Actividad en Clase

1. ¿Es apropiado considerar que el software es más que programas que son ejecutados por los usuarios finales de un sistema?

Para llegar al final tuvo que pasar por un proceso de análisis, investigación y desarrollo de la lógica del producto y la arquitectura dependiendo del modelo del negocio

No es apropiado reducir el software a simples programas que corren en un computador. Antes de llegar al usuario final, el software atraviesa un ciclo de análisis, investigación, diseño, pruebas y despliegue. Es decir, detrás de cada aplicación hay una arquitectura pensada para el modelo de negocio, lo que garantiza que la solución no sea solo funcional, sino también escalable, mantenible y segura.  
Asesoramiento al cliente: Invertir en un proceso estructurado asegura que la solución no se quede en “programas”, sino en un activo estratégico que potencia su negocio.

2. ¿Cuáles son las diferencias entre el desarrollo de un producto de software genérico y el desarrollo de un software personalizado?

software genérico se orienta a un mercado masivo, con funcionalidades estándar para cubrir necesidades comunes, requisitos únicos

 el software personalizado se diseña desde cero para un cliente o e

* Genérico: pensado para un mercado masivo, con funciones estándar (ejemplo: Microsoft Word, SAP Business One).
* Personalizado: diseñado a la medida de un cliente, enfocado en requisitos únicos y en la ventaja competitiva de ese negocio.

Asesoramiento al cliente: Si buscas rapidez y bajo costo inicial, el software genérico puede ser opción; pero si lo que deseas es diferenciarte de la competencia y cubrir necesidades específicas, el software personalizado es la mejor inversión, porque se alinea 100% a tu operación.

3. ¿Cuáles son los cuatro atributos importantes que todos los productos de software deben tener? Sugiera otros cuatro atributos que pueden ser significativos.

Los cuatro básicos son:

1. Confiabilidad
2. Usabilidad
3. Eficiencia
4. Mantenibilidad

Otros cuatro muy significativos pueden ser:

* Escalabilidad (crecer sin perder rendimiento).
* Seguridad (proteger datos y operaciones).
* Portabilidad (funcionar en diferentes entornos).
* Interoperabilidad (integrarse con otros sistemas).

Asesoramiento al cliente: Diseñar con estos atributos asegura que el software no solo funcione hoy, sino que siga aportando valor en el futuro.

4. ¿Cuál es la diferencia entre un modelo del proceso del software y un proceso del software? Sugiera dos formas en las que un modelo del proceso del software ayuda en la identificación de posibles mejoras del proceso.

* Modelo de proceso: representación abstracta del ciclo de desarrollo (ej. Cascada, Iterativo, Ágil).
* Proceso de software: las actividades reales llevadas a cabo por el equipo.

Dos formas en que el modelo ayuda a mejorar:

1. Permite identificar cuellos de botella (ej. demasiadas pruebas al final).
2. Facilita introducir mejoras graduales (ej. adoptar integración continua).

Asesoramiento al cliente: Un modelo bien elegido da visibilidad y control, lo que reduce riesgos de sobrecostos y retrasos.

5. Explique por qué los costos de pruebas de software son particularmente altos para productos de software genéricos que se venden a un mercado amplio.

En un producto masivo, el software debe funcionar en múltiples entornos, sistemas operativos y dispositivos. Esto multiplica la complejidad de las pruebas, ya que no se puede fallar en un mercado amplio.

Asesoramiento al cliente: Invertir en pruebas robustas asegura que el producto sea confiable y competitivo, evitando pérdidas económicas por errores masivos.

6. ¿Qué es una herramienta CASE?

Son herramientas de Ingeniería de Software Asistida por Computadora, que ayudan en el diseño, modelado, pruebas y mantenimiento (ejemplo: Bizagi, Rational Rose, Enterprise Architect).

Asesoramiento al cliente: Usar CASE mejora la productividad, reduce errores humanos y estandariza el desarrollo, garantizando entregables de calidad.

7. Comente si los ingenieros profesionales deben atestiguar de la misma forma que los doctores o los abogados.

Así como doctores o abogados responden por su trabajo, los ingenieros deben responder éticamente por la calidad y confiabilidad del software.

Asesoramiento al cliente: Al contratar ingenieros certificados y responsables, se asegura que su inversión estará respaldada por buenas prácticas profesionales y ética laboral.

Nosotros como ingenieros debemos considerar la seguridad, el bienestar y los derechis de las personas que se involucren en los proyectos que realicemos, pues dentro de estos proyectos debemos tomar decisiones informadas que tendrán consecuencias significativas para los usuarios y la sociedad

Hablando de la ingeniera de software esta disciplina rwquiere que los profesionales actúen dentro de un marco que considere las responsabilidades sociales, primando el bienestar común

8. Para contrarrestar al terrorismo, muchos países están desarrollando sistemas informáticos que siguen la pista de un gran número de sus ciudadanos y de sus acciones. Desde luego, esto tiene implicaciones sobre la privacidad. Comente la ética de desarrollar este tipo de sistema.

El dilema ético central radica en encontrar un equilibrio entre la seguridad y la libertad. Una posible solución es establecer marcos legales estrictos y transparentes que regulen cómo se recopilan, almacenan y usan los datos. Esto podría incluir:

* Supervisión judicial independiente para autorizar y monitorear la vigilancia.
* Mecanismos de rendición de cuentas para las agencias gubernamentales.
* Limitaciones claras sobre el tipo de datos que se pueden recopilar y el tiempo que se pueden retener.
* Derechos de los ciudadanos para saber si su información está siendo recopilada y para impugnar su uso.

En última instancia, la discusión ética gira en torno a si los beneficios potenciales de la seguridad superan los riesgos reales de la pérdida de privacidad y libertad. La implementación de estos sistemas sin un debate público robusto y sin salvaguardias adecuadas podría sentar un precedente peligroso para el futuro de las sociedades democráticas.

9. Se le ha nombrado gestor de proyecto dentro de una organización de sistemas de información. Su trabajo es construir una aplicación que es bastante similar a otras que ha construido su equipo, aunque ésta es mayor y más compleja. Los requisitos han sido detalladamente documentados por el cliente. ¿Qué estructura de equipo elegiría y por qué? ¿Qué modelo(s) de proceso de software elegiría y por qué?

En cuanto al equipo lo estructuraría por roles de manera jerárquica para mantener orden en el ciclo de la vida del desarrollo del software y en cuanto a los modelos del proceso de software escogería un modelo de proceso incremental, primero porque la envergadura del proyecto, segundo porque los requisitos ya están debidamente definidos y documentados lo que permitiría trabajar bajo ese modelo, minimizando riesgo y asegurando un cumplimiento exacto de los requisitos.

10. Se le ha nombrado gestor de proyecto de una pequeña compañía de productos software. Su trabajo consiste en construir un producto innovador que combine hardware de realidad virtual con software innovador. Puesto que la competencia por el mercado de entretenimiento casero es intensa, hay cierta presión para terminar el trabajo rápidamente. ¿Qué estructura de equipo elegiría y por qué? ¿Qué modelo(s) de proceso de software elegiría y por qué?

Debido a las circunstancias en la que me han nombrado y la necesidad de entregar resultados con rapidez, implementaría una estructura de equipo ágil y autofuncional y de trabajo ágil puesto que este modelo de trabajo le permitiría al equipo tener una comunicación directa, rápida y flexible en casos de cambios. Adaptable. En cuanto al modelo del proceso, la opcion mas optima para cumplir con lo estipulado es la scrum ya que sus entregas son incrementales, contando con un feedback para manejar las falencias y minimizando riesgos de lanzar un producto que no satisfaga al mercado

* Por qué:
  + Ciclos de Desarrollo Cortos: Scrum opera en Sprints (ciclos cortos, generalmente de 2-4 semanas), permitiendo lanzar versiones funcionales tempranas del producto y recibir retroalimentación del usuario de manera continua.
  + Gestión de la Incertidumbre: Dado que es un producto innovador, el resultado esperado puede no ser del todo claro desde el principio. Scrum permite explorar diferentes enfoques y ajustar la dirección del proyecto basándose en lo aprendido en cada Sprint.
  + Priorización de la Innovación: Al enfocarse en entregar valor en cada iteración, se fomenta la experimentación y la incorporación de nuevas ideas, características y tecnologías que mantengan el producto a la vanguardia.
  + Reducción de Riesgos: La entrega temprana de prototipos y versiones funcionales permite identificar problemas de integración entre hardware y software, o de usabilidad, mucho antes de que se conviertan en obstáculos mayores.
  + Acelera el Time-to-Market: Al iterar rápidamente y centrarse en las funcionalidades de mayor valor, Scrum ayuda a entregar un producto con el que los usuarios puedan interactuar pronto, lo cual es importante en un mercado competitivo.