

Herramienta para la Elaboración del Documento de Especificación de Requerimientos de Software: HEDERS

Liliana Velázquez Bello, María de los Ángeles Sumano López

Facultad de Estadística e Informática, Universidad Veracruzana, Av. Xalapa esquina Av. Manuel Ávila Camacho, Xalapa, Veracruz, México.
lily_vb16@hotmail.com, masumano04@yahoo.com.mx

Paper received on 03/08/08, accepted on 10/09/08.

Resumen. Son muchos factores que deben tomarse en cuenta al desarrollar un sistema de software. Por ello, es necesario antes de empezar la elaboración de un software, que se logren acuerdos justos entre cliente y desarrollador basándose en un documento de requerimientos de software. Dada la tardanza en la elaboración del documento de Especificación de Requerimientos de Software, se concluyó que lo mejor sería crearlo automáticamente para obtenerlo en tiempo y forma. Este documento estará basado en el estándar IEEE 830-1993. Aquí se reporta el desarrollo de tal herramienta en donde se usó: Áncora para el análisis de Requerimientos, Proceso Unificado de Desarrollo de Software para los modelos de análisis, diseño e implementación y programando en Java bajo el ambiente de MyEclipse.

1. Introducción

A pesar de los avances significativos en cuanto a tecnologías de información muchos sistemas de software son abandonados por no haber contado con una buena especificación de requerimientos ni su correspondiente documento. Por ello es necesario que, al comenzar a desarrollar un sistema, se especifiquen los requerimientos de software, éstos deberán ser correctos, completos, estructurados y entendibles tanto por el equipo desarrollador como por el cliente y los usuarios, ya de este entendimiento compartido dependerán tanto el adecuado desarrollo como la aceptación y funcionamiento del software.

Los requerimientos son una especificación de lo que debería ser implementado. Son una descripción de cómo el sistema debería comportarse o de una propiedad o atributo del sistema [1].

En la práctica existen pocas herramientas que ayuden a la especificación adecuada de los requerimientos, tanto en el lenguaje del cliente como para el desarrollador.

El proyecto HEDERS puede realizar un documento de especificación de requerimientos de software bajo el estándar IEEE 830-1998 [4], después de que el Ingeniero de Requerimientos ha establecido los modelos del mismo usando la Base de Datos de Áncora-Soft o capturado los detalles necesarios mediante interfaces especiales creadas ex profeso en HEDERS.

2. Herramientas y estándar para la Especificación de Requerimientos de Software

En la actualidad existen herramientas que permiten la Especificación de Requerimientos de Software (ERS) las cuales ayudan al logro del avance y a la sobre la especificación de requerimientos, pero por lo regular estas herramientas son costosas de adquirir. Algunos ejemplos son:

1. **RequisitePro:** herramienta para la gestión de requerimientos, es utilizado por los equipos de desarrollo de proyectos que quieren gestionar sus necesidades, escribir casos de uso, mejorar la trazabilidad, fortalecer la colaboración de equipo, reducir riesgos del proyecto y aumentar la calidad.
2. **SpeeDev:** está especialmente diseñado para proyectos distribuidos, donde los desarrolladores, usuarios y administradores están ubicados en diferentes lugares.
3. **Telelogic AB:** provee soluciones para la automatización y el soporte de mejores prácticas para la organización. Telelogic brinda soluciones para mejorar la toma de decisiones, administración de procesos empresariales, ingeniería de sistemas y desarrollo de software.

2.1. Estándar IEEE 830-1998

Para poder entender un documento de Especificación de Requerimientos de Software ERS se tiene que llevar a cabo una estructura definida, existen varias alternativas propuestas y guías para su llenado, entre ellas destacan: la guía IEEE P1233/D3, estándar IEEE 830-1998, estándar ISO/IEC 12119-1994 y el estándar IEEE 1362-1998

La importancia de utilizar un estándar en Ingeniería de Software radica en que agrupan lo mejor y más apropiado de las buenas prácticas y usos del desarrollo de software, además, engloban los conocimientos que son patrimonio de una organización, proporcionan un marco para implementar procedimientos de aseguramiento de calidad y proporcionan continuidad entre el trabajo de distintas personas.

En este proyecto se optó por el estándar de la IEEE 830-1998, ya que parece estar más difundido, está relacionado con los procesos de desarrollo del software y su esquema es bastante amplio, éste se estructura de la siguiente manera:

1. **Introducción.** En esta sección se proporciona una pequeña introducción de todo el documento ERS y contiene varios apartados como:
 - 1.1. **Propósito.** Aquí se define el propósito del documento ERS y se especifica a quien va dirigido el documento.
 - 1.2. **Ámbito del Sistema.** Se especifica el nombre del futuro sistema, se da una explicación de lo que éste hará y lo que no cubrirá, se describirán los beneficios, objetivos y metas que se esperan alcanzar con el futuro sistema.
 - 1.3. **Definiciones.** Acrónimos y Abreviaturas. Aquí se definen todos los términos, acrónimos y abreviaturas en el desarrollo de la ERS.
 - 1.4. **Referencias.** Se presentan todas las referencias de los documentos relacionados con la ERS.

- 1.5. **Apreciación Global.** Descripción breve de los contenidos y la organización del resto de la ERS.
2. **Descripción Global.** En este apartado se describen todos aquellos factores que afectan al producto y a sus requisitos.
 - 2.1. **Perspectiva del producto.** Aquí se describe el futuro del sistema, su relación con otros productos, características de los usuarios, restricciones, suposiciones y futuros requerimientos. Esta parte se subdivide en diferentes puntos importantes tales como: interfaces (del sistema, de usuario, con otros software y hardware), comunicaciones, limitaciones de memoria y de operaciones.
 - 2.2. **Funciones del producto.** Aquí se proporciona un resumen de las funciones principales que el software debe llevar a cabo. Estas funciones deberán de estar organizadas de forma que el cliente o cualquier otro involucrado las entienda.
 - 2.3. **Características de los Usuario.** Aquí se indica el tipo de usuario al que se dirige la aplicación, su experiencia, su nivel de conocimiento.
 - 2.4. **Restricciones.** Aquí se indica las limitaciones y las políticas de la empresa, como: estándares de hardware, seguridad, protocolos de comunicación, interfaz con otros sistemas.
 - 2.5. **Suposiciones y dependencias.** En este apartado aparecen cualquier factor que puede afectar a los requerimientos.
 - 2.6. **Requisitos futuros.** Aquí se indican las posibles mejoras del sistema a futuro.
3. **Requerimientos específicos.** En esta parte del documento ERS se detallan todos los requerimientos tanto funcionales como no funcionales del sistema, especificando el nombre del requisito, tipo de requisito y su prioridad.
 - 3.1. **Interfaces Externas.** En esta sección se definen los requisitos que afectan a la interfaz de usuario, interfaces de hardware, interfaces de software y la interfaz de comunicación.
 - 3.2. **Requerimientos Funcionales.** Aquí se especifican cada uno de los requerimientos funcionales del sistema con una breve descripción de cada uno de ellos.
 - 3.3. **Requerimientos no Funcionales.** En esta sección se especifican los requerimientos no funcionales. Al igual que en el apartado de arriba, se realiza una descripción detallada de cada uno de ellos, definiéndolos según el tipo, como serían: rendimiento, seguridad, fiabilidad o disponibilidad.

3. Descripción de las metodologías y herramientas usadas

La metodología Áncora [2] ha sido creada para apoyar el análisis de los requerimientos de software y los desarrolladores que la han utilizado han obtenido los resultados buscados, es decir varios artefactos que conjuntamente establecen el modelado de requerimientos. Por lo anterior se utilizó en este proyecto, cubriendo así la el modelado inicial de requerimientos. Apoyados por el Proceso Unificado de Desarrollo de Software (PUDS) [3], también conocido como RUP, se realizaron los

modelos de: análisis, diseño e implementación y, finalmente, se programó en Java bajo el ambiente de MyEclipse.

Los artefactos que se producen en cada modelo fueron trazados inicialmente por el guión de Áncora a Casos de Uso y mediante PUDS se realizaron los Casos de Uso a los diversos artefactos que poco a poco llevan a la implementación.

4. Desarrollo de HEDERS

En esta sección se muestran algunos artefactos creados en los diversos flujos fundamentales para desarrollar HEDERS.

4.1. Análisis de Requerimientos de HEDERS

Para poder tener asentadas todas las peticiones pertinentes, HEDERS indica paso a paso la realización del documento de Especificación de Requerimientos de Software, provocando que a lo largo del sistema los involucrados queden satisfechos con este acuerdo. Los requerimientos para esta herramienta son los siguientes:

1. Construir el documento de Especificación de Requerimientos de Software (ERS) de acuerdo a la estructura de la IEEE 830-1998 [4]. Esta parte es la principal dentro del desarrollo de este proyecto, ya que aquí se incluye la especificación de los requerimientos y podrá ser empleado por el cliente y por el desarrollador.
2. Importar elementos sobre requerimientos de otras herramientas desarrolladas en la UV y basadas en Áncora, como son: Redes Semánticas Naturales, Guiones-Áncora y Puntos de Función. En estos artefactos se encuentran datos importantes que constituirán el documento ERS.
3. Especificar los requerimientos dentro de la herramienta HEDERS clasificados de acuerdo a su tipo, los cuales van a ser estipulados en el documento de ERS.
4. Especificar otros elementos relacionados con el proyecto como: datos de la empresa, del cliente y de los usuarios.

En la Figura 1 se muestra el Guión-Áncora de la propuesta computacional de la herramienta HEDERS, en él se muestran los requerimientos funcionales y sus interacciones con el de esta herramienta en cada una de las escenas.

4.2. Caso de Uso que conforma el diseño de la herramienta

Para tener una visión más clara del diseño de la herramienta HEDERS, se muestra en la Figura 2, el diagrama general de casos de uso. Estos constituyen los requerimientos funcionales del sistema los cuales son: Capturar Proyecto, Importar requerimientos, Capturar datos del Cliente, Modificar Requerimientos, Generar documento ERS y Exportar reportes.

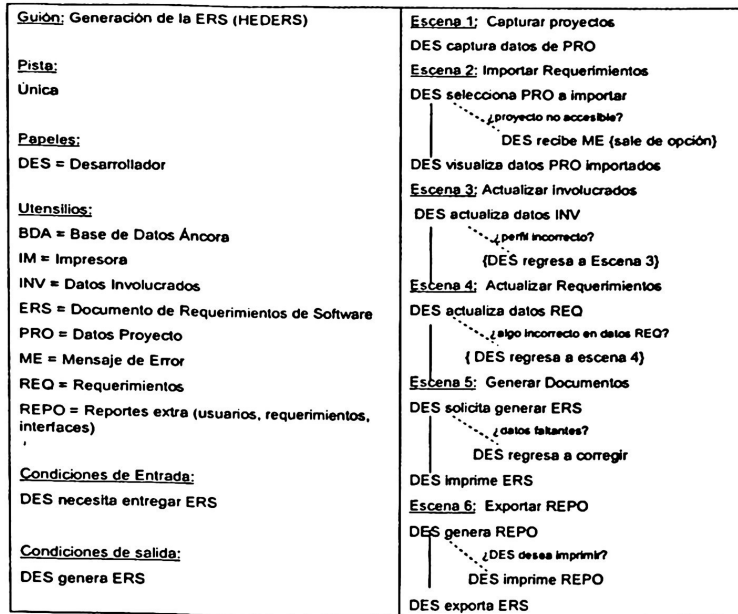


Fig. 1. Guión-Áncora de la propuesta computacional de HEDERS

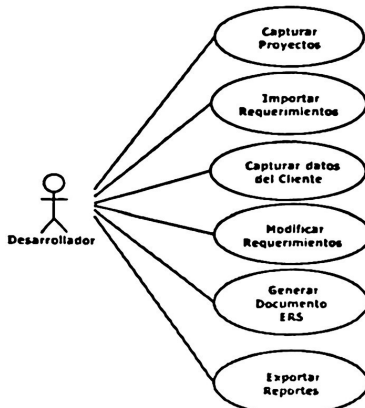


Fig. 2. Casos de Uso de HEADRES

4.3. Arquitectura de HEDERS

Para que esta herramienta logre un buen funcionamiento que ayude a muchas personas, se requiere de una adecuada estructura, además de un buen análisis. Ésto ayudará a crear documentos de Especificación de Requerimientos ERS, con un valor de calidad al momento de realizarlo. En consecuencia, esta herramienta cuenta con la siguiente estructura y se representa mediante un diagrama de cuatro capas en donde se muestra la estructura, (ver Figura 3).

En la Figura 3 se muestra un bosquejo importante de las principales capas de esta herramienta. En la parte superior la capa específica de aplicación en esta contiene la parte principal del sistema que es el documento de Especificación de Requerimientos de Software.

En la capa general de la aplicación se muestra las aplicaciones que interactúan con el sistema, incluyendo guiones, gestión de requerimientos, Puntos de Función, Usuarios y BDANCORA, los cuales son componentes reutilizables dentro de este sistema.

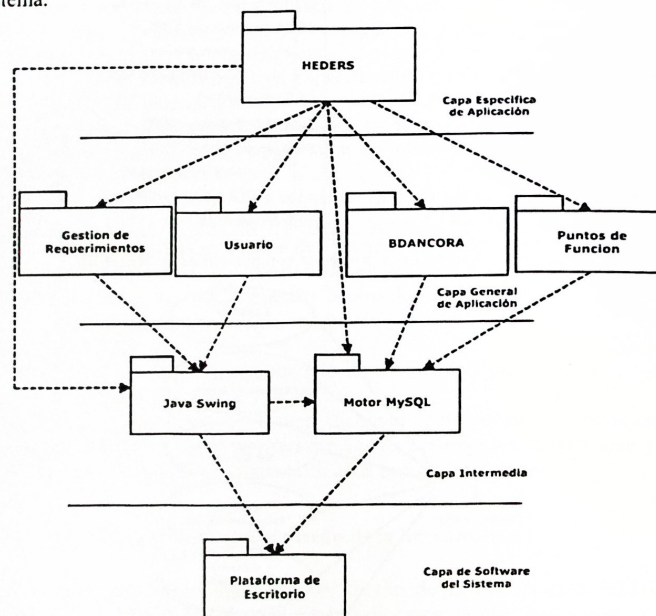


Fig. 3. Arquitectura de HEDERS

4.4. Realización del Caso de Uso Generar Documento ERS

El diagrama de secuencia ayuda a establecer la interacción entre las clases que realizan un caso de uso, es decir del funcionamiento de una parte de la herramienta. En la Figura 4 se muestra el diagrama de secuencia del Caso de Uso: Generar documento ERS.

Flujo de Sucesos del Diagrama: El desarrollador da un clic en botón importar de HEDERS y selecciona proyecto que desea importar; se realiza la conexión con la base de datos del proyecto y se ejecutan las operaciones SQL que llevan a cabo la consulta y se devuelven los resultados de la operación; se integra y se visualiza información al documento ERS y, si falta información del documento ERS, se rellenan los campos faltantes; se guarda información en la BD y se manda un mensaje de información guardada; luego se presiona el botón generar Reporte, se pasan los parámetros y se genera reporte.

Requerimientos no funcionales del Caso de Uso Generar documento ERS: Los requerimientos no funcionales o restricciones asociadas a este caso de uso son:

1. Seguridad en la importación de la información. Es decir, sólo la persona que está registrada en el sistema HEDERS como responsable del proyecto puede generar el ERS
2. Velocidad de respuesta. Se espera que el tiempo de respuesta para generar el ERS no sobrepase los cinco minutos para sistemas de hasta veinte casos de uso.
3. Bloqueo de impresora. No se permitirá la impresión de documentos en cola hasta que se haya impreso la ERS.
4. Fácil uso de sistema. Se creará una interfaz basada en ventanas gráficas con datos asociados por sección del estándar.

5. Implementación de HEDERS

Para implementar HEDERS Se desarrolló un prototipo con elementos como los que siguen:

Los requerimientos funcionales. En la Figura 5, se muestran la ventana inicial de la GUI de HEDERS, donde además de pedir los datos del proyecto a analizar se muestran las funciones que conforman HEDERS.

1. Los requerimientos funcionales. En la Figura 5, se muestran la ventana inicial de la GUI de HEDERS, donde además de pedir los datos del proyecto a analizar se muestran las funciones que conforman HEDERS.
2. Las interfaces con otras componentes. Dado que HEDERS es sólo un componente más de un proyecto global de apoyo a desarrolladores de software, se tuvo que programar en Java y respetando las características de interfaz establecidas. Cabe aclarar que, gracias a esta interacción, se reduce el llenado de los elementos que conforman el documento de ERS, que es extenso y tedioso, mediante la importación de artefactos de otras componentes. Sin embargo, esta operación complicó la implementación de HEDERS, pues deben respetarse los estándares de datos establecidos.

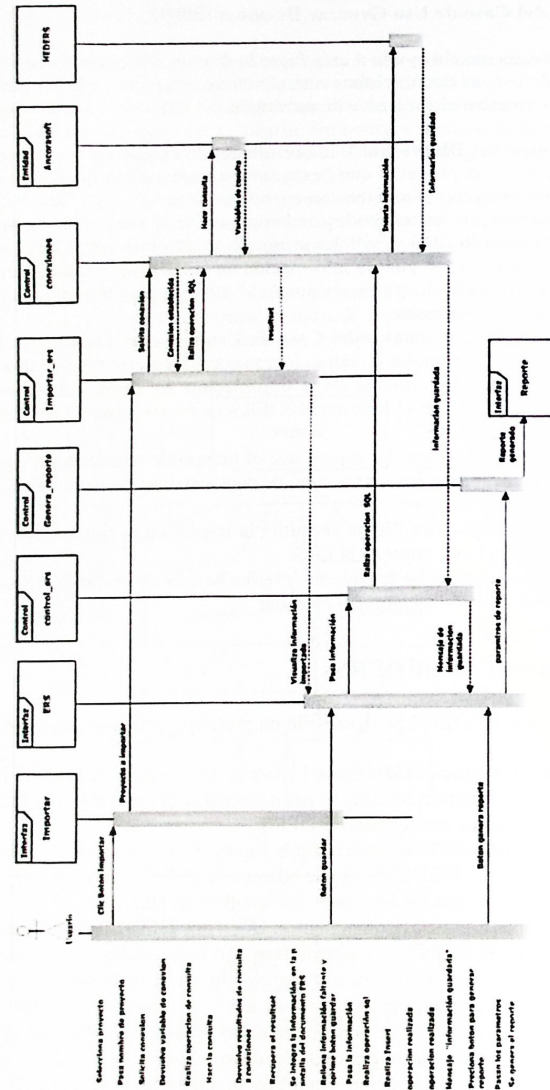


Fig. 4. Generar documento ERS de HEDERS

3. Además, pensando en que HEDERS también puede correr de forma aislada se consideraron ventanas para la captura de los datos, como la que se muestra en la Figura 6.

6. Conclusiones

Construir el documento de Especificación de Requerimientos de Software (ERS) es una tarea cansada y lenta, que se ve agravada por el poco tiempo y presupuesto que se le asigna a esta parte del desarrollo de software.

Así pues, contar con una herramienta como HEDERS que facilite su elaboración resulta una gran ventaja.

Al término de la escritura de este artículo se lleva un 80% del desarrollo de HEDERS y se alcanzan a vislumbrar sus ventajas, entre las que están: se desarrolla como una herramienta de uso libre y gratuito, facilitará la ERS y con ello se garantizará el establecer desde un principio una base sólida de entendimiento sobre lo que será el nuevo software entre cliente y desarrollador.

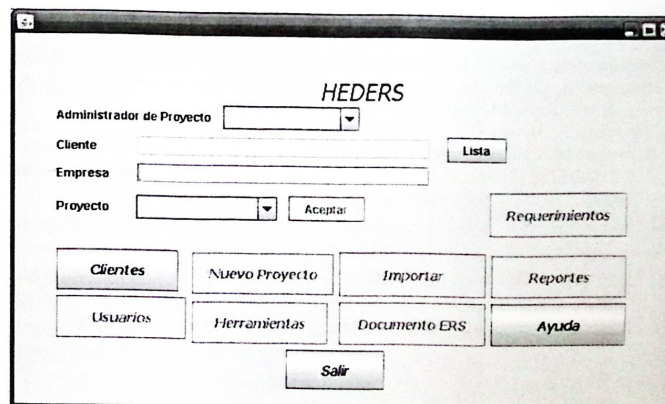


Fig. 5. Ventana inicial de la GUI de HEDERS

The image shows a screenshot of a software requirements template form. The form is divided into several sections, each with a heading and a description. The sections are: INTRODUCTION, PROPÓSITO, ALCANCE, DEFINICIONES, ACRÓNIMOS Y ABBREVIATURAS, and REFERENCIAS. The form includes various input fields, checkboxes, and a table for references. The form is titled 'PARTE 1' and 'PARTE 2'.

INTRODUCCIÓN
 Descripción:

PROPÓSITO
 Se requiere un sistema que permita la gestión de personal de DETEC.

ALCANCE
 El sistema:

DEFINICIONES, ACRÓNIMOS Y ABBREVIATURAS

ACRÓNIMO	DEFINICIÓN
DET	Detec
DET	Detec

REFERENCIAS BIBLIOGRÁFICAS

Nº REFERENCIA	TÍTULO	PAÍS	FECHA	AUTOR
1				
2				
3				
4				

REFERENCIAS BIBLIOGRÁFICAS

1. Sommerville I. (2005). Ingeniería de Software (7ª. Edición.). Madrid: Addison Wesley.

2. Sumano, A. (2006). Ancora: Análisis de requerimientos de software conducente al re- uso de artefactos. México: Universidad Veracruzana.

3. Jacobson I., Booch G., Rumbaugh J. (1999). El proceso unificado de desarrollo de Software. Madrid: Addison Wesley.

4. IEEE Std 1223 (1998). IEEE Recommended practice for Software Requirements Specifications

Fig. 6. Captura de la sección 1 del estándar

Referencias

1. Sommerville I. (2005). Ingeniería de Software (7ª. Edición.). Madrid: Addison Wesley.
2. Sumano, A. (2006). Ancora: Análisis de requerimientos de software conducente al re- uso de artefactos. México: Universidad Veracruzana.
3. Jacobson I., Booch G., Rumbaugh J. (1999). El proceso unificado de desarrollo de Software. Madrid: Addison Wesley.
4. IEEE Std 1223 (1998). IEEE Recommended practice for Software Requirements Specifications