



Métodos y Modelos en Ingeniería de Software I (95.20)

Cuestionarios

Versión 5.0 - Marzo 2024



Objetivo



Proponer un conjunto de actividades que **estimulen** la lectura y permitan afianzar conocimientos teóricos.



- **Por favor, leer** el material de lectura obligatoria y ver los vídeos de los capítulos **antes** de resolver las actividades.
- Las actividades incluidas en esta guía están basadas en las lecturas y en los vídeos de las presentaciones de cada uno de los capítulos del curso.
- Todas las actividades se irán poniendo gradualmente a disposición en la plataforma Google Classroom.
- Salvo indicación en contrario, las entregas de las resoluciones deben hacerse mediante Google Classroom.
- Las actividades están organizadas por capítulos. Al comienzo de cada sección se indican las lecturas obligatorias, que también se encuentran incluidas en el documento MMI-Referencias Bibliográficas disponibles en Google Drive.
- Todo el material de lectura obligatoria (un subconjunto de las referencias bibliográficas) se encuentra disponible en Google Drive, junto al resto del material del curso.
- No hay ejercicios “con trampa”. Hemos puesto especial empeño en que los enunciados estén claramente redactados. Sugerimos leerlos detenidamente y procurar interpretarlos de acuerdo con el contexto.



Capítulo 1

Introducción a la Ingeniería de Software

Capítulo 1: Introducción a la ingeniería de software

Lecturas

- [Washizaki] Hironori Washizaki (editor). *Guide to the Software Engineering Body of Knowledge 4.0*. IEEE Computer Society, 2024 (Public Review Version, Jan 2024)
- [Brooks] Brooks, Frederick P. "No silver bullet." IEEE Computer 20.4 (1987): 10-19.
- [Fowler] Fowler, Martin. "The new methodology." Wuhan University Journal of Natural Sciences 6.1-2 (2001): 12-24.
- [INCOSE] Interational Council on Systems Engineering 2019, Overview, <<https://www.incose.org/about-systems-engineering/system-and-se-definition>>

Capítulo 1: Introducción a la ingeniería de software

Actividades

Sistemas e ingeniería de sistemas

- Actividad 1.1 (Q)

Según las definiciones de [INCOSE], seleccionar las respuestas más adecuadas:

Un sistema es...

Marca solo un óvalo.

- ☐ un conjunto de elementos
- ☐ un conjunto de elementos que exhiben comportamiento
- ☐ un conjunto de elementos que exhiben significado
- ☐ un grupo de partes o elementos que en conjunto exhiben un comportamiento o significado que los elementos individuales no tienen

Ejemplos válidos de sistemas diseñados (engineered systems) son los siguientes:

Marca solo un óvalo.

- ☐ Un banco
- ☐ El sistema circulatorio
- ☐ Una aplicación móvil
- ☐ El servicio de atención al cliente de un hotel
- ☐ La atención a alumnos en una universidad

Capítulo 1: Introducción a la ingeniería de software

Actividades

Sistemas e ingeniería de sistemas

- Actividad 1.1 (Q)

Responder V/F de acuerdo [INCOSE]

El software es un sistema del tipo conceptual que no necesita de un sistema físico para su transporte o almacenamiento

Marca solo un óvalo.

☐ Verdadero

☐ Falso

Capítulo 1: Introducción a la ingeniería de software

Actividades

Software e Ingeniería de Software

- Actividad 1.2 (Q)

Seleccionar la mejor respuesta:

Ingeniería es

Marca solo un óvalo.

- ☐ la creación de soluciones
- ☐ la creación de soluciones costo efectivas a problemas prácticos mediante la aplicación de conocimiento codificado para construir cosas al servicio de la humanidad.
- ☐ la aplicación de un enfoque sistemático al desarrollo de soluciones a problemas complejos

La Ingeniería de Software es

Marca solo un óvalo.

- ☐ la creación de soluciones basadas en software
- ☐ la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software.
- ☐ una ingeniería como las restantes, sin nada en particular que la diferencie del resto

Capítulo 1: Introducción a la ingeniería de software

Actividades

Software e Ingeniería de Software

- Actividad 1.2 (Q) (cont)

Seleccionar la mejor respuesta:

Indicar a qué área de conocimiento del SWEBOK [Washizaki] es MÁS probable que pertenezcan las siguientes actividades que realiza una persona que desarrolla software en la compañía PSA:

	Requisitos	Diseño	Construcción	Prueba	G.Configuración	G.Ingeniería de software
Llega 8:45 a la oficina. Se sirve un café y se sienta al escritorio a revisar su correo y a repasar el estado de las tareas a su cargo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A las 9:15 se reúne con su equipo de proyecto para la Daily Stand Up Meeting. En ella, cada uno de los participantes expone brevemente qué tareas realizó el día anterior, cuáles tiene planificadas para el día y si hay algún obstáculo o problema que deba ser abordado.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A las 10:00 mantiene, junto a su equipo de proyecto, una reunión de trabajo con el usuario que cubre el rol de product owner para clarificar ciertos aspectos de la funcionalidad de la aplicación que están desarrollando.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Luego de la reunión retorna a su escritorio y se dispone a modificar unas componentes que desarrolló el día anterior. Trabaja con ellas por espacio de una hora, modificando varias líneas de código y haciendo algunas pruebas manuales.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Antes de salir a almorzar, ejecuta el script para compilar las components y desplegarlas en el ambiente de pruebas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Capítulo 1: Introducción a la ingeniería de software

Actividades

Software e Ingeniería de Software

- Actividad 1.2 (Q) (cont)

Seleccionar la mejor respuesta:

Pruebas y revisiones de software son actividades relacionadas con los

Marca solo un óvalo.

- ☐ costos de prevención de la calidad
- ☐ costos de falla internos
- ☐ costos de falla externos
- ☐ costos de evaluación de la calidad
- ☐ costos de aseguramiento de la calidad
- ☐ costos del desarrollo de software

Un proceso es

Marca solo un óvalo.

- ☐ un enfoque sistemático para la especificación, diseño, construcción, prueba y verificación de software.
- ☐ un conjunto de actividades que transforma una entrada en una salida y que consume recursos
- ☐ un mecanismo para entender, modelar y comunicar.

Capítulo 1: Introducción a la ingeniería de software

Actividades

- Actividad 1.3 (Q)

- A partir del artículo de [Brooks], responder:

La parte más compleja en la construcción de software es llevar adelante la ejecución de lo que se decidió construir *

Marca solo un óvalo.

- ☐ Falso
☐ Verdadero

Las características esenciales del software (complejidad, maleabilidad, conformidad, invisibilidad) se pueden resolver plenamente mediante:

Marca solo un óvalo.

- ☐ Lenguajes de programación más poderosos
☐ Ambientes de desarrollo más poderosos, con ayuda de herramientas de inteligencia artificial
☐ Mayor capacitación del personal involucrado en el desarrollo
☐ Programación basada en modelos
☐ MBSE (Model Based Software Engineering)
☐ Todas las opciones
☐ Ninguna de las opciones

Un banco necesita manejar altos estándares de seguridad en todas sus transacciones. ¿Qué le recomendaría Brooks?

Marca solo un óvalo.

- ☐ Desarrollar su propio sistema de encriptado, para tener todo bajo control
☐ Comprar a terceros el sistema de encriptado
☐ Contratar más gente para que el conocimiento esté distribuido

Capítulo 1: Introducción a la ingeniería de software

Actividades

- Actividad 1.3 (Q) (cont.)

- A partir del artículo de [Brooks], responder:

El desarrollo de lenguajes de alto nivel fue uno de los avances más importantes en lo que refiere al trabajo sobre las dificultades accidentales del software *

Marca solo un óvalo.

☐ Verdadero

☐ Falso

Una metodología que se concentre durante una década en atacar las problemáticas esenciales del software permitiría mejorar en un orden de magnitud la productividad en el desarrollo de software. *

Marca solo un óvalo.

☐ Falso

☐ Verdadero

No existen ni existieron balas de plata en el desarrollo de hardware que permitan obtener mejoras en un orden de magnitud en una década en lo que refiere a la productividad, simplicidad y confiabilidad. *

Marca solo un óvalo.

☐ Falso

☐ Verdadero

Capítulo 1: Introducción a la ingeniería de software

Actividades

- **Actividad 1.4 (Q)**
 - A partir de [Fowler], contestar las siguientes preguntas:
 - a) Según [Fowler] ¿cuál de las siguientes actividades se ajusta mejor a la tarea de programar el software?
 - Diseño
 - Construcción
 - Verificación
 - Validación

Capítulo 1: Introducción a la ingeniería de software

Actividades

- **Actividad 1.4 (Q)**

b) De acuerdo a [Fowler], indicar cuál o cuales de las siguientes afirmaciones son verdaderas en relación a la predictibilidad del software:

- Existen proyectos que por su naturaleza nos permiten utilizar una metodología predictiva.
- La importancia de utilizar iteraciones es mayor en metodologías adaptativas que en predictivas.
- Los cambios en los requisitos de un programa son consecuencia de un mal trabajo de ingeniería de requisitos.
- Incorporar cambios tardíos en los requisitos es un problema para los clientes.



Capítulo 2

Introducción a la Ingeniería de Requisitos

Capítulo 2: Introducción a la ingeniería de requisitos

Lecturas

- **[Wiegers]** Wiegers, Karl, and Joy Beatty. Software requirements. Pearson Education, 2013. Capítulos 1, 2 y 14.

Capítulo 2: Introducción a la ingeniería de requisitos

Actividades

Requisitos e ingeniería de requisitos

- Actividad 2.1
 - Responder de acuerdo a lo planteado en el capítulo 2 y en el apéndice 2.1:
 - Los requisitos son un diseño del sistema a construir (V/F)
 - En los desarrollos predictivos los requisitos se van elaborando progresivamente a lo largo del ciclo de vida (V/F)
 - Business Analysis y Design Thinking son nombres alternativos para la ingeniería de requisitos (V/F)
 - En la exploración del espacio de la solución solamente hay convergencia (V/F)

Capítulo 2: Introducción a la ingeniería de requisitos

Actividades

Requisitos e ingeniería de requisitos

- Actividad 2.2
 - A partir de los capítulos 1 y 2 de [Weigers], contestar las siguientes preguntas:
 - a) Un requisito es una descripción de cómo el sistema deberá comportarse, o una propiedad o atributo que deberá poseer (V/F)
 - b) Seleccionar qué definición corresponde a cada uno de los términos:

	Un objetivo de negocio de alto nivel que tiene la organización que construye un producto o un cliente que lo adquiere	Una o más capacidades del sistema, lógicamente relacionadas, que proporcionan valor a un usuario y se describen mediante un conjunto de requisitos funcionales.	Una descripción del comportamiento que un sistema exhibirá bajo condiciones específicas.	Un requisito de nivel superior para un producto que contiene múltiples subsistemas, que podría ser todo software o software y hardware.	Un objetivo o tarea que clases específicas de usuarios deben poder realizar con un sistema o un atributo de producto deseado.
Requerimiento de negocio	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Requisito de usuario	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Requisito funcional	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Requisito del sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Feature (funcionalidad/prestación)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Capítulo 2: Introducción a la ingeniería de requisitos

Actividades

- Actividad 2.2 (cont.)
 - A partir de los capítulos 1 y 2 de [Weigers], contestar las siguientes preguntas (cont.):
 - c) Las reglas de negocio son en sí mismas requisitos porque existen más allá de los límites de cualquier aplicación de software. (V/F)
 - d) Seleccione qué término corresponde con cada una de las opciones:

Marca solo un óvalo por fila.

	Req. No funcional	Req. Funcional	Funcionalidad (feature)	Req. del usuario	Req. del sistema
Localización	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Portabilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pagar una factura	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Actualización automática del listado de firmas de virus en un producto antimalware	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Se deben poder recolectar los datos del sensor de temperatura	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Capítulo 2: Introducción a la ingeniería de requisitos

Actividades

- Actividad 2.2 (cont.)

- A partir de los capítulos 1 y 2 de [Weigers], contestar las siguientes preguntas:

- e) El objetivo del desarrollo de requisitos es acumular entendimiento y conocimiento compartido que sea lo suficientemente bueno como para permitir la construcción de la siguiente porción del producto en un nivel de riesgo aceptable. (V/F)

- f) Seleccione la o las disciplinas que componen el desarrollo de requisitos:

- Especificación
 - Validación
 - Exploración
 - Evaluación
 - Documentación
 - Confirmación
 - Análisis
 - Obtención/Descubrimiento (Elicitation)

- g) Planificar iteraciones de exploración de requisitos, refinándolos progresivamente desde un alto nivel a uno con más precisión y detalle, y confirmando las modificaciones con los usuarios, es una de las claves para el éxito del desarrollo de requisitos. (V/F)

Capítulo 2: Introducción a la ingeniería de requisitos

Actividades

- Actividad 2.2 (cont.)
 - A partir de los capítulos 1 y 2 de [Weigers], contestar las siguientes preguntas:
 - h) Un interesado es un individuo u organización que obtiene un beneficio directo o indirecto de un producto. (V/F)
 - i) Es suficiente con que algunos de los interesados se comprometan a aplicar las mejores prácticas en el desarrollo y gestión de requisitos para producir productos de calidad superior (V/F)
 - j) Marque todas aquellas opciones que demuestran un bajo nivel de aplicación de buenas prácticas de requisitos en una organización
 - Existen funcionalidades ya desarrolladas en el software que nadie aprovecha o utiliza.
 - Al momento de programar los desarrolladores encontraron ambigüedades y falta de información por lo que tuvieron que adivinar y hacer suposiciones.
 - Los usuarios se concentran en las pantallas o características de la interfaz del usuario y no en lo que ellos necesitan que cumpla el software.
 - El equipo de desarrollo interactúa de forma constante con los usuarios representativos para entender sus necesidades.

Capítulo 2: Introducción a la ingeniería de requisitos

Actividades

Requisitos no funcionales

- Actividad 2.3
 - De acuerdo a [Wiegers], capítulo 14:
 - La portabilidad es un atributo de calidad externa (V/F)
 - La robustez es un atributo que indica cuán confiable es un sistema (V/F)
 - La integridad trata de prevenir la pérdida de información (V/F)
 - El desempeño (performance) y la eficiencia son atributos que no están relacionados entre sí (V/F)
 - Los requisitos de seguridad (Safety) buscan evitar daños a los datos de las personas (V/F)
 - Aumentar la usabilidad generalmente impacta en forma negativa la portabilidad (V/F)



Capítulo 3

Técnicas de Descubrimiento de Requisitos

Capítulo 3: Técnicas de descubrimiento de requisitos

Lecturas

- **[Alexander-2004]** Alexander, Ian, and Suzanne Robertson. "Understanding project sociology by modeling stakeholders." IEEE Software, 21.1 (2004): 23-27.
- **[PMI-2015]** Project Management Institute. Business Analysis for Practitioners: A Practice Guide. Project Management Institute, Incorporated, 2015.

Lecturas complementarias

- **[Brennan]** Brennan, Kevin, ed. A Guide to the Business Analysis Body of Knowledge. IIBA, 2009.
- **[Bartyzel]** Bartyzel, Michal. Conversation Patterns for Software Professionals. Lulu.com, 2016.
Disponble en: <https://www.infoq.com/minibooks/conversation-patterns/>

Capítulo 3: Técnicas de descubrimiento requisitos

Actividades

- Actividad 3.1
 - ¿Cuáles técnicas es MÁS PROBABLE que se utilicen en los siguientes escenarios?

	Entrevistas.	Observación.	Taller de requisitos	Encuestas.	Grupos de discusión/Focus group	Análisis de documentos.	Análisis de reglas de negocio.	Análisis de causa raíz.	Tormenta de ideas.	Prototipado.	Open space.
Una empresa especializada en software para empresas está analizando agregar a su suite de gestión un nuevo sistema de CRM (Customer Relationship Management). Para ello, planea convocar a expertos en la materia y a los responsables del área de Marketing.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Un banco tiene un plan para reemplazar su sistema desarrollado internamente a lo largo de las últimas tres décadas por un software bancario comercializado por una empresa de renombre internacional. El nuevo software deberá replicar la funcionalidad del sistema actual e incorporar nuevas características que han estado en lista de espera por años.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Capítulo 3: Técnicas de descubrimiento requisitos

Actividades

- Actividad 3.1 (cont.)
 - ¿Cuáles técnicas es MÁS PROBABLE que se utilicen en los siguientes escenarios?

	Entrevistas.	Observación.	Taller de requisitos	Encuestas.	Grupos de discusión/Focus group	Análisis de documentos.	Análisis de reglas de negocio	Análisis de causa raíz.	Tormenta de ideas.	Prototipado.	Open space.
Una startup de cuatro socios con un producto orientado al público masivo detecta que el crecimiento de usuarios empieza a declinar, incluso a generar una contracción. Uno de los socios sugiere analizar qué funcionalidades se pueden agregar para revertir la situación.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Un gerente comercial solicita al área de sistemas de su empresa el desarrollo de una nueva aplicación que le permita planificar y realizar el seguimiento de planes comerciales.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Una empresa de aviación planea el desarrollo de una nueva aplicación para el entretenimiento de sus pasajeros abordo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Capítulo 3: Técnicas de descubrimiento requisitos

Actividades

●Actividad 3.3

- Contestar las siguientes preguntas de acuerdo a las técnicas descriptas en ~~[IBA-2009]~~ y [PMI-2015]
 - a) Son dueños de una organización que se dedica a exportar software. Construyen un análisis FODA. ¿Cuáles de las siguientes incluyen dentro de las Oportunidades?
 - Contrataron a una persona de ventas que tiene contactos con importantes firmas del exterior
 - Se espera un aumento considerable del dólar para el siguiente año
 - El principal competidor del mercado decidió especializarse en el mercado interno.
 - El sector de I+D (investigación y desarrollo) diseñó un proyecto base que aumentará la velocidad de comienzo de cada proyecto.
 - b) ¿Cuáles de las siguientes afirmaciones del diagrama de espina de pescado son correctas?
 - El diagrama de espina de pescado se focaliza en resolver el problema identificado
 - Se comienza identificando el problema a tratar(efecto)
 - Cada causa a su vez puede tener otras causas

Capítulo 3: Técnicas de descubrimiento requisitos

Actividades

●Actividad 3.3 (cont.)

- Contestar las siguientes preguntas de acuerdo a las técnicas descriptas en [~~IBA-2009~~] y [PMI-2015]
 - c) Una técnica utilizada para identificar reglas de negocio es:
 - Brainstorming
 - La construcción de árboles de decisión
 - Prototipado
 - Cuestionarios
 - d) Indicar cuáles de los siguientes ítems referidos a técnicas para descubrir requisitos son correctos
 - La técnica de brainstorming, a diferencia de los focus groups y talleres de requerimientos, no requiere de un facilitador o moderador
 - En los talleres de requerimientos toda los participantes pertenecen a un único sector. Precisamente el sector sobre el que se desean descubrir los requerimientos
 - El facilitador en un focus group y en un taller de requerimientos debe tomar acciones concretas(si es necesario) para asegurarse que cada participante tenga su espacio para opinar
 - La técnica utilizada para recibir feedback de prototipos suele ser el taller de requerimientos
 - Una de las ventajas de la técnica de observación es que las personas realizan sus tareas correctamente cuando son observadas

Capítulo 3: Técnicas de descubrimiento requisitos

Actividades

● Actividad 3.3 (cont.)

- Contestar las siguientes preguntas de acuerdo a las técnicas descritas en ~~[HBA-2009]~~ y [PMI-2015]
 - e) ¿Qué riesgo asumimos al aplicar técnicas como talleres de requerimientos o focus group sin haber hecho previamente un análisis de interesados o stakeholders?
 - No incluir a stakeholders cuya experiencia y conocimiento del problema puede llevar a la solución de un problema.
 - Invitar a personas que pueden intentar sabotear la actividad.
 - Ofender a interesados que no fueron invitados a la actividad.



Capítulo 4

Técnicas de análisis, modelado y especificación de requisitos

Capítulo 4: Técnicas de análisis, modelado y especificación de requisitos

Lecturas

- **[Hofer]** Hofer, Stefan, and Henning Schwentner. Domain Storytelling: A Collaborative, Visual, and Agile Way to Build Domain-Driven Software. Pearson Education, Limited, 2021 (capítulo 1)
- **[Larman]** Larman, Craig. Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development. Pearson Education India, 2012 (Capítulos 9 y 32)
- **[Lucassen]** Lucassen, Garm, et al. "Forging high-quality user stories: towards a discipline for agile requirements." 2015 IEEE 23rd international requirements engineering conference (RE). IEEE, 2015.
- **[PMI-2015]** Project Management Institute. Business Analysis for Practitioners: A Practice Guide. Project Management Institute, Incorporated, 2015. (técnicas correspondientes)
- **[Rose]** Rose, Seb and Gaspar Nagy. "Formulation: Document examples with Given/When/Then (Volume 2)." (2021) (capítulos 1 y 3)
- **[Wiegers]** Wiegers, Karl, and Joy Beatty. Software requirements. Pearson Education, 2013. Disponible en <http://wdz.eng.br/LivroRequirements3Ed.pdf> (Capítulos 5, 8, 14 y 15)

Lectura complementaria

- **[Brennan]** Brennan, Kevin, ed. A Guide to the Business Analysis Body of Knowledge. IIBA, 2009 (técnicas que correspondan)

Capítulo 4: Técnicas de análisis, modelado y especificación de requisitos

Actividad 4.1

- Contestar de acuerdo a lo planteado en el vídeo del capítulo 4:
Los requisitos pueden ser especificados desde al menos tres puntos de vista diferentes. Cada punto de vista describe:

	Qué es lo que el usuario debe poder hacer con el sistema/software	Por qué la organización necesita el sistema/software	Qué es lo que deben implementar los desarrolladores para que los usuarios puedan hacer sus tareas
Requisito de negocio	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Requisito de usuario	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Requisito de software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Capítulo 4: Técnicas de análisis, modelado y especificación de requisitos

Actividad 4.1 (cont.)

- Contestar de acuerdo a lo planteado en el vídeo del capítulo 4:

Clasificar los requisitos que se detallan a continuación :

	Funcional	No funcional	No es requisito
Para cada producto, el sistema deberá informar código, descripción, precio y cantidad en stock	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Diariamente, el sistema recibirá las novedades (nuevos productos, baja de productos, modificaciones de productos) del sistema de inventario	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
El desarrollo deberá esta terminado en el primer trimestre del año	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Para validar la transacción de pago con tarjeta de crédito, el sistema solicitará la autorización de la operación al gateway de pagos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

La interfaz del usuario debe ser fácil de usar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
El número de tarjeta de crédito no debe mostrarse en ninguna pantalla. Solamente deben exhibirse los últimos cuatro dígitos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cada cajero automático podrá interrumpir su servicio por tareas de mantenimiento no más de quince minutos por día	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Con el nuevo sistema, se espera reducir los costos de operación en un 10%	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Al final del día, el sistema deberá enviar al gateway de pago las transacciones realizadas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Un usuario deberá poder hacer la instalación del producto en 15 minutos o menos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Capítulo 4: Técnicas de análisis, modelado y especificación de requisitos

Actividad 4.6 – Historias de Usuario

Para cada una de las historias de usuario, determinar cuál es el criterio de calidad MÁS GRAVE que incumple ([Lucassen])

	Atómica	Mínima	Bien formada	Libre de conflictos	Conceptualmente razonable	Orientada al problema	No ambigua	Dependencias explícitas	Oración completa	Independiente	Escalable	Uniforme	Única
Como gateway de pago quiero recibir todas las transacciones para poder proceder con las mismas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Como alumno, quiero presionar un botón en la pantalla para inscribirme a una materia para poder cursarla	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Como sistema, quiero enviar las transacciones de pago al sistema de tarjetas, para que se procesen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Capítulo 4: Técnicas de análisis, modelado y especificación de requisitos

Actividad 4.6 - Historias de Usuario (cont.)

Para cada una de las historias de usuario, determinar cuál es el criterio de calidad MÁS GRAVE que incumple ([Lucassen])

	Atómica	Mínima	Bien formada	Libre de conflictos	Conceptualmente razonable	Orientada al problema	No ambigua	Dependencias explícitas	Oración completa	Independiente	Escalable	Uniforme	Única
Como sistema quiero instalar los nuevos comercios en los servidores propios para dar acceso a nuevos compradores	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Como comprador quiero consultar los productos ofrecidos y preparar el pedido sin necesidad de registrarse	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Como comprador quiero poder agregar o quitar productos de la consulta así como variar la cantidad de los mismos para tener más información	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



Capítulo 5

Técnicas de verificación y validación de requisitos

Capítulo 5: Técnicas de verificación y validación de requisitos

Lecturas

- **[Wiegers]** Wiegers, Karl, and Joy Beatty. Software requirements. Pearson Education, 2013. (Capítulo 17)

Capítulo 5: Técnicas de verificación y validación de requisitos

Actividad 5.1

- Sobre la base de lo propuesto por [Weigers], responder las siguientes preguntas:
 - Cuando desarrolladores y clientes trabajan en conjunto sobre la definición de los casos de prueba, aumenta la confianza acerca de que los requisitos son correctos. (V/F)
 - Los test conceptuales no son independientes de la implementación (V/F)
 - Los criterios de aceptación permiten determinar únicamente si el sistema desarrollado cumple con los requisitos documentados (V/F)



Capítulo 6

Prototipado y experiencia del usuario

Capítulo 6: Prototipado y experiencia del usuario

Lecturas

- **[Wiegers]** Wiegers, Karl, and Joy Beatty. Software requirements. Pearson Education, 2013. (Capítulo 15)

Capítulo 6: Prototipado y experiencia del usuario

Actividad 6.1

- A partir de lo planteado en el vídeo del capítulo y por [Weigers], responder:
 - La razón principal de crear prototipos es resolver incertidumbres tempranamente (V/F)
 - Una prueba de concepto se concentra en la experiencia del usuario (V/F)
 - Un wireframe es un tipo particular de prototipo evolutivo (V/F)
 - Un mock-up se puede usar para explorar diferentes enfoques para la interfaz del usuario (V/F)
 - Uno de los riesgos del prototipado es que los usuarios presionen para liberarlo a producción (V/F)
 - Una interfaz de usuario es una prueba de concepto (V/F)
 - Los principios de la experiencia del usuario son:
 - La interfaz del usuario es comunicación
 - Las tareas deben explicarse clara y concisamente, como si fuera en persona
 - Todos los elementos de la interfaz del usuario pueden ser evaluados por qué comunica y cómo lo hace
 - La interfaz del usuario siempre debe ofrecer ayuda en línea
 - Hay que ser amable, respetuoso e inteligente
 - Si la interfaz del usuario se siente como una comunicación natural, profesional y amigable, probablemente sea por su buen diseño



Capítulo 8

Diseño de software

Capítulo 8: Diseño de software

Lecturas

- **[Kruchten-1995]** Kruchten, Philippe B. "The 4+ 1 view model of architecture." *Software*, IEEE 12.6 (1995): 42-50.
- **[Larman]** Larman, Craig. *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development*. Pearson Education India, 2012 (Capítulos 13 y 33; se recomiendan también los capítulos 17, 18, 20 si no se está familiarizado con patrones GRASP)
- **[Richards]** Richards, Mark. *Software architecture patterns*. Vol. 4. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Incorporated, 2015.

Capítulo 8: Diseño de software

Actividades

- Actividad 8.1
 - A partir del paper de [Kruchten], contestar las siguientes preguntas:
 - ¿En cuáles de las siguientes vistas incluirían a la consideración de diseño: tolerancia a las fallas?:
 - Vista lógica
 - Vista física
 - Vista de desarrollo
 - Vista de procesos
 - En ninguna

Capítulo 8: Diseño de software

Actividades

- Actividad 8. 1 (cont.)
 - A partir del paper de [Kruchten], contestar las siguientes preguntas (cont.):
 - ¿Cuál de las siguientes vistas cree que es la que más le interesaría a un usuario final?:
 - Vista física
 - Vista de procesos
 - Vista lógica
 - Vista de estado

Capítulo 8: Diseño de software

Actividades

- Actividad 8.1 (cont.)
 - A partir del paper de [Kruchten], contestar las siguientes preguntas (cont.):
 - Indique cuáles de las siguientes afirmaciones son correctas respecto de la vista de escenarios:
 - Permite visualizar todas las funcionalidades del sistema
 - En un enfoque guiado por escenarios, se toma en cuenta el riesgo y la criticidad para ver cuales formarán parte de la iteración
 - Los escenarios son instanciaciones concretas de un caso de uso
 - Se utiliza para descubrir elementos de la arquitectura de la solución

Capítulo 8: Diseño de software

Actividades

- Actividad 8.1 (cont.)
 - A partir del paper de [Kruchten], contestar las siguientes preguntas (cont.):
 - ¿Cuál de las siguientes vistas se puede utilizar como base para medir el avance del proyecto?:
 - Vista lógica
 - Vista de desarrollo
 - Vista de desarrollo
 - Vista de procesos
 - El modelo implica el uso de la notación UML (V/F)

Capítulo 8: Diseño de software

Actividades

Arquitectura

- Actividad 8.2
 - A partir de los capítulos 13 y 33 de [Larman], contestar las siguientes preguntas:
 - Los principios de cohesión, acople y protección de variaciones se aplican a nivel de diseño de objetos únicamente. (V/F)
 - En una arquitectura en capas, la capa de lógica de la aplicación incluye las clases que implementan los objetos del modelo de dominio. (V/F)
 - El principio de Separación Modelo-Vista -en el que se basa el patrón MVC- establece que las clases que implementan los objetos de dominio deben tener visibilidad de las clases relacionadas con la interfaz del usuario. (V/F)
 - El análisis arquitectónico implica:
 - Identificar y analizar los requisitos no funcionales
 - Identificar y analizar los requisitos funcionales que tengan impacto en la arquitectura
 - Identificar y analizar los requisitos no funcionales que tengan impacto en la arquitectura
 - Analizar alternativas y crear soluciones para resolver el impacto de cada uno de los requisitos funcionales y no funcionales
 - Analizar alternativas y crear soluciones para resolver el impacto de cada uno de los requisitos arquitectónicamente significativos

Capítulo 8: Diseño de software

Actividades

Arquitectura

- Actividad 8.3
 - Según [Richards]:
 - La principal ventaja de la arquitectura en capas es su agilidad (V/F)
 - Los sistemas con arquitectura event-driven son más fáciles de probar que los de arquitectura orientada a microservicios (V/F)
 - Las arquitecturas de microservicios:
 - Son distribuídas
 - Presentan alto acople y baja cohesión
 - Están compuestas por componentes de servicios
 - Cada unidad es desplegada en forma independiente
 - Por su naturaleza distribuida, son difíciles de probar.
 - Surgieron con el propósito de resolver los problemas con las aplicaciones monolíticas en capas y con las aplicaciones distribuidas desarrolladas con SOA



Capítulo 11

Gestión del desarrollo de software y modelos de proceso

Capítulo 11: Gestión del desarrollo de software y modelos de proceso

Lecturas

- [Boehm] Boehm, Barry W. "A spiral model of software development and enhancement." Computer 21.5 (1988): 61-72.
- [Royce] Royce, Winston W. "Managing the development of large software systems." Proceedings of IEEE WESCON. Vol. 26. No. 8. 1970.

Capítulo 11: Gestión del desarrollo de software y modelos de proceso

Actividad 11.1 – Ciclos de vida y modelos de proceso

- A partir de los artículos de [Royce] y [Boehm] responder las siguientes preguntas:
 - a) Según [Boehm], las funciones primarias de un modelo de proceso de desarrollo de software son (marcar una o más opciones):
 - Describir los entregables
 - Determinar el orden de las actividades
 - Determinar el orden de las etapas
 - Describir cuáles son los criterios de transición de una actividad a otra
 - Describir cuáles son los criterios de transición de una etapa a otra
 - b) Según [Boehm], los principales problemas del modelo Code & Fix son (marcar una o más opciones)
 - Dificultades para estimar el esfuerzo necesario
 - Falta de alineamiento con las necesidades del usuario
 - Dificultades para identificar casos de prueba
 - Mala estructura del Código
 - Alto costo de las modificaciones debido a la poca preparación para pruebas y modificaciones

Capítulo 11: Gestión del desarrollo de software y modelos de proceso

Actividad 11.1 – Ciclos de vida y modelos de proceso

- c) Según [Royce], el mayor problema del ciclo de vida en cascada es su elevado:
 - a) Costo
 - b) Riesgo
 - c) Tiempo
- d) ¿Qué propone Royce para mitigar los riesgos de desarrollar software siguiendo el ciclo de cascada?
 - a) Probar cada secuencia de pasos lógica del sistema construido
 - b) Construir un prototipo antes de comenzar el análisis del sistema
 - c) Elegir una metodología ágil que se enfoque en maximizar el valor otorgado al cliente
 - d) Estimar cada tarea antes de llevarla adelante
 - e) Agregar una etapa de diseño preliminar antes de llevar adelante el análisis
 - f) Documentar claramente el equipo de trabajo que construirá la solución

Capítulo 11: Gestión del desarrollo de software y modelos de proceso

Actividad 11.1 – Ciclos de vida y modelos de proceso

d) Indicar cuáles de las siguientes afirmaciones acerca del modelo del espiral propuesto en [Boehm] son correctas:

- La dimensión radial indica cuánto se ha progresado en una fase
- El costo acumulado está representado por la dimensión vertical
- En cada comienzo de ciclo se debe identificar un objetivo que cada ciclo que pasa debe ser más específico
- Si es falsa la hipótesis del espiral en un determinado momento, se debe dar por concluido el ciclo actual y comenzar uno nuevo
- La incertidumbre es mayor en el inicio de los proyectos. Por esta razón, en los ciclos iniciales, se debe hacer un análisis de riesgo que deja de ser necesario en los ciclos finales
- Ninguna de las anteriores es correcta



Capítulo 12

Peopleware

Capítulo 12: Peopleware

Lecturas

- **[Wrobel]** Wrobel, Michal R. "Emotions in the software development process." *2013 6th International Conference on Human System Interactions (HSI)*. IEEE, 2013.

Capítulo 12: Peopleware

Actividad 12.1

En no más de una carilla, resumir las principales ideas del artículo de [Wrobel].