

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

**METODOLOGÍA DE VERIFICACIÓN Y VALIDACIÓN DE
ADQUISICIÓN EN LA ETAPA DE ANÁLISIS DE SISTEMAS DE
INFORMACIÓN DESARROLLADOS A LA MEDIDA PARA LA
ADQUISICIÓN EN PEQUEÑOS CONTEXTOS**

Anexos

Daniel Alonso Farfán Molina

**ASESOR: Abraham Dávila
CO-ASESOR: Karin Meléndez**

Lima, Agosto del 2015

Tabla de contenido

1. ANEXOS	3
1.1. REGISTRO DE ACTIVIDADES	3
1.2. LISTA DE TÉCNICAS DE IDENTIFICACIÓN Y OBTENCIÓN DE REQUISITOS	3
1.3. LISTA DE MODELOS PARA DEFINIR REQUISITOS DE SOFTWARE	7
1.4. PLANTILLA DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE	15
1.5. LISTA DE COMPROBACIÓN DE LA ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE	19
1.6. DOCUMENTO DE ACEPTACIÓN PARA LA ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE	22
1.7. LISTA DE COMPROBACIÓN DEL PLAN DE ACEPTACIÓN	23
1.8. GUÍA DE REALIZACIÓN DE REVISIÓN ENTRE PARES	24
1.9. INVENTARIO DE ACTIVIDADES DE VERIFICACIÓN Y VALIDACIÓN (V & V)	25
1.10. GUÍA DE CONTROL DE CAMBIOS	30
1.11. INVENTARIO DE PRUEBAS	32
1.12. GUÍA DE TRAZABILIDAD DE REQUISITOS	38
1.13. LISTA DE COMPROBACIÓN DE ENTREGABLES DEL PROVEEDOR	41
1.14. PLANTILLA PARA VALIDACIÓN POR JUICIO EXPERTO	42
1.15. CONSTANCIA DE VALIDACIÓN	48
2. REFERENCIAS BIBLIOGRÁFICAS	49

1. Anexos

1.1. Registro de actividades

El presente artefacto es utilizado con motivo de dejar constancia de la realización o falta de realización de las diversas actividades que conforman la metodología. Los estados sugeridos a utilizar en el registro de actividades son: No realizado – Realizado – Inconforme. A continuación se presenta el formato.

Registro de actividades

Nombre del proyecto:

Fecha de actualización:

Nr o	Actividades	Esta do	Rol Responsable	Fecha de realización	Observacion es
1	Revisar especificación de requisitos de software.				
2	Generar plan de aceptación.				
3	Realizar análisis de trazabilidad de requerimientos.				
4	Revisar informes.				
5	Contrastar resultados.				

1.2. Lista de técnicas de identificación y obtención de requisitos

Con motivo de cumplir eficientemente los objetivos en la reunión propuesta de *Stakeholders* del lado del Adquiriente, en la que se identificaran y obtendrán requisitos, se propone el presente conjunto de técnicas para recabar requisitos.

Cabe resaltar que, tal como establece Sommerville (2005), los requisitos son cambiantes a lo largo del proyecto debido a diversos motivos, tales como una mejor comprensión de objetivos del software, cambios en la organización adquiriente o cambios en el hardware, software o entorno organizacional. Como consecuencia, la metodología establece como actividad posterior una revisión del Adquiriente y Proveedor en la que se incorpore la visión y objetivos de ambas partes.

La realización de la “Lista de técnicas de identificación y obtención de requisitos” está basada en técnicas obtenidas de Sommerville (2005), Pressman (2010) y Kendall & Kendall (2011)

Lista de técnicas de identificación y obtención de requisitos

Nombre de proyecto:

Nombre del responsable(es):

Rol(es) responsable(es):

Fecha de selección:

Nro	Técnica	Descripción	Proceso/Recomendaciones	Selección	Observación
1	Punto de vista	Es un enfoque que utiliza puntos de vista para organizar la obtención de requerimientos y los requerimientos. Se busca identificar y luego organizar los puntos de vista para a partir de ellos identificar los requisitos del sistema. Los puntos pueden ser interactuadores, indirectos y de dominio dependiendo de si interactúan con el sistema de manera directa, indirecta o si son características o restricciones propias del dominio	<ol style="list-style-type: none"> 1) Identificar puntos de vista 2) Organizar y estructurar los puntos de vista por jerarquía. 3) Identificar requisitos a partir de puntos de vista. 	<input type="checkbox"/>	
2	Entrevistas	Esta técnica consiste en aplicar un conjunto de preguntas a los stakeholders sobre el sistema a desarrollar y el posible sistema que utilizan, obteniendo los requisitos a través de las respuestas a dichas preguntas. Se aplican 2 clases de entrevistas, cerradas en donde los stakeholders responden a un conjunto predefinido de preguntas y abiertas donde no hay programa predefinido. Normalmente es usada como una técnica complementaria.	<ol style="list-style-type: none"> 1) Preparar un conjunto de preguntas para iniciar la entrevista, ya que las entrevistas completamente abiertas no suelen salir bien. 2) Evitar prejuicios e ideas preconcebidas sobre los requisitos. 3) Estar dispuestos a escuchar a los stakeholders. 	<input type="checkbox"/>	

Nro	Técnica	Descripción	Proceso/Recomendaciones	Selección	Observación
			4) Hablar en un contexto definido en lugar de en términos generales.		
3	Escenarios	Se propone un escenario de cómo se podría interactuar con un sistema software, con la finalidad de comprenderlo y discutir sobre el y de esta manera obtener información que facilite la identificación y obtención de los requisitos. Son considerados como descripciones de interacción con el sistema. Los casos de uso son una variante estructurada de los escenarios.	1) Incluir una descripción de lo que esperan el sistema y los usuarios cuando el escenario comienza. 2) Incluir una descripción del flujo formal de eventos. 3) Incluir una descripción de lo que puede ir mal y cómo manejarlo. 4) Incluir información de otras actividades que se podrían llevar a cabo al mismo tiempo. 5) Incluir una descripción del estado del sistema cuando el escenario termina.	<input type="checkbox"/>	
4	Etnografía	La etnografía es una técnica de observación que se puede utilizar para entender requisitos sociales y organizacionales, consiste en observar el trabajo diario y anotar las tareas reales de los involucrados. Ayuda a descubrir requisitos implícitos que reflejan los procesos reales más que los formales en los que la gente está involucrada. Se centran en el usuario final y es usada como técnica complementaria.	1) Es recomendable su combinación con prototipos para suministrarles información. 2) Es aplicable cuando se desea descubrir requisitos que se derivan de la forma en la que la gente realmente trabaja en lugar de lo que dicta el proceso formal. 3) Es aplicable cuando se desea descubrir requisitos que se	<input type="checkbox"/>	

Nro	Técnica	Descripción	Proceso/Recomendaciones	Selección	Observación
			derivan de la cooperación y conocimiento de las actividades de la gente.		
5	Despliegue de la función de calidad	Es una técnica que traduce las necesidades del cliente en requisitos a partir de la maximización de la satisfacción del cliente en el proceso de ingeniería de software. Se centra en entender lo que resulta valioso para el cliente. Clasifica los requisitos en normales, esperados y emocionantes dependiendo si los requisitos son acordados con el cliente, son implícitos o superan las expectativas del cliente.	1) Utilizar técnicas de entrevistas, observación, encuestas y/o datos históricos para recabar requisitos. 2) Realizar una tabla de requisitos a revisar con el cliente y otros participantes (tabla de la voz del cliente). 3) Utilizar diagramas, matrices y métodos de evaluación para extraer requisitos esperados y tratar de percibir requisitos emocionantes.	<input type="checkbox"/>	
6	Diseño de aplicación conjunta (JAD)	Es una técnica alternativa a las entrevistas desarrolladas por IBM, consiste en realizar un análisis de requisitos y diseño de interfaz en forma conjunta con los usuarios en un ambiente de grupo. Su aplicación permite mejorar la calidad de requisitos de información y mejorar la identificación del usuario con el nuevo sistema de información.	1) Es recomendable usar JAD como solución innovadora a un problema común cuando los usuarios deseen algo nuevo. 2) Es recomendable usar JAD cuando la cultura organizacional promueva solución de problemas en grupo sin distinguir jerarquía. 3) Es recomendable usar JAD cuando el flujo de trabajo permita disponibilidad de personal en conjunto. 4) Es recomendable usar JAD	<input type="checkbox"/>	

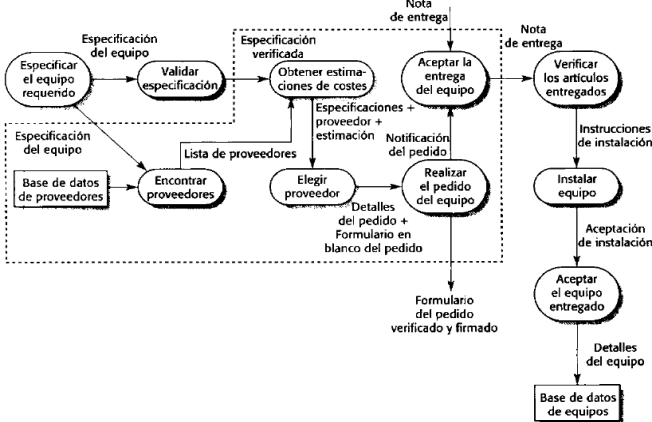
Nro	Técnica	Descripción	Proceso/Recomendaciones	Selección	Observación
			cuando se pronostica mayor cantidad de ideas generadas en conjunto que individualmente.		
7	Cuestionarios	Es una técnica complementaria de recopilación de información que permite cuantificar lo que se encontró con otras técnicas. También Permite estudiar posturas, comportamiento y características de personas clave con respecto al sistema propuesto.	1) Planificar uso de cuestionarios (participantes esparcidos, cantidad de participantes, estudio de opinión). 2) Redactar preguntas (preguntas abiertas, cerradas, terminología, uso de escalas). 3) Diseño de cuestionarios (incluir espacios para respuestas, mantener un estilo consistente, orden de las preguntas) 4) Administrar cuestionarios (convocatoria, entrega personal, anonimidad, fecha de entrega, entrega vía correo)	<input type="checkbox"/>	

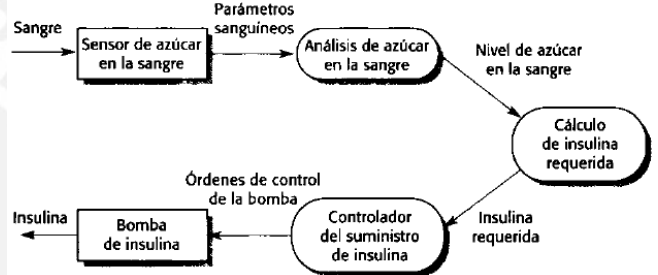
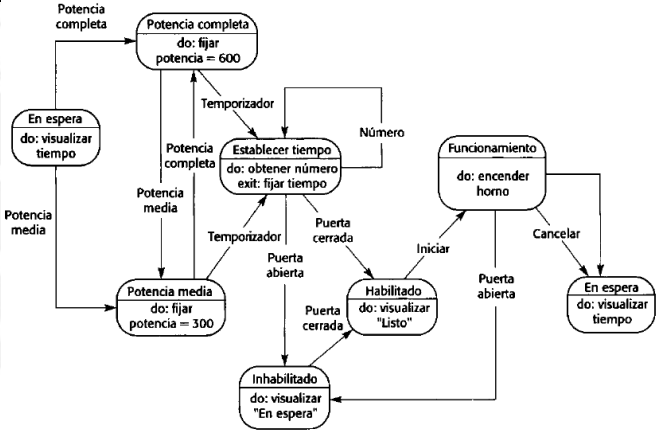
1.3. Lista de modelos para definir requisitos de software

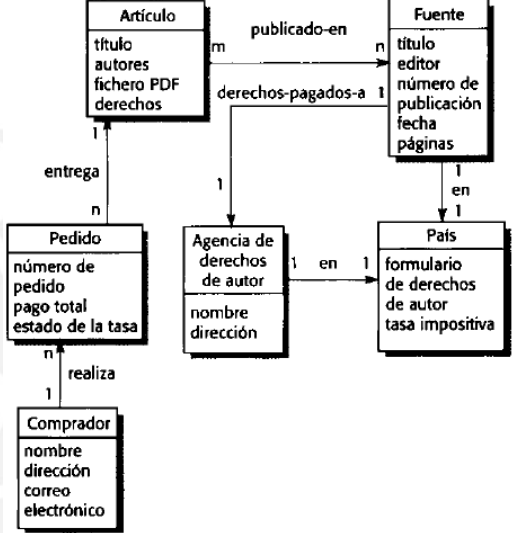
El presente artefacto ha sido elaborado con la finalidad de presentar un conjunto de tipos de modelos que permiten expresar los requisitos de sistema de forma más detallada y técnica. Además, dichos modelos son representaciones gráficas, motivo por el cual muchas veces son más comprensibles que las descripciones detalladas en lenguaje natural de los requisitos del sistema (Sommerville, 2005) y por medio de ellos se establece una base para generar un diseño de software en base a requisitos que puedan ser validados una vez contruido el software (Pressman, 2010).

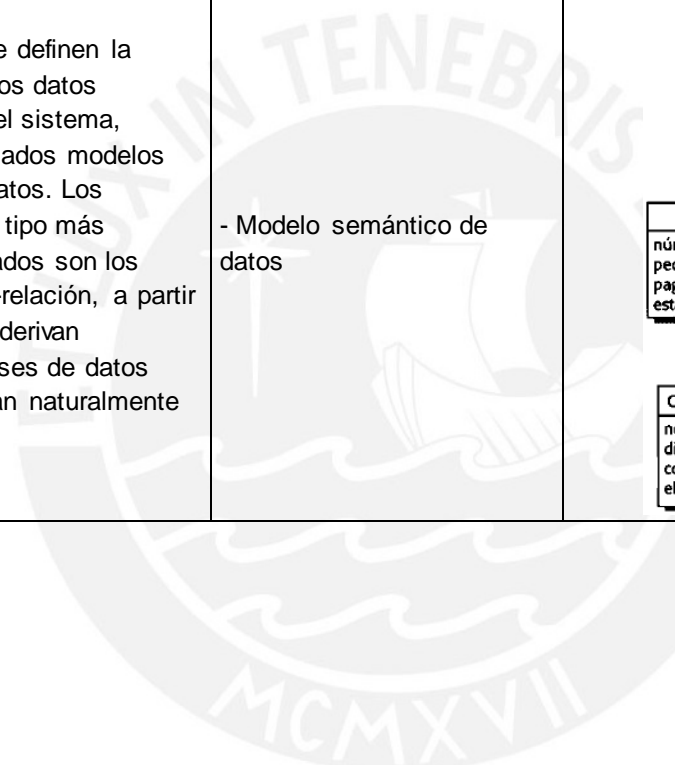
La “Lista de modelos para definir requisitos de software” fue elaborada a partir de modelos extraídos de Sommerville (2005) y Pressman (2010).

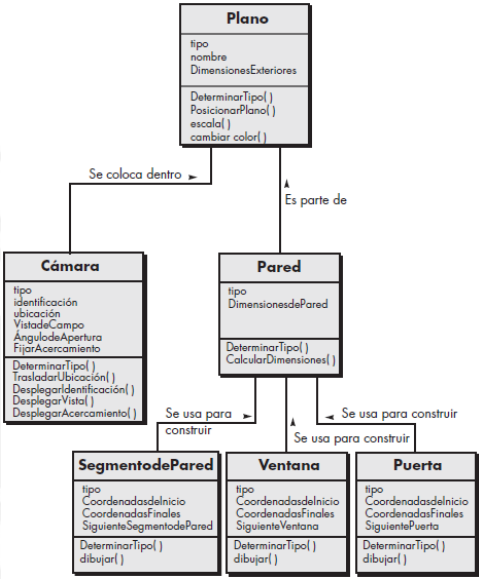
Lista de modelos para definir requisitos de software



Nro	Tipo Modelo	Descripción	Aplicable a través de	Diagrama	Selección
1	Modelo de contexto	Son modelos en los que se define los límites del sistema, es decir donde se distingue el entorno del propio sistema, además también se incluye la definición de las dependencias con el entorno. Son expresados mediante modelos arquitectónicos simples o a través de modelos de flujos de datos en los que se delimita el sistema.	<ul style="list-style-type: none"> - Modelo arquitectónico - Modelo de flujo de datos con delimitación del sistema 		<input type="checkbox"/>
2	Modelo de comportamiento	Son modelos que se utilizan para describir el comportamiento del sistema en su totalidad. Existen 2 tipos de modelos de comportamiento, los modelos de flujo de datos y los modelos de máquinas de estado, los primeros modelan el procesamiento de los datos en el sistema, mientras que los otros	<ul style="list-style-type: none"> - Modelo de flujo de datos - Modelo de máquinas de estado 	-	<input type="checkbox"/>

Nro	Tipo Modelo	Descripción	Aplicable a través de	Diagrama	Selección
2.1	Modelo de flujos de datos	<p>modelan la reacción de los sistemas a los eventos.</p> <p>Los modelos de flujo de dato se usan para mostrar como los datos son procesados por un sistema a través de una secuencia de pasos. Mediante su notación se representa el procesamiento funcional, almacenes de datos y el flujo de datos entre funciones.</p>	<p>- Modelo de flujo de datos</p>		<input type="checkbox"/>
2.2	Modelo de máquinas de estado	<p>Los modelos de máquinas de estado describen como responde un sistema a eventos internos o externos. Se muestra los estados del sistema y los eventos que provocan las transiciones de un estado a otro. Comúnmente se usan para modelar sistemas en tiempo real, ya que estos responden a eventos procedentes del entorno del sistema.</p>	<p>- Modelo de máquinas de estado</p>		<input type="checkbox"/>

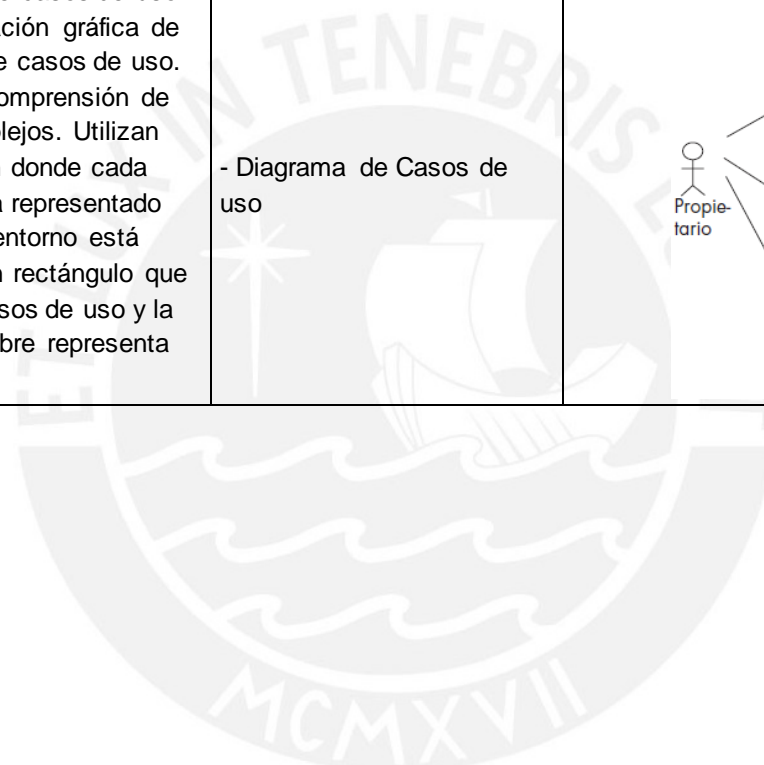
Nro	Tipo Modelo	Descripción	Aplicable a través de	Diagrama	Selección
3	Modelo de datos	<p>Son modelos que definen la forma lógica de los datos procesados por el sistema, también son llamados modelos semánticos de datos. Los modelos de este tipo más ampliamente usados son los modelos entidad-relación, a partir de los cuales se derivan esquemas de bases de datos que se encuentran naturalmente normalizados.</p>	<p>- Modelo semántico de datos</p>	 <pre> erDiagram Artículo --} Fuente : publicado-en Artículo --} Fuente : derechos-pagados-a Pedido --} Artículo : entrega Comprador --} Pedido : realiza Agencia --} País : en País --} Fuente : en </pre>	<div style="border: 1px solid black; width: 30px; height: 30px; margin: auto;"></div>

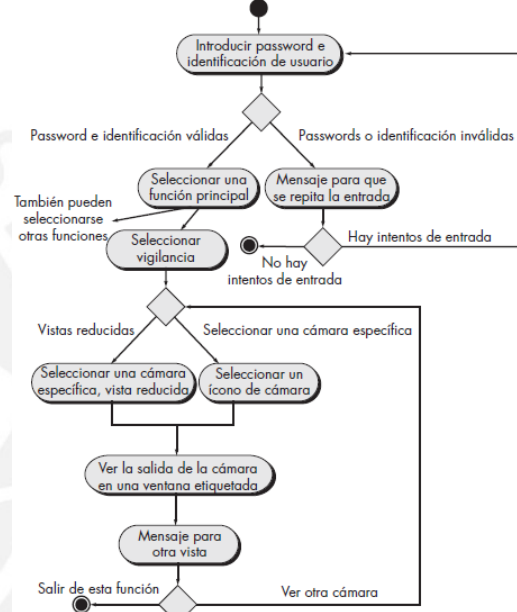


Nro	Tipo Modelo	Descripción	Aplicable a través de	Diagrama	Selección
4	Modelo basados en clases	<p>El modelo basado en clases representa los objetos que manipulará el sistema, las operaciones con los objetos de dichas clases (métodos), relaciones entre objetos y colaboraciones entre clases. Se Tiene como elementos a las clases, objetos, atributos y operaciones. Se aplican varios elementos de la notación UML para definir jerarquías, asociaciones, relaciones, agregaciones y dependencias entre clases.</p>	<p>- Modelo de clases de análisis</p>	 <pre> classDiagram class Plano { tipo nombre DimensionesExteriores DeterminarTipo() PosicionarPlano() escala() cambiar color() } class Cámara { tipo identificación ubicación Vista de Campo Angulo de Apertura Fijar Acercamiento DeterminarTipo() Trasladar Ubicación() Desplegar Vista() Desplegar Acercamiento() } class Pared { tipo Dimensiones de Pared DeterminarTipo() Calcular Dimensiones() } class SegmentodePared { tipo Coordenadas de Inicio Coordenadas Finales Siguiendo Segmento de Pared DeterminarTipo() dibujar() } class Ventana { tipo Coordenadas de Inicio Coordenadas Finales Siguiendo Ventana DeterminarTipo() dibujar() } class Puerta { tipo Coordenadas de Inicio Coordenadas Finales Siguiendo Puerta DeterminarTipo() dibujar() } Plano < -- Pared Cámara --> Pared : Se coloca dentro Pared < -- SegmentodePared Pared < -- Ventana Pared < -- Puerta Pared --> SegmentodePared : Se usa para construir Pared --> Ventana : Se usa para construir Pared --> Puerta : Se usa para construir </pre>	<input type="checkbox"/>
5	Modelado basado en escenarios	<p>El modelo basado en escenarios se aplica cuando se desea entender como interactúan los usuarios finales con el sistema. Para su modelado se usa comúnmente UML. Se crean escenarios mediante casos de uso y diagramas de actividades.</p>	<p>- Casos de Uso - Diagrama de Casos de uso - Diagrama de actividades</p>	<p>-</p>	<input type="checkbox"/>

Nro	Tipo Modelo	Descripción	Aplicable a través de	Diagrama	Selección
5.1	Casos de uso	<p>Los casos de uso son un tipo de modelo, basado en texto, que describe escenarios específicos desde el punto de vista de un actor. Capta las interacciones que hay entre los productores y consumidores de información y el sistema en si. Para redactar casos de uso son necesarios los participantes, el alcance del problema, los objetivos, tener claro los requisitos funcionales y conocer las actividades realizadas por un actor específico.</p>	<p>- Casos de Uso</p>	<p> <i>Formato de caso de uso para vigilancia</i> Caso de uso: Acceder a la vigilancia con cámaras por internet, mostrar vistas de cámaras (AVC-MVC).</p> <p>Iteración: 2, última modificación: 14 de enero por V. Ramon.</p> <p>Actor principal: Propietario.</p> <p>Objetivo en contexto: Ver la salida de las cámaras colocadas en la casa desde cualquier ubicación remota por medio de internet.</p> <p>Precondiciones: El sistema debe estar configurado por completo; deben obtenerse las identificaciones y claves de usuario apropiadas.</p> <p>Disparador: El propietario decide ver dentro de la casa mientras está fuera.</p> <p>Escenario:</p> <ol style="list-style-type: none"> 1. El propietario se registra en el sitio web <i>Productos CasaSegura</i>. 2. El propietario introduce su identificación de usuario. 3. El propietario proporciona dos claves (cada una con longitud de al menos ocho caracteres). 4. El sistema despliega todos los botones de las funciones principales. 5. El propietario selecciona "vigilancia" entre los botones de funciones principales. 6. El propietario escoge "seleccionar una cámara". 7. El sistema muestra el plano de la casa. 8. El propietario selecciona un icono de cámara en el plano de la casa. 9. El propietario pulsa el botón "vista". 10. El sistema muestra la ventana de la vista de la cámara identificada. 11. El sistema presenta una salida de video dentro de la ventana de vistas, con una velocidad de un cuadro por segundo. <p>Excepciones:</p> <ol style="list-style-type: none"> 1. La identificación o las claves son incorrectas o no se reconocen (véase el caso de uso Validar identificación y claves). 2. La función de vigilancia no está configurada para este sistema (el sistema muestra el mensaje de error apropiado; véase el caso de uso Configurar la función de vigilancia). 3. El propietario selecciona "Mirar vistas reducidas de todas las cámaras" (véase el caso de uso Mirar vistas reducidas de todas las cámaras). 4. No se dispone o no se ha configurado el plano de la casa (se muestra el mensaje de error apropiado y véase el caso de uso Configurar plano de la casa). 5. Se encuentra una condición de alarma (véase el caso de uso Condición de alarma encontrada). <p>Prioridad: Moderada, por implementarse después de las funciones básicas.</p> <p>Cuando estará disponible: En el tercer incremento.</p> <p>Frecuencia de uso: Frecuencia moderada.</p> <p>Canal al actor: A través de un navegador con base en PC y conexión a internet.</p> <p>Actores secundarios: Administrador del sistema, cámaras.</p> <p>Canales a los actores secundarios:</p> <ol style="list-style-type: none"> 1. Administrador del sistema: sistema basado en PC. 2. Cámaras: conectividad inalámbrica. <p>Asuntos pendientes:</p> <ol style="list-style-type: none"> 1. ¿Qué mecanismos protegen el uso no autorizado de esta capacidad por parte de los empleados de <i>Productos CasaSegura</i>? 2. Es suficiente la seguridad? El acceso ilegal a esta característica representaría una invasión grave de la privacidad. 3. ¿Será aceptable la respuesta del sistema por internet dado el ancho de banda que requieren las vistas de las cámaras? 4. ¿Desarrollaremos una capacidad que provea el video a una velocidad más alta en cuadros por segundo cuando se disponga de conexiones con un ancho de banda mayor? 	

Nro	Tipo Modelo	Descripción	Aplicable a través de	Diagrama	Selección
5.2	Diagrama de Casos de uso	Los diagramas de casos de uso son la representación gráfica de los escenarios de casos de uso. Sirven para la comprensión de escenarios complejos. Utilizan notación UML en donde cada caso de uso está representado por un ovalo, el entorno está delimitado por un rectángulo que encierra a los casos de uso y la figura de un hombre representa al actor.	- Diagrama de Casos de uso		<input type="checkbox"/>



Nro	Tipo Modelo	Descripción	Aplicable a través de	Diagrama	Selección
5.3	Diagrama de actividades	La Función del diagrama de actividades es enriquecer un caso de uso al proporcionar una representación gráfica del flujo de interacción en un escenario específico. Se usan cuadrados redondeados para representar funciones del sistema, flechas para los flujos entre funciones, rombos para las decisiones y líneas continuas para actividades en paralelo.	- Diagrama de actividades	 <p>The diagram illustrates the user authentication and camera selection process. It starts with a start node leading to the activity 'Introducir password e identificación de usuario'. A decision diamond follows, branching into 'Password e identificación válidas' and 'Passwords o identificación inválidas'. The 'válidas' path leads to 'Seleccionar una función principal', which then leads to 'Seleccionar vigilancia'. The 'inválidas' path leads to 'Mensaje para que se repita la entrada', which then leads to a decision diamond 'Hay intentos de entrada'. This diamond branches into 'No hay intentos de entrada' (leading to the start of 'Seleccionar vigilancia') and 'Hay intentos de entrada' (leading to a loop back to 'Introducir password e identificación de usuario'). From 'Seleccionar vigilancia', a decision diamond branches into 'Vistas reducidas' and 'Seleccionar una cámara específica'. 'Vistas reducidas' leads to 'Seleccionar una cámara específica, vista reducida', and 'Seleccionar una cámara específica' leads to 'Seleccionar un ícono de cámara'. Both paths merge and lead to 'Ver la salida de la cámara en una ventana etiquetada', followed by 'Mensaje para otra vista'. A final decision diamond branches into 'Salir de esta función' (leading to an end node) and 'Ver otra cámara' (leading to a loop back to 'Seleccionar una cámara específica').</p>	<input data-bbox="1948 678 2004 726" type="checkbox"/>

1.4. Plantilla de especificación de requisitos de software

La incorporación de la especificación de requisitos de software (ERS) tiene la función de cubrir la necesidad de describir con mayor detalle los requisitos y todos los aspectos del software a desarrollar por la empresa proveedora. La especificación de requisitos de software (ERS) está destinada a ser usada por diferentes usuarios, por lo que en palabras de Sommerville (2005) "...el documento de requerimientos tiene que presentar un equilibrio entre la comunicación de los requisitos a los clientes, la definición de los requisitos en detalle para los desarrolladores y probadores, y la inclusión de información sobre la posible evolución del sistema."

El presente artefacto basado en IEEE 830 (1998), Wiegers, K. (2003) y Sommerville (2005) pretende ser un modelo genérico de especificación de requisitos de software (ERS) a partir del cual se pueda adaptar un documento de ERS que se ajuste a la organización y a las dimensiones de proyectos orientados a pequeños contextos.

Plantilla de especificación de requisitos de software

- Estructura propuesta de la ERS

- 1) Introducción

- 1.1) Propósito
- 1.2) Alcance
- 1.3) Tabla de contenido e índice
- 1.4) Descripción del documento

- 2) Descripción General

- 2.1) Perspectiva del producto
- 2.2) Funciones del producto
- 2.3) Restricciones
- 2.4) Arquitectura del sistema

- 3) Requisitos

- 3.1) Requisitos de interfaz
- 3.2) Requisitos funcionales
- 3.3) Requisitos no funcionales

- 4) Modelos de análisis

- 5) Apéndices

- Apéndice A: Glosario de términos y abreviaciones
- Apéndice B: Evolución del sistema
- Apéndice C: Otra información relevante

- **Descripción de las secciones**

- 1) Introducción

Mediante esta sección se brindará una visión general del documento de ERS a través de las subsecciones que lo componen.

- 1.1) Propósito

En esta subsección se identifica y delimita el producto cuyos requisitos se especificaran en las siguientes subsecciones.

- 1.2) Alcance

A través de esta subsección se delimitara el alcance que tendrá el producto software por medio de una descripción de lo que el producto software realizara, sus beneficios y objetivos. Se guardara con otros documentos de especificación de ser el caso.

- 1.3) Tabla de contenido e índice

Esta subsección muestra la composición general del contenido y facilitar su uso.

- 1.4) Descripción del documento

Mediante esta subsección se explica la organización del documento.

- 2) Descripción General

A través de las subsecciones que la componen, esta sección de la ERS se encarga de describir los factores generales que afectan el producto y sus requisitos.

- 2.1) Perspectiva del producto

En esta subsección se establece si el producto es independiente o si es un componente de un producto software más grande, de ser este el caso se describen los requisitos del producto software mayor y las interfaces que establecen la relación entre el componente y el software.

- 2.2) Funciones del producto

Esta subsección se encarga de presentar un resumen de las principales funciones que realizará el software de forma comprensible y/o gráfica para los posibles lectores del documento.

- 2.3) Restricciones

La intención de esta subsección de la ERS es brindar una relación y descripción de los temas que limiten de alguna manera el desarrollo del producto software. Entre las

posibles restricciones se toman en cuenta políticas, limitaciones de hardware, protocolos de comunicaciones, criticidad de la aplicación, consideraciones de seguridad, entre otros.

2.4) Arquitectura del sistema

Mediante esta subsección se muestran la distribución de funciones en los módulos del software, así como los componentes que forman el software. Es recomendable incluir cualquier otro concepto que contribuya a la visión general de la arquitectura.

3) Requisitos

El objetivo de esta sección de la ERS es detallar de forma escrita los requisitos de software. Es recomendable que por cada requisito se asocien sus entradas al sistema, las salidas y las funciones relacionadas. Los requisitos serán clasificados para su especificación de acuerdo a las siguientes subsecciones.

3.1) Requisitos de interfaz

En esta subsección se definen los requisitos asociados a las interfaces del sistema. Estos requisitos están asociados a estándares de interfaz de usuario a seguirse, restricciones de diseño, estándares de botones, mensajes, entre otros.

3.2) Requisitos funcionales

En esta subsección se detallan los requisitos asociados a lo que debe hacer el sistema al aceptar determinadas entradas y producir ciertas salidas. Esto incluye verificación de entradas, secuencia de operaciones, respuesta a flujos alternativos de ejecución, relaciones de entradas a salidas, entre otros.

3.3) Requisitos no funcionales

En esta subsección se incluyen el resto de requisitos no incluidos en las subsecciones anteriores, tales como requisitos de desempeño, requisitos relacionados a base de datos, requisitos de restricciones de diseño, requisitos de adecuación a estándares, requisitos relacionados a atributos del sistema, requisitos de seguridad, entre otros.

4) Modelos de análisis

En esta sección se exponen los modelos del sistema que expresan de forma más detallada y técnica los requisitos, componentes y procesos del sistema. En esta sección se pueden incluir cualquiera de los modelos definidos en el artefacto llamado "Lista de modelos para definir requisitos de software".

5) Apéndices

En esta sección se proporciona información relacionada a la ERS y/o al producto software.

Apéndice A: Glosario de términos y abreviaciones

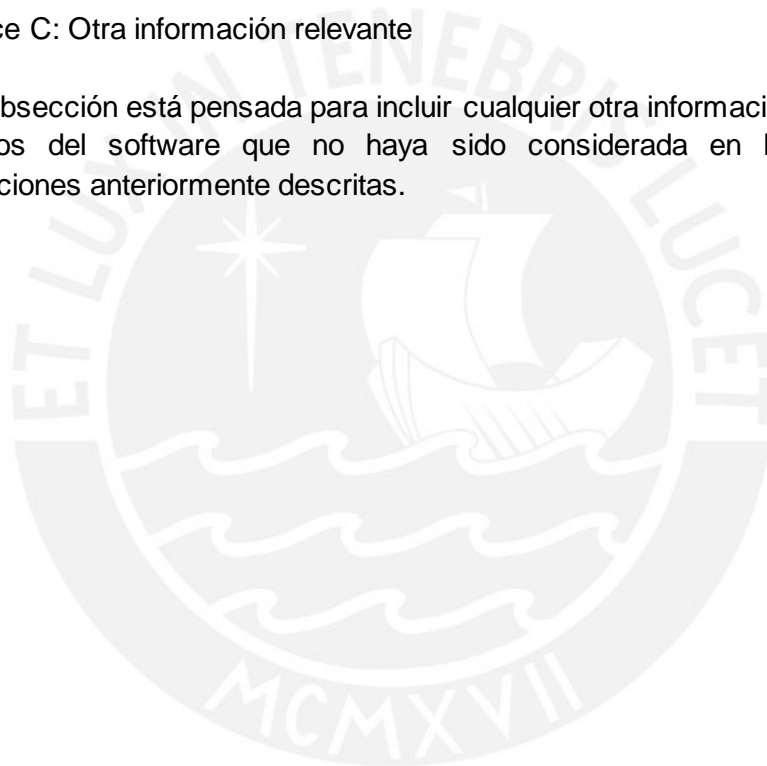
En esta subsección se definen los términos y abreviaciones utilizados a lo largo de la ERS.

Apéndice B: Evolución del sistema

En esta subsección se deben establecer los cambios previstos en el software debido a hardware, usuarios, suposiciones, entre otros.

Apéndice C: Otra información relevante

Esta subsección está pensada para incluir cualquier otra información asociada a los requisitos del software que no haya sido considerada en las secciones y subsecciones anteriormente descritas.



1.5. Lista de comprobación de la especificación de requisitos de software

Con la finalidad de asegurar un conjunto de características que reflejen las necesidades del adquiriente y la visión del proveedor se establece un listado de criterios que debe cumplir la especificación de requerimientos de software y/o los requisitos que la componen, dicho listado de criterios y características a cumplir es la “Lista de comprobación de la especificación de requisitos de software”.

La “Lista de comprobación de la especificación de requisitos de software” ha sido creada a partir de características, criterios y recomendaciones tomadas de diversos autores en temas de elicitación de requisitos e ingeniería de requisitos, entre ellos IEEE 830 (1998), Brian Berenbach et. al (2009), Elizabeth Hull et. al (2005) y Vega (2012).

Lista de comprobación de la especificación de requisitos de software

Nombre de proyecto:

Nombre del revisor:

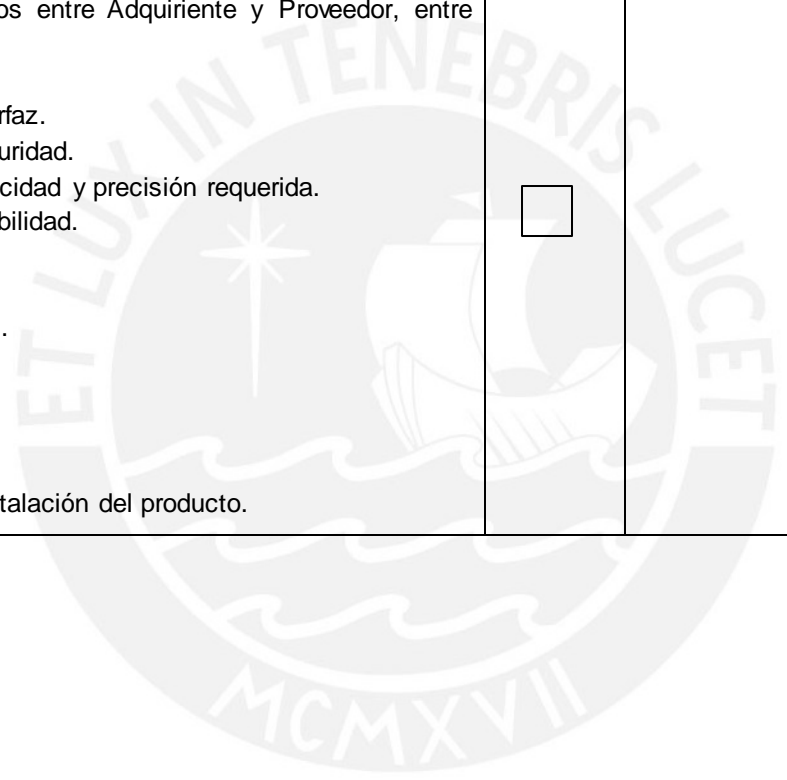
Rol del revisor:

Fecha de revisión:

Nro	Criterio	Revisión	Observaciones
1	Cada requisito incluido en la ERS es no ambiguo, es decir solo tienen una posible interpretación clara y concisa.	<input type="checkbox"/>	
2	La especificación de requisitos de software (ERS) es modificable; es decir, no existen redundancias y realizar un cambio en un requisito no requiere cambios en cascada.	<input type="checkbox"/>	

Nro	Criterio	Revisión	Observaciones
3	Cada requisito es trazable; es decir, es posible identificar su fuente y su relación con otras características del producto.	<input type="checkbox"/>	
4	La especificación de requisitos de software (ERS) es concisa; es decir, si se remueve un requerimiento, cambia la definición del producto o sistema.	<input type="checkbox"/>	
5	Cada requisito es coherente con los objetivos generales del sistema o producto.	<input type="checkbox"/>	
6	La especificación de requisitos de software esta expresada en un nivel informal entendible por los participantes del lado del Adquiriente y del Proveedor.	<input type="checkbox"/>	
7	Cada requisito es necesario para cumplir con los objetivos del sistema; es decir, no representan alguna característica trivial o agregada.	<input type="checkbox"/>	
8	No existen conflictos entre requisitos.	<input type="checkbox"/>	
9	Cada requisito es realizable técnicamente, considerando costes y planificación.	<input type="checkbox"/>	
10	Cada requisito es verificable; es decir, existe una forma de comprobar que el producto cumple con el requisito.	<input type="checkbox"/>	

Nro	Criterio	Revisión	Observaciones
11	<p>La especificación de requisitos incorpora todos los requisitos no funcionales relevantes acordados entre Adquiriente y Proveedor, entre ellos:</p> <ul style="list-style-type: none"> a. Requisitos de interfaz. b. Requisitos de seguridad. c. Desempeño, capacidad y precisión requerida. d. Requisitos de usabilidad. e. Confiabilidad. f. Disponibilidad. g. Tolerancia a fallos. h. Mantenibilidad. i. Portabilidad. j. Reusabilidad. k. Interoperabilidad. l. Facilidad en la instalación del producto. 	<input type="checkbox"/>	



1.6. Documento de aceptación para la Especificación de Requisitos de Software.

El documento de aceptación para la especificación de requisitos de software es realizado después que se realiza la revisión de la especificación de requisitos de software con un resultado satisfactorio que evidencia que cumple con las características deseadas, a continuación el artefacto.

Aceptación de Especificación de Requisitos de Software

Nombre del proyecto:

Fecha de Aceptación:

Mediante el presente documento se deja constancia de la revisión a la especificación requisitos de software, con resultado satisfactorio, que tuvo como responsable al Sr(a) Nombre del responsable, como participantes de la revisión a los Sres(as): Nombre de los participantes de la revisión y fecha Fecha de la revisión. Como consecuencia se da por ACEPTADO EL DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE posibilitando la ejecución de las siguientes etapas del desarrollo del proyecto Nombre del proyecto.

Firma del Adquiriente

Firma del encargado de calidad del cliente

1.7. Lista de comprobación del Plan de aceptación

Con la finalidad de brindar un conjunto de características deseables y viables para el producto software se establece un listado de criterios que debe cumplir el Plan de aceptación, dicho listado de criterios y características a cumplir es la “Lista de comprobación del Plan de aceptación”.

La “Lista de comprobación del Plan de aceptación” se elaboró a partir de (Vega, 2012) y criterios propios relacionados al cumplimiento y posibilidad de ejecución de las tareas y artefactos que componen la actividad “Generar Plan de aceptación”, por parte del Proveedor.

Lista de comprobación del Plan de aceptación

Nombre de proyecto:

Nombre del revisor:

Rol del revisor:

Fecha de revisión:

Nro	Criterio	Revisión	Observaciones
1	Las actividades de V & V consideradas como parte del Plan de aceptación fueron determinadas a partir del Inventario de actividades de V & V.		
2	Las actividades de V & V seleccionadas pueden ser desarrolladas y cumplidas con recursos e infraestructura asignados al proyecto.		
3	Las herramientas y el proceso a seguir para el control de cambios incluidos en el Plan de aceptación fueron determinados a partir de la Guía de control de cambios.		
4	El proceso de control de cambios puede ser implementado y cumplido con los recursos e infraestructura asignados al proyecto.		
5	Las técnicas de pruebas consideradas como parte del Plan de aceptación fueron determinadas a partir del Inventario de pruebas propuesto.		
6	Las pruebas determinadas pueden ser desarrolladas y cumplidas con recursos e infraestructura asignados al proyecto.		

Nro	Criterio	Revisión	Observaciones
7	Las condiciones de entorno e infraestructura para para las pruebas y actividades de V & V son los mismos que para el entorno e infraestructura final del producto.		
8	Los entregables solicitados al proveedor están pensados para contener evidencia de la ejecución de las actividades de V & V seleccionadas.		
9	La organización responsable de la validación del producto y ejecución de las pruebas se encuentra plenamente identificada.		

1.8. Guía de realización de revisión entre pares

El presente artefacto es elaborado con la finalidad de definir y guiar en la aplicación de una revisión entre pares, la cual es aplicada como técnica de verificación y validación en el proceso de revisión de un determinado documento o artefacto de la metodología. La realización de la guía de realización de revisión entre pares fue adaptada de Pierre Robillard et. al (2007).

Guía para Revisión entre pares

En primer lugar se presenta la definición de revisión, “Una revisión es un proceso o reunión durante el que un producto software, conjunto de productos software, o proceso software es presentado al personal del proyecto, administradores, usuarios, clientes, usuarios representativos, auditores u otra parte interesada en la evaluación, observación u aprobación” (IEEE 10208 2008).

Los roles a desempeñarse por los participantes de la revisión entre pares son:

- Responsable: Encargado de la preparación de la ejecución y difusión de los resultados de la revisión entre pares.
- Moderador: Encargado de la conducción de la revisión entre pares.
- Evaluador: Encargado de encontrar defectos en el artefacto, documento o producto.

Para la realización de la revisión entre pares considerar los siguientes pasos:

- 1) Planificación: El responsable de la actividad relacionada a la revisión entre pares designa un responsable para dicha revisión. En esta etapa se asignaran los roles al resto de los participantes de la revisión entre pares.
- 2) Preparación previa: El moderador informa y brinda a los participantes de la revisión entre pares el contexto y el artefacto, documento o producto a evaluarse,

así como determina y brinda un tiempo a los participantes de la revisión para prepararse y revisar la información provista.

- 3) Reunión de detección de defectos: El moderador se encarga de estructurar, clasificar los temas de discusión y controlar los tiempos invertidos en cada tema. Posteriormente a la discusión de los temas, y si se dispone de una herramienta de evaluación, se procederá a la evaluación del artefacto, documento o producto correspondiente que posteriormente se encargara de analizar el Responsable, de lo contrario se procede a una aprobación por consenso.
- 4) Resultado: El responsable se encarga de analizar los resultados, en caso sea necesario, y de la posterior difusión de los resultados.

1.9. Inventario de actividades de verificación y validación (V & V)

El presente artefacto presenta un conjunto de actividades de verificación y validación aplicables al ciclo de vida de desarrollo de software, tiene como finalidad asegurar la calidad de software a través de la selección de actividades y el compromiso del Proveedor para realizar dichas actividades durante el desarrollo de software, en el contexto de una adquisición de software.

Las actividades contenidas en el inventario de V & V se basan en el standard IEEE 1012 (2012) que propone un conjunto de actividades de verificación y validación mínimas para procesos del ciclo de vida de sistemas, software y hardware.

Una característica destacable del estándar IEEE 1012 es la clasificación de las actividades en 4 niveles de integridad asociados a la criticidad del producto software, su uso y la importancia del sistema que sustenta. El conjunto de actividades consideradas para el inventario pertenecen a los niveles 1 y 2 de integridad, debido al nivel de criticidad propio de los sistemas de información realizados en pequeños contextos.

Inventario de actividades de verificación y validación

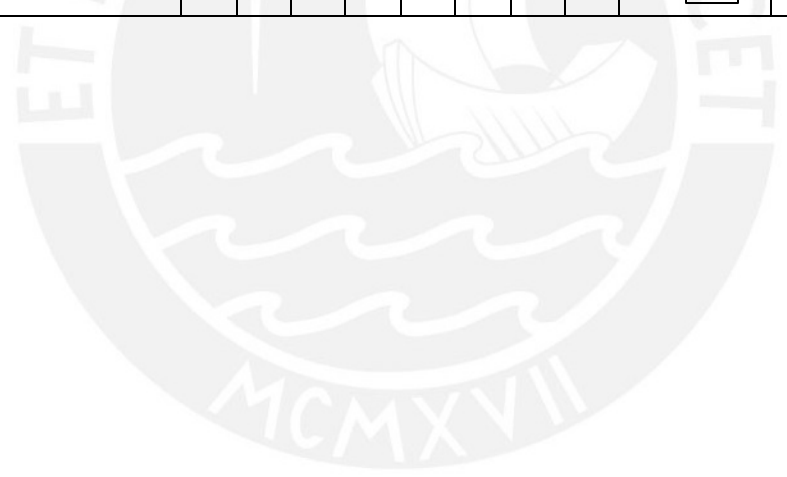
Nombre de proyecto:
 Nombre del responsable(es):
 Rol(es) responsable(es):
 Fecha de selección:

Nro	Actividades	Conceptos	Requisitos	Diseño	Implementación	Pruebas	Instalación	Operación	Mantenimiento	Selección	Observación
1	Evaluación de la documentación de la etapa de Conceptos	X								<input type="checkbox"/>	
2	Evaluación de requisitos		X							<input type="checkbox"/>	
3	Evaluación del diseño			X						<input type="checkbox"/>	
4	Evaluación de documentación de código fuente				X					<input type="checkbox"/>	
5	Evaluación de nuevas restricciones							X		<input type="checkbox"/>	
6	Análisis de criticidad	X	X	X	X				X	<input type="checkbox"/>	
7	Análisis de trazabilidad	X	X	X	X	X				<input type="checkbox"/>	

Nro	Actividades	Conceptos	Requisitos	Diseño	Implementación	Pruebas	Instalación	Operación	Mantenimiento	Selección	Observación
8	Análisis de interface		X	X	X					<input type="checkbox"/>	
9	Iteración de tareas								X	<input type="checkbox"/>	
10	Revisión del plan de V & V								X	<input type="checkbox"/>	
11	Evaluación de anomalías								X	<input type="checkbox"/>	
Verificación y validación de casos de pruebas de software											
12	Aceptación				X					<input type="checkbox"/>	
13	Componentes				X					<input type="checkbox"/>	
14	Integración				X					<input type="checkbox"/>	
Verificación y validación de procedimiento de pruebas de software											
15	Aceptación					X				<input type="checkbox"/>	

Nro	Actividades	Conceptos	Requisitos	Diseño	Implementación	Pruebas	Instalación	Operación	Mantenimiento	Selección	Observación
16	Componentes				X					<input type="checkbox"/>	
17	Integración				X					<input type="checkbox"/>	
Verificación y validación de ejecución de pruebas de software											
18	Aceptación					X				<input type="checkbox"/>	
19	Componentes				X					<input type="checkbox"/>	
20	Integración					X				<input type="checkbox"/>	
Verificación y validación del diseño de pruebas de software											
21	Aceptación			X						<input type="checkbox"/>	
22	Componentes			X						<input type="checkbox"/>	
23	Integración			X						<input type="checkbox"/>	
Verificación y validación del plan de pruebas de software											

Nro	Actividades	Conceptos	Requisitos	Diseño	Implementación	Pruebas	Instalación	Operación	Mantenimiento	Selección	Observación
24	Aceptación		X							<input type="checkbox"/>	
25	Componentes			X						<input type="checkbox"/>	
26	Integración			X						<input type="checkbox"/>	



1.10. Guía de Control de cambios

El control de cambios surge debido al cambio de los requisitos y las necesidades organizacionales, trayendo como consecuencia cambios en el software y documentos asociados que tienen que ser analizados en coste y beneficio antes de su implementación (Sommerville 2005).

Mediante el presente artefacto se establece las herramientas y el proceso a seguir para realizar una correcta gestión de cambios, adecuada a pequeños contextos, cuya importancia está basada en asegurar que los cambios se realicen en forma correcta.

Para el desarrollo de la guía de control de cambios se usó como referencia a Sommerville (2005) y Vega (2012).

Guía para el control de cambios

Para realizar el control de cambios se toma en cuenta su proceso, el cual es descrito a través del diagrama presentado en la Figura Anx1.

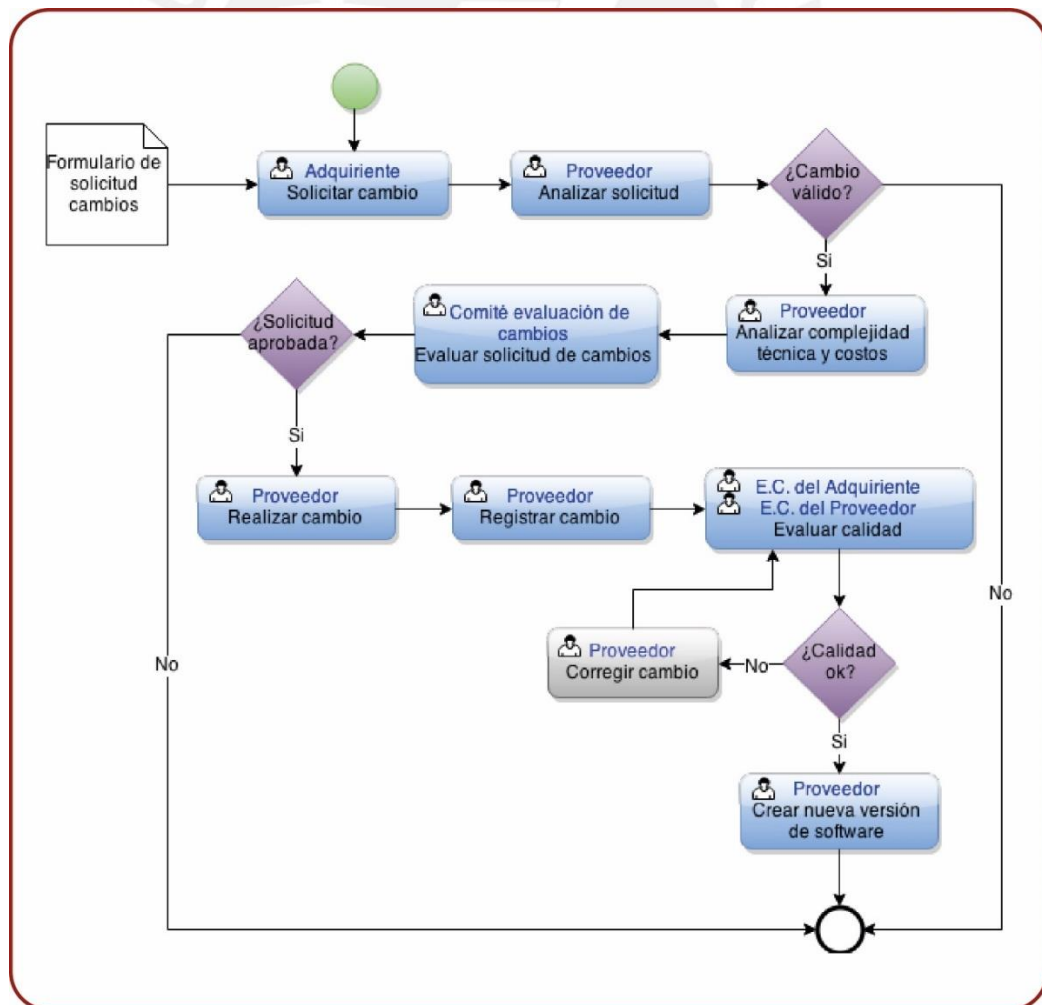


Figura Anx1. Proceso de Gestión de cambios. Elaboración propia. Basado en (Sommerville 2005)

Tal como se puede observar en la Figura Anx1 el proceso de Control de cambios seguido incluye 2 herramientas importantes: el formulario de solicitud de cambios y el comité de solicitud de cambios.

- Formulario de solicitud de cambios:** Esta herramienta tiene como finalidad señalar los cambios requeridos en el software, su descripción, costos y fechas pertinentes, así como evidenciar su aprobación o desaprobación, la Figura Anx2 muestra un formulario de solicitud de cambios a considerar como base.

General	
Nombre proyecto:	
Nombre solicitante:	Fecha:
Cambio solicitado:	
Análisis	
Analista asignado al cambio:	Fecha de análisis:
Componentes afectados:	
Componentes asociados:	
Evaluación del cambio	
Descripción:	
Prioridad:	
Implementación del cambio	
Fecha evaluación de comité:	Decisión comité:
Analista responsable del cambio:	Fecha de inicio:
Fecha revisión de calidad:	Decisión de evaluación de calidad:
Observaciones:	

Figura Anx2. Formulario de solicitud de cambios. Elaboración propia. Basado en Sommerville (2005)

- Comité de evaluación de cambios:** Es un grupo de participantes del proyecto que tienen a su cargo la decisión de evaluar las solicitudes de cambio que afectaran la línea base del software y/o la documentación asociada. Según Vega (2012) la conformación de dicho comité varía de 1 a 3 personas en proyectos de pequeña escala, para la presente metodología dicho comité estará compuesto por el encargado de calidad del Adquiriente, el encargado de calidad del proveedor y opcionalmente un usuario

1.11. Inventario de pruebas

El inventario de pruebas tiene como propósito proveer un conjunto de técnicas de pruebas que, una vez seleccionadas, son usadas en el diseño de pruebas e implementación a través de su aplicación en la construcción de casos de prueba.

La realización y posterior ejecución de casos de prueba nos brinda evidencia de la satisfacción de requisitos o presencia de defectos en el componente de prueba que forma parte del sistema de información o software afecto a dichas pruebas.

El desarrollo del formato de inventario de pruebas se basó íntegramente en el estándar *Software and Systems Engineering - Software Testing - Part 4: Test Techniques* (2013) que tiene como objetivo el proveer a los interesados en el proyecto la habilidad de realizar pruebas de software en cualquier organización.

Los tipos de pruebas incluidos en el inventario de pruebas están basados en especificación, basados en la estructura y en la experiencia, en donde los documentos que especifican al componente de pruebas, la estructura del componente de pruebas y el conocimiento o experiencia son fuentes primarias respectivamente.

Inventario de pruebas

Nombre de proyecto:

Nombre del responsable(es):

Rol(es) responsable(es):

Fecha de selección:

Nro	Tipo de prueba	Nombre	Descripción	Selección	Observación
1	Basado en especificación	Particionamiento equivalente	Se particiona las entradas y salidas del componente de pruebas en particiones equivalentes en base a las que se realizan las pruebas.	<input type="checkbox"/>	
2	Basado en especificación	Método de clasificación de árbol	Se particiona las entradas del componente de pruebas y se les representa gráficamente en forma de árbol. Las particiones son clasificaciones disjuntas que en suma forman todas la entradas, las pruebas se realizan en base a la combinación de clasificaciones.	<input type="checkbox"/>	
3	Basado en especificación	Análisis de valor límite	Se particiona las entradas y salidas del componente de pruebas en un numero ordenado de conjuntos con limites identificables, las pruebas se realizan en base a un valor en el límite y al valor más pequeño considerado el cual se aleja incrementalmente del límite.	<input type="checkbox"/>	

Nro	Tipo de prueba	Nombre	Descripción	Selección	Observación
4	Basado en especificación	Prueba de sintaxis	A partir de un modelo de sintaxis definido se aplican un conjunto de reglas de entradas a un componente de pruebas, cada regla define el formato de un parámetro de entrada entre elementos de sintaxis, las pruebas se realizan en base a las opciones permitidas en la sintaxis.	<input type="checkbox"/>	
5	Basado en especificación	Prueba de pares acertados	Se aplica a componentes de prueba con parámetros restringidos. Para las pruebas se considera la combinación de parámetros (P) y sus valores (V). En este caso P = todos, V = algunos seleccionados.	<input type="checkbox"/>	
6	Basado en especificación	Prueba de cada elección	Se aplica a componentes de prueba con parámetros restringidos. Para las pruebas se considera la combinación de parámetros (P) y sus valores (V). En este caso P = todos, V = todos.	<input type="checkbox"/>	
7	Basado en especificación	Prueba de elección base	Se aplica a componentes de prueba con parámetros restringidos. Para las pruebas se considera la combinación de parámetros (P) y sus valores (V). En este caso se establece un parámetro base inicial y un parámetro base final, V = todos.	<input type="checkbox"/>	

Nro	Tipo de prueba	Nombre	Descripción	Selección	Observación
8	Basado en especificación	Prueba de tabla de decisiones	Se aplica un modelo de regla de decisiones al componente de pruebas, es decir se consideran condiciones (causas) y acciones (efectos), cada condición es una entrada o combinación de entradas y cada acción es una salida o combinación de salidas. Las pruebas se basan en las combinaciones únicas de causas y efectos.	<input type="checkbox"/>	
9	Basado en especificación	Gráfico causa - efecto	Se aplica un modelo de regla de decisiones al componente de pruebas, ilustrándolo a través de operadores booleanos, es decir se consideran condiciones (causas) y acciones (efectos), cada condición es una entrada o combinación de entradas y cada acción es una salida o combinación de salidas. Las pruebas se basan en las combinaciones únicas de causas y efectos.	<input type="checkbox"/>	
10	Basado en especificación	Prueba de transición de estados	El componente de pruebas se representa a través de un modelo de estados discreto, identificable y finito con transiciones entre estados, eventos que causan transiciones y acciones resultantes de transiciones. Las pruebas se basan en los estados y transiciones y varían de acuerdo al enfoque adoptado.	<input type="checkbox"/>	

Nro	Tipo de prueba	Nombre	Descripción	Selección	Observación
11	Basado en especificación	Prueba de escenario	Se usa un modelo de pruebas de interacciones entre el componente de pruebas y otros sistemas para probar flujos de uso involucrados con el componente de pruebas. Las pruebas se basan en escenarios principales que son una secuencia de acciones esperada y escenarios alternativos que son secuencias de acciones alternativas.	<input type="checkbox"/>	
12	Basado en especificación	Prueba aleatoria	Se elige una distribución estadística para generar valores de entradas aleatorios para el componente de pruebas y realizar las pruebas.	<input type="checkbox"/>	
13	Basado en la estructura	Prueba de instrucción	Se identifica como ejecutables y no ejecutables las instrucciones del código fuente del componente de pruebas. Las pruebas se basan en las instrucciones ejecutables.	<input type="checkbox"/>	
14	Basado en la estructura	Prueba de ramificación	Se utiliza un modelo de flujo de control del componente de pruebas en el que se identifican ramas. Las pruebas están basadas en las ramas. Una rama es una transferencia de control condicional entre nodos, transferencia sin condiciones entre nodos, transferencia de control de punto de entrada.	<input type="checkbox"/>	

Nro	Tipo de prueba	Nombre	Descripción	Selección	Observación
15	Basado en la estructura	Pruebas de decisión	Se utiliza un modelo de flujo de control del componente de pruebas en el que se identifican decisiones que son puntos con 2 o más posibles salidas. Las pruebas están basadas en las salidas de cada decisión.	<input type="checkbox"/>	
16	Basado en la estructura	Prueba de condición de ramas	Se utiliza un modelo de flujo de control del componente de pruebas en el que se identifican decisiones y condiciones. Las pruebas se basan en los valores de verdad de las condiciones y las decisiones.	<input type="checkbox"/>	
17	Basado en la estructura	Prueba de combinación de condición de ramas	Se utiliza un modelo de flujo de control del componente de pruebas en el que se identifican decisiones y condiciones. Las pruebas se basan en cada combinación factible de valores de verdad de las condiciones dentro de cada decisión.	<input type="checkbox"/>	
18	Basado en la estructura	Prueba de cobertura de decisión y condición modificada	Se utiliza un modelo de flujo de control del componente de pruebas en el que se identifican decisiones y condiciones. Las pruebas se basan en cada combinación factible de valores de verdad de las condiciones dentro de una decisión que permita una condición booleana afectar la salida independientemente.	<input type="checkbox"/>	

Nro	Tipo de prueba	Nombre	Descripción	Selección	Observación
19	Basado en la experiencia	Tanteo de error	Las pruebas involucran el diseño de una lista de comprobación de errores que posiblemente existan en el componente de pruebas, de tal manera que se permita identificar entradas que causen fallas.	<input type="checkbox"/>	

1.12. Guía de trazabilidad de requisitos

El presente artefacto tiene como propósito definir un conjunto de pasos que sirvan de guía en la práctica de trazabilidad de diversos objetos de la etapa de análisis. En detalle, la trazabilidad se refiere a la asociación o rastreabilidad de objetos de la etapa de análisis, en especial los requisitos, hacia objetos y componentes del sistema propios de otras etapas del desarrollo de sistemas y del análisis mismo. (Tabares et. al. 2006)

Como artefacto resultante de aplicar la Guía de trazabilidad de requisitos se obtendrá una o más matrices de trazabilidad, es recomendable su gestión a través de alguna herramienta CASE (*Computer aided Software Engineering*) o software de ofimática. Estas matrices, a partir de sus elementos, además de servir como registro de la relación o dependencia entre ellos, sirven para analizar ciertas características tales como nivel de especificación de requisitos, nivel de participación de usuarios, costo por etapa de desarrollo, arquitectura, entre otras (Tabares et. al. 2006)

La guía de trazabilidad de requisitos fue desarrollada a partir de (Loucopoulos Et. al. 2005) y de la descripción de los modelos de (Lindvall y Sandahl 1996) y (Ramesh y Jarke 2001), encontrados en (Tabares et. al. 2006)

Guía de trazabilidad de requisitos

Para la realización de la trazabilidad del modelo, donde por modelo se entiende el proyecto software correspondiente, se consideran los siguientes pasos:

- 1) Identificar los elementos de trazado existentes correspondientes al modelo.

Los elementos de trazado se clasifican en:

- **Fuentes:** Las fuentes es el artefacto físico donde la información es mantenida, por ejemplo la ERS, documento de diseño, memorándum, entre otros.
 - **Stakeholders:** Representan a los involucrados en el proyecto de desarrollo que tienen roles sobre los objetos y vínculos, como ejemplos tenemos al Adquiriente, el analista de sistemas, el gestor de proyectos, entre otros.
 - **Objetos:** Son los diferentes tipos de información que se relacionan entre ellos, como ejemplo se tiene a los requisitos, componentes del sistema, decisiones, entre otros.
- 2) Identificar los vínculos de trazado entre los diferentes elementos del modelo.

Los principales vínculos usados en relación con las fuentes, *stakeholders* y objetos se presentan a continuación:

- Objeto “trazable a” Objeto
- *Stakeholder* “tiene rol en” Objeto
- *Stakeholder* “gestiona” Fuente
- Fuente “documenta” Objeto

Otros vínculos a considerar son:

- “depende de”, para dependencia entre objetos.
- “evoluciona a”, para representar cambios de etapa o niveles del proceso de desarrollo de software.

3) Creación de matrices de trazabilidad a partir de los elementos y vínculos identificados en el modelo.

Como ejemplos de aplicación se presentan los siguientes casos:

- Objeto: Requisito “trazable a” Objeto: Caso de uso

Casos de uso Requisitos	CU 1	CU 2	CU 3	CU 4
Requisito 1		J		
Requisito 2				J
Requisito 3	J			
...				
Requisito n			J	

- Stakeholder “tiene rol en” Objeto: Componente del sistema

Componente Stakeholder	Componente 1	Componente 2	Componente 3	Componente 4
Desarrollador 1		J		J
Desarrollador 2	J		J	
Gestor de proyecto	J	J	J	J
...				
Analista 1			J	J
Analista 2	J	J		
Diseñador BD		J	J	J

1.13. Lista de comprobación de entregables del proveedor

El presente artefacto es creado con motivo de proveer un conjunto de características y criterios a tenerse en cuenta para la revisión de los entregables del proveedor. Dichos criterios toman en cuenta lo establecido en el Plan de aceptación para la etapa de análisis.

La Lista de comprobación de entregables del proveedor está basada en criterios propios y criterios obtenidos de (Vega, 2012).

Lista de comprobación de entregables del proveedor

Nombre de proyecto:

Nombre del responsable(es):

Rol(es) responsable(es):

Fecha de revisión:

Criterio	Revisión	Observaciones
Las actividades correspondientes a la etapa de análisis que aparecen en los entregables fueron definidas en el "Inventario de actividades de V & V".	<input type="checkbox"/>	
Las actividades que figuran en los entregables son consistentes con el "Inventario de pruebas" definido.	<input type="checkbox"/>	
Las actividades presentadas en los entregables muestran los resultados obtenidos.	<input type="checkbox"/>	
De presentarse algún resultado irregular, sus causas se encuentran identificadas.	<input type="checkbox"/>	
Los mecanismos que resuelven las anomalías o resultados irregulares son adecuados y consistentes con las causas identificadas.	<input type="checkbox"/>	

Criterio	Revisión	Observaciones
Se identifican lecciones aprendidas y buenas prácticas a partir de los resultados irregulares.	<input type="checkbox"/>	
Existen evidencias de la ejecución de las actividades desarrolladas.	<input type="checkbox"/>	
Existen evidencias de la resolución de las anomalías o resultados irregulares.	<input type="checkbox"/>	

1.14. Plantilla para validación por juicio experto

La plantilla de validación por juicio experto fue usada como instrumento en la validación por jueces, permitiendo, a través de las calificaciones en las diversas categorías y observaciones, evaluar y obtener retroalimentación en la metodología.

Plantilla para validación por juicio experto

Estimado Revisor, me es grato dirigirme a usted con la finalidad de solicitar su colaboración en la evaluación de la *Metodología de verificación y validación de adquisición en la etapa de análisis de sistemas de información desarrollados a la medida en pequeños contextos*, que forma parte de uno de los proyectos de fin de carrera de la especialidad de ingeniería informática de la PUCP para el ciclo 2015 – I.

La evaluación de los ítems planteados contribuirá directamente a las conclusiones sobre la aplicabilidad y validez de la metodología; sin embargo la determinación de la validez de la metodología propiamente dicha, se determinará posteriormente a través de consenso entre revisores.

Nombres y apellidos del Revisor: _____
 Formación académica: _____
 Áreas de experiencia profesional: _____

Objetivo del Juicio de expertos: Validar el contenido y aplicabilidad de una metodología de adquisición de software en la que se incluyen técnicas de verificación y validación y parámetros como su aplicación a sistemas hechos a la medida y a pequeños contextos

Objetivo de la validación: Usar los resultados obtenidos de la validación para la obtención de conclusiones con respecto a cada dimensión. Se usará valores promedio o representativos obtenidos a partir de la calificación de ambos revisores.

De acuerdo con los siguientes indicadores califique cada uno de los ítems según corresponda:

CATEGORIA	CALIFICACION	INDICADOR
<p>SUFICIENCIA</p> <p>Los ítems que pertenecen a una misma dimensión bastan para obtener la medición de esta.</p>	<p>1. No cumple con el criterio</p> <p>2. Bajo nivel</p> <p>3. Moderado nivel</p> <p>4. Alto nivel</p>	<p>1. Los ítems no son suficientes para medir la dimensión.</p> <p>2. Los ítems miden algún aspecto de la dimensión pero no corresponden con la dimensión total.</p> <p>3. Se debe incrementar algunos ítems para poder evaluar la dimensión completamente.</p> <p>4. Los ítems son suficientes</p>

CATEGORIA	CALIFICACION	INDICADOR
<p style="text-align: center;">CLARIDAD</p> <p>El ítem es entendible, es decir su identificación y comprensión son sencillas.</p>	<ol style="list-style-type: none"> 1. No cumple con el criterio 2. Bajo nivel 3. Moderado nivel 4. Alto nivel 	<ol style="list-style-type: none"> 1. El ítem no es claro en la metodología. 2. El ítem requiere bastantes modificaciones o una modificación muy grande para expresarse con claridad en la metodología. 3. Se requiere una modificación específica de algunos de los aspectos del ítem en la metodología. 4. El ítem es totalmente claro en la metodología.
<p style="text-align: center;">COHERENCIA</p> <p>El ítem tiene relación lógica con la metodología y está presente en ella, y más específicamente está presente en la dimensión que se está midiendo.</p>	<ol style="list-style-type: none"> 1. No cumple con el criterio 2. Bajo nivel 3. Moderado nivel 4. Alto nivel 	<ol style="list-style-type: none"> 1. El ítem no tiene relación lógica y/o no está presente en la dimensión. 2. El ítem guarda poca relación y/o esta apenas presente en la dimensión. 3. El ítem tiene una relación moderada con la dimensión y/o está presente en la dimensión. 4. El ítem está totalmente relacionado con la dimensión que está midiendo y/o notoriamente presente.

CATEGORIA	CALIFICACION	INDICADOR
<p style="text-align: center;">RELEVANCIA</p> <p>Mide la importancia del ítem en la metodología con la consecuente necesidad de su inclusión.</p>	<ol style="list-style-type: none"> 1. No cumple con el criterio 2. Bajo nivel 3. Moderado nivel 4. Alto nivel 	<ol style="list-style-type: none"> 1. El ítem puede ser eliminado sin que se vea afectada la metodología y la dimensión. 2. El ítem tiene alguna relevancia en la metodología y dimensión, pero puede estar incluido en otro. 3. El ítem es relativamente importante en la metodología y dimensión. 4. El ítem es muy relevante en la metodología y dimensión, por lo que su presencia imprescindible.



Dimensiones de la metodología	Ítems de la metodología a calificarse	Coherencia	Relevancia	Claridad	Suficiencia	Observaciones
Estructura	Roles					
	Actividades					
	Uso de diagramas					
	Artefactos					
Adquisición de Software	Uso de Buenas prácticas					
	Relación entre Adquiriente y Proveedor					
	Integración de las fases de adquisición					
Verificación & Validación	Uso de técnicas de V & V					
	Integración al proceso de adquisición					
Pequeños contextos	Adaptabilidad a pymes					
	Complejidad					

Dimensiones de la metodología	Ítems de la metodología a calificarse	Coherencia	Relevancia	Claridad	Suficiencia	Observaciones
Etapa de Análisis	Calidad en Gestión de requisitos					



1.15. Constancia de validación

La constancia de validación es un formato que constituye la prueba de que el revisor ha realizado la validación de la metodología; asimismo da a conocer el resultado final de la validación.

Formato de Constancia de validación

Nombres y apellidos del Revisor: _____

Formación académica: _____

Áreas de experiencia profesional: _____

Nombres y apellidos del Revisor: _____

Formación académica: _____

Áreas de experiencia profesional: _____

Por medio de la presente hacemos constar que, a través del instrumento correspondiente, se revisó con fines de validación la *Metodología de verificación y validación de adquisición en la etapa de análisis de sistemas de información desarrollados a la medida en pequeños contextos*, habiéndose determinado por consenso que dicha metodología (es válida y aplica | no es válida y no aplica) como tal para su fin correspondiente.

Lima, xx de Junio del 2015.

Firma

Firma

2. Referencias bibliográficas

- ASSMANN, D. y T. PUNTER
2003 *Towards partnership in Software subcontracting. Elsevier.*
- AKBAR, Uzair; BLEKINGE, Raja y Tekniska HÖGSKOLA
2009 “Empirical Studies of Requirements Validation Techniques”.*2nd International Conference on. Ronneby, Sweden.*
- BERENBACH, Brian; PAULISH, Daniel; KAZMEIER, Juerguen y Arnold RUDORFER
2009 *Software & Systems Requirements Engineering in Practice.*
- BIOLCHINI, Jorge; GOMES MIAN, Paula; CÁNDIDA, Ana; CRUZ, Natali y Guilherme HORTA TRAVASSOS,
2005 *Systematic Review in Software Engineering. Rio de Janeiro: PESC, pp. 1*
- BONIFACIO ELOY MENDOZA ORTIZ
2009 *Definición de planes de acción como resultado de las evaluaciones de procesos Software en las pequeñas y medianas empresas Software. Tesis para obtener el grado de maestro en ingeniería de Software. Oaxaca: Universidad Tecnológica de la Mixteca*
- CALVO-MANZANO, José; GARZÁS, Javier; PIATTINI, Mario; PINO, Francisco; SALILLAS Jesús y José Luis SÁNCHEZ
2008 “Perfiles del ciclo de vida del Software para pequeñas empresas: los informes técnicos ISO/IEC 29110 ” en Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.4, No. 2.
- CINCOM SYSTEMS INC
2008 *To Build or Buy? A Question of Application Development for Compliance and Quality Systems. White Paper.*
- ESCOBAR PEREZ; Jazmine y Ángela CUERVO MARTINEZ
2008 *Validez de Contenido y juicio de expertos: Una aproximación a su utilización*
- GASCA, Gloria y Gonzalo CUEVAS
2008 “Un análisis crítico comparativo de modelos y estándares relacionados con la adquisición del Software”. *VII Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento. Guayaquil, p. 117-125.*
- GENDREAU, Olivier; ROBILLARD, Pierre y Pierre LABRÈCHE,
2007 *Peer review as a V & V Standard Compliance Technique, An Aviation Industry Case Study. Electrical and Computer Engineering.*

- HULL, Elizabeth; JACKSON, Ken y Jeremy DICK
2005 *Requirements Engineering, second edition.*
- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE)
1998 *IEEE Std 1062. Recommended Practice for Software Acquisition.*
- INEI
2013 Micro, Pequeñas y Medianas empresas concentran mas del 20% de las ventas. Consulta: 20 de Marzo del 2015.
<<http://www.inei.gob.pe/prensa/noticias/micro-pequenas-y-medianas-empresas-concentran-mas-/>>
- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS
1998 "IEEE Std 1062: 1998 Recommended Practice for Software Acquisition".
Software Engineering Standards Committee of the IEEE Computer Society.
- 1998 *IEEE 830 Recommended Practice for Software Requirements Specifications.*
- 2008 *IEEE 10208. Stándar for software reviews and audits.*
- INFORMATION SYSTEM PROCUREMENT LIBRARY (ISPL)
Consulta: 2 de Marzo del 2015.
< <http://projekte.fast.de/ISPL/>>
- INSTITUTO NACIONAL DE TECNOLOGIAS DE LA COMUNICACIÓN (INTECO)
2009 *Guía de Validación y Verificación.* Madrid: Laboratorio Nacional de Calidad del Software.
- SOFTWARE ENGINEERING METAMODEL FOR DEVELOPMENT
METHODOLOGIES ISO/IEC
- 2006 *NTP-ISO/IEC 12207.*
- 2007 *ISO/IEC 24744.*
- 2008 *ISO/IEC 12207*
- 2010 *ISO/IEC TR 24774*
- 2011 *ISO/IEC 29110*
- JOHAN C.
2005 *IT Services Procurement, Based on ISPL - a Pocket Guide.* Van Haren Publishing.

- KENDALL, Kenneth E. y Julie E. KENDALL
2011 *Análisis y Diseño de Sistemas*. Octava edición.
- KITCHENHAM, Bárbara
2004 "Procedures for performing systematic reviews". Vol. 33. Keele, pp. 1- 3.
- LADRÓN DE GUEVARA
GERVERA, Michele; HINCAPIÉ, Jhoana; JACKMAN, Joseph; HERRERA, Omar
y Carlo Vinicio CABALLERO URIBE
2008 "Revisión por pares: ¿Qué es y para qué sirve?". *Salud*. Barranquilla,
v.24, n.2.
<http://www.scielo.co/scielo.php?script=sci_arttext&pid=S0120-55522008000200011&lng=es&nrm=iso>
- LALIOTI, Vali
1997 "Animation for Validation of Business System Specifications". [System Sciences, Proceedings of the Thirtieth Hawaii International Conference. Maui County, volumen 2](#), pp. 220 – 229.
- LINDVALL M. y SANDAHL
1996 "Practical implications of traceability". *Journal of Software Practice and Experience*. pp. 1161-1180.
- LOUCOPOULOS, Pericles Et. al.
2005 *Engineering and Managing Software Requirements*. Editorial Springer
- MCGARTLAND, D Et. al
2003 "Objectifying content validity: Conducting a content validity study in social work research". *Social Work Research*. Número 27, volume 2, pp. 94 -104.
- MINISTERIO DE ADMINISTRACIONES PÚBLICAS
2001 *Introducción a Métrica Versión 3*. Madrid.
- MISHRA, D., and MISHRA, A.
2007 "Efficient Software review process for small and medium enterprises". *IET Software*. Pp. 132-142.
- PETERSEN, Kai; FELDT Robert, MUJTABA Shahid y Michael MATTSSON
2008 "Systematic Mapping Studies in Software Engineering". School of Engineering, Blekinge Institute of Technology.
- PMBOK
2013 *Guide, A Guide to the project Management Body of Knowledge*. Quinta Edition.

- POHL, Klaus
2010 *Requirements Engineering: Fundamentals, Principles, and Techniques.* Springer Publishing Company.
- PRESSMAN, Roger
2010 *Ingeniería del software, Un enfoque práctico.*
- RAMESH B. y JARKE
2001 "Toward reference models for requirements traceability". *IEEE Transactions on Software Engineering.* Vol. 27, No. 1.
- RUNESON, P., ANDERSSON, C. Y THELIN, T.
2006 *What Do We Know about Defect Detection Methods.* IEEE Software
- OMG
2011 *Documento normativo formal/2011-01-03 asociado con Business Process Model and Notation (BPMN) Versión 2.0.*
- SANZ ESTEBAN, Ana
2012 *Marco metodológico para la mejora de las actividades de Verificación y Validación de Productos Software.* Tesis doctoral. Madrid: Universidad Carlos III, Departamento de informática.
- SIMMONS, R.A.
1989 "SQE vs SQA vs IV&V". *Aerospace and Electronics Conference.* Dayton: IEEE.
1990 "Software Quality Assurance (SQA) early in the acquisition process". *Aerospace and Electronics Conference.* Dayton: IEEE.
- SINGH, Raghu
1995 *International Standard ISO/IEC 12207 Software Life Cycle Processes.* Washington.
- SKJONG, R. y WENTWORTH, B.
2000 *Expert Judgement and risk perception*
- SOFTWARE ENGINEERING INSTITUTE OF CARNEGIE MELLON UNIVERSITY
2002 *SA-CMM. Software Acquisition Capability Maturity Model. Version 1.03.*
2007 *CMMI for Acquisition, Version 1.2*
2010 *CMMI for development version 1.3.*
2010 *CMMI for Acquisition. Version 1.3.*

- SOMMERVILLE, Ian
2005 *Ingeniería del software*. 7 ma edición.
- STANDISH GROUP INTERNATIONAL
2009 “The 10 Laws of CHAOS”. *CHAOS Summary*.
- SVENNBERG, Daniel
2001 “Software Acquisition Management Guidelines”. *The Department of Computer and Information Science*. Suecia, p. 16.
- TABARES, Marta Silvia; ARANGO, Fernando y Raquel ANAYA
2006 “Una revisión de modelos y semánticas para la trazabilidad de requisitos”. *Revista EIA*. Medellín, Número 6, p. 33-42.
- TRAN VAN, Hung; VAN LAMSWEERDE, Axel; MASSONET, Philippe y Cristophe PONSARD
2004 “Goal-oriented requirements animation”. *Proceedings of the 12th IEEE International Requirements Engineering Conference*.
- TRAYLOR, Polly y LLOYD, David
2006 “To build or to buy IT applications?”. *Build vs. Buy. Scientific Computing & Instrumentation*.
- TOKAR, R., y S. MANKEFORTS
2003 “A survey on testing and reuse”. *The Proceedings of the IEEE International Conference on Software – Science, Technology and Engineering. Israel*, pp. 164-173.
- UTKIN, L. V.
2006 “A method for processing the unreliable expert judgments about parameters of probability distributions”. [Versión Electrónica]. *European Journal of Operational Research*. Numero 175, volumen 1, pp. 385 – 398.
- VAN BON, Jan y Verheijen TIENEKE
2006 *Frameworks for IT Management*. Vigésimo Segundo Edición. Netherland: Van Haren Publishing.
- VEGA ZEPEDA, Vianca
2012 *Metodología para el aseguramiento de la calidad en la adquisición del Software (proceso y producto) y servicios correlacionados. (MACAD-PP)*. Tesis doctoral. Madrid: Universidad Politécnica de Madrid, Facultad de informática.

WIEGERS, K.

2003

Software Requirements. Segunda Edición. Microsoft Press.

