****

MÓDULO 4. ANÁLISIS Y DISEÑO DE SISTEMAS

**ACTIVIDAD 2 Módulo 4 (Análisis y diseño de sistemas)**

**Contesta las siguientes preguntas:**

1.- Qué son los Modelos de estimación?

Los modelos de estimación son técnicas que ayudan a estimar con precisión el tiempo, el presupuesto y los recursos necesarios para completar un proyecto con éxito. Estos cálculos son esenciales para garantizar que el equipo y las partes interesadas del proyecto acuerden y establezcan lo que necesitan antes de que comience un proyecto. **Las estimaciones precisas son esenciales para una planificación efectiva del proyecto**.

2.- Investiga sobre diferentes modelos de estimación utilizados en la industria del desarrollo de software, como el modelo COCOMO o el método de puntos de función. Compara y contrasta dos modelos de estimación y explica en qué situaciones serían más adecuados.

En la industria del desarrollo de software, se utilizan varios modelos de estimación para predecir el tiempo, el costo y los recursos necesarios para completar un proyecto. Dos de los modelos más conocidos son el **COCOMO (Constructive Cost Model)** y el **método de Puntos de Función**.

A continuación, compararé y contrastaré estos dos modelos y explicaré en qué situaciones serían más adecuados.**COCOMO** es un modelo ampliamente utilizado que se basa en el tamaño del software y en factores de coste y escala para realizar estimaciones. Este modelo es adecuado para proyectos donde se dispone de información detallada sobre el tamaño y la naturaleza del software, así como de los factores que influyen en los costos y la escala del proyecto

3.- A que llamamos la estimación de un proyecto de software?

La estimación de un proyecto de software se refiere al proceso de predecir el tiempo, el costo y los recursos necesarios para desarrollar o mantener un sistema de software. Este proceso es fundamental para planificar y ejecutar proyectos de manera efectiva, ya que proporciona una guía para estimar los recursos necesarios para alcanzar el éxito en la finalización de un proyecto. En el contexto del desarrollo de software, la estimación puede implicar el cálculo del tamaño del software, la medición de la funcionalidad del sistema, el análisis de datos históricos de proyectos pasados, la opinión de expertos y el uso de modelos matemáticos paramétricos. La estimación de proyectos de software es una tarea compleja que requiere disponer de información detallada del proyecto y puede involucrar diferentes técnicas, como el método COCOMO, el método de Puntos de Función y la estimación analógica, entre otros.

4.- Enumera y explica brevemente los principios esenciales para el desarrollo de sistemas, como la modularidad, la cohesión, el acoplamiento y la abstracción. Elige uno de los principios y da un ejemplo de cómo se podría aplicar en el desarrollo de un sistema.

* **Modularidad**: La modularidad se refiere a la propiedad de un sistema que se ha descompuesto en un conjunto de módulos cohesivos y poco acoplados. Este principio implica descomponer un problema en un conjunto de módulos para reducir la complejidad general del sistema. Un ejemplo de aplicación de la modularidad en el desarrollo de un sistema sería la creación de módulos independientes para diferentes funcionalidades, como un módulo para la gestión de usuarios, otro para la generación de informes, etc. Cada módulo debería ser autónomo y tener interacciones limitadas con otros módulos

5.- Crea una lista de las partes interesadas (stakeholders) que podrían estar involucradas en un proyecto de desarrollo de software para una aplicación móvil. Para cada parte interesada, identifica sus roles, responsabilidades y posibles expectativas con respecto al proyecto.

1. **Clientes/Usuarios Finales**:
   * Roles: Proporcionar requisitos, retroalimentación y validación del producto.
   * Responsabilidades: Expresar claramente sus necesidades y expectativas.
   * Expectativas: Esperan un producto que satisfaga sus necesidades y sea fácil de usar.
2. **Equipo de Desarrollo**:
   * Roles: Diseñar, desarrollar, probar y mantener la aplicación.
   * Responsabilidades: Entregar un producto funcional y de calidad dentro del plazo y presupuesto establecidos.
   * Expectativas: Esperan claridad en los requisitos y apoyo para resolver problemas técnicos.
3. **Patrocinadores/Inversionistas**:
   * Roles: Proporcionar financiamiento y apoyo estratégico.
   * Responsabilidades: Asegurar el retorno de la inversión y la alineación con los objetivos comerciales.
   * Expectativas: Buscan un producto exitoso que genere beneficios o cumpla con los objetivos estratégicos.
4. **Equipo de Marketing**:
   * Roles: Promover la aplicación y asegurar su adopción.
   * Responsabilidades: Desarrollar estrategias de marketing efectivas.
   * Expectativas: Esperan una aplicación que sea atractiva para el mercado y que cumpla con las promesas de marketing.
5. **Equipo de Soporte Técnico**:
   * Roles: Brindar soporte y resolver problemas técnicos.
   * Responsabilidades: Garantizar la estabilidad y usabilidad del producto.

Expectativas: Esperan una aplicación que sea fácil de mantener y que minimice las consultas de soporte.

6.- Diseña un plan de proyecto en el que se definan las diferentes fases y actividades necesarias para desarrollar un sistema de gestión de inventario. Asocia cada actividad con su respectiva fase y establece las dependencias lógicas entre ellas.

**Fase 1: Análisis de Requisitos**

* Actividades:
  1. Reunión con el cliente para recopilar requisitos.
  2. Realizar un estudio de viabilidad.
  3. Documentar los requisitos funcionales y no funcionales.
* Dependencias: La actividad 2 depende de la conclusión de la actividad 1.

**Fase 2: Diseño del Sistema**

* Actividades:
  1. Crear el diseño de la base de datos.
  2. Diseñar la interfaz de usuario.
  3. Definir la arquitectura del sistema.
* Dependencias: La actividad 1 depende de la conclusión de la actividad 2 de la Fase 1.

**Fase 3: Desarrollo**

* Actividades:
  1. Desarrollar la base de datos del inventario.
  2. Implementar la lógica de negocio.
  3. Codificar la interfaz de usuario.
* Dependencias: La actividad 1 depende de la conclusión de la actividad 1 de la Fase 2.

**Fase 4: Pruebas**

* Actividades:
  1. Realizar pruebas de integración del sistema.
  2. Realizar pruebas de rendimiento.
  3. Realizar pruebas de usuario.
* Dependencias: Todas las actividades de esta fase dependen de la conclusión de las actividades de desarrollo en la Fase 3.

**Fase 5: Implementación**

* Actividades:
  1. Instalar el sistema en el entorno de producción.
  2. Capacitar a los usuarios finales.
  3. Realizar ajustes finales.
* Dependencias: La actividad 1 depende de la conclusión de la actividad 2 de la Fase 4.

**Fase 6: Mantenimiento**

* Actividades:
  1. Ofrecer soporte post-implementación.
  2. Realizar actualizaciones según sea necesario.
* Dependencias: La actividad 1 depende de la conclusión de la actividad 1 de la Fase 5.

7.- Elabora un conjunto de normas y estándares que deben seguirse en el desarrollo de software para garantizar la consistencia en la codificación y la documentación. Proporciona ejemplos concretos de cómo se aplicarían estas normas en la práctica.

1. **CMMI (Capability Maturity Model Integration)**:
   * Esta norma ayuda a mejorar los procesos de construcción de software y proyectos de TI, lo que garantiza la calidad y madurez en la gestión de proyectos.
   * Ejemplo de aplicación: Implementar prácticas de gestión de configuración para garantizar la consistencia en el control de versiones del código fuente.
2. **IEEE (Institute of Electrical and Electronics Engineers)**:
   * Proporciona una serie de estándares para la documentación y el desarrollo de software, estableciendo criterios para medir la funcionalidad, fiabilidad, usabilidad, rendimiento y seguridad del software.
   * Ejemplo de aplicación: Seguir las directrices de documentación de IEEE para la creación de especificaciones técnicas y manuales de usuario.
3. **ISO/IEC 12207**:
   * Esta norma establece un marco para el ciclo de vida del software, abordando procesos como la adquisición, suministro, desarrollo, operación y mantenimiento del software.
   * Ejemplo de aplicación: Seguir los procesos definidos en ISO/IEC 12207 para el desarrollo y mantenimiento del sistema de gestión de inventario.

### Ejemplos Prácticos de Aplicación:

* **Estándares de Codificación**:
  + Establecer reglas para el formato del código, nombres de variables, comentarios y estructura de los archivos.
  + Ejemplo: Requerir el uso de un estándar de nomenclatura de variables, como camelCase o snake\_case, para garantizar consistencia en todo el código.
* **Estándares de Documentación**:
  + Definir formatos y estructuras para la documentación técnica, como especificaciones de diseño y manuales de usuario.
  + Ejemplo: Utilizar plantillas predefinidas para la documentación técnica, asegurando que todos los documentos sigan un formato coherente.
* **Estándares de Proceso**:
  + Establecer procesos y prácticas recomendadas para el desarrollo, pruebas y mantenimiento del software.
  + Ejemplo: Implementar un proceso de revisión de código para garantizar la calidad y consistencia del código desarrollado.

***Nota: Llenar en este mismo documento y enviarlo en formato pdf en el ítem correspondiente.***

***¡Espero su entrega!***

***Neftalina Jimenez***

***Tutora Virtual***