Fase de Elaboración

Tema Nº4:RUP Y SUS FASES

Indicador de logro Nº4:Diseña y estructura el modelado de requerimientos con los diagramas del lenguaje de modelamiento unificado para el diseño de un software.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº1:**

CAPTURA DE REQUISITOS

El esfuerzo principal en esta disciplina es desarrollar un modelo del sistema que se va a construir. La utilización de los casos de uso es una forma adecuada de crear ese modelo. Esto es debido a que los requisitos funcionales se estructuran de forma natural mediante casos de uso. Los casos de uso proporcionan un medio intuitivo y sistemático para capturar los requisitos funcionales con un énfasis especial en el valor añadido para cada usuario individual o para cada sistema externo. Un caso de uso puede contener uno o más requisitos funcionales.

El modelo de casos de uso es construido a través de un proceso iterativo durante el cual las discusiones entre los desarrolladores del sistema y los clientes (y/o usuarios finales) llevan a una especificación de requisitos en la que todos estén de acuerdo.

Así, los propósitos de la disciplina Captura de requisitos son:

**Subtema 1.1:**

Artefactos de la captura de requisitos

El conjunto completo de artefactos de la captura de requisitos, mostrado en la figura, sirven como entrada y referencia para el análisis, diseño, implementación y pruebas del sistema.

Imagen que contiene Icono

Descripción generada automáticamente

La propuesta del curso, para una solución de mediana envergadura, es crear los artefactos proporcionados en la tabla:

|  |  |
| --- | --- |
| Artefacto | Descripción |
|  | Documento que define la opinión de los stakeholders del producto que se desarrollará, especificada en términos de necesidades y características claves de los stakeholders. Contiene un esquema de los requisitos previstos, el cual proporciona la base contractual para los requisitos técnicos más detallados. |
|  | La especificación de requisitos de software es un documento que enfoca la organización completa de los requisitos del proyecto. Comúnmente conocido como SRS por sus iniciales en inglés. Contiene la lista de requisitos funcionales y no funcionales. |
|  | Es una colección de casos de uso, de actores, de relaciones, de diagramas, y de otros paquetes, de ser necesario; es utilizado para estructurar el modelo de casos de uso dividiéndolo en piezas más pequeñas. |
|  | Es una funcionalidad específica del sistema con identidad propia, el cual define una secuencia de acciones que el sistema realiza para un actor en particular. Un caso de uso contiene uno o más requisitos funcionales. |
|  | Representa un rol (humano, hardware o software) externo al sistema con el que se establece intercambio directo de información. Puede ser asociado a uno ó más casos de uso. |
|  | Es un modelo que captura los requisitos funcionales de los usuarios a un alto nivel y establece la estructura fundamental del sistema. Es un input esencial para las actividades en análisis, diseño y pruebas. |
|  | Es un documento que contiene información de los actores identificados en el modelo de casos de uso. |
|  | Documento que contiene las características de un caso de uso. Contiene, primordialmente, una descripción del flujo de eventos que describen la interacción entre los actores y el sistema. La especificación, también, contiene otra información, tal como precondiciones, poscondiciones, requisitos especiales y prototipos. Se realiza una especificación por caso de uso. |
|  | Documento que especifica los requisitos funcionales que no son traducidos a casos de uso y los requisitos no funcionales. |

**Subtema 1.1:**

Actividades para realizar la captura de requisitos

Según RUP, la captura de requisitos comprende las siguientes actividades:

Diagrama

Descripción generada automáticamente

|  |  |
| --- | --- |
| Analizar el problema | El documento visión es el principal artefacto en el cual el análisis del problema es documentado. Para determinar el alcance inicial del proyecto, los límites del sistema deben ser definidos. El analista de sistema identifica usuarios y sistemas, representado por actores, los cuales interactúan con el El documento visión es el principal artefacto en el cual el análisis del problema es documentado. Para determinar el alcance inicial del proyecto, los límites del sistema deben ser definidos. El analista de sistema identifica usuarios y sistemas, representado por actores, los cuales interactúan con el |
| El artefacto principal es un documento refinado de la visión. También, los requisitos son discutidos y expresados en términos de casos de uso y actores. Los requisitos no funcionales, que no son representados en el modelo de casos de uso deberán ser documentados en especificaciones suplementarias.  El analista se relaciona con los stakeholders utilizando técnicas para capturar requisitos, tales como las entrevistas si se encuentra en las primeras iteraciones de esta disciplina y prototipos si se encuentra en las últimas iteraciones. Los stakeholders son un grupo de personas cuyas necesidades deben ser satisfechas por el proyecto. El papel puede ser desempeñado por cualquier persona que es (o será potencialmente) afectado por los resultados del proyecto. Por lo tanto, son fuentes de requisitos, por ejemplo, usuarios finales del sistema, gerentes, accionistas, reguladores quiénes certifican la aceptabilidad del sistema. | Entender las necesidades del stakeholder |
| Definir el sistema | En definir el sistema, se enfoca en identificar a los actores y los casos de uso completamente para obtener un modelo de casos de uso refinado y expandir los requisitos no funcionales definidos en los documentos de especificaciones suplementarias. |
| El alcance del proyecto es definido por el conjunto de requisitos definidos para éste. La clave para manejar un proyecto exitoso es administrar el alcance del proyecto cumpliendo con los recursos disponibles tales como el tiempo, la gente y el dinero. La priorización los casos de uso, desarrollado por el arquitecto de software, permite planificar el proyecto. | Administrar el alcance del sistema |
| Refinar la definición del sistema | El resultado de este flujo de trabajo del RUP es una comprensión más profunda de la funcionalidad del sistema expresada en casos de uso detallados y documentos de especificaciones suplementarias detallados. Si es necesario, una especificación de requisitos de software formal puede ser desarrollada, además de los documentos detallados de casos de uso y especificaciones suplementarias. |
| Los cambios a los requisitos impactan los modelos producidos en la disciplina de análisis y diseño, el modelo de pruebas creado en la disciplina de pruebas y el material de soporte al usuario final de la disciplina de despliegue. Las relaciones de trazabilidad son establecidas para identificar las relaciones entre los requisitos y otros artefactos. Las relaciones de trazabilidad son la clave para entender el impacto del cambio de los requisitos. | Administrar los cambios de requisitos |

**TEMA Nº2:**

REQUISITOS

Un requisito se define como una condición o capacidad a la que debe ajustarse el sistema que se construye para satisfacer un contrato, norma, especificación u otro documento formalmente impuesto. El proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un sistema es llamado ingeniería de requisitos.

La meta de la Ingeniería de requisitos (IR) es entregar una especificación de requisitos de software correcta y completa. Algunos otros conceptos de Ingeniería de requisitos son: Según Pressman “Ingeniería de Requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software”.

Por otro lado, Sommerville define que “La ingeniería de requisitos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema”. En síntesis, el proceso de ingeniería de requisitos se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requisitos para un producto de software determinado, donde es muy importante tomar en cuenta que el aporte de la IR vendrá a ayudar a determinar la viabilidad de llevar a cabo el software (si es factible llevarlo a cabo o no), pasando posteriormente por un subproceso de obtención y análisis de requisitos, su especificación formal, para finalizar con el subproceso de validación donde se verifica que los requisitos realmente definen el sistema que quiere el cliente.

**Subtema 2.1:**

Tipos de requisitos

**Subtema 2.2:**

Requisitos FURPS+

Este es un tipo de clasificación de requisitos especificado en la documentación de RUP. Se utiliza el acrónimo FURPS (por las siglas en inglés) para describir las principales categorías de requisitos:

El símbolo "+" en FURPS+ hace referencia a que se deben incluir otros requisitos, tales como:

**Subtema 2.2.1:**

Funcionales

Los requisitos funcionales deben incluir:

* Conjunto de características
* Capacidades
* Seguridad

Por ejemplo, para un Sistema de Ventas:

* R1: Mostrar descripción y precio de productos
* R2: Registrar venta de productos
* R3: Reducir stock cuando se realiza la venta
* R4: Identificar al cajero utilizando un usuario y una clave

**Subtema 2.2.2:**

Facilidad de uso

Deben incluir subcategorías tales como:

* Factores humanos
* Estéticos • Consistencia de interfaz de usuario
* Ayuda en línea o “context-sensitive”
* Asistentes (“wizards”)
* Documentación de usuario
* Materiales de capacitación/entrenamiento

Por ejemplo:

* R1: El sistema deberá proporcionar ayudas en línea para orientar en el uso de las interfaces.
* R2: Maximizar eficiencia mediante la navegación con teclado.
* R3: El sistema debe tener interfaces gráficas de administración y de operación en idioma español y en ambiente 100% Web, para permitir su utilización a través de navegadores de Internet

**Subtema 2.2.3:**

Confiabilidad

* Frecuencia de fallos
* Capacidad de recuperación a fallos
* Posibilidades de predicción del programa
* Precisión
* • Tiempo medio de fallos

Por ejemplo:

* R1: El sistema debe registrar los pagos a crédito autorizados que se hagan a las cuentas por cobrar en un plazo de 24 horas, aun cuando se produzcan fallas de energía o del equipo.
* R2: La cuenta del usuario se bloqueará por un lapso de 30 minutos luego de 4 intentos fallidos para evitar vulnerabilidades en la seguridad del sistema.

**Subtema 2.2.4:**

Rendimiento

Condiciones impuestas a requisitos funcionales, tales como:

* Velocidad
* Eficiencia
* Disponibilidad
* Tiempo de respuesta
* Tiempo de recuperación
* Uso de recursos

**Subtema 2.2.5:**

Soporte

Es la capacidad que tiene el software de ser modificado fácilmente para adecuar mejoras o cambios. Incluye:

* Adaptabilidad
* Facilidad de mantenimiento
* Compatibilidad
* Configurabilidad
* Facilidad de instalación Internacionalización

Por ejemplo:

* R1: El sistema debe operar de manera independiente del navegador que se utilice (Microsoft Internet Explorer 6.0 o superior, Netscape 6.0 o superior, Mozilla FireFox).
* R2: El sistema deberá estar orientado a que las actualizaciones sólo se hagan en el sitio del servidor.

**Subtema 2.2.6:**

Restricciones de diseño

Especifican o restringen el diseño de un sistema.

Por ejemplo:

* R1: El sistema deberá considerar, en su arquitectura, un modelo tres capas, donde se definen tres componentes lógicos de manera independiente: servicios de presentación o interfaz de usuario, servicios de funcionalidad y servicios de datos.

**Subtema 2.2.7:**

Requisitos de implementación

Especifica restricciones de codificación o de construcción del sistema:

* Estándares requeridos
* Lenguajes de implementación
* Políticas para la integridad de bases de datos
* Límite de recursos
* Ambientes de operación

Por ejemplo:

* R1: El sistema debe desarrollarse con el lenguaje JAVA V1.6.

**Subtema 2.2.8:**

Requisitos de interfaz

Especifica:

* Elemento externo con el que el sistema debe interactuar
* Restricciones o formatos, tiempos u otros factores usados en tales interacciones

Por ejemplo:

* R1: El sistema deberá proporcionar, para los diferentes reportes solicitados, salidas en documentos electrónicos (Word, Excel o Acrobat Reader).
* R2: En una visión preliminar de impresión se consideraría que todos los textos estarán relacionados con un visor de PDF’s, las estadísticas y resultados de consultas estarán relacionados con Excel 2016.

**Subtema 2.2.9:**

Requisitos físicos

Especifican características físicas que el sistema debe poseer; por ejemplo, material, forma, tamaño y peso. Pueden especificarse los requisitos de hardware.

Por ejemplo:

* R1: Para que un cliente de la aplicación pueda ejecutar procesos, en línea, considerados en el sistema el punto de acceso deberá cumplir con los siguientes requisitos mínimos.
  + Procesador 1.0 GHz.
  + Memoria 128 MB.
  + Disco duro 10 GB. o Sistema Operativo Windows XP
  + Linux. o Navegador internet Explorer 6.0 o posterior, Mozilla Firefox 2.X, Netscape Navigator 6.X o posterior con plug ins para Macromedia Flash y Java. o Conexión a Internet, mínimo 56Kbps

**TEMA Nº3:**

CAPTURA DE REQUISITOS A SOLICITUD DEL CLIENTE

Si el equipo de desarrollo tiene un conocimiento de la estructura de la organización, de las metas, de la visión y de los clientes/usuarios o si sólo se está añadiendo una nueva característica a un sistema existente, entonces, RUP no recomienda que se empiece con un modelado del negocio. En ese caso, RUP recomienda que se empiece con la captura de requisitos.

Esta actividad consiste en identificar todas las necesidades de stakeholders. Como se explicó anteriormente, el término stakeholder se utiliza para referirse a cualquier persona o grupo que está interesado por los resultados del proyecto. Obtener y comprender los requisitos de los stakeholders es difícil por varias razones:

* Los stakeholders, a menudo, no conocen lo que desean obtener del sistema informático, excepto en términos muy generales; puede resultarles difícil expresar lo que quieren que haga el sistema o pueden hacer demandas irreales debido a que no conocen el costo de sus peticiones.
* Los stakeholders expresan los requisitos distintos con sus propios términos de forma natural y con un conocimiento implícito de su trabajo.
* Diferentes requisitos de diferentes stakeholders tienen concordancia y algunos generan conflictos.

En la figura, se muestra un modelo general para obtener y analizar requisitos. Con cada vuelta del ciclo de este modelo, la comprensión de los requisitos por parte del analista mejorará. Cada equipo de desarrollo tendrá su propia versión o instancia de este modelo dependiendo de la habilidad del personal, el tipo de sistema a desarrollar y los estándares utilizados.

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

Los pasos para seguir son:

**TEMA Nº3:**

CAPTURA DE REQUISITOS A PARTIR DEL DIAGRAMA DE NEGOCIO:

Mediante la utilización del modelo del negocio como entrada, el analista emplea una técnica sistemática para crear un modelo de casos de uso tentativo. Para ello, construirá un diagrama de casos de uso tomando como punto de partida los diagramas de actividades de los casos de uso del negocio.

En primer lugar, se obtienen los requisitos funcionales a partir de las actividades candidatas a ser informatizadas. Luego, con estos requisitos, se crean los casos de uso. Las actividades que no serán soportadas por el sistema no les corresponderán un caso de uso. Los actores se identificarán a partir de los roles (trabajadores o actores del negocio) que realizan las actividades del negocio a informatizar.

Diagrama

Descripción generada automáticamente

Del modelo del negocio al modelo de casos de uso:

Diagrama

Descripción generada automáticamente

Es importante documentar el seguimiento de estos elementos: actividades a informatizar, requisitos funcionales y casos de uso; más aún, si se trata de seguir requisitos funcionales de casos de uso, el cual es un proceso complicado por el hecho de que existe una relación muchos a muchos entre ellos. Un caso de uso puede tratar muchos requisitos y un requerimiento funcionales puede estar presente en varios casos de uso diferentes.

Afortunadamente, existen herramientas de ingeniería de requisitos, como el RequisitePro y DOORS. Pero si no tiene ningún soporte de herramienta de modelado, tiene que hacer el trabajo manualmente. Un buen enfoque es crear una matriz denominada Matriz de actividades Vs. requisitos. Estas matrices se crean a menudo en hojas de cálculo (Excel). La plantilla se proporciona en la Tabla:

Tabla

Descripción generada automáticamente

Una matriz de **actividades Vs. requisitos** es una herramienta manual utilizada para obtener **requisitos funcionales** a partir de **actividades del negocio** que se van a informatizar. Una vez identificado los requisitos funcionales, se crean los casos de uso. Por otro lado, **los actores** son a partir de los **responsables de las actividades del negocio** que se tienen en la matriz.

Examinemos un ejemplo para capturar requisitos a partir de un diagrama de actividades del negocio para un sistema de créditos XYZ. En la figura, se muestra el diagrama de actividades del caso de uso de negocio “otorgamiento de Créditos”; las actividades sombreadas se identificaron para ser informatizadas.

Diagrama

Descripción generada automáticamente

La matriz de actividades Vs. requisitos para el sistema de créditos XYZ es el siguiente:

Tabla

Descripción generada automáticamente

Observaciones:

* Note que, para el requerimiento R02 y R04, se crea el mismo caso de uso, esto es, debido a que ambos requisitos dan lugar al criterio de búsqueda del CU Consultar Solicitudes por Estado.
* Por otro lado, fue necesario crear el actor Visualizador de Solicitudes para el Inspector y Jefe de Créditos, pues dos actores no pueden iniciar el mismo caso de uso.
* Por último, los requisitos R05, R06, R07 y R08 dieron lugar al caso de uso Evaluar Crédito porque serán definidos como subflujos en la Especificación del CU Evaluar Crédito.

Luego, el diagrama de casos de uso inicial resultante será el siguiente:

Diagrama

Descripción generada automáticamente

Este diagrama de casos de uso debe ser refinado. Nuevos casos de uso serán detectados al describir cada caso de uso obtenido; para ello, se utiliza el documento Especificación de caso de uso.

Por último, los nuevos casos de uso serán agregados en un diagrama denominado Diagrama General de Casos de Uso, consiguiendo de esta manera, la versión final del modelo de casos de uso.

ACTIVIDAD PROPUESTA

1. De la lista, clasifique los requisitos según las categorías de FURPS+.

**R01.** El sistema deberá garantizar que su despliegue se pueda realizar, ya sea en el lado del servidor o del cliente, sobre plataforma hardware de 32 bits o de 64 bits sin que esto afecte el rendimiento del mismo.

**R02.** El sistema debe contar con un Manual Técnico de Referencia para la Aplicación, el cual estará orientado a profesionales capacitados en aspectos técnicos del área de sistemas.

**R03.** La clave de los usuarios considerará las siguientes políticas:

* + Expirar según parametrización del sistema
  + Tener mínimo una longitud de 8 caracteres
  + Contener letras y números.
  + No puede contener espacios.
  + No pueden repetirse las 3 últimas contraseñas.

- No contendrá el nombre o apellido de la persona dueña del usuario.

**R04**. El código fuente del sistema deberá cumplir con el estándar de codificación definido en el documento “Estándares y Nomenclaturas de Código Fuente”.

**R05.** Utilizar el idioma castellano para los mensajes y textos en la Interfaz.

**R06.** El sistema será utilizado por clientes de todo el mundo. Adicionalmente, la Organización Pro-Turismo exige que, para anunciar servicios en su portal, éstos deben estar provistos en español, inglés y portugués. Estos tres idiomas deben ser soportados por el producto desarrollado.

**R07**. El sistema deberá permitir la creación, modificación, activación, desactivación y autorización de los roles de usuarios definidos.

**R08.** El sistema deberá prever contingencias que pueden afectar la prestación estable y permanente del servicio. La siguiente es la lista de las contingencias que se deben tener en cuenta y se pueden considerar críticas:

• Sobrecarga del sistema por volumen de usuarios

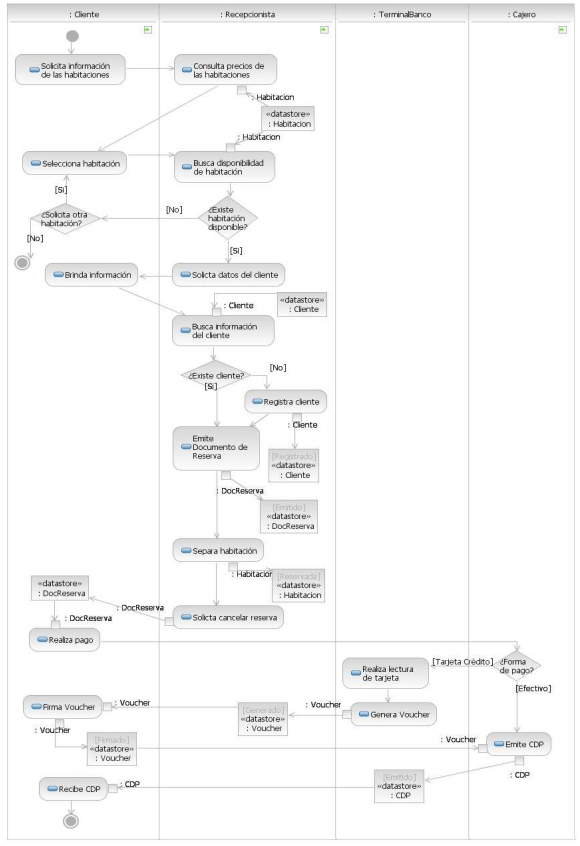
• Caída del sistema por sobrecarga de procesos

• Caída del sistema por sobrecarga de transacciones

• Caída del sistema por volumen de datos excedido en la base de datos

**R09.** El sistema deberá contar con el manual de FAQ (Frequent Asked Questions), en línea e impreso, que es un resumen con las respuestas a las preguntas más frecuentes que se hacen los usuarios. R10. El sistema debe considerar, en su arquitectura, el patrón de diseño MVC.

1. A partir del **diagrama de actividades del negocio del proceso** “Reserva de Habitaciones” para un **Sistema de Reservaciones** de un Hotel realice la matriz de **actividades Vs. requisitos** y el diagrama de casos de uso (Inicial).



RECORDAR

* En la captura de requisitos se describe las condiciones o capacidades que el sistema debe cumplir.
* La identificación de los requisitos funcionales llevará a la proyección de las funciones del sistema (casos de uso).
* La descripción de los requisitos no funcionales facilitará la construcción de la plataforma del sistema.
* Los stakeholders son un grupo de interés cuyas necesidades deben ser satisfechas por el proyecto, por ejemplo, usuarios finales del sistema, gerentes, accionistas, reguladores quiénes certifican la aceptabilidad del sistema, etc.
* El Modelo de obtención y análisis de requisitos se utiliza para capturar requisitos a partir de los stakeholders.
* La matriz de actividades Vs. requisitos es una técnica utilizada para documentar la trazabilidad de actividades Vs. requisitos funcionales y de requisitos funcionales Vs. casos de uso.
* Si desea saber más acerca de estos temas, puede consultar los siguientes libros. “Ingeniería del Software” de Ian Sommerville En el capítulo 6, encontrará información sobre requisitos del sistema; y en el capítulo 7, se profundiza el tema de la ingeniería de requisitos. “UML 2” de Jim Arlow e Ila Neustadt En el capítulo 3, encontrará información sobre la captura de requisitos.

**Desarrollar el Ejercicio propuesto:**

1. Desarrollar lo realizado basado en el problema del Club Náutico del Perú.
   1. Indicaciones
      1. i. Se efectuará en grupo de hasta cuatro integrantes
      2. Será de entrega digital, Se debe desarrollar sobre la estructura realizada en este manual.

**BIBLIOGRAFÍA**

FUENTES BIBLIOGRÁFICAS

* Charette, R. N. (1989), Software Engineering Risk Analysis and Management, McGraw-HillDntertext.
* G. Kotonya and I.(2000) Sommerville, Requirements Engineering: Processes and Techniques, John Wiley & Sons
* IEEE Computer Society, (2014), SWEBOK (Guide to the Software Engineering Body of Knowledge), Version 3.0. ISBN-10: 0-7695-5166-1
* Jacobson, Ivar; Booch, Grady; Rumbaugh, James (2000) (en Español). El Proceso Unificado de Desarrollo de Software. Pearson Addisson-Wesley.
* Piattini, Mario G. (1996), Análisis y diseño detallado de aplicaciones informáticas de gestión. 1ª ed. RAMA Editorial, Madrid, 1996
* Pressman, R. (2010). Ingeniería de Software un Enfoque práctico, Séptima Edición. ISBN 978-607-15- 0314-5.
* Sommerville, Lan. (2011) Ingeniería de software, novena edición. Pearson, México. ISBN 0137035152 | 9780137035151
* Sommerville, Lan (2005), Ingeniería de software, séptima edición, Pearson Educación, Madrid (España) ISBN: 84-7829-074-5
* [Wasserman, 1996] Wasserman, A. “Toward a Discipline of Software Engineering”. IEEE Software, 13(6):23-31. November/December 1996
* •PRESSMAN, Roger. “Ingeniería del Software. Un Enfoque Práctico”. 5ª ed. México: McGraw-Hill Latinoamericana, 2002. ISBN: 8448132149.
* •BOOCH, Grady et tal. “El Proceso Unificado de Desarrollo de Software”. 1ª ed. España: Editorial Addison-Wesley.
* •LARMAN, Craig. UML y Patrones – Introducción al Análisis y Diseño Orientado a Objetos. 1ª ed. España: Pearson Educación.
* •SENN, James. “Análisis y Diseño de Sistemas de Información”. México: Mc Graw Hill. ISBN: 9684229917.
* •BRAUDE, J. “Ingeniería de Software: Una Perspectiva Orientada a Objetos” Ra-ma. ISBN: 8478975756. ISBN-13: 9788478975754.
* •SOMMERVILLE, Ian “Ingeniería de Software: Un enfoque practico”, Eddison Wesley, México, 692 p.
* •FONTELA, CARLOS (2011), UML Modelado de Software para profesionales. Editorial Alfa Omega.
* CRAIG LARMAN (2000), “UML y Patrones”, Editorial Prentice Hall.
* JACOBSON, BOOCH, RUMBAUGH (2000), “El Lenguaje Unificado de Modelado”. Editorial Addison-Wesley.
* KENDALL y KENDALL (2010), “Análisis y Diseño Sistemas”, 8va Ed. Editorial Prentice Hall Hispanoamericana S.A.
* LAUDON, Kenneth y LAUDON, Jane (2002),”Administración de los Sistemas de Información”. Editorial Prentice Hall Hispanoamericana S.A.
* JACOBSON, BOOCH, RUMBAUGH (2000), “El proceso unificado de desarrollo de Software”. Editorial Addison-Wesley,
* Martin FOWLER - Kendall SCOTT (1999), “UML Gota a Gota”. Editorial Pearson.
* PRESMAN, Roger S (2005), “Ingeniería de Software”, 5ta Ed.; Mc Graw Hill.

FUENTES DIGITALES

* Barzana, A & Menéndez, R. (2005), Gestión de Riesgos en Ingeniería de Software. Consultado el 29 de febrero de 2016 de http://www.wikilearning.com/curso\_gratis/gestion\_de\_riesgos\_en\_ingenieria\_del\_softwareintroduccion/3620-1
* IEEE Computer Society, (2004), Software Engineering Body of Knowledge, Consultado el 09 de noviembre de 2015 de: <http://www.swebok.org>
* IEEE Computer Society, (2014), SWEBOK (Guide to the Software Engineering Body of Knowledge), Version 3.0. ISBN-10: 0-7695-5166-1, consultado el 09 de noviembre de 2015 de: http://www.computer.org/web/swebok/v3-guide
* ISO/IEC 25000, SQuaRE, (2014). Consultado el 09 de noviembre de 2015 de: System and Software Quality Requirements and Evaluation.
* Real Academia Española, Asociación de Academias de la Lengua Española. Diccionario de la lengua española, 23.ª ed., Edición del Tricentenario, [en línea]. Madrid: Espasa, 2014.
* Diccionario WordReference Copyright / derecho de autor © 2016 WordReference.com. http://www.wordreference.com/definicion/dise%C3%B1o

FUENTES BASES DE DATOS ESPECIALIZADAS

* <http://scholar.google.es/>
* <http://dialnet.unirioja.es/>
* <http://www2.ebsco.com/es-es/Pages/index.aspx>
* <https://cgrw01.cgr.go.cr/rup/RUP.es/SmallProjects/core.base_rup/workproducts/rup_design_model_2830034D.html>
* <http://www.redalyc.org/>