | **4, 19, 20** | **24 y 26** |
| 1. Medición 2. Factores de calidad. 3. Establecimiento de un programa de métricas de software. 4. Medidas, métricas e indicadores. | 1.Cap 4.1 |  |  | 1.  2.Cap 19.1  3.Cap 4.2  4.Cap 4.1 | 4.Cap 24.4 |
| **5.2- Mediciones del software** |  |  |  |  |  |
| 1. Métricas técnicas. 2. Métricas manuales. 3. Métricas orientadas al tamaño y a la función. 4. Métricas de producto, proceso y proyecto. 5. Métricas ampliadas de puntos función, de puntos función de casos de uso y Lite. 6. Métricas de calidad. 7. Integración de métricas y líneas base. 8. Aplicación de las métricas en Administración de Proyectos. 9. Métricas del modelo de análisis, diseño, desarrollo, pruebas y mantenimiento. Métricas orientadas a objetos. 10. Factores de ajuste. | 1.  2.  3. Cap 5.4.1 y 5.4.2  4. Cap 4 hasta 4.4.1  5. Cap 5.5  6.  7.  8.  9. | 1.  2.Cap 3  3.  4.  5.  6.  7.  8.Cap 22  9.  10.Cap 12 | 1.  2.  3.  4.Cap 25  5.  6.  7.  8.  9. | 1.Cap 19  2.  3.Cap 4.3  4.  5.  6.Cap 4.5  7.Cap 4.6  8.  9.Cap 4.2, 19.3,.4,.5,.6 y .7 Cap 20.4.2 | 4.Cap 26.2 |
| **5.3- Introducción a la estimación de costos de software** |  |  |  |  |  |
| 1. Funcionamiento de las herramientas de estimación de costos de software. 2. Omisiones accidentales. 3. Estimación de costos. 4. Historia de la estimación. 5. Goal Question Metrics. 6. Seis formas de estimar costos de software. | 1.  2.  3.  4.  5. Cap 2.2 hasta 2.2.4  6. | 1.Cap 1 y Cap. 4  2.Cap 1  3.Cap 1  4.Cap 2  5.  6. Cap 3 |  |  | 5.Cap26.2 |
| **5.4- Estimaciones de costos de software** |  |  |  |  |  |
| 1. Estimación de requisitos. 2. Estimación de prototipos. 3. Estimación de especificaciones y diseño de software. 4. Estimaciones de inspecciones de diseño. 5. Estimaciones de codificación. 6. Estimación del control de configuración de software. 7. Estimación de pruebas de software. 8. Estimación de documentación del usuario y del proyecto. 9. Estimación de la administración de proyectos. 10. Estimación de costos de mantenimiento y mejoras. 11. Herramientas de medición de la calidad de un producto | 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.Cap.9 | 1.Cap 13  2.Cap 14  3.Cap 15  4.Cap 16  5.Cap 17  6.Cap 19  7.Cap 20  8.Cap 21  9.Cap 22  10. Cap 23  11. | 1.Cap 23  2.  3.Cap 23  4.  5.Cap 23  6.  7.Cap 23  8.  9.  10.  11. |  | 9.Cap23.5 |