



Universidad Nacional Autónoma de México
Facultad de Ingeniería
Ingeniería de Software



Grupo 4
Ing. Orlando Zaldívar Zamorategui

Cuestionario con respuestas

425490518 - Barrancos Parada Tiago
320304435 - Cabrera Rojas Oscar
322308381 - Calva Gatica Ángel David
319154780 - Esquivel Rosales Josselyn Anaid
320318467 - Estrada Martínez Mitzy Naomi

1. Cuestionario

A continuación se presentan las preguntas de evaluación sobre los fundamentos de los requerimientos funcionales. En cada pregunta, la opción correcta aparece resaltada en verde con la marca (*resp. correcta*).

1. Según los autores, ¿por qué los requerimientos son la base de todo proyecto de software?

- a) **Porque describen las funcionalidades y características que el software debe tener para satisfacer las necesidades de los usuarios y/o clientes (Celi-Párraga et al., 2023, p. 3). (*resp. correcta*)**
- b) Porque definen las pruebas unitarias que el equipo de calidad ejecutará durante la fase de verificación.
- c) Porque establecen el lenguaje de programación y la arquitectura de despliegue del producto final.
- d) Porque determinan el presupuesto y el cronograma de entrega comprometido con el cliente.

2. ¿Cuál es la advertencia que hacen Contreras & Hernández respecto a la dicotomía funcional/no funcional?

- a) Que los requerimientos no funcionales deben redactarse siempre antes que los funcionales.
- b) **Que con frecuencia se mencionan dos grandes tipos de requerimientos, funcionales y no funcionales, aunque conviene tener cuidado con estas definiciones, puesto que no es raro encontrar que exista una relación entre ambas más cercana de lo que se espera (Contreras Bernal & Hernández Ruiz, 2022, p. 93). (*resp. correcta*)**
- c) Que la separación entre funcionales y no funcionales es absoluta y nunca debe relativizarse.
- d) Que los requerimientos no funcionales son siempre opcionales mientras que los funcionales son obligatorios.

3. ¿Cuál es la frase central de Sommerville sobre los requerimientos funcionales que casi todos los autores latinoamericanos recogen?

- a) Los requerimientos funcionales son atributos de calidad del sistema que deben ser medibles cuantitativamente.
- b) **Los requerimientos funcionales para un sistema refieren lo que el sistema debe hacer (Sommerville, 2011, §4.1.1, p. 85). (*resp. correcta*)**
- c) Los requerimientos funcionales describen exclusivamente las restricciones operacionales del sistema.
- d) Los requerimientos funcionales son las condiciones contractuales entre el cliente y el desarrollador.

4. ¿Cómo formula Sommerville de manera extendida los requerimientos funcionales?

- a) Como diagramas UML que representan la estructura interna del sistema y sus dependencias.
- b) Como listas de chequeo derivadas de los estándares ISO 9001 aplicados al desarrollo de software.
- c) **Como enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas; en algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema (Sommerville, 2011, §4.1, p. 84). (*resp. correcta*)**

- d) Como métricas de rendimiento que miden la latencia y el throughput del sistema bajo carga.

5. Según Celi-Párraga et al., ¿cómo se definen los requerimientos funcionales?

- a) Como restricciones impuestas por el hardware sobre la lógica de negocio del sistema.
- b) Como criterios de aceptación firmados por el patrocinador del proyecto.
- c) Como contratos legales entre el equipo de desarrollo y el área de operaciones.
- d) **Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas; en algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema (Celi-Párraga et al., 2023, §3.3, p. 39). (resp. correcta)**

6. ¿Qué frase-resumen ofrecen Celi-Párraga et al. tomada directamente de Sommerville?

- a) **Los requerimientos funcionales para un sistema refieren lo que el sistema debe hacer (Celi-Párraga et al., 2023, §3.3, p. 39). (resp. correcta)**
- b) Los requerimientos funcionales son los componentes mínimos verificables del sistema.
- c) Los requerimientos funcionales son la traducción técnica de los objetivos del negocio.
- d) Los requerimientos funcionales son siempre derivados de los requerimientos no funcionales.

7. ¿Cómo describe Laplante los *functional requirements* (FRs) de manera industrial pero compacta?

- a) Como las propiedades de calidad medibles del producto en producción.
- b) Como los algoritmos específicos que el equipo de desarrollo implementará.
- c) **Los *functional requirements* (FRs) describen los servicios que el sistema debe proveer y cómo el sistema reaccionará ante sus entradas (Laplante, 2017, Cap. 1, p. 6). (resp. correcta)**
- d) Como las interfaces gráficas con las que el usuario interactuará.

8. ¿Qué énfasis añade Laplante, igual que Sommerville, sobre lo que el sistema no debe hacer?

- a) **Los requerimientos funcionales necesitan establecer explícitamente ciertos comportamientos que el sistema no debe ejecutar (Laplante, 2017, Cap. 1, p. 6). (resp. correcta)**
- b) Los requerimientos funcionales solo deben describir comportamientos positivos del sistema.
- c) Los requerimientos funcionales nunca deben mencionar restricciones negativas porque generan ambigüedad.
- d) Los requerimientos funcionales deben enumerar exclusivamente los casos de éxito del sistema.

9. Según SWEBOK, ¿qué describen los requerimientos funcionales?

- a) Las propiedades de mantenibilidad y portabilidad que el software debe exhibir.
- b) **Los requerimientos funcionales describen las funciones que el software debe ejecutar; por ejemplo, formatear un texto o modular una señal; a veces se les conoce como *capabilities* o *features* (Bourque & Fairley, 2014, Software Requirements KA, §1.3, p. 1-3). (resp. correcta)**
- c) Los protocolos de comunicación entre los componentes distribuidos del sistema.

d) La estructura de la base de datos relacional y sus restricciones de integridad.

10. ¿Cuál es el criterio de identificación operativo que aporta SWEBOK para los requerimientos funcionales?

- a) Que sean comprensibles únicamente por el equipo de arquitectura del sistema.
- b) Que se redacten exclusivamente con verbos en voz pasiva para evitar ambigüedad.
- c) Que estén numerados según el estándar ANSI/IEEE 830 sin excepciones.
- d) **Un requerimiento funcional puede describirse como aquél para el cual puede escribirse un conjunto finito de pasos de prueba que validen su comportamiento (Bourque & Fairley, 2014, Software Requirements KA, §1.3, p. 1-3). (resp. correcta)**

11. Según SWEBOK, ¿cuál es la propiedad esencial de todos los requerimientos de software?

- a) Que estén traducidos al idioma del cliente final como mínimo en dos versiones.
- b) Que sean escritos por un ingeniero certificado en metodologías ágiles.
- c) **Una propiedad esencial de todos los requerimientos de software es que sean verificables como característica individual en el caso de un requerimiento funcional, o a nivel del sistema en el caso de un requerimiento no funcional (Bourque & Fairley, 2014, Software Requirements KA, §1.1, p. 1-2). (resp. correcta)**
- d) Que tengan asociado un costo estimado en horas-persona desde su redacción inicial.

12. ¿Cuál es la formulación más breve y precisa de un requerimiento funcional según Wieggers & Beatty?

- a) **Un requerimiento funcional es una descripción de un comportamiento que un sistema exhibirá bajo condiciones específicas (Wieggers & Beatty, 2013, Tabla 1-1, p. 7). (resp. correcta)**
- b) Un requerimiento funcional es una promesa contractual del proveedor frente al cliente.
- c) Un requerimiento funcional es la implementación detallada de un caso de uso del sistema.
- d) Un requerimiento funcional es una métrica cuantitativa del rendimiento del sistema.

13. ¿Qué precisa la versión ampliada de la definición de Wieggers sobre los tres niveles de requerimientos?

- a) Que los requerimientos funcionales deben escribirse después del código fuente.
- b) Que los requerimientos funcionales se redactan únicamente por el equipo de pruebas.
- c) **Los requerimientos funcionales especifican los comportamientos que el producto exhibirá bajo condiciones específicas, describiendo lo que los desarrolladores deben implementar para que los usuarios puedan cumplir sus tareas (requerimientos del usuario), satisfaciendo así los requerimientos del negocio; esta alineación entre los tres niveles de requerimientos es esencial para el éxito del proyecto (Wieggers & Beatty, 2013, p. 9). (resp. correcta)**
- d) Que los requerimientos funcionales son irrelevantes para los requerimientos del negocio.

14. Según Robertson & Robertson, ¿qué es en esencia un requerimiento funcional?

- a) Una restricción técnica derivada de la plataforma elegida para el despliegue.
- b) Un atributo de calidad medible bajo condiciones de prueba controladas.
- c) Un fragmento de código documentado que implementa una función del sistema.

- d) **En esencia, un requerimiento funcional es algo que el producto debe hacer para soportar el negocio de su propietario (Robertson & Robertson, 2012, Cap. 1, Truth 10, p. 8). (resp. correcta)**

15. ¿Cuál es la definición canónica de apertura del Capítulo 10 de Robertson & Robertson?

- a) **Los requerimientos funcionales especifican lo que el producto debe hacer: las acciones que debe realizar para satisfacer las razones fundamentales de su existencia (Robertson & Robertson, 2012, Cap. 10, p. 223). (resp. correcta)**
- b) Los requerimientos funcionales son los atributos no negociables del contrato comercial.
- c) Los requerimientos funcionales describen únicamente la interfaz gráfica del usuario final.
- d) Los requerimientos funcionales son las restricciones legales aplicables al producto.

16. Según Contreras & Hernández, ¿cómo se definen los requerimientos funcionales?

- a) Como las pruebas de aceptación firmadas por el cliente al final del proyecto.
- b) Como los componentes reutilizables del catálogo organizacional de patrones.
- c) **Aquellos que describen con detalle lo que el sistema debe hacer, es decir, su comportamiento a partir de indicaciones sobre cuáles son las excepciones, entradas y salidas del mismo; por lo anterior, dependen mucho del tipo de software, de los usuarios, así como de lo específico y particular que pueda ser el negocio de la organización (Contreras Bernal & Hernández Ruiz, 2022, p. 93). (resp. correcta)**
- d) Como las cláusulas penales del contrato de desarrollo de software.

17. ¿Cuál es el esquema de identificación que propone Pressman para clasificar requisitos?

- a) A = aceptación, B = base de datos, C = código, D = despliegue, E = entrega.
- b) P = prueba, Q = calidad, R = rendimiento, S = seguridad, T = tiempo.
- c) U = usuario, V = vista, W = workflow, X = excepción, Y = yield.
- d) **F = requisito funcional, D = requisito de datos, C = requisito de comportamiento, Z = requisito de interfaz, y S = requisito de salida; un requisito identificado como F09 indica que se trata de un requisito funcional y que tiene asignado el número 9 dentro de los citados requisitos (Pressman, 2002, §10.5.6, p. 175). (resp. correcta)**

18. ¿Cómo define Pressman la calidad del software por contraste?

- a) Como la satisfacción del cliente medida en encuestas posteriores al lanzamiento del producto.
- b) **La calidad del software se define como concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente (Pressman, 2002, Cap. 8, p. 135). (resp. correcta)**
- c) Como el cumplimiento exclusivo del cronograma de entrega original del proyecto.
- d) Como la cantidad de líneas de código documentadas con comentarios en línea.

19. Según Sommerville, ¿de qué dependen los requerimientos funcionales?

- a) Del estilo personal de redacción del analista de requerimientos asignado al proyecto.

- b) **Los requerimientos funcionales dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos (Sommerville, 2011, §4.1.1, p. 85). (resp. correcta)**
- c) Exclusivamente de la plataforma tecnológica seleccionada para el despliegue.
- d) Únicamente del presupuesto disponible para el proyecto en el ciclo fiscal.

20. ¿Cuál es la formulación de Sommerville sobre el modelo binario clásico usuario/sistema?

- a) Que los requerimientos del usuario son siempre más detallados que los del sistema.
- b) Que los requerimientos funcionales solo aplican al nivel de sistema, nunca al de usuario.
- c) **Al expresarse como requerimientos del usuario, los requerimientos funcionales se describen por lo general de forma abstracta que entiendan los usuarios del sistema; sin embargo, requerimientos funcionales más específicos del sistema detallan las funciones del sistema, sus entradas y salidas, sus excepciones, etcétera (Sommerville, 2011, §4.1.1, p. 85). (resp. correcta)**
- d) Que el modelo binario fue descartado por la literatura moderna en favor de uno ternario.

21. ¿Cómo formulan Celi-Párraga la diferencia entre requerimientos del usuario y del sistema?

- a) **Los requerimientos del usuario son enunciados, en un lenguaje natural junto con diagramas, acerca de qué servicios esperan los usuarios del sistema, y de las restricciones con las cuales éste debe operar; los requerimientos del sistema son descripciones más detalladas de las funciones, los servicios y las restricciones operacionales del sistema de software (Celi-Párraga et al., 2023, §3.2, pp. 37–38). (resp. correcta)**
- b) Los requerimientos del usuario son atributos de calidad mientras que los del sistema son funcionales.
- c) Los requerimientos del usuario son verbales mientras que los del sistema son siempre matemáticos.
- d) Los requerimientos del usuario y del sistema son sinónimos intercambiables sin diferencia técnica.

22. Según Sommerville, ¿qué rango interno tienen los requerimientos funcionales del sistema?

- a) **Los requerimientos funcionales del sistema varían desde requerimientos generales que cubren lo que tiene que hacer el sistema, hasta requerimientos muy específicos que reflejan maneras locales de trabajar o los sistemas existentes de una organización (Sommerville, 2011, §4.1.1, p. 85). (resp. correcta)**
- b) Los requerimientos funcionales del sistema son uniformes y siempre del mismo nivel de detalle.
- c) Los requerimientos funcionales del sistema son siempre más generales que los del usuario.
- d) Los requerimientos funcionales del sistema solo aplican a sistemas embebidos críticos.

23. ¿Cómo reproduce Laplante la estratificación clásica usuario/sistema?

- a) Indicando que los requerimientos funcionales son siempre del mismo nivel sin estratificación posible.
- b) Indicando que la estratificación clásica solo aplica a sistemas distribuidos de gran escala.

- c) **Los requerimientos funcionales pueden ser de alto nivel y generales —en cuyo caso son requerimientos del usuario— o pueden ser detallados, expresando entradas, salidas, excepciones, etcétera —en cuyo caso son requerimientos del sistema (Laplante, 2017, Cap. 1, p. 6). (resp. correcta)**
- d) Indicando que los requerimientos detallados son siempre del usuario y los generales del sistema.

24. ¿Qué tres niveles distintos de requerimientos del software introducen Wiegers & Beatty?

- a) Niveles de prioridad alta, media y baja según el impacto en el negocio.
- b) Niveles de redacción técnica, semitécnica y no técnica según la audiencia.
- c) **Los requerimientos del software incluyen tres niveles distintos: requerimientos de negocio, requerimientos del usuario y requerimientos funcionales; adicionalmente, todo sistema tiene una colección de requerimientos no funcionales (Wiegers & Beatty, 2013, p. 7). (resp. correcta)**
- d) Niveles de validación, verificación y aceptación a lo largo del ciclo de vida.

25. Según Wiegers & Beatty, ¿qué es una *feature*?

- a) **Una *feature* consiste en una o más capacidades del sistema lógicamente relacionadas que proveen valor a un usuario y son descritas por un conjunto de requerimientos funcionales (Wiegers & Beatty, 2013, p. 7). (resp. correcta)**
- b) Una *feature* es un fragmento de código en un repositorio Git aislado.
- c) Una *feature* es un atributo no funcional medido en términos de tiempo de respuesta.
- d) Una *feature* es un documento legal que firma el cliente al recibir el producto.

26. ¿Cómo describen Wiegers & Beatty la distinción de perspectiva entre requerimientos del usuario y funcionales?

- a) Los requerimientos del usuario son técnicos y los funcionales son comerciales.
- b) **Los requerimientos del usuario describen la vista del usuario; los requerimientos funcionales describen la vista del desarrollador (Wiegers & Beatty, 2013, p. 89). (resp. correcta)**
- c) Los requerimientos del usuario son obligatorios y los funcionales son opcionales.
- d) Los requerimientos del usuario y los funcionales coinciden completamente en su perspectiva.

27. ¿Qué jerarquía no subjetiva proponen Robertson & Robertson?

- a) Una jerarquía de tres niveles: estratégico, táctico y operacional con asignación de roles.
- b) Una jerarquía de cuatro niveles basada en el modelo de madurez CMMI clásico.
- c) **Una jerarquía no subjetiva de cinco niveles —contexto de trabajo, evento de negocio, caso de uso de producto, paso del caso de uso de producto, y requerimiento atómico—. Los requerimientos funcionales bien redactados son requerimientos atómicos: ayuda a pensar en los requerimientos atómicos como el nivel más bajo de especificación de requerimientos (Robertson & Robertson, 2012, Cap. 10, p. 238). (resp. correcta)**
- d) Una jerarquía plana sin subordinación entre tipos de requerimientos.

28. ¿Cómo describen Celi-Párraga los requerimientos no funcionales?

- a) **Los requerimientos no funcionales son limitaciones sobre servicios o funciones que ofrece el sistema; incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares; los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema (Celi-Párraga et al., 2023, §3.3, p. 40). (resp. correcta)**
- b) Los requerimientos no funcionales son siempre opcionales y pueden omitirse del SRS.
- c) Los requerimientos no funcionales describen únicamente la interfaz visual del producto.
- d) Los requerimientos no funcionales son siempre derivables a partir del código fuente final.

29. Según Celi-Párraga, ¿qué dice el complemento sobre los requerimientos no funcionales?

- a) Que los requerimientos no funcionales coinciden completamente con los servicios al usuario.
- b) **Los requerimientos no funcionales, como indica su nombre, son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios (Celi-Párraga et al., 2023, §3.3, p. 41). (resp. correcta)**
- c) Que los requerimientos no funcionales solo aplican a sistemas en la nube.
- d) Que los requerimientos no funcionales son sustitutos directos de los funcionales.

30. ¿En qué tres grandes categorías se subdividen los NFR según Celi-Párraga?

- a) Categorías de prioridad alta, prioridad media y prioridad baja según importancia.
- b) **Los NFR se subdividen en tres grandes categorías: del producto (usabilidad, eficiencia, dependibilidad, seguridad), organizacionales (entorno, organizacionales, desarrollo) y externos (regulatorios, éticos, legislativos) (Celi-Párraga et al., 2023, §3.3, p. 41). (resp. correcta)**
- c) Categorías cuantitativas, cualitativas y mixtas según su forma de medición técnica.
- d) Categorías obligatorias, recomendables y opcionales según el contrato firmado.

31. ¿Qué advertencia da Sommerville sobre la frontera difusa entre FR y NFR?

- a) Que los NFR siempre deben expresarse junto con los FR y nunca por separado.
- b) **Si los requerimientos no funcionales se expresan por separado de los requerimientos funcionales, las relaciones entre ambos serían difíciles de entender (Sommerville, 2011, Cap. 4, p. 90). (resp. correcta)**
- c) Que los NFR y FR son completamente independientes y no requieren ser correlacionados.
- d) Que los NFR siempre deben omitirse cuando los FR ya están bien documentados.

32. Según SWEBOK, ¿cómo se relacionan estructuralmente los requerimientos de calidad con los funcionales?

- a) Los requerimientos de calidad sustituyen completamente a los requerimientos funcionales.
- b) Los requerimientos de calidad son independientes de los requerimientos funcionales.
- c) **Los requerimientos de calidad del software son en realidad atributos de (o restricciones sobre) los requerimientos funcionales (lo que el sistema hace); este KA busca claridad usando software quality en el sentido más amplio y usando los requerimientos de calidad del software como restricciones sobre los requerimientos funcionales (Bourque & Fairley, 2014, Software Quality KA, p. 10-1). (resp. correcta)**
- d) Los requerimientos de calidad solo aplican al hardware del sistema, no al software.

33. ¿Cómo caracteriza SWEBOK los requerimientos no funcionales?

- a) Los requerimientos no funcionales son aquellos que actúan para restringir la solución; a veces se les conoce como *constraints* o *quality requirements*; pueden clasificarse adicionalmente en requerimientos de rendimiento, mantenibilidad, seguridad (safety), confiabilidad, seguridad (security), interoperabilidad u otros (Bourque & Fairley, 2014, Software Requirements KA, §1.3, p. 1-3). (*resp. correcta*)
- b) Los requerimientos no funcionales son sinónimos exactos de los requerimientos del usuario.
- c) Los requerimientos no funcionales solo se refieren a aspectos legales del sistema.
- d) Los requerimientos no funcionales se clasifican exclusivamente en dos categorías: técnicos y comerciales.

34. ¿Cómo se asocian los NFR con los requerimientos funcionales según Laplante?

- a) Los NFR pueden asociarse a los requerimientos funcionales, por ejemplo: «solo usuarios autorizados deberían poder acceder a la funcionalidad X del sistema» (Laplante, 2017, p. 10). (*resp. correcta*)
- b) Los NFR nunca pueden asociarse a los requerimientos funcionales por su naturaleza.
- c) Los NFR siempre se asocian uno a uno con cada requerimiento funcional sin excepciones.
- d) Los NFR son completamente equivalentes a los requerimientos funcionales sin distinción.

35. ¿Qué ejemplo paradigmático dan Wiegers & Beatty sobre la trazabilidad NFR→FR?

- a) Los requerimientos de seguridad para autenticación de usuarios dan lugar a requerimientos funcionales derivados que podrían implementarse a través de funcionalidad de contraseñas o biometría (Wiegers & Beatty, 2013, p. 290). (*resp. correcta*)
- b) Los requerimientos de rendimiento siempre se traducen literalmente en código sin pasar por FR.
- c) Los requerimientos de usabilidad nunca generan requerimientos funcionales derivados.
- d) Los requerimientos de mantenibilidad son sustituidos directamente por los funcionales.

36. ¿Cuál es la taxonomía de tres tipos que propone Laplante?

- a) Requerimientos técnicos, comerciales y legales según el ámbito de aplicación.
- b) Requerimientos del cliente, del proveedor y del usuario final según los actores.
- c) Otra taxonomía para las especificaciones de requerimientos se enfoca en el tipo de requerimiento de la siguiente lista de posibilidades: requerimientos funcionales (FRs), requerimientos no funcionales (NFRs) y requerimientos de dominio (Laplante, 2017, Cap. 1, p. 6). (*resp. correcta*)
- d) Requerimientos obligatorios, deseables y opcionales según la prioridad asignada.

37. ¿Cómo describe Laplante los requerimientos de dominio?

- a) Los requerimientos de dominio son derivados del dominio de aplicación; estos tipos de requerimientos pueden consistir en nuevos requerimientos funcionales o restricciones a requerimientos funcionales existentes, o pueden especificar cómo deben realizarse cómputos particulares (Laplante, 2017, p. 11). (*resp. correcta*)

- b) Los requerimientos de dominio son requerimientos derivados exclusivamente del marketing del producto.
- c) Los requerimientos de dominio son una invención hispanoamericana sin equivalente internacional.
- d) Los requerimientos de dominio son sinónimos exactos de los requerimientos no funcionales.

38. ¿Cómo formulan Celi-Párraga el principio rector sobre la redacción de requerimientos del usuario?

- a) Que los requerimientos del usuario deben escribirse en notación matemática formal exclusivamente.
- b) Que los requerimientos del usuario deben incluir detalles de la arquitectura del sistema.
- c) Que los requerimientos del usuario deben usar lenguaje técnico especializado para garantizar precisión.
- d) **Los requerimientos del usuario para un sistema deben describir los requerimientos funcionales y no funcionales, de forma que sean comprensibles para los usuarios del sistema que no cuentan con un conocimiento técnico detallado; el documento de requerimientos no debe incluir detalles de la arquitectura o el diseño del sistema; en consecuencia, si se escriben los requerimientos del usuario, no se debe usar jerga de software, anotaciones estructuradas o formales; deben escribirse los requerimientos del usuario en lenguaje natural, con tablas y formas sencillas, así como diagramas intuitivos (Celi-Párraga et al., 2023, §3.5, p. 44). (resp. correcta)**

39. Según Celi-Párraga, ¿cómo cambian las herramientas a nivel del sistema?

- a) **Los requerimientos del usuario se escriben casi siempre en lenguaje natural, complementado con diagramas y tablas adecuados en el documento de requerimientos; los requerimientos del sistema se escriben también en lenguaje natural, pero de igual modo se utilizan otras notaciones basadas en formas, modelos gráficos del sistema o modelos matemáticos del sistema (Celi-Párraga et al., 2023, §3.5, p. 44). (resp. correcta)**
- b) Los requerimientos del sistema deben escribirse exclusivamente en pseudocódigo verificable.
- c) Los requerimientos del sistema solo pueden escribirse en inglés técnico estandarizado.
- d) Los requerimientos del sistema y del usuario deben escribirse en formatos completamente disjuntos.

40. ¿Cuáles son los cinco lineamientos que recomienda Sommerville para minimizar la interpretación errónea?

- a) (i) uso obligatorio de pseudocódigo; (ii) todos los verbos en pasado; (iii) sin negaciones; (iv) máximo cinco palabras por requerimiento; (v) sin numeración.
- b) **(i) formato estándar uniforme para todos los requerimientos —un requerimiento, una oración— asociado con su razón y fuente; (ii) distinguir obligatorio de deseable mediante verbos consistentes («debe» vs. «debería»); (iii) resaltar las partes clave con negrilla, cursiva o color; (iv) evitar la jerga técnica; (v) asociar una razón a cada requerimiento de usuario para facilitar el cambio futuro (Sommerville, 2011, p. 96). (resp. correcta)**
- c) (i) exclusivamente lenguaje natural; (ii) sin formato visual; (iii) con jerga técnica detallada; (iv) sin razones; (v) sin estructura uniforme.
- d) (i) siempre en idioma inglés; (ii) con notación UML obligatoria; (iii) sin distinguir prioridad; (iv) con verbos imperativos; (v) con anexos legales obligatorios.

41. ¿Qué información debe incluir la plantilla estructurada de siete campos de Sommerville?

- a) (1) hora; (2) fecha; (3) lugar; (4) persona; (5) equipo; (6) tarea; (7) estado.
- b) (1) costo; (2) plazo; (3) recursos; (4) prioridad; (5) riesgo; (6) impacto; (7) beneficio.
- c) **(1) función o entidad —descripción de qué se especifica; (2) entradas y procedencias; (3) salidas y destinos; (4) requiere —datos u otras entidades del sistema necesarios para el cálculo; (5) acción —qué se va a hacer; (6) precondición y postcondición; (7) efectos colaterales (Sommerville, 2011, §4.3.2, p. 98). (resp. correcta)**
- d) (1) título; (2) autor; (3) fecha; (4) versión; (5) estado; (6) aprobador; (7) revisor.

42. Según Robertson & Robertson, ¿cuál es el nivel de detalle revelador en sus ejemplos?

- a) **Los requerimientos están escritos como una única oración con un único verbo; cuando se escribe esta única oración, se hace el requerimiento mucho más fácil de probar y mucho menos susceptible a la ambigüedad (Robertson & Robertson, 2012, Cap. 10, p. 228). (resp. correcta)**
- b) Los requerimientos están escritos como párrafos extensos con múltiples verbos coordinados.
- c) Los requerimientos están escritos exclusivamente como diagramas UML sin texto narrativo.
- d) Los requerimientos están escritos sin verbos para evitar ambigüedad sintáctica.

43. ¿Cuál es la forma canónica que recomiendan Robertson & Robertson?

- a) «Could be considered to...» para mantener flexibilidad en la implementación.
- b) «It would be nice if...» para no presionar al equipo de desarrollo.
- c) «The system might possibly...» para dejar abierta la prioridad del requerimiento.
- d) **La forma canónica es «The product shall...»: hace la oración activa y se enfoca en comunicar lo que el producto pretende hacer (Robertson & Robertson, 2012, p. 228). (resp. correcta)**

44. ¿Por qué desaconsejan Robertson & Robertson mezclar verbos modales como shall, must, will, might, could?

- a) Porque la lengua inglesa no admite combinaciones de modales en un mismo documento.
- b) **A veces las personas usan una mezcla de «shall», «must», «will», «might», «could», etc., para indicar la prioridad de un requerimiento; esta práctica resulta en confusión semántica y se aconseja contra ella; en su lugar, debe usarse una forma consistente para escribir las descripciones de los requerimientos — «The product shall...» es la más común— y usar un atributo separado del requerimiento para identificar su prioridad (Robertson & Robertson, 2012, p. 229). (resp. correcta)**
- c) Porque los verbos modales están prohibidos por el estándar IEEE 830 vigente.
- d) Porque los verbos modales solo aplican a requerimientos de calidad pero no a funcionales.

45. ¿Qué exigen Robertson & Robertson sobre el *rationale* en cada requerimiento?

- a) Que el *rationale* sustituya por completo a la descripción del requerimiento.
- b) Que el *rationale* solo se incluya cuando el requerimiento sea de baja prioridad.
- c) **Se sugiere —fuertemente— que se añada un *rationale* a los requerimientos para mostrar por qué existe el requerimiento; en algunos casos puede ser obvio, pero en muchas circunstancias es un componente crucial del requerimiento (Robertson & Robertson, 2012, p. 229). (resp. correcta)**

- d) Que el *rationale* debe redactarse en pseudocódigo formal exclusivamente.

46. ¿Cuál es el ejemplo concreto de Description + Rationale dado por Robertson & Robertson?

- a) **Description: «The product shall record roads that have been treated.» Rationale: «To be able to schedule untreated roads and highlight potential danger.» (Robertson & Robertson, 2012, pp. 229–230). (resp. correcta)**
- b) Description: «El sistema procesará pagos.» Rationale: «Porque el cliente lo necesita.»
- c) Description: «The product manages users.» Rationale: «It is required by law.»
- d) Description: «Sistema de inventario.» Rationale: «Para vender más productos.»

47. Según Laplante (IEEE 29148), ¿qué deben capturar los requerimientos funcionales?

- a) **Los requerimientos funcionales deben capturar todas las entradas del sistema y la secuencia exacta de operaciones y respuestas (salidas) ante situaciones normales y anormales para cada posibilidad de entrada (Laplante, 2017, Cap. 4, p. 98). (resp. correcta)**
- b) Los requerimientos funcionales deben capturar exclusivamente los casos de éxito felices.
- c) Los requerimientos funcionales deben capturar únicamente las restricciones legales aplicables.
- d) Los requerimientos funcionales deben capturar solo los flujos de la interfaz gráfica.

48. Según Laplante, ¿cuáles son las opciones organizacionales que provee IEEE 29148 para los requerimientos funcionales?

- a) **IEEE 29148 provee un número de opciones organizacionales para los requerimientos funcionales; los requerimientos funcionales pueden organizarse por: modo del sistema, clase de usuario, objeto, característica, estímulo, jerarquía funcional (Laplante, 2017, p. 99). (resp. correcta)**
- b) IEEE 29148 obliga a organizar los requerimientos funcionales únicamente por orden alfabético del autor.
- c) IEEE 29148 prohíbe cualquier organización por estímulo o respuesta del sistema.
- d) IEEE 29148 solo permite organización por costo estimado y prioridad de implementación.

49. ¿Qué convención más rica que la de Pressman ofrecen Wiegers & Beatty para identificar requerimientos?

- a) Identificación obligatoria mediante hash criptográfico SHA-256 generado del texto.
- b) Identificación únicamente por nombre del autor seguido de la fecha de redacción.
- c) **El enfoque más simple da a cada requerimiento un número de secuencia único, como UC-9 o FR-26; el prefijo indica el tipo de requerimiento, como FR para requerimiento funcional (Wiegers & Beatty, 2013, p. 187). (resp. correcta)**
- d) Identificación obligatoria mediante código de barras impreso en cada documento.

50. ¿Qué buena convención sugieren Wiegers & Beatty para los requerimientos jerárquicos?

- a) Que los requerimientos padre y los hijos sean idénticos para mantener consistencia.
- b) **Una buena convención es escribir el requerimiento padre para que parezca un título, encabezado o nombre de feature, en lugar de parecerse a un requerimiento funcional en sí mismo; los requerimientos hijos de ese padre, en agregado, entregan la capacidad descrita en el padre (Wiegers & Beatty, 2013, p. 188). (resp. correcta)**

- c) Que los requerimientos jerárquicos no deben usarse en proyectos profesionales.
- d) Que el requerimiento padre debe ser el más detallado y los hijos los más generales.

51. ¿Cuáles son los atributos individuales de un FR excelente según Wieggers & Beatty?

- a) Brevedad, oscuridad, ambigüedad, abstracción, generalidad, opcionalidad y rigidez.
- b) **Completitud (*Complete*), corrección (*Correct*), factibilidad (*Feasible*), necesidad (*Necessary*), priorización (*Prioritized*), no ambigüedad (*Unambiguous*) y verificabilidad (*Verifiable*) (Wieggers & Beatty, 2013, Cap. 11, pp. 204–205). (resp. correcta)**
- c) Velocidad, escalabilidad, seguridad, usabilidad, accesibilidad, portabilidad y reusabilidad.
- d) Económico, rápido, barato, simple, popular, atractivo y comercial.

52. Según Wieggers & Beatty, ¿cuáles son los atributos del conjunto de FRs?

- a) Que los FRs deben ser idénticos entre sí para garantizar uniformidad sin excepción.
- b) **Completitud (no faltan FRs), consistencia (no se contradicen entre sí), modificabilidad (se pueden modificar manteniendo histórico) y trazabilidad (bidireccional) (Wieggers & Beatty, 2013, p. 206). (resp. correcta)**
- c) Que los FRs deben ser inmutables y nunca modificarse después de redactados.
- d) Que los FRs deben numerarse en orden estrictamente alfabético sin repetición.

Referencias

- Bourque, P., & Fairley, R. E. (Eds.). (2014). *Guide to the Software Engineering Body of Knowledge, Version 3.0 (SWEBOK Guide V3.0)* [Producido por la IEEE Computer Society. Editado por Pierre Bourque (École de technologie supérieure) y Richard E. Fairley (Software and Systems Engineering Associates)]. IEEE Computer Society. <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
- Celi-Párraga, R. J., Boné-Andrade, M. F., Mora-Olivero, A. P., & Sarmiento-Saavedra, J. C. (2023, abril). *Ingeniería del Software I: Requerimientos y Modelado del Software* (1.ª ed.) [Texto evaluado por pares doble ciego externos. Licencia Creative Commons CC BY-NC-SA 4.0. Autores afiliados a la Universidad Técnica Luis Vargas Torres de Esmeraldas, Sede Santo Domingo de los Tsáchilas (Ecuador)]. Editorial Grupo AEA.
- Contreras Bernal, C. A., & Hernández Ruiz, C. M. (2022). Ingeniería de requerimientos, camino a un proyecto exitoso. *Revista Tecnología, Investigación y Academia (TIA)*, 10(2), 90-115.
- Laplante, P. A. (2017). *Requirements Engineering for Software and Systems* (3.ª ed.). CRC Press, Auerbach Publications.
- Pressman, R. S. (2002). *Ingeniería del Software: Un Enfoque Práctico* (5.ª ed.) [Adaptación europea por Darrel Ince. Traducción de *Software Engineering: A Practitioner's Approach, 5th Edition, European Adaptation* (ISBN 0-07-709677-0), McGraw-Hill, 2001]. McGraw-Hill Interamericana de España.
- Robertson, S., & Robertson, J. (2012). *Mastering the Requirements Process: Getting Requirements Right* (3.ª ed.) [Reference book for the Volere requirements methodology]. Addison-Wesley Professional.
- Sommerville, I. (2011). *Ingeniería de Software* (9.ª ed.) [Traducción de *Software Engineering, 9th Edition*, Addison-Wesley, 2011]. Pearson Educación.
- Wieggers, K., & Beatty, J. (2013). *Software Requirements* (3.ª ed.). Microsoft Press.