



Universidad de las Ciencias Informáticas  
Facultad 4

## Desarrollo del Subsistema Ayuda para la Plataforma ZERA

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autores:**

Suramis Conte Torres  
Rolando Vázquez Escuela

**Tutoras:**

Ing. Yuneikys Recio Miranda  
Msc. Yaillet Martínez Pérez

La Habana, 20 de Junio de 2011



## Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro Fortes de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Suramis Conte Torres

Autora

\_\_\_\_\_  
Rolando Vázquez Escuela

Autor

\_\_\_\_\_  
Ing. Yuneikys Recio Miranda

Tutora

\_\_\_\_\_  
Msc. Yaillet Martínez Pérez

Tutora



## Dedicatoria

### **De Suramis:**

*A mis abuelitos: Angarica y Luis D. (mi socio). Mi corazón es su casa para siempre.*

### **De Rolando:**

*Dedico este Trabajo de Diploma:*

*A la persona que ha sido capaz de darlo todo sin recibir nada. A ti que has mantenido la confianza en mí cuando todos los demás la han perdido. **A ti mami.***

*A mi papá por su ejemplo y sacrificio, al que siempre he admirado y respetado.*

*A mi hermana Diany para que se inspire y llegue a donde he llegado.*

*A mis abuelos que también se merecen este triunfo porque han sido como mis padres. Muy especialmente a la memoria de mi abuelo Vázquez. Donde quiera que estés siéntete tan orgulloso de tu nieto como lo estuviste de tu hijo.*



## Agradecimientos

### **Suramis y Rolando:**

*Gracias a todos los que aportaron ideas y opiniones para el desarrollo de este trabajo.*

*A Yaillet y Yune por confiar en nosotros desde el primer momento y mantenerse en el barco, aun cuando todos creían que se hundiría. ¡Gracias por todo!*

*A Raúl y Anisbert por las horas dedicadas a mejorar la calidad de nuestras ideas.*

### **De Suramis:**

*Tata sé que soy difícil de entender y sobrellevar en ocasiones, pero quiero que sepas que siempre te estaré agradecida por muchas cosas: darme la vida, hacerme la mujer que soy y sobre todo por apoyarme siempre (aun cuando yo misma no lo noto).*

*A mi tía Rosita, muchas gracias por ser mi lucecita cuando todo lo demás se me hace tinieblas. Gracias por estar junto a mí en todas mis locuras.*

*A tita Silvia, mi segunda mami, también muchas gracias. Contigo he aprendido mucho y he vivido suficiente para saber que eres de oro. ¡Juntas llegaremos lejos!*

*Al negro Adonis que ha estado conmigo, incluso estando lejos, en los caminos enredados de mi vida. Gracias por darme recuerdos tan bonitos de la amistad, por ser el hermano que no he tenido y por todo el esfuerzo que has puesto en hacer pensar mi cabecita loca.*

*A las mimis especiales de mi vida en la escuela y que seguirán conmigo hasta el fin, las quiero mucho a todas. Gracias por estar ahí a Yerisilla, Moni y Nani.*



*A los titos Emily y Rober por ser super guays conmigo. Gracias por su preocupación, las terapias y todo lo que me han enseñado.*

*A mi compañero de tesis que ha subido conmigo las escaleras del destino, manteniendo fuerzas para crear cosas buenas y diseñar otras mejores. Gracias por tenerme paciencia y no abandonar la lucha Roly.*

*Siempre que te pregunten si puedes hacer un trabajo, contesta que sí y ponte enseguida a aprender cómo se hace. No solo con la tesis, hasta conmigo has tenido la voluntad de entender, comprender y querer. Gracias por aceptarme con todas mis “cosillas” cariño Juan Luis.*

*Nunca encontrarás tiempo para nada. Debes crearlo. Gracias por crear ese tiempo para mí y por encontrar energías al final del día para asistir este trabajo a Josefina, Hino y Marta.*

### **De Rolando:**

*El éxito no significa nada si no hay nadie con quien compartirlo es por eso que deseo expresar lo que mi corazón siente. Gracias a la vida, a la Revolución por darme la oportunidad de estudiar, a mis amigos que más quiero y a toda mi familia de manera general, pero muy en especial a:*

*A mi mamá: En el cielo hay una Estrella que presume de bonita pero estando tu a su lado la dejas muy pequeñita. Eres la única persona del mundo que siempre está a mi lado, de forma incondicional no importa la distancia. Si te rechazo, me perdonas. Si me equivoco, me acoges. Si estoy feliz, celebras conmigo. Tu fuerza y tu amor me guiaron hasta aquí y gracias a ti hoy llegué hasta donde estoy.*

*A mi papá: Personas que interfieran en la vida de uno, hay muchas, pero padres no hay más que uno, y tú has sabido ser el mejor entre todos ellos a pesar de todo.*

*A Luga: por ayudarme tanto en la vida y por preocuparse por mí, te quiero como a una madre más.*



*A tí: Cuando todo me parece imposible, horrible e inalcanzable, pienso en tu sonrisa y la fuerza que me da, y de repente todo es posible. Gracias por todo. Estaría perdido sin ti.*

*A todos los que me apoyaron en cualquier momento sin pedir nada a cambio. Si no fuera por ustedes mi sueño no lo habría cumplido, esta tesis es también un logro de Uds. Reciban todos mis más modestos y sinceros agradecimientos. Muchas gracias a...*

*A mis amigos: pues cuando me dolía mirar hacia atrás y me daba miedo mirar adelante, miraba hacia mi izquierda o mi derecha y allí estaban ustedes, a mi lado: Raúl, Alexy, Robert, Mónica, Daily, Yula. Gracias por su apoyo y amistad. Sé que aguantarme a lo largo de estos 5 años no ha sido nada fácil.*

*A mi Gente de Zona: Anisbert, Ginita, Luis Raciél, Maité, Adrianet, Manuel, Celia, Yoisy.*

*Quiero agradecer a todas las personas que de una manera u otra han contribuido a mi formación y preparación como ingeniero. A los Profes: Alexy, Lesyanis, Jorge Alberto, Mayi, Sasha, Reynier Cartaya y Greisy. La enseñanza que deja huella no es la que se hace de cabeza a cabeza, sino de corazón a corazón. Uds. dejaron esa huella.*

*A Yusniel, Raúl, Anisbert, Josefina, Yaismel, Adrián por las ideas, opiniones, críticas y el apoyo brindado en el desarrollo de la tesis.*

*A Suri: por su inmensa paciencia conmigo y por haberse convertido en la amiga que hoy es. No ha sido fácil llegar, se ha necesitado ahínco, lucha y deseo, pero sobre todo apoyo como el que he recibido de ti durante este tiempo. Gracias por lo que hemos logrado.*



## Resumen

Hoy en día la educación ha incorporado el uso de las Tecnologías de la Información y las Comunicaciones a los procesos de enseñanza y aprendizaje, dando lugar al surgimiento y aplicación de los Entornos Virtuales de Aprendizaje. Estos sistemas resultan tan amplios que mayormente se requiere un estudio previo por el usuario para explotar al máximo sus funcionalidades. Al incorporar sistemas de Ayuda a los mismos se garantiza que sus consumidores puedan consultar y aclarar sus dudas en cualquier momento durante su interacción con el software. El trabajo abarca el proceso de desarrollo de un sistema de Ayuda que será utilizado en la plataforma educativa ZERA y permitirá que esta posea elementos de asistencia informativa y personalizada para todos sus usuarios. Para la solución se crea un plugin que encapsule las clases, archivos de configuración y hojas de estilos CSS; que son de uso común para todos los subsistemas de dicha plataforma a los cuales se les incluye la Ayuda. Se desarrolla un sistema para la administración de los contenidos de Ayuda, que es para uso exclusivo del equipo de desarrollo. Para garantizar la calidad del producto entregado, este fue sometido a un proceso de pruebas para la validación de sus funcionalidades.

Palabras clave: asistencia, Ayuda, educación, plataforma ZERA, plugin, administración.



## Índice

<b>Introducción</b> .....	<b>1</b>
<b>Capítulo 1: Fundamentación Teórica</b> .....	<b>5</b>
1.1. Introducción .....	5
1.2. Conceptos asociados al dominio del problema .....	5
1.2.1. Software Educativo .....	5
1.2.2. Plataformas Educativas .....	7
1.3. La Ayuda .....	10
1.3.1. Tendencias actuales de las Ayudas para plataformas Educativas .....	12
1.4. Metodologías, herramientas y estándares de posible utilización como propuesta de modelación a utilizar en la solución.....	16
1.4.1. Metodologías de desarrollo de software.....	16
1.4.2. Herramientas para el modelado de sistemas .....	19
1.5. Herramientas para la confección del contenido del Subsistema Ayuda .....	20
1.6. Herramientas y lenguajes a utilizar en el entorno de desarrollo.....	21
1.6.1. Servidor Web .....	22
1.6.2. Sistemas Gestores de Bases de Datos .....	22
1.6.3. Lenguajes de programación para la web.....	24
1.6.4. Framework.....	26
1.6.5. Entornos de desarrollo integrados (en inglés IDE) .....	29
1.7. Conclusiones .....	31
<b>Capítulo 2: Características del Subsistema Ayuda de la Plataforma ZERA</b> .....	<b>32</b>
2.1. Introducción .....	32
2.2. Características de la plataforma para la gestión del aprendizaje ZERA.....	32
2.3. Planificación del Contenido a incluir en la Ayuda.....	33
2.4. Especificaciones acerca del Diseño Gráfico del Subsistema Ayuda .....	33
2.5. Organización del Contenido .....	34
2.5.1. Formato y presentación de las imágenes adjuntas al contenido .....	34





2.5.2. Redacción e integración del contenido de la Ayuda .....	35
2.5.3. Descripción de las pautas de formato y redacción para el contenido de la Ayuda.....	36
2.6. Modelo de dominio.....	36
2.7. Especificación de los requerimientos del Subsistema Ayuda.....	37
2.7.1. Requerimientos Funcionales.....	38
2.7.2. Requerimientos No Funcionales .....	38
2.8. Modelo del Sistema.....	39
2.8.1. Actores .....	39
2.8.2. Casos de Uso .....	40
2.9. Conclusiones .....	46
<b>Capítulo 3: Análisis y Diseño del Subsistema Ayuda para la Plataforma ZERA.....</b>	<b>47</b>
3.1. Introducción .....	47
3.2. Modelo de Datos.....	47
3.2.1. Descripción de las entidades presentes en el Modelo Entidad – Relación .....	47
3.3. Modelo de análisis .....	48
3.3.1. Descripción de las clases del análisis .....	48
3.3.2. Diagramas de interacción del análisis .....	49
3.4. Modelo de diseño.....	49
3.4.1. Patrones utilizados.....	49
3.4.2. Descripción de las clases del diseño.....	50
3.4.3. Diagramas de interacción del diseño .....	51
3.5. Conclusiones .....	51
<b>Capítulo 4: Implementación y Pruebas del Subsistema Ayuda para la Plataforma ZERA .....</b>	<b>52</b>
4.1. Introducción .....	52
4.2. Flujo de Trabajo de Implementación.....	52
4.2.1. Diagrama de componentes .....	52
4.2.2. Diagrama de despliegue .....	53
4.2.3. Plugin sfAOHelpPlugin.....	54
4.2.4. Sistema para la Administración de la Ayuda .....	56
4.3. Pruebas de Software.....	56



---

4.3.1. Descripción de los Casos de Prueba del Subsistema Ayuda .....	57
4.3.2. Resultados de las validaciones realizadas .....	61
4.4. Conclusiones .....	61
<b>Conclusiones .....</b>	<b>63</b>
<b>Recomendaciones .....</b>	<b>64</b>
<b>Glosario de Términos .....</b>	<b>65</b>
<b>Referencias Bibliográficas .....</b>	<b>66</b>



## Índice de Figuras

<b>Figura 1. Artefactos Generados y Tareas del Rol Escritor Técnico en RUP.¡Error! Marcador no definido.</b>	
<b>Figura 2. Modelo del dominio.....</b>	<b>37</b>
<b>Figura 3. Diagrama de Casos de Uso del Subsistema Ayuda.....</b>	<b>40</b>
<b>Figura 4. Diagrama Entidad – Relación de la Base de Datos del Subsistema Ayuda. ....</b>	<b>48</b>
<b>Figura 5. Diagrama de clases del análisis del caso de uso Administrar Ayuda.¡Error! Marcador no definido.</b>	
<b>Figura 6. Flujo de trabajo de Symfony..... ¡Error! Marcador no definido.</b>	
<b>Figura 7. Diagrama de clases del análisis del caso de uso Administrar Ayuda.¡Error! Marcador no definido.</b>	
<b>Figura 8. Diagrama de Componentes del Subsistema Ayuda.....</b>	<b>53</b>
<b>Figura 9. Diagrama de Despliegue del Subsistema Ayuda. ....</b>	<b>54</b>
<b>Figura 9. Gráfica de las No Conformidades detectadas en las pruebas funcionales.¡Error! Marcador no definido.</b>	

## Índice de Tablas

Tabla 1. Descripción de los actores del Subsistema Ayuda. ....	39
Tabla 2. Descripción textual del Caso de Uso Administrar Ayuda. ....	41
Tabla 3. Descripción del flujo básico de eventos de la sección Incluir Contenido del CU Administrar Ayuda.....	41
Tabla 4. Descripción del flujo alternativo de eventos de la sección Incluir Contenido del CU Administrar Ayuda.....	42
Tabla 5. Descripción del flujo básico de eventos de la sección Ver Datos de Contenido del CU Administrar Ayuda. ....	43
Tabla 6. Descripción del flujo alternativo de eventos de la sección Ver Datos de Contenido del CU Administrar Ayuda.....	44
Tabla 7. Descripción del flujo básico de eventos de la sección Modificar Datos de Contenido del CU Administrar Ayuda. ....	44
Tabla 8. Descripción del flujo alternativo de eventos de la sección Modificar Datos de Contenido del CU Administrar Ayuda. ....	45
Tabla 9. Descripción del flujo básico de eventos de la sección Eliminar Contenido del CU Administrar Ayuda. ....	45
Tabla 10. Descripción del flujo alternativo de eventos de la sección Eliminar Contenido del CU Administrar Ayuda. ....	45
Tabla 11. Escenario Incluir Contenido del Caso de Prueba Administrar Ayuda. ....	57
Tabla 12. Escenario Ver Contenido del Caso de Prueba Administrar Ayuda.....	59
Tabla 13. Escenario Modificar Contenido del Caso de Prueba Administrar Ayuda.....	59
Tabla 14. Escenario Eliminar Contenido del Caso de Prueba Administrar Ayuda.....	60



## Introducción

La rápida evolución de las Tecnologías de la Información y las Comunicaciones (TIC) ha traído consigo un aumento en la productividad de las áreas donde estas se aplican, optimizando los procesos que se llevan a cabo en las diferentes esferas. La educación es uno de los sectores que recientemente se ha incorporado al uso de las TIC; permitiendo a las personas mediante la comprensión de las nuevas tecnologías, entender el mundo en que viven, aumentar sus conocimientos y su capacidad intelectual. El desarrollo de las TIC representa una mejora en la calidad de la educación al revolucionar las metodologías, formas de organización y métodos de enseñanza-aprendizaje. Al incorporar tecnología de punta a la educación se enriquece la pedagogía y surge un nuevo tipo de software: el software educativo.

Este tipo de software está siendo aceptado rápidamente en los centros educacionales donde es puesto en práctica debido a su amplia respuesta a los requisitos del sistema educacional. En Cuba se ha desarrollado una fuerte industria de producción de software educativo para todas las edades, contando con colecciones de libros digitales, multimedias y plataformas para la gestión del aprendizaje.

Uno de los objetivos principales de cualquier software educativo que se desarrolla es tener una buena arquitectura de información y una interfaz clara y agradable para el usuario; así como también, un sistema de Ayuda adicional para proporcionar la información y asistencia necesaria al usuario en el proceso de enseñanza-aprendizaje. Inclusive si las interfaces son simples y claras, la existencia de usuarios con distintos niveles de conocimiento y diferentes objetivos hacen necesario el añadir sistemas de apoyo para proveerles asistencia.

En Cuba, la Universidad de las Ciencias Informáticas (en lo adelante UCI) tiene como misión producir software y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación. Desde su surgimiento en el año 2002, la UCI ha jugado un rol fundamental en el desarrollo de la informatización de la sociedad cubana y se encuentra entre las primeras que hoy se dedican a la producción de aplicaciones con tecnología multimedia como apoyo a la docencia; muchas de ellas para la propia universidad. La plataforma para la gestión del aprendizaje ZERA desarrollada en el centro Fortes<sup>1</sup>

---

<sup>1</sup> Centro de Tecnologías para la Formación.



por el proyecto Alfaomega, está dirigida a usuarios que de manera general poseen básicos conocimientos de los conceptos informáticos. Esta plataforma posee una interfaz gráfica muy novedosa y diferente a las de su tipo en el mundo, por lo que pudiera convertirse en una dificultad el entendimiento de los íconos y la navegabilidad a través de la misma. Esto puede ocasionar que el usuario deba dedicar mucho tiempo al aprendizaje previo del uso y las funcionalidades más complejas de la plataforma; o deba limitarse a usar solo un pequeño porcentaje de su potencial. Por lo que se hace necesario que los usuarios, independientemente de su clasificación, obtengan un apoyo que les permita interactuar de forma inmediata con el software y esclarecer sus dudas. Debido a esto se requiere que el proyecto cuente con los elementos de ayuda informativa y personalizada para los usuarios de la plataforma.

Dicha ayuda permitirá aminorar la problemática anterior y facilitaría el acceso de todos los usuarios a la información de una manera más eficiente disminuyendo así el tiempo de búsqueda de la misma. Por todo lo anteriormente expuesto se plantea el siguiente **problema a resolver**: ¿Cómo garantizar que los usuarios de la plataforma ZERA cuenten con elementos de asistencia informativa y personalizada de manera que facilite su interacción con el sistema?

Como **objeto de estudio** se tiene a los Sistemas de Ayuda y se define como **objetivo general** desarrollar un subsistema de ayuda para la plataforma ZERA que facilite la interacción del usuario con la aplicación. Como **campo de acción** se presenta la Ayuda de la plataforma ZERA.

En apoyo a la investigación del presente trabajo se trazan los siguientes **objetivos específicos**:

- ☑ Elaborar el marco teórico de la investigación.
- ☑ Analizar y diseñar el proceso de desarrollo de la Ayuda para la plataforma ZERA.
- ☑ Elaborar la propuesta de solución para la plataforma ZERA.
- ☑ Validar el subsistema de Ayuda desarrollado.

Para dar cumplimiento a los objetivos planteados se determinan las siguientes **tareas**:

- ☑ Realización de un estudio del estado del arte sobre las tendencias actuales y estándares de los Sistemas de Ayuda en la Industria del Software en general.
- ☑ Realización de un estudio y comparación de formato, presentación, navegación, métodos de acceso y herramientas utilizadas en Sistemas de Ayuda de software similares.



- ✓ Descripción y selección de las metodologías, tecnologías, herramientas y estándares para el desarrollo de la Ayuda.
- ✓ Realización de un estudio exhaustivo de la arquitectura de la plataforma ZERA.
- ✓ Elaboración de los artefactos para modelar el subsistema de Ayuda según los estudios anteriormente realizados.
- ✓ Implementación del subsistema de Ayuda.
- ✓ Integración del subsistema de Ayuda con los diferentes subsistemas y módulos de la plataforma ZERA.
- ✓ Realización de pruebas al subsistema de Ayuda.

La **idea a defender** es: Con el desarrollo de un subsistema de Ayuda para la plataforma ZERA se facilitará la interacción del usuario con el sistema, contando así con elementos de asistencia informativa y personalizada.

Los métodos científicos que se usarán en la investigación serán:

- ✓ Histórico-Lógico: en el estudio de la evolución de los conceptos, estándares y herramientas para la selección de cuáles utilizar.
- ✓ Analítico Sintético: para realizar el análisis y la síntesis de los conceptos, estándares y herramientas en la investigación sobre los Sistemas de Ayuda en uso actualmente.
- ✓ Inductivo-Deductivo: Se utilizará en la descripción de las funcionalidades.
- ✓ Observación: A través de este método se domina la esencia de la problemática definida, además permite conocer el proceso definido como objeto de estudio, lo cual influye a la hora de tener un conocimiento más detallado de lo que se quiere, lo que hace falta hacer y cómo se hace.
- ✓ Entrevista: Se utilizará a través de diálogos con el cliente para obtener información sobre los requisitos para el diseño del Subsistema Ayuda.

La estructura del documento es de 4 capítulos, incluyendo introducción, conclusiones, recomendaciones, anexos, referencias bibliográficas distribuidos de la forma siguiente:

**Capítulo 1:** Fundamentación Teórica.



Se realiza una descripción de los conceptos fundamentales relacionados con los Sistemas de Ayuda para Software, también se exponen los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. Se presenta una descripción, análisis y selección de las herramientas utilizadas para el diseño y desarrollo de los Sistemas de Ayuda en la actualidad.

### **Capítulo 2:** Características del Subsistema Ayuda de la plataforma ZERA.

En este capítulo se describe la situación actual que presenta la plataforma ZERA y se especifican los requisitos que debe cumplir el Subsistema Ayuda.

### **Capítulo 3:** Análisis y Diseño del Subsistema Ayuda para la plataforma ZERA.

Se detallan los pasos a seguir para comenzar el proceso de desarrollo del Subsistema Ayuda. Así como la estructura y organización del contenido de Ayuda, los responsables de su elaboración y las normas a seguir para la uniformidad de su formato. Se diseña la propuesta del Subsistema Ayuda.

### **Capítulo 4:** Implementación y Pruebas del Subsistema Ayuda para la plataforma ZERA.

En este capítulo se especifican los detalles de la implementación del subsistema, se realiza una validación de los requisitos a implementar y de las pautas de diseño de la Ayuda.





## Fundamentación Teórica

# 1 Capítulo

### 1.1. Introducción

En este capítulo se explican los conceptos asociados al dominio del problema que se deben conocer para un mejor entendimiento de la solución. Se expone una investigación sobre los Sistemas de Ayuda para plataformas educativas a nivel mundial y en Cuba. Se describen las metodologías y estándares de modelados escogidos como propuesta para modelar la solución. Se detallan las características de las metodologías y herramientas a utilizar en el modelado y desarrollo de Sistemas de Ayuda actualmente, señalando aquellas que son propuestas para su utilización en desarrollo de la Ayuda para la plataforma ZERA.

### 1.2. Conceptos asociados al dominio del problema

A continuación se analizan y detallan los dos conceptos fundamentales a estudiar en el dominio del problema a resolver. Se describe la evolución de la noción de software educativo, a la vez que se señalan las principales ideas asociadas a este. La definición de plataformas educativas se aborda desde un ángulo diferente, enumerando las distintas clasificaciones que presentan estos sistemas en la actualidad.

#### 1.2.1. Software Educativo

Luego de consultar y analizar criterios de varios autores en sus definiciones sobre software y software educativo, se citan las siguientes definiciones por ser las que mejor proyectan este concepto. Software es un conjunto de instrucciones que contienen las órdenes con las que trabaja la computadora y que le indican cómo manipular los datos, en general se puede decir que condiciona el funcionamiento del hardware. Está dividido en las categorías: sistema operativo, software de aplicación y lenguajes de programación. En la categoría de software de aplicación se encuentran aquellos que son diseñados para cumplir una tarea específica, ejemplo son los sistemas gestores de bases de datos, tratamiento de textos y software educativos, entre otros. (1)

*“El software educativo no es más que el conjunto de recursos informáticos diseñados con la intención de ser utilizados en el proceso de enseñanza-aprendizaje.”* (2) Es una herramienta didáctica muy joven y que



resulta muy útil tanto para el estudiante como para el profesor, pues permite controlar las tareas docentes (individuales o colectivas) además de constituir una nueva, atractiva y dinámica fuente de conocimientos. Se caracterizan por ser altamente interactivos a partir del empleo de recursos multimedia que apoyan las funciones de evaluación y diagnóstico. Los recursos empleados se clasifican en:

**Multimedia:** Son todos aquellos sistemas que utilizan más de un medio de comunicación al mismo tiempo en la presentación de la información, como el texto, la imagen, la animación, el video y el sonido. *“Este concepto es tan antiguo como la comunicación humana ya que al expresarnos en una charla normal hablamos (sonido), escribimos (texto), observamos a nuestro interlocutor (video) y accionamos con gestos y movimientos de las manos (animación).”* (3)

**Hipertexto:** Es un documento digital o no, que se puede leer de manera no secuencial o lineal de acuerdo con la necesidad establecida. Un hipertexto tiene los siguientes elementos: secciones, enlaces o hipervínculos y anclajes. Las secciones o nodos son los componentes del hipertexto o hiperdocumento. Los enlaces son las uniones entre nodos que facilitan la lectura secuencial o no secuencial del documento. Los anclajes son los puntos de activación de los enlaces. El hipertexto utiliza *“palabras calientes”* contenidas en el cuerpo del documento para acceder a otros textos relacionados con dicha palabra. Así se controla el orden de lectura y la aparición de los datos en la pantalla de una manera más parecida a nuestro modo de relacionar pensamientos. Si algo caracteriza al hipertexto es el dinamismo y la interactividad que conlleva frente a lo estático del texto impreso ya que permite navegar por un entramado de nodos, de acuerdo a las necesidades que se tengan en un momento dado. (3)

**Hipermedia:** Combinación de los conceptos de hipertexto y multimedia. Tecnología de construcción de documentos con facilidades de búsqueda de la información necesitada. A través de enlaces entre los diferentes elementos de información multimedia, dígase texto, audio, video, gráfico, entre otros. Un documento hipermedia es siempre una multimedia, pero no al revés. Cuando se está en presencia de un documento que brinde la posibilidad de usar interconexiones para moverse y localizar la información por el mismo, entonces se está frente a un documento hipermedia. (3)

La hipermedia conjuga los beneficios de las tecnologías hipertexto y multimedia. Mientras que la multimedia proporciona una gran riqueza en los tipos de datos, dotando de mayor flexibilidad a la expresión de la información, el hipertexto aporta una geometría que permite que estos datos puedan ser explorados y presentados siguiendo diferentes secuencias, de acuerdo con las necesidades del usuario. Siguiendo con el mismo ejemplo, el hiperdocumento estaría diseñado de forma que cada nodo sería una



presentación multimedia que incluiría enlaces a conceptos relacionados. De esta manera, el usuario podría disfrutar de la secuencia de lectura más apropiada, navegando de una forma sencilla y rápida, sin tener que preocuparse dónde se encuentra esa información.

Con el surgimiento de internet, el software educativo se abrió camino hacia muchas áreas que hasta el momento habían permanecido sin explotar. La red se presenta como una nueva herramienta metodológica y didáctica en el proceso educativo, dando lugar al surgimiento de nuevos conceptos y formas de modelar dicho proceso que se fueron adoptando en todo el mundo. Uno de esos conceptos es e-learning. Entiéndase como e-learning la formación a distancia a través de las TIC, en concreto Internet. (4) Facilita la creación, adopción y distribución de contenidos, la adaptación del ritmo de aprendizaje, la disponibilidad de las herramientas de aprendizaje independientemente de límites horarios o geográficos y el intercambio de opiniones y aportes a través de la plataforma. (5)

En este tipo de enseñanza se evidencia como característica el uso masivo de los medios técnicos y que el alumno se centra en una formación independiente y flexible. El aprendizaje virtual de conjunto con los objetos de aprendizaje<sup>2</sup> representa la última ampliación del concepto de software educativo, mayormente apreciado en las plataformas educativas.

### 1.2.2. Plataformas Educativas

El software educativo tiene rasgos esenciales básicos y una estructura general común, lo que no impide que tengan también características muy diversas. Hasta finales del pasado siglo existían dos tipos de aplicaciones educativas: algorítmicas (sistemas tutoriales, sistemas entrenadores y libros electrónicos) y heurísticas (simuladores, juegos educativos y sistemas expertos). A principios del presente siglo surge una nueva tendencia: integrar en un mismo sistema todas o algunas de las tipologías antes mencionadas. A este nuevo modelo de software se le denomina Hiperentorno Educativo o Hiperentorno de Aprendizaje, también conocidos como Entornos Virtuales de Enseñanza y Aprendizaje (EVEA). Son aplicaciones desarrolladas sobre los protocolos de red e incluyen herramientas para cubrir las necesidades del centro educativo que las implemente. Estos sistemas reciben el nombre de plataformas y actualmente algunas de

---

<sup>2</sup> Los objetos de aprendizaje son recursos digitales que se pueden reutilizar en diferentes contextos para lograr un objetivo de aprendizaje particular, en donde los estudiantes los utilizan para generar nuevo contenido educativo.



ellas están estandarizadas mientras que otras son completamente personalizadas, existiendo la posibilidad de comercializar ambas.

Los EVEA, están basados en el principio de aprendizaje colaborativo y se apoyan de herramientas multimedia para hacer más dinámico el proceso de enseñanza y aprendizaje. Pasando de ser simplemente un texto en línea a un entorno interactivo que permite gestionar contenidos educativos. Luego de consultar diferentes fuentes acerca de los sistemas que permiten gestionar distintos tipos de contenido, se definen los siguientes conceptos siguiendo solamente los criterios que mejor los ilustran.

#### **1.2.2.1. Sistema de Gestión de Contenido (por sus siglas en inglés CMS)**

Un Sistema de Gestión de Contenido permite la creación y administración de contenidos principalmente en páginas web. Consiste de una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de forma independiente el contenido por una parte y el diseño por otra. Existen varios tipos diferentes de CMS agrupados en las categorías:

- ☑ Foros (sitio que permite el debate en línea, donde los usuarios pueden reunirse y discutir temas en los que están interesados).
- ☑ Blogs (publicación de noticias o artículos en orden cronológico con espacio para comentarios y discusión).
- ☑ Wikis (sitio web donde todos los usuarios pueden colaborar en los artículos, los cuales permiten espacios para discusiones).
- ☑ e-Commerce (sitio web para comercio electrónico).
- ☑ Sitios web (sitio web con contenido y funcionalidad diversa que sirve como fuente de información o como soporte a una comunidad).
- ☑ Galería (permite administrar y generar automáticamente un portal o sitio web que muestra contenido audiovisual). (6)

#### **1.2.2.2. Sistema de Gestión del Aprendizaje (por sus siglas en inglés LMS)**

Sistema de Gestión del Aprendizaje o Learning Management System (LMS) es una aplicación que se ubica en un servidor y está diseñada para administrar, controlar y distribuir diferentes actividades de formación presencial o de educación virtual. Estos sistemas son diferentes entre ellos, pues cada uno es diseñado para cubrir las necesidades de la organización que lo utilizará. El objetivo principal de un LMS es



dar seguimiento al aprendizaje, participación y desempeño de los estudiantes. Los LMS presentan funciones como: registro de usuarios, organización de los diferentes cursos en un catálogo y almacenamiento de datos sobre los usuarios. (7)

Provee al educador de un mecanismo para distribuir contenido, monitorear la participación de los estudiantes y evaluar su desempeño. También ofrecen a los estudiantes el uso de los mecanismos de interacción como foros de discusión, videoconferencias o servicios de mensajería instantánea. Generalmente no incluyen posibilidades de crear contenido educativo propio, sino que gestiona contenidos creados por diferentes fuentes. Trabaja con bases de datos para la gestión de usuarios así como para gestionar y planificar el aprendizaje. Existen sistemas de gestión de aprendizaje en línea de código abierto, que son muy populares en el mundo y están disponibles para cualquier organización que desee utilizarlos; aunque en su mayoría los LMS son sistemas propietarios comercializados.

### **1.2.2.3. Sistema de Gestión de Contenidos de Aprendizaje (por sus siglas en inglés LCMS)**

Un Sistema de Gestión de Contenidos de Aprendizaje (también conocido por Learning Content Management System, LCMS) es independiente o integrado con el LMS, que gestiona y administra los contenidos educativos. Una vez que los contenidos están incorporados en el LCMS ya pueden ser combinados, asignados a distintos cursos, descargados desde el archivador electrónico, etc.

Los LCMS pueden ser considerados como plataformas que incorporan la gestión de contenidos para personalizar los recursos a cada alumno y añaden técnicas de gestión del conocimiento al modelo LMS. Son ambientes estructurados diseñados para que las organizaciones puedan implementar mejor sus procesos y prácticas con el apoyo de cursos, materiales y contenidos en línea. Permiten una creación mucho más eficiente, evitan redundancia y permiten administrar también la participación de diversos desarrolladores, expertos colaboradores o instructores que estén involucrados en la creación de contenidos. El contenido creado es reutilizable a lo largo de cursos, currículos y transferible entre organizaciones; así como se puede publicar en diversos formatos a la vez que se almacena en una base de datos centralizada. (8)

Las plataformas educativas están especialmente diseñadas para facilitar la gestión de cursos virtuales a los profesores. Fueron desarrolladas originalmente para cursos a distancia y actualmente vienen siendo utilizadas como suplemento de cursos presenciales. Los servicios que normalmente incluyen son control de acceso, elaboración del contenido educativo, herramientas de comunicación, y la administración de



grupos de estudiantes y/o profesores. También presentan plantillas para la elaboración de foros, ejercicios de diversos tipos, cuestionarios y mensajería que luego serían habilitados para su uso por los estudiantes. Algunos sistemas recientes presentan blogs y lector de noticias Really Simple Syndication (RSS). Debido a la presencia de numerosas funcionalidades y para aumentar la facilidad de uso de los EVEA, sus creadores han incorporado Sistemas de Ayuda a los mismos.

### 1.3. La Ayuda

El término Ayuda visto en la informática refiere a aquellos sistemas diseñados para brindar asistencia en el uso de un software o sistema operativo, enfocando la información en las necesidades del usuario sin importar su nivel de experiencia. (9)

Los Sistemas de Ayuda en Línea son aplicaciones que brindan atención y solución a las necesidades de los usuarios al momento de estos interactuar con el software a través de la web. Tienen como objetivo principal ofrecer la información necesaria para que los usuarios estén en capacidad de utilizar en forma eficiente las potencialidades que les ofrezca dicha aplicación. Estos Sistemas de Ayuda en Línea son muy útiles a los usuarios de aplicaciones web que no están familiarizados con las mismas.

La Ayuda en Línea debe poseer simplicidad en la búsqueda de información y facilidad de acceso, esto permite que el intercambio del usuario con la aplicación sea intuitivo y brinde un entorno amigable de trabajo. La usabilidad que puedan tener estos sistemas permite que resulte eficaz para el usuario, además mejora el aprendizaje y efectividad de la aplicación que lo contiene.

Es necesaria una continua actualización del contenido de la ayuda para lograr una evolución conjunta con el software y sus usuarios. Se entiende por contenido todos aquellos fragmentos de información u objetos informativos a organizar, estructurar y clasificar, por ejemplo textos e imágenes. La selección de la información a incluir en la Ayuda debe ser mediante un proceso minucioso, pues esta debe ser demostrativa, significativa y útil para el usuario en su interacción con el software. (10)

La información a mostrar no debe ser en páginas extensas pues los usuarios tienden a desechar aquello que les resulta en extremo tedioso; aunque no es recomendable fragmentar demasiado el contenido porque esto resulta en demoras para acceder al mismo. Se propone utilizar pequeñas piezas de información para los enlaces posibilitando que el usuario encuentre datos puntuales relacionados con el tema de dicho enlace. Cuando la Ayuda está directamente ligada a las acciones del usuario se le llama Ayuda sensible al contexto.



La Ayuda sensible al contexto, en comparación con la Ayuda en Línea, no tiene necesariamente que estar disponible para su lectura total por los usuarios. Cada tema o característica del software es descrita y mostrada independientemente según las necesidades del usuario. Conociendo de qué trata un sistema de Ayuda Contextual el siguiente paso es dominar las características y requisitos que debe cumplir y que se analizarán a continuación:

- ☑ Se integran completamente en una aplicación determinada.
- ☑ Se centran en las tareas que permite realizar la aplicación.
- ☑ Permiten el acceso eficiente a la información asociada a las tareas.
- ☑ Permiten extraer y asimilar fácilmente la información, para aplicarla inmediatamente a la tarea.

Además, la interfaz de usuario juega un papel fundamental en estos asistentes, ya que influye de forma decisiva en la sencillez de su uso. (11)

Como la descripción de las características y requisitos ideales de los Sistemas de Ayuda Contextual es tan amplia, se expondrán brevemente algunas de las propiedades que habitualmente no poseen los sistemas de Ayuda comerciales. En primer lugar, no son sistemas tutoriales que explican interactivamente los conceptos funcionales básicos de las aplicaciones. No son manuales textuales de descripción de aplicaciones, por lo que no puede esperarse de ellos que sirvan como introducciones de manejo de las aplicaciones a las que asisten. Estos tampoco presentan mensajes de detección de errores cada vez que el usuario realiza una operación no aceptada por la aplicación. No aparecen espontáneamente, sin una invocación previa por parte del usuario. Por último, no son fuentes primarias de información (tales como bases de datos o enciclopedias interactivas). (11)

Para concluir, si se considera el grado de integración y adaptabilidad entre el sistema de Ayuda Contextual y la interfaz de usuario de la aplicación, se podrían enunciar los siguientes principios de diseño para los Sistemas de Ayuda Contextual:

- ☑ La ayuda nunca debería ser sustituta de un buen diseño de la interfaz de la aplicación que la contiene.
- ☑ La ayuda debería producir la menor interrupción posible en la tarea del usuario.
- ☑ La ayuda debería facilitar a los usuarios el planteamiento de sus preguntas y proporcionar diferentes tipos de ayuda para diferentes preguntas.
- ☑ Los Sistemas de Ayuda Contextual deberían proporcionar respuestas adecuadas a las necesidades de los usuarios.
- ☑ Los usuarios no deberían necesitar ayuda para conseguir ayuda. (9)



Los Sistemas de Ayuda Contextual tienen evidentes ventajas sobre la documentación escrita, la principal es la disponibilidad simultánea para varios usuarios, así como la facilidad de acceso a la información de ayuda y el bajo costo de almacenamiento, distribución, reproducción y actualización de la información. Como desventajas, se debe notar la imposibilidad de trabajar en la aplicación al mismo tiempo que se accede a la información de ayuda y los límites impuestos por el tamaño de la ventana de ayuda.

### 1.3.1. Tendencias actuales de las Ayudas para plataformas Educativas

El uso del software educativo se ha popularizado alrededor del mundo en menos de una década, gracias al aporte en calidad y rendimiento que hace a los procesos educacionales. Un ejemplo de lo anterior se logra ver actualmente en Internet, donde se pueden encontrar gran cantidad de plataformas educativas para ser personalizadas y ajustadas a las necesidades de cada proceso educativo.

Tanto la administración como el uso de esas plataformas resultan relativamente novedosos para algunos de sus usuarios. A pesar de que los EVEA están regidos mayormente por los mismos modelos y principios, pueden presentar funcionalidades específicas a una organización; y como consecuencia sus consumidores pueden no estar familiarizados con su explotación. En la actualidad conviven en el espacio cibernético infinidad de entornos virtuales de aprendizaje que serán detallados a continuación:

**Sakai** es un entorno de colaboración y aprendizaje para la educación superior y que proporciona al usuario flexibilidad en su gestión y un fácil mantenimiento. La navegación por Sakai es sencilla, ya que únicamente hay que ir seleccionando los sucesivos enlaces, botones y otros elementos de la interfaz. Posee un subsistema de Ayuda en Línea integrado a la aplicación y disponible para los usuarios autenticados en todas las interfaces de la aplicación y a la que se accede a través de un vínculo en el menú de opciones lateral. El contenido de la Ayuda se encuentra separado por temas afines; para visualizar el mismo se debe clicar el nombre del tema deseado y luego el nombre del contenido. También permite la búsqueda según el criterio introducido por el usuario, ya sea una palabra o una frase. Mientras se utiliza alguna de las herramientas de Sakai se puede acceder directamente al contenido de la Ayuda sobre esta solamente haciendo clic en el vínculo Ayuda del menú lateral de opciones; los contenidos que se mostrarán están en dependencia también del rol del usuario autenticado, por lo que podemos definir esta Ayuda como un subsistema personalizado de acuerdo al rol de los actores y el espacio de trabajo activo en el momento de su ejecución.





**Dokeos** es un entorno de eLearning y una aplicación de administración de contenidos de cursos. Es una herramienta de aprendizaje especialmente recomendada a usuarios que tengan nociones mínimas de computación y cuyo objetivo sea la preocupación por el contenido. Cuenta con un sistema de Ayuda accesible solamente en su página principal, antes de autenticarse el usuario. El sistema está dividido en 2 categorías: Manual del Profesor y Tutoriales Flash. El Manual del Profesor es un documento .pdf<sup>3</sup> que resume y explica las funcionalidades que el educador debe conocer para un trabajo óptimo con la herramienta. En la sección de tutoriales están ubicadas una serie de animaciones flash que muestran cómo trabajar con Dokeos paso a paso. Una vez que se accede a estos tutoriales se visualiza toda la documentación del entorno de aprendizaje que incluye un mapa del sitio y una sección de trucos.

**Helvia** es una plataforma educativa de software libre desarrollada para los centros TIC de Andalucía. Esta plataforma presenta un sistema de Ayuda en forma de manual de usuario basado en HTML que integra un índice, tabla de contenidos y una sección de búsqueda. Este manual de usuario se puede obtener a su vez en formato .pdf de forma íntegra. Para acceder a la Ayuda se tiene un vínculo en el menú situado debajo del banner, este vínculo está disponible en cada una de las interfaces del software.

**ATutor** es un entorno de creación y gestión de cursos en línea de código abierto de la Universidad de Toronto en Canadá. El acceso a su sistema de Ayuda es mediante el menú de opciones ubicado en la esquina superior de la página y que se mantiene visible en todo el sitio. Al acceder a la Ayuda se muestra una página con las 3 secciones principales en que se divide y los contenidos correspondientes a cada una de ellas. Las secciones son: Ayuda Externa, Accesibilidad y Contactar con el Soporte.

**Eleven** es una plataforma educativa que permite a los centros educativos el acceso a todo tipo de contenidos digitales, herramientas de gestión didáctica y funcionalidades docentes. La plataforma dispone de una interfaz muy intuitiva y fácil de utilizar para cualquier usuario. Presenta un sistema de Ayuda ubicado en una barra con funcionalidades básicas para el usuario y que se mantiene visible durante la navegación por el sitio.

Luego de un análisis de los diferentes Sistemas de Ayuda implementados en las plataformas y entornos educativos en internet se observa una tendencia a minimizar su contenido y dar información básica sobre los productos. Notorio es también la accesibilidad de estos Sistemas de Ayuda en las diferentes interfaces

---

<sup>3</sup> Formato gráfico creado por la empresa Adobe que reproduce cualquier tipo de documento en forma digital idéntica, permitiendo así la distribución electrónica de los mismos a través de la red garantizando su integridad.



de las aplicaciones que los contienen, de manera que resulta visible en todo momento al usuario. Menos común pero aún muy utilizado es la inclusión de foros en el contenido de la Ayuda, creando una comunidad entre los usuarios de las aplicaciones y obteniendo su aporte, que resulta valioso al tener una interacción frecuente con el producto. Otras de las funcionalidades encontradas son las secciones de preguntas frecuentes (FAQ) y de soporte técnico, ambas resultan provechosas al usuario pues la primera puede resultar en un menor empleo de tiempo de su parte navegando por el contenido de la Ayuda; y la segunda permite dar atención a una duda personalizada y la cual, para el usuario, no fue cubierta por el contenido de la Ayuda.

Desde los primeros años de la Revolución, Cuba se ha visto inmersa en una serie de transformaciones con el fin de mejorar la calidad de la educación en todos sus niveles. Para ello se destinan una gran cantidad de recursos y tecnologías como eslabón fundamental en el perfeccionamiento del proceso enseñanza-aprendizaje. Esto se puede observar en la gran variedad de software educativos, que se han creado desde los comienzos de la informática en Cuba hasta nuestros días, específicamente la creación de aplicaciones con tecnología multimedia destinadas a la enseñanza. A continuación se listan algunas de ellas con sus funcionalidades y un análisis de cómo ha sido diseñado su sistema de Ayuda al usuario.

**HAEduc:** es una herramienta de autor multiplataforma destinada al desarrollo de recursos educativos informáticos, utilizando distintos tipos de objetos multimedia. A pesar de ser una aplicación de escritorio, esta herramienta presenta un sistema de Ayuda novedoso y creativo basado en agentes, que interactúan con el usuario brindándoles mensajes e información de una manera agradable e inteligente.

Estos Agentes o también llamados Mascotas, no son más que personajes animados que pueden ser insertados en las aplicaciones y permiten comunicar información al usuario. Estos tienen la capacidad de moverse por HAEduc y mostrar mensajes en forma de texto (se planea mostrarlos con grabaciones de voz). Un agente es capaz de brindar Ayuda diferenciada por pantallas y atendiendo a diferentes tipos de usuario. Además son empleados para realizar recorridos dirigidos por HAEduc.

**SADHEA-Web:** Es un Sistema de Autor para el Desarrollo de Hiperentornos de Aprendizaje en formato web. Cuenta con un sistema de Ayuda al usuario que se caracteriza por estar contextualizado y en formato hipermedial. Esta Ayuda familiariza al usuario con el trabajo con el Hiperentorno de Aprendizaje, aun cuando el sistema es lo suficientemente sencillo y fácil de usar.

**La Colección Multisaber:** Los diferentes programas educativos incluidos en esta colección se respaldan en el concepto de Hiperentornos de aprendizaje. Cada uno de ellos cuenta con una Mascota (Agente) con



nombres atractivos para sus usuarios, que son niños en educación primaria. Se proporcionan también vías de orientación, ubicación y localización (señalizadores, títulos de cada pantalla y contadores de páginas) en las diferentes partes del programa. Como parte de este sistema de orientación cuentan además con una ayuda para cada contexto, a través de la cual se explican los elementos que integran las interfaces de cada entorno y la forma de operar con ellas para optimizar la interacción.

Los Sistemas de Ayuda al usuario generalmente no son utilizados en las plataformas y herramientas educativas, en las herramientas desarrolladas en Cuba se exhibe una preferencia hacia el uso de sistemas guías integrados a la aplicación y que interactúan con el usuario en todo momento. Estos agentes o mascotas deben ser vistos a su vez como Sistemas de Ayuda al usuario pues tienen incorporado el contenido necesario para esto y al ser altamente interactivos la posibilidad de éxito con los usuarios.

En la Universidad de las Ciencias Informáticas se desarrollan software para diversos clientes y con diversos propósitos, el Centro Fortes alberga el proyecto **Multisaber – Navegante** que consiste en el desarrollo de 2 colecciones de software educativo. Estas son aplicaciones web y solo la colección Multisaber contiene un sistema de ayuda contextual para sus usuarios.

Se observa que aunque en la UCI existen numerosos proyectos para el desarrollo de software educativo, la inclusión de un sistema de ayuda al usuario no ha sido prioridad en sus requisitos.

Luego de analizar las diferentes clasificaciones y particularidades de los Sistemas de Ayuda, así como las tendencias actuales en las ayudas para plataformas educativas; se decide hacer uso de las características de los Sistemas de Ayuda Contextual como propuesta de solución al problema científico planteado. Este sistema debe incorporar aspectos comunes de las aplicaciones anteriormente analizadas. La inclusión de una sección de preguntas frecuentes (FAQ por sus siglas en inglés) brinda valor agregado a los usuarios del sistema, además se debe tener en cuenta la presencia del acceso a la Ayuda desde todas las interfaces del mismo. Se adopta también la tendencia a minimizar el contenido a mostrar a los usuarios, garantizando que estos obtengan solo lo necesario y puedan continuar trabajando con el sistema a la mayor brevedad posible.



## 1.4. Metodologías, herramientas y estándares de posible utilización como propuesta de modelación a utilizar en la solución

### 1.4.1. Metodologías de desarrollo de software

Las metodologías de desarrollo de software son el conjunto de técnicas y procedimientos que nos permiten conocer los elementos necesarios para construir un software. El uso de metodologías en el desarrollo del mismo aumenta su calidad final y facilita el trabajo del equipo de desarrollo en gran medida a la vez que permite un mayor control de todo el proceso de desarrollo, obteniendo los resultados esperados en el tiempo planificado. Están clasificadas en: Estructuradas, Para Sistemas de Tiempo Real y Orientadas a Objetos. Estas últimas fomentan la reutilización de componentes y facilitan la división del sistema en varios subsistemas independientes. Las metodologías orientadas a objetos que gozan de mayor popularidad serán listadas a continuación:

#### **Proceso Unificado de Desarrollo de Software (RUP)**

El Proceso Unificado de Desarrollo (Rational Unified Process, RUP), es un proceso robusto de desarrollo de software que fue creado por Ivar Jacobson, James Rumbaugh y Grady Booch que junto con el Lenguaje Unificado de Modelado (UML) constituyen la metodología estándar de desarrollo de software más utilizada en la actualidad para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es la integración de las mejores prácticas empleadas por otras metodologías, enfocándose generalmente en desarrollar grandes y complejos proyectos aplicando el paradigma de la Programación Orientada a Objetos. (12)

Los autores del Proceso Unificado de Desarrollo destacan que el proceso de software propuesto por RUP tiene tres características esenciales:

**Dirigido por Casos de Uso:** Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. En RUP, los Casos de Uso constituyen la forma de especificar los requisitos del sistema, guiar su diseño, implementación y prueba. Constituyen además un elemento integrador y una guía del trabajo, pues inician el proceso de desarrollo y proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

**Centrado en la arquitectura:** La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo. En el caso



de RUP, además de utilizar los Casos de Uso para guiar el proceso, se presta especial atención al establecimiento temprano de una buena arquitectura, que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño, por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás. Para RUP, todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura, el cual recibe este nombre porque lo forman: la Vista Lógica, Vista de Implementación, Vista de Proceso y Vista de Despliegue, más la Vista de Casos de Uso que ofrece cohesión a todas.

**Iterativo e incremental:** El trabajo se divide en partes más pequeñas o mini proyectos, permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto. De esta forma, durante todo el proceso de desarrollo se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales), de la cual se obtiene un incremento que produce un crecimiento en el producto.

RUP utiliza como lenguaje de modelado el Unified Modelling Language (UML) para describir todo el proceso. Se divide en ciclos de trabajo, teniendo un producto superior como resultado de cada ciclo. Estos se componen en su interior por varias fases, en la cuales se llevan a cabo un conjunto de flujos para el desarrollo de todo el proyecto. (12)

El ciclo de vida de RUP está dividido en 4 fases (inicio, elaboración, construcción y transición) en las que dependiendo de la magnitud del proyecto se realizarán las iteraciones que sean necesarias siguiendo un modelo secuencial o de cascada orientado por los flujos de trabajos. RUP define flujos de trabajos básicos que se sustentan en la creación consistente del producto y otros tres complementarios dedicados a las tareas de soporte. Cada flujo de trabajo está asociado con uno o varios modelos para los cuales UML propone una vista distinta del sistema en cada caso. Estos modelos son la abstracción del sistema que permite que los usuarios y desarrolladores entiendan qué se debe hacer, por lo tanto constituyen en su mayoría artefactos entregables para el cliente. RUP se ha convertido en un estándar para líderes y gerentes, en la medida en que estos se interesan por mejorar los procesos de desarrollo de software que tienen lugar en sus organizaciones.

### **Extreme Programming (XP)**

La metodología Programación Extrema o *Extreme Programming (XP)*, es conocida como una metodología ágil o ligera, orientada al cliente, y de iteraciones cortas. La base para el desarrollo de software que usa



esta metodología son las llamadas historias de usuario (User Stories), historias escritas por el cliente en las que describe escenarios sobre el funcionamiento del sistema y que no sólo están limitados a los formularios de usuario, sino que también pueden describir modelos. Estas historias de usuario, junto a la arquitectura que se persigue, sirven de base para crear un plan de “entregas de software” entre el equipo de desarrollo y el cliente.

Propone que en el equipo de desarrollo se necesita un representante constante del cliente que conozca perfectamente el negocio y que esté a disposición para cualquier duda o necesidades de los desarrolladores. El cliente se mantiene todo el tiempo informado paso por paso de las actividades que se están desarrollando y a medida que se dé la liberación de cualquier entregable, se discutirá con el representante y se repite la nueva iteración del software. (13)

La programación del software, siempre se define en pareja con el objetivo principal de lograr mayores resultados y los menores errores posibles, mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta.

### **Proceso Ágil Unificado (AUP)**

El Proceso Ágil Unificado (Agil Unified Process, AUP), es una versión simplificada de RUP que describe de forma simple y fácil de comprender el uso de técnicas ágiles para el desarrollo de aplicaciones que permanezcan dentro de los conceptos de RUP. AUP promueve principios ágiles a la vez que mantiene las cuatro fases propuestas por RUP (inicio, elaboración, construcción y transición), no así para los flujos de trabajo donde el Modelo, primer flujo establecido por AUP, abarca las disciplinas Modelo de Negocio, Requerimientos, Análisis y Diseño. (14)

Luego de realizar un estudio de las diferentes metodologías teniendo en cuenta sus particularidades se comprende que son aplicables a diferentes problemáticas. Para lograr seleccionar cuál de ellas utilizar, se debe hacer una comparación entre todas para una mejor modelación de los procesos en el desarrollo de la Ayuda. Se hace notable que la metodología RUP utilizada de conjunto con el lenguaje UML provee de un proceso robusto de desarrollo de software, donde al final de cada iteración se obtendrá una versión del producto que podrá ser evaluada por el cliente. Esta metodología facilita abundante documentación lo que resulta provechoso para futuras modificaciones del subsistema. Por estas razones se elige el Proceso Unificado de Desarrollo de Software como la metodología para guiar el desarrollo del Subsistema Ayuda.



### 1.4.2. Herramientas para el modelado de sistemas

Luego de definir que metodología será usada, se procede a hacer un análisis de las herramientas más utilizadas actualmente para modelar los procesos de software.

**Rational Rose** es la herramienta CASE que comercializan los desarrolladores de UML. “Proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente. Características adicionales que incluye se puede mencionar su capacidad de análisis de calidad de código, modelado UML para trabajar diseños de base de datos y generación de informes para optimizar la comunicación dentro del equipo” (15)

**Visual Paradigm** es una herramienta CASE profesional y soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Es una herramienta que genera la documentación del proyecto automáticamente en formato Web o .pdf, soporta la realización de ingeniería tanto directa como inversa y permite control de versiones. Presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas que soportan la Ingeniería de Requerimientos. Cabe destacar igualmente su robustez, usabilidad y portabilidad. (16)

**Pencil Project** es un programa libre y de código abierto para la creación de diagramas y prototipos de interfaces de usuario (GUI por sus siglas en inglés) de forma rápida y sencilla. Entre sus funcionalidades cuenta con plantillas para crear diagramas y prototipos, vínculos entre páginas y cuenta con un plugin<sup>4</sup> para ser ejecutado en Firefox<sup>5</sup>. Una vez instalado lo siguiente es ejecutar la aplicación y comenzar un proyecto nuevo, contando con numerosas herramientas y opciones de personalización. Resulta muy útil para crear maquetas de aplicaciones web aunque también puede ser usado para aplicaciones de escritorio. (17)

En cuanto a las herramientas se debe tener en cuenta las características del ambiente de desarrollo, el cual está basado en software libre y de código abierto por lo que la herramienta elegida es el Visual Paradigm for UML 6.4 para modelar los procesos del sistema. El Pencil Project 1.2 resulta muy útil para la

---

<sup>4</sup> Un complemento (plugin) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

<sup>5</sup> Navegador web de la fundación Mozilla que se caracteriza por su navegación por pestañas. Es libre y de código abierto, siendo un software multiplataforma, disponible para Microsoft Windows, Mac y GNU/Linux.



creación de maquetas de interfaz de usuario, además de contar con plantillas para la creación de diagramas.

### 1.5. Herramientas para la confección del contenido del Subsistema Ayuda

Primeramente el redactor del manual del usuario debe tener acceso mediante un navegador web a una versión actualizada de la plataforma para así poder probar y ejecutar las funcionalidades que va a explicar. Una vez logrado esto procederá a redactar el contenido de Ayuda en la herramienta para procesar texto que provee la Suite Open Office, el Open Office Writer. Se ha seleccionado esta herramienta por la facilidad de uso que presenta para las notas simples y al ser un software de licencia de código abierto elimina las preocupaciones del cumplimiento de la licencia así como la amenaza de las auditorías de software. Se necesita además de una herramienta para la captura y edición de imágenes para complementar la redacción del contenido. En la investigación surgieron varias que se listan a continuación:

**RapiCapWin** es una herramienta para los entornos Windows 98, NT, ME, 2000 y XP en idioma español que permite realizar capturas de pantalla (pantalla completa, ventana activa o área personalizada) dando la posibilidad de salvarlas en los formatos JPG, GIF o BMP. También admite copiar la captura directamente al portapapeles. (18)

**Free Screen Capture** es una herramienta para los entornos Windows 98, NT, ME, 2000 y XP que permite realizar captura de la pantalla completa, captura solo de la ventana activa, de video, de sonido, un área determinada de pantalla e incluso captura con desplazamiento (para capturar un área más grande que la pantalla). Además soporta todos los formatos gráficos habitualmente usados. (19)

**JShot** es una herramienta para los entornos GNU/Linux que captura imágenes de la pantalla de la computadora y que luego da la posibilidad de editarlas antes de guardarlas. (20)

El siguiente estudio se centrará principalmente en las herramientas de confección de Sistemas de Ayuda basados en HTML, ya que el producto que se está desarrollando es un sistema web. De esta forma se puede construir una Ayuda más personalizada e interactiva.

Los Sistemas de Ayuda basados en HTML son de carácter abierto y multiplataforma, esto quiere decir que un mismo hipertexto podría ser leído por cualquier usuario equipado con un navegador capaz de interpretar HTML estándar, independientemente del tipo de sistema operativo que esté utilizando. La característica principal de estos sistemas es que toman como base textos escritos en HTML en lugar de





textos en formato RTF (Formato Rico en Texto - RTF, siglas en inglés). Entre las herramientas que permiten la creación de páginas HTML se estudiaron las más populares en la actualidad:

**BlueFish Editor** es una herramienta especialmente dirigida a los diseñadores y programadores de aplicaciones web con variadas opciones para programar webs y scripts<sup>6</sup>. Soporta diferentes lenguajes de programación y va especialmente dirigido a la edición de sitios dinámicos e interactivos. Es una herramienta libre y multiplataforma. (21)

**Amaya** es una herramienta de creación de páginas Web, equipada con una interfaz de diseño muy sencilla. Posee un menú de herramientas muy intuitivo, una amplia variedad de funciones y utilidades para dar formato al texto, introducir imágenes, crear enlaces, insertar tablas, cambiar colores y muchas otras. Trabaja básicamente con HTML, pero también tiene soporte para documentos en formato CSS, MML y SVG. Es una herramienta libre y en idioma español. (22)

En el desarrollo del Sistema de Ayuda son necesarias herramientas que provean de un entorno de desarrollo con una licencia libre. Las herramientas a utilizar a la hora de elaborar el contenido que será acoplado en la Ayuda también se rigen por este criterio. Por tanto para lograr obtener un mayor beneficio de este trabajo, se deben seleccionar herramientas compatibles con el Sistema Operativo utilizado en el desarrollo: GNU/Linux. Dichas herramientas también deben proveer la facilidad de crear y editar páginas HTML y hojas de estilos CSS, además de posibilitar la captura y edición de las imágenes. Tomando como referencia estos criterios se proponen las herramientas Jshot 1.4.0.1 y BlueFish 2.0 porque se ajustan a los requerimientos antes mencionados y son las más sencillas de utilizar ahorrando tiempo de capacitación al equipo de trabajo.

## 1.6. Herramientas y lenguajes a utilizar en el entorno de desarrollo

Es de gran importancia la selección de un ambiente adecuado para poder explotar todos los beneficios de este y agilizar el desarrollo del software. En esta sección se define el conjunto de herramientas, lenguajes y plataformas tecnológicas (especificando en cada caso su versión). Se especifica y describen los lenguajes de programación, gestor de base datos y otras herramientas de trabajo que son necesarias para el desarrollo del Subsistema Ayuda, así como la integración entre estas.

---

<sup>6</sup> Un script (cuya traducción literal es 'guion') o **archivo de órdenes** es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.



### 1.6.1. Servidor Web

Un servidor de páginas Web es un programa que permite acceder a páginas Web alojadas en un ordenador. Los servidores Web suministran páginas Web a los navegadores que lo solicitan. En términos más técnicos, los servidores Web soportan el Protocolo de Transferencia de Hipertexto como HTTP (HyperText Transfer Protocol), el estándar de Internet para comunicaciones Web.

**Apache** es el servidor Web hecho por excelencia, su robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa haciéndolo el servidor web más utilizado del mundo. Dispone de varios módulos que no están incluidos en el núcleo del servidor, algunos de estos módulos permiten que Apache brinde las funcionalidades básicas para un servidor web y otras lo convierten en mucho más que un servidor web básico ya que optimiza el rendimiento y la rapidez del código. (23)

**Cherokee** es un servidor web multiplataforma. Su objetivo es ser rápido y completamente funcional, sin dejar de ser liviano comparado con otros servidores web. Entre sus características destacan que soporta la configuración de servidores virtuales, permite la realización de redirecciones y dispone de un panel de administración desde la web. (24)

Se ha escogido el servidor web Apache pues hoy en día se encuentra muy por encima de sus competidores, tanto gratuitos como comerciales. También debido a que es un software de código abierto que funciona sobre cualquier plataforma. Además de brindar las funcionalidades básicas para un servidor web, optimiza el rendimiento y la rapidez del código

### 1.6.2. Sistemas Gestores de Bases de Datos

Una base de datos es un “almacén” que permite guardar grandes cantidades de información de forma organizada para luego poderla encontrar y utilizar fácilmente. Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Entre las principales características de los sistemas de base de datos se pueden mencionar independencia lógica y física de los datos, redundancia mínima, acceso concurrente por parte de múltiples usuarios, integridad de los datos, consultas complejas optimizadas, seguridad de acceso y auditoría y acceso a través de lenguajes de programación estándar. (25)



En los procesos que se pretende modelar se describe el uso de una base de datos para almacenar los contenidos de la Ayuda, logrando así una mayor eficiencia al momento de responder las dudas de los usuarios siendo más fácil ubicar el contenido a la vez de tener mayor organización.

El uso de una base de datos ofrece beneficios tales como control sobre la redundancia de datos, mejora la accesibilidad de los datos, el mantenimiento y aumenta la concurrencia.

Los Sistemas de Gestión de Base de Datos<sup>7</sup> sirven de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. (25)

Como propuesta de Sistemas de Gestión de Bases de Datos se exponen los siguientes:

**MySQL Database Server** es el DBMS de código fuente abierto más usado del mundo. Es un servidor de base de datos muy rápido y sencillo de usar, multiusuario, multiplataforma (Windows, Linux, Mac OS) Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. (26)

**PostgreSQL** es una potente fuente de sistema de bases de datos de código abierto objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de confiabilidad, integridad de los datos y corrección. También soporta el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o video. Como cliente de PostgreSQL se utiliza pgAdmin 3, que es la plataforma de código abierto más popular para la administración y desarrollo para PostgreSQL. (27)

Presenta una serie de características que se especifican a continuación:

- ☑ **Atomicidad (Indivisible)** es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- ☑ **Consistencia** es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
- ☑ **Aislamiento** es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.

---

<sup>7</sup> En inglés: Database Management System DBMS



- ✓ Durabilidad es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.
- ✓ Documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios.
- ✓ Comunidades muy activas, varias comunidades en castellano. (28)

Analizando las características de los dos gestores de bases de datos se puede concluir que MySQL carece de soporte para transacciones y subconsultas y no implementa una buena escalabilidad. No es viable para su uso con grandes bases de datos, como la de la plataforma ZERA, a la que se accede continuamente. El hecho de que MySQL no maneje la integridad referencial, hace de este gestor una solución pobre. Sin embargo, PostgreSQL implementa todos estos requisitos e incluye la posibilidad de almacenar imágenes, funciona en todos los principales sistemas operativos, incluyendo Linux, Unix (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows. Además tiene pleno apoyo a las claves foráneas, uniones, vistas, desencadenadores y procedimientos almacenados (en varios idiomas). Como resultado de este análisis se concluye que la herramienta PostgreSQL 8.4 garantiza la confiabilidad, consistencia e integridad de los datos del sistema a desarrollar, por lo que es seleccionada para ser utilizada.

### 1.6.3. Lenguajes de programación para la web

Las máquinas en general, y las computadoras en particular, necesitan de un lenguaje propio para poder interpretar las instrucciones que se les dan y para que nosotros podamos controlar su comportamiento. Ese lenguaje que permite esta relación con las computadoras es el lenguaje de programación. *“En el dominio de la red, los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP y PERL.”* (29)

#### 1.6.3.1. Lenguajes del lado servidor

**ASP** (Active Server Pages) es un lenguaje derivado del Visual Basic desarrollado por Microsoft. Evidentemente su empleo se realiza sobre plataformas funcionando bajo sistema Windows. (29)

**PHP** es el lenguaje libre más utilizado para desarrollar aplicaciones web a nivel mundial. Se distribuye en formato open-source y es gratuito. PHP es el acrónimo de Hipertext Preprocesor y es un lenguaje interpretado de alto nivel introducido en páginas HTML y se ejecuta del lado del servidor en la Arquitectura Cliente – Servidor (esto significa que PHP funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario). (30)



PHP es muy utilizado por los desarrolladores de páginas web dinámicas<sup>8</sup>, pues tiene incorporado un grupo de mejoras en cuanto a programación orientada a objetos, funciones, extensiones y muchos detalles más. **PERL** (Practical Extracting and Reporting Language) es muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Es un lenguaje rápido y potente que requiere un aprendizaje más largo. Por lo que resulta más reservado para personas ya familiarizadas con la programación avanzada. (31)

El lenguaje PHP 5.3.0, se eligió dado que es un lenguaje de libre distribución lo cual permite que la aplicación pueda ser ejecutada bajo cualquier plataforma. No presenta inconvenientes cuando es utilizado con otros lenguajes como HTML y JavaScript. Así que es posible generar HTML dinámico desde PHP, con el fin de construir páginas web interactivas.

### 1.6.3.2. Lenguajes del lado cliente

**JavaScript** se utiliza principalmente para crear páginas web dinámicas e interactivas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar el código para poder ejecutarlo. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador web sin necesidad de procesos intermedios. JavaScript reduce la cantidad de transacciones que se efectúan a través del protocolo http y las posibilidades de que se genere un error durante la inserción de datos. También puede leer y escribir cookies, una operación que hasta hace poco únicamente podía desarrollar el servidor Web. (32)

**XHTML**, acrónimo inglés de Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es lograr una web semántica, donde la información y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo CSS su aspecto y diseño en distintos medios (computadoras, PDAs, teléfonos móviles, impresoras y dispositivos utilizados por personas discapacitadas) y para JavaScript su comportamiento. (33)

---

<sup>8</sup> Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.



**CSS** son las siglas de *Cascading Style Sheets* (Hojas de estilos en cascada) y es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. (34)

#### 1.6.4. Framework

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (35)

Una de las muchas preguntas que puede hacerse un desarrollador es: ¿Qué framework se puede utilizar? La decisión se vuelve más difícil por dos cuestiones básicas: la primera es que no es muy sencillo cambiar de framework ya que hay todo un proceso de aprendizaje previo que conlleva al uso del mismo. Por otro lado, la segunda cuestión es que cada framework tiene sus características, ventajas y desventajas; y comparar entre tantas opciones se vuelve algo difícil a veces. Existe una web que ayuda en el proceso de comparar los frameworks web, se trata de [www.bestwebframeworks.com](http://www.bestwebframeworks.com). La selección de los frameworks, que a continuación se estudian en este epígrafe, está basada en el criterio que ofrece este sitio web mediante unas tablas comparativas con características relevantes de los distintos frameworks web según el tipo de tecnología (PHP, JavaScript, CSS).

##### 1.6.4.1. Frameworks PHP

Los framework PHP más populares actualmente son<sup>9</sup>: Zend Framework, CakePHP y Symfony.

**Zend Framework** es uno de los frameworks Open Source más conocido y más utilizado profesionalmente para desarrollo de aplicaciones Web y servicios Web con PHP5. Brinda soluciones para construir sitios web robustos y seguros. (36)

**CakePHP** es un framework de desarrollo rápido para PHP que ofrece una arquitectura extensible para el desarrollo, mantenimiento y despliegue de aplicaciones. Sigue el modelo MVC (Modelo Vista Controlador) y facilita al usuario la interacción con la base de datos mediante el uso de ActiveRecord. Es compatible

---

<sup>9</sup> Tomado de la web [www.bestwebframeworks.com](http://www.bestwebframeworks.com)



con PHP 4 y 5 y se recomienda para proyectos no demasiado complejos, en ocasiones combinado con algún componente de Zend. (37)

**Symfony** es un framework completamente diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web con PHP. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de la aplicación. Symfony está desarrollado completamente con PHP 5 y es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar no sólo en plataformas libres como Linux, sino también en plataformas Windows.

Symfony emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web. El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos. La vista transforma la información obtenida por el modelo en las páginas web a las que acceden los usuarios. El controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista. Symfony tiene integrado el ORM (Mapeo de objetos-relacional) Doctrine, el cual se encarga del acceso a datos brindando un conjunto de clases mapeadas que permiten la interacción con la base de datos a través de objetos, evitando la incompatibilidad entre los distintos gestores de bases de datos.

Symfony puede ser completamente personalizado para cumplir con los requisitos, las políticas y reglas dispuestas para la gestión de proyectos y la programación de aplicaciones y automatiza la mayoría de los elementos comunes de los proyectos web, como por ejemplo:

- ✓ La capa de presentación utiliza plantillas y *layouts*<sup>10</sup>
- ✓ Los *helpers* incluidos permiten minimizar el código utilizado en la presentación, pues encapsulan grandes bloques de código.
- ✓ Los formularios incluyen validación automatizada y relleno automático de datos ("*repopulation*"),
- ✓ Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.

---

<sup>10</sup> Son diferentes tipos de área donde se colocan los controles, estos son elementos de interacción con el usuario, como botones, entrada de texto o etiquetas.



✓ Las interacciones con Ajax son muy fáciles de implementar mediante los *helpers* que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código. Los plugin de Symfony se crean mediante el lenguaje PHP. Permiten encapsular clases, filtros, archivos de configuración, tareas, módulos, esquemas y extensiones para el modelo, archivos estáticos y también agrupar todo el código diseminado por diferentes archivos y reutilizar este código en otros proyectos. (38)

#### 1.6.4.2. Frameworks JavaScript

Los frameworks JavaScript más utilizados por los desarrolladores en sus proyectos son:

**MooTools** es un Framework web orientado a objetos para JavaScript, de código abierto, compacto y modular. El objetivo de MooTools es aportar una manera de desarrollar JavaScript sin importar en qué navegador se ejecute de una manera elegante. (39)

**JQuery** es una librería open source de JavaScript rápida y concisa, que simplifica las operaciones sobre el DOM, el manejo de eventos, animaciones e interacciones AJAX para el desarrollo en la WEB. JQuery encabeza la lista de las librerías JavaScript más utilizadas gracias a su sencillez, flexibilidad y rendimiento. Incorpora efectos visuales, drag and drop, auto-completar, animaciones etc. El desarrollar todos estos conceptos desde cero puede resultar complicado sobre todo si se tiene que presentar la solución con muy poco tiempo, en este tipo de situaciones el empleo de librerías como jQuery puede facilitar el desarrollo de la aplicación. (40)

#### 1.6.4.3. Frameworks CSS

Un framework CSS permite hacer interfaces web estéticamente mucho más placenteras y eliminar todos los problemas derivados de la compatibilidad con diferentes navegadores. Entre los frameworks CSS más populares en la actualidad se encuentran: YAML y Blueprint.

**YAML (Yet Another Multicolumn Layout)** está configurado para desarrollar diseños flexibles y es ampliamente utilizado. Tiene una licencia libre Creative Commons y una documentación sólida. También posee una comunidad de usuarios activos en caso de que necesiten ayuda. (41)

**Blueprint** es un framework CSS que sirve de apoyo en el diseño y maquetación de webs 2.0, gracias a su rejilla y otros estilos de tipografía e impresión. Se trata de una serie de librerías de Hojas de Estilo en Cascada (CSS) que contienen código útil para maquetar una página web y aplicar otros tipos de estilos tipográficos o de impresión, algo que comúnmente conocemos como Framework CSS. (42)





Este sistema ofrece una serie de archivos con código CSS que podremos incluir en las páginas web para aplicar estilos de una manera rápida, y solucionar también muchos de los problemas de compatibilidad entre navegadores que pueden surgir durante la etapa de maquetación. Existen diversos Framework CSS disponibles de manera gratuita en Internet, creados por varios desarrolladores distintos. Blueprint es quizás el más conocido, o por lo menos uno de los más populares y prueba de ello es que se han creado incluso algunas variantes o productos derivados para ajustar Blueprint a las distintas necesidades de los desarrolladores.

El framework CSS Blueprint está creado sobre una sólida base de código CSS que ayudará a desarrollar el diseño y maquetación de sitios web. (42)

Con todo esto se consigue una base con la que se puede comenzar fácilmente a hacer páginas web complejas y que serán más compatibles en distintos navegadores. Además, Blueprint se completa con algunos plugin creados por terceras personas que sirven para hacer cosas más avanzadas o diferentes. Por ejemplo hay plugin para conseguir íconos, para hacer páginas que se adaptan a la anchura de la ventana del navegador, etc.

Luego de analizar todos los frameworks según sus clasificaciones se selecciona como framework de desarrollo el Symfony 1.4.11 pues puede ser completamente personalizado para cumplir con los requisitos, las políticas y reglas dispuestas para la gestión de proyectos y la programación de aplicaciones; conjuntamente automatiza la mayoría de los elementos comunes de los proyectos web. El framework javascript JQuery 1.3.2 se selecciona para las interacciones AJAX complejas y el framework CSS Blueprint 2.0 para elaborar la presentación de la plataforma.

### 1.6.5. Entornos de desarrollo integrados (en inglés IDE)

Los Entornos de Desarrollo Integrado son aplicaciones informáticas que engloban varias herramientas que son muy útiles a los desarrolladores de software porque facilitan y optimizan su tarea de escribir programas, corregirlos y ejecutarlos.

**Netbeans** es un producto libre y gratuito sin restricción de uso y un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. Se puede utilizar para programar en otros lenguajes además de PHP. Aparte de las funciones más básicas con las que debería contar cualquier IDE, como resaltado de sintaxis, autocompletado, formateo de código o depurador (xDebug), también cuenta con



otras funcionalidades menos comunes como la integración con PHPUnit para las pruebas unitarias y con CVS, Subversion y Mercurial para el control de versiones.

Gracias a esta herramienta se puede construir rápidamente, editar de forma visual y depurar aplicaciones ricas para Internet (En inglés, Rich Internet Applications). El NetBeans 6.9 brinda soporte para PHP 5.3 así como también para Symfony. Entre sus mejoras y novedades se destacan:

- ☑ JavaFX Composer, una herramienta visual para construir aplicaciones JavaFX.
- ☑ Soporte de la plataforma OSGI.
- ☑ Mejoras en el soporte de los frameworks JavaFX SDK 1.3, PHP Zend, Ruby on Rails 3.0 y Spring Framework 3.0.
- ☑ Mejoras en los editores y debuggers<sup>11</sup> Java. (43)

**Eclipse** es un entorno de desarrollo integrado libre desarrollado originalmente por IBM. Actualmente es mantenido por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. Una de las características más importantes de Eclipse es su modularidad, pues se pueden añadir plugin según las necesidades de cada desarrollador. Eclipse dispone de un editor de texto con resaltado de sintaxis. La compilación es en tiempo real. (44)

**Zend Studio** es un entorno integrado de desarrollo especializado en la tecnología de servidores en PHP, que funciona en plataformas como Microsoft Windows, Mac OS X y GNU/Linux y se desarrolla bajo licencia comercial. El programa está escrito en java lo que en ocasiones provoca que no funcione con la rapidez de otras aplicaciones de uso diario. Dispone de características como: resaltado de sintaxis, autocompletado de código, ayuda de código, lista de parámetros de funciones y métodos de clase, inserción automática de paréntesis y corchetes de cierre así como su emparejamiento dentro del código, sangrado automático, detención de errores de sintaxis en tiempo real, phpDoc integrado, manual de PHP integrado, soporte para control de versiones como CVS o subversión, soporte para la navegación de bases de datos y sentencias SQL, entre otras. (45)

Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Aunque

---

<sup>11</sup> Programa que permite la ejecución controlada de otros programas, mostrando simultáneamente el código que se está ejecutando.



esta ayuda contextual no solo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que se vayan creando. A pesar de presentar funcionalidades favorables para el equipo de desarrollo, este IDE no es distribuido bajo una licencia gratuita, por lo que no podrá ser utilizado en el proceso de desarrollo.

Al analizar las herramientas mencionadas anteriormente se opta por utilizar el IDE Netbeans 6.9 por las enormes prestaciones que ofrece y por su integración con el framework de desarrollo seleccionado.

### **1.7. Conclusiones**

En este capítulo se hizo un análisis sobre la evolución del concepto de software educativo y de la utilización de sistemas de ayuda para estos. Se realizó un estudio sobre las tendencias actuales en Cuba y el mundo para el diseño de Sistemas de Ayuda para software educativo. Se caracterizaron diferentes metodologías para la descripción de los procesos para desarrollar una aplicación, así como las herramientas disponibles para esto. Se describieron las posibles herramientas para desarrollar la Ayuda. Se seleccionaron las metodologías que se ajustan a las necesidades del equipo de trabajo; así como las herramientas que mejor se adaptan a los requerimientos de los procesos a modelar.



## Características del Subsistema Ayuda de la Plataforma ZERA



### 2.1. Introducción

En este capítulo se exponen las características del sistema informático al cual se le desarrolla el Subsistema Ayuda, haciendo énfasis en los procesos que tienen lugar en el mismo. Se especifican los requerimientos a cumplir para la creación del subsistema y se detallan los pasos seguidos para comenzar el proceso de diseño del Subsistema Ayuda.

### 2.2. Características de la plataforma para la gestión del aprendizaje ZERA

ZERA es una plataforma web para la gestión del aprendizaje, que brinda soporte a los diferentes contenidos educativos desarrollados por la empresa Alfaomega S.A de los Estados Unidos Mexicanos. A la vez que gestiona el acceso a los mismos, así como su creación o modificación. La plataforma cuenta con numerosas funcionalidades para garantizar el buen funcionamiento del proceso de aprendizaje en línea; como es proporcionar a profesores, directores de escuelas y tutores, el seguimiento y control de las actividades académicas de los estudiantes que matriculan en los programas de estudios. Para aquellos usuarios que comienzan a utilizar ZERA o que no estén familiarizados con esta les puede resultar difícil sacar el máximo provecho a su trabajo. También es notable el empleo de una mayor cantidad de tiempo en la realización de tareas de cualquier complejidad.

Una forma de brindar asistencia a los usuarios es mostrando en todo momento pequeñas descripciones en forma de párrafo acerca de las funcionalidades que se muestran en la pantalla activa. Estas descripciones pueden ser mostradas u ocultadas según la voluntad del usuario y podrán ser visualizadas por todos los roles. Para lograr una asistencia lo más completa posible se debe conocer que tema desea consultar el usuario. Generalmente, al invocar una Ayuda, los usuarios desean comprender como ejecutar una función específica. Por tanto según sea la acción siendo ejecutada por el usuario al momento de consultar la Ayuda, será el contenido que se le mostrará al mismo; eliminando la navegación entre los temas de Ayuda. Esto garantiza que accedan a un contenido específico solo los usuarios con permisos para ello; ya que las opciones de manejo de la plataforma y el contenido a mostrar dependerán del perfil del usuario



activo. La presentación del contenido de Ayuda será en una ventana emergente; pues, de esta manera, se preserva el trabajo siendo realizado por el usuario al momento de invocar la Ayuda.

### **2.3. Planificación del Contenido a incluir en la Ayuda**

El contenido a incluir en la Ayuda está compuesto por todos aquellos fragmentos de información a estructurar y publicar en la misma; los más comunes son texto e imágenes. Debe notarse que la información que se le va a ofrecer al usuario sea primeramente necesaria, concisa y suficiente.

Al visualizar una interfaz de la aplicación, se incluye una descripción sobre sus funcionalidades y la forma de trabajar con ellas; logrando así que los usuarios conozcan mejor la aplicación. Esta descripción puede ser ocultada por decisión del usuario si este no desea visualizarla, obteniendo con esto una asistencia de cierto modo dinámica.

Al ejecutar acciones más complejas en el Sistema las descripciones antes nombradas resultan insuficientes para cubrir las dudas que puedan surgir en los usuarios. Por esto se crean contenidos específicos a cada una de estas acciones, los cuales serán mostrados únicamente cuando estas sean ejecutadas. El contenido propio de las acciones que puede realizar un usuario se corresponde con la descripción textual de las mismas (que se pueden encontrar en las descripciones textuales de los Casos de Uso del Sistema).

El paso más importante de esta etapa es definir los módulos en que se divide el contenido del Subsistema Ayuda, que representan cada uno de los subsistemas de la aplicación.

La selección de la información que se desea incluir en la Ayuda debe ser de manera rigurosa, pues ésta debe ser significativa y útil para el usuario en su interacción con el software. El lenguaje que se debe utilizar para el contenido de la Ayuda debe ser conciso, sin ambigüedades y entendible para los usuarios finales.

### **2.4. Especificaciones acerca del Diseño Gráfico del Subsistema Ayuda**

El diseño de la Ayuda es una parte fundamental dentro del proceso de desarrollo y se debe tener en cuenta que al realizarlo no se conoce qué partes de la Ayuda serán las más utilizadas; por tanto se debe prestar la misma importancia a cada una de ellas. Las pautas de diseño del Subsistema Ayuda siguen los principios establecidos por la plataforma ZERA, siempre que no interfiera con los suyos propios. La Ayuda estará disponible en cada una de las interfaces de la aplicación ya sea en forma de descripción o como



apoyo al usuario. Se debe mantener igualdad de una pantalla a otra, así como sus parámetros de diseño. De esta manera el usuario desarrolla un modelo mental sobre todas las áreas de la aplicación sobre las cuales trabaja. También es objetivo lograr que el entorno de trabajo de la Ayuda sea similar al entorno de ZERA, por lo que al usuario le resultará familiar trabajar con ambos software.

## 2.5. Organización del Contenido

Se evitan las páginas demasiado largas porque tienden a desorientar al usuario ya que a medida que van avanzando se ven obligados a recordar elementos que no pueden ver en pantalla. Aunque tampoco es necesario subdividir la información demasiado pues los usuarios pueden cansarse de pulsar muchos enlaces para encontrar la información. Se utilizan pequeñas piezas de información para los enlaces y el usuario debe encontrar datos precisos de la información relacionada con el tema al pulsar dicho enlace.

Se utiliza un formato uniforme para la organización y presentación del contenido; permitiendo a los usuarios usar su experiencia anterior en la Ayuda en el momento de buscar una nueva información.

La información mostrada será generalmente en forma de pasos lógicos que se deben seguir para cumplir su objetivo. Para explicar los pasos a seguir se usan viñetas, evitando así el exceso de párrafos. Los párrafos que contenga la Ayuda deberán ser cortos (máximo 3 oraciones). Se utilizarán imágenes para fortalecer la explicación de las funciones de la aplicación y para explicarlas se usan indicadores numéricos. En el **Anexo 1** se puede consultar una interfaz de ejemplo junto con la explicación de los indicadores numéricos.

### 2.5.1. Formato y presentación de las imágenes adjuntas al contenido

Las imágenes serán capturadas por los escritores técnicos a medida que estos van redactando el contenido de Ayuda. Los criterios para tomar una imagen son los siguientes:

- Cuando se explican los pasos a seguir en alguno de los menús de la aplicación.
- En cada una de las interfaces que presente una opción que sea importante para el flujo básico del Caso de Uso siendo explicado.
- Cuando se llegue a la interfaz a explicar.

Una vez tomada la captura de pantalla se procede a la edición de la imagen, adicionándole los indicadores numéricos que sean necesarios para cubrir todas las opciones que esta contenga. El formato de estos indicadores así como su distribución y numeración se especifican en las Pautas de Escritura del



Contenido de la Ayuda.

### 2.5.2. Redacción e integración del contenido de la Ayuda

Para las acciones de confección, mantenimiento e integración del contenido de Ayuda para el subsistema en cuestión se designa al Escritor Técnico. Este rol definido en la metodología RUP es encargado de producir el material de Ayuda del usuario final; por ejemplo las guías del usuario y textos de Ayuda.

Los escritores técnicos tienen la tarea de redactar los contenidos de la Ayuda, para luego llevarlos al formato HTML y añadirlos al subsistema. Para lograr la uniformidad en la redacción se les provee con un documento que reúne todos los requerimientos en cuanto a formato, estilo de redacción y diseño para cada contenido; así como las plantillas más utilizadas y que presentan elementos comunes entre sí. Este rol se caracteriza por adquirir un conocimiento profundo de la aplicación, sus funcionalidades y las diferentes formas de acceder a una misma acción. De esta forma pueden identificar y transmitir efectivamente la información que necesita el usuario considerando sus experiencias previas y habilidades. Para desarrollar su trabajo el escritor técnico debe apoyarse en los programadores y analistas de las funcionalidades que describirá, a la vez que se adueña de sus conocimientos.

Las tareas a cumplir por los escritores técnicos durante el proceso de desarrollo del Subsistema Ayuda son las siguientes:

- ☑ Estudiar las Pautas de Escritura de los Contenidos de Ayuda, en la cual se numeran los elementos que deben aparecer en el sistema de Ayuda y como deben hacerlo.
- ☑ Lograr un total dominio de las herramientas que se utilizan en el desarrollo de la Ayuda, teniendo a su disposición la documentación Web existente sobre ellas.
- ☑ Mantener actualizado el contenido del subsistema. Debe mantener la paridad entre la aplicación y el contenido de Ayuda a medida que vaya avanzando el proyecto.
- ☑ Especificar el formato que será usado y aplicarlo a las páginas de la Ayuda. Basándose en los estilos CSS definidos anteriormente.
- ☑ Asegurarse que las páginas cumplen con las pautas definidas en el documento “Pautas de Escritura del Contenido de la Ayuda” antes de incluirlas en el subsistema.

Las metas a alcanzar son las siguientes:

- ☑ Tener accesible y organizado el Subsistema Ayuda y su documentación general durante el proceso de desarrollo.



- ✓ El usuario final debe disponer de un sistema de Ayuda actualizado que le permita operar correctamente el software, conociendo sus funciones.

Para las tareas de administración de la Ayuda, los escritores técnicos deberán interactuar directamente con las funcionalidades del subsistema; por lo que para el modelado del mismo será tomado en cuenta como el actor Administrador de Ayuda. Como uno de los objetivos es conocer el documento “Pautas de Escritura del Contenido de la Ayuda”, se considera necesario argumentar su contenido.

### 2.5.3. Descripción de las pautas de formato y redacción para el contenido de la Ayuda

Las pautas que deben tener en cuenta al momento de redactar los contenidos son:

- ✓ Definición del contenido de las páginas: el tamaño del texto es mínimo, se realizan páginas de tamaño poco extenso ya que no es conveniente presentarle al usuario textos muy largos pues se corre el riesgo de que no los lea. Se usan colores y gráficos, variando el tamaño de la letra para realzar la página, siempre cuidando su semejanza con el diseño de ZERA.
- ✓ Reducción del tamaño de las imágenes: las imágenes son elementos que dan un aspecto más atractivo a la página y que además refuerzan una explicación textual. Las imágenes que se reflejan en la Ayuda son capturadas directamente de la aplicación. Deben ser guardadas en formato .PNG y de ser necesario comprimida; de esta manera se minimiza el tiempo de carga de las páginas de la Ayuda.
- ✓ Establecimiento de la longitud de la página: el contenido que se muestra en cada página es referente a un Caso de Uso o una sección del mismo. La información se encuentra distribuida de manera que el usuario encuentre lo que necesita en cada una de ellas.

Las restantes definiciones de las Pautas de Escritura para los contenidos de ayuda pueden ser consultadas en el **Anexo 2** del documento.

## 2.6. Modelo de dominio

Los procesos que tienen lugar en el negocio no están definidos claramente; por lo cual se elabora un modelo de dominio para identificar los conceptos y eventos del ambiente del problema. Este artefacto refleja los conceptos propios de la realidad que rodea el sistema que se pretende construir. Está compuesto por las clases conceptuales o entidades, sus relaciones y/o atributos. En el modelo se incluyen las siguientes clases:





Plataforma para la Gestión del Aprendizaje ZERA: representa el sistema al cual se pretende agregar la opción de Ayuda.

Ayuda: representa el subsistema a diseñar.

Consultor de Ayuda: Representa la generalización de todos los roles que tienen acceso a la plataforma ZERA, y su función es acceder al contenido de Ayuda.

Subsistema, modulo, acción: Cada uno gestiona funciones específicas para el trabajo de la plataforma y a su vez se relacionan con la Ayuda, la cual brinda contenidos concretos a cada uno de ellos.

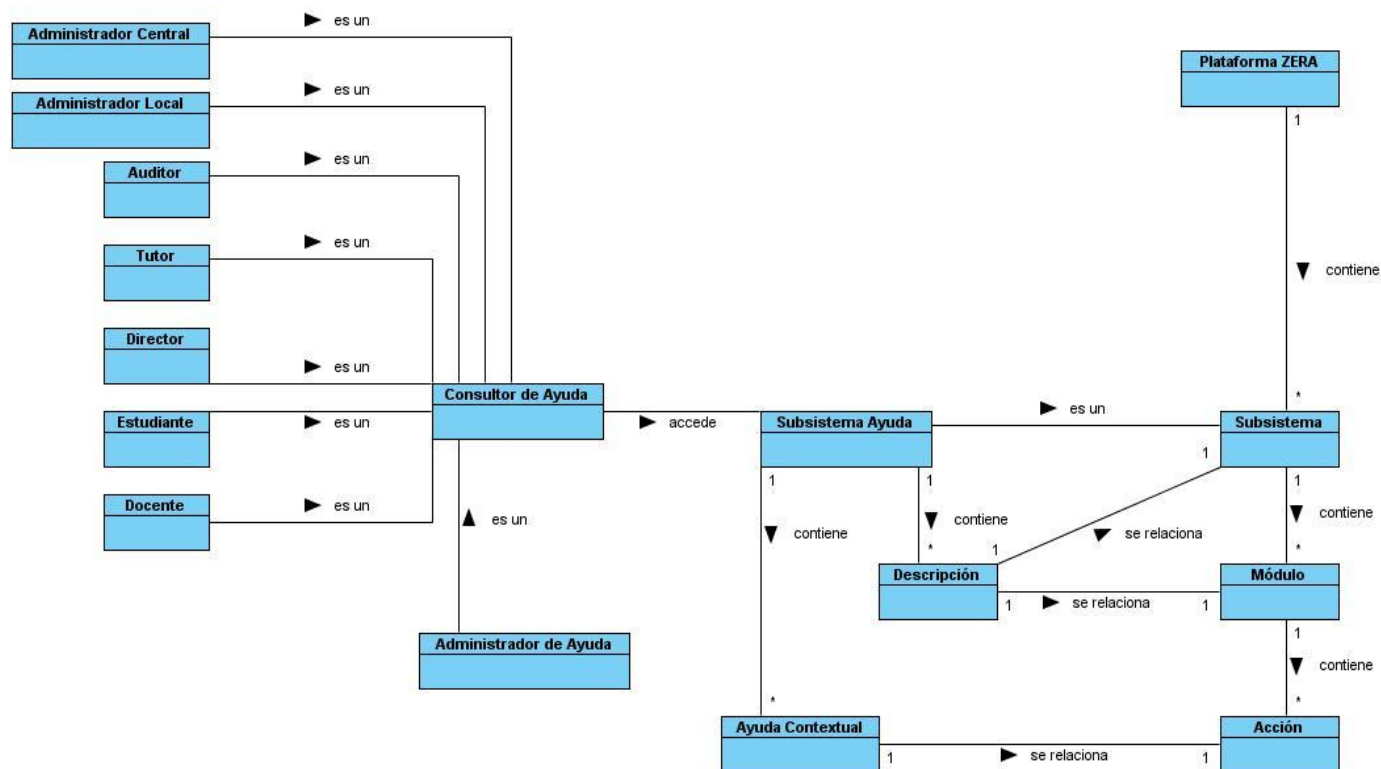


Figura 1. Modelo del dominio.

## 2.7. Especificación de los requerimientos del Subsistema Ayuda

Los requerimientos de un software son capacidades o condiciones que este debe cumplir para satisfacer las necesidades del cliente. La meta principal en esta etapa es identificar y documentar lo que en realidad se necesita, en una forma en que pueda fácilmente ser transmitido al cliente y al equipo de desarrollo. Para el desarrollo del Subsistema Ayuda se realizaron entrevistas con un representante del cliente; en las que se obtuvieron las características funcionales que este desea que cumpla la Ayuda.



Los requerimientos se dividen en 2 categorías:

### 2.7.1. Requerimientos Funcionales

Definen las funciones que el sistema será capaz de realizar y son únicos para cada producto de software.

Los requisitos funcionales (en lo adelante RF) para el subsistema se listan a continuación:

RF 1 – Mostrar Ayuda Contextual

RF 2 – Gestionar Descripciones

RF 2.1 – Mostrar descripciones

RF 2.2 – Ocultar descripciones

RF 3 – Mostrar Ayuda General

RF 4 – Administrar Contenido de Ayuda

RF 4.1 – Incluir Contenido de Ayuda

RF 4.2 – Ver Contenido de Ayuda

RF 4.3 – Modificar los datos del Contenido de Ayuda

RF 4.4 – Eliminar Contenido de Ayuda

RF 4.5 – Mostrar listado de Contenidos de Ayuda

### 2.7.2. Requerimientos No Funcionales

Definen las características que de alguna manera puedan limitar el sistema al momento de su desarrollo; no describen información a guardar, ni funciones a realizar. Tienen numerosas clasificaciones entre las cuales están:

Accesibilidad

El subsistema debe ser generado en tecnología web para ser accesible a través de Internet.

Usabilidad

La Ayuda podrá ser accedida desde cualquier interfaz de la aplicación.

Los elementos gráficos como los iconos deberán contar con un mensaje flotante que señalen el tipo de recurso al que se refiere.

Seguridad

La información solo será visible para los usuarios autorizados.

La base de datos deberá estar encriptada.



### Eficiencia

- ✓ El trabajo realizado por el usuario previo a la invocación de la Ayuda debe ser conservado.
- ✓ La Ayuda debe evolucionar en conjunto con la aplicación.

### Restricciones de Diseño

- ✓ El diseño del subsistema debe seguir los principios de diseño de la plataforma ZERA, siempre que no interfiera con los suyos propios.

### Software

- ✓ El subsistema usará como motor de base de datos PostgreSQL 8.4.
- ✓ Correr sobre cualquier navegador, siendo como mínimo compatible con: Explorer 7.0 o superior, Mozilla Firefox 3.6 o superior, Opera 10.00 o superior, Chrome 7.0 o superior y Safari 4.0 o superior.

## 2.8. Modelo del Sistema

El modelado de Casos de Uso es la técnica más utilizada y sencilla para representar los requisitos del sistema desde la perspectiva del usuario. Los CU generalmente son el punto de partida hacia el análisis de los sistemas orientados a objetos, aunque ellos en sí mismos conllevan un análisis previo de lo que se quiere hacer. Durante el modelado de casos de uso se deben tener en cuenta dos conceptos importantes:

### 2.8.1. Actores

Los actores de un Caso de Uso representan a usuarios u otros sistemas que interactúan con el sistema siendo analizado. El nombre que se le dé a un actor debe reflejar el papel que tiene este en el sistema. Mayormente los actores representan un tipo de usuario, dígase rol, que puede ser ocupado por varias personas a la vez. Esto significa que cuando una persona interactúa con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores.

En el Subsistema Ayuda se definen 2 actores que se corresponden entre sí con una relación de tipo generalización/especialización; donde el Administrador de Ayuda es la especialización del Consultor de Ayuda pues puede iniciar las funcionalidades de este más las suyas propias.

**Tabla 1. Descripción de los actores del Subsistema Ayuda.**

Consultor de Ayuda	Es la persona que está utilizando la plataforma y decide consultar la Ayuda de la misma.
Administrador de Ayuda	Es el miembro del equipo de desarrollo con permisos especiales que puede modificar, incluir o eliminar los componentes del Subsistema Ayuda.

Luego de definir los actores del subsistema se analiza que casos de uso inicia cada uno, lo que da lugar a la creación del diagrama de casos de uso para este subsistema. Inicialmente se genera un caso de uso por cada requerimiento funcional identificado, luego se procede a aplicar los patrones de casos de uso. En este caso se hace uso del patrón CRUD que agrupa las funcionalidades: crear, modificar, ver y eliminar contenidos de ayuda en el caso de uso Administrar Ayuda. Por la afinidad que presenta la funcionalidad de listar los contenidos de ayuda también se asocia con este caso de uso. También se emplea el patrón Múltiples actores-roles comunes, en el cual se relacionan en un rol común aquellos actores que inician simultáneamente un mismo caso de uso. Una vez creado este diagrama, se obtiene una visión general de la Ayuda a nivel de casos de uso.

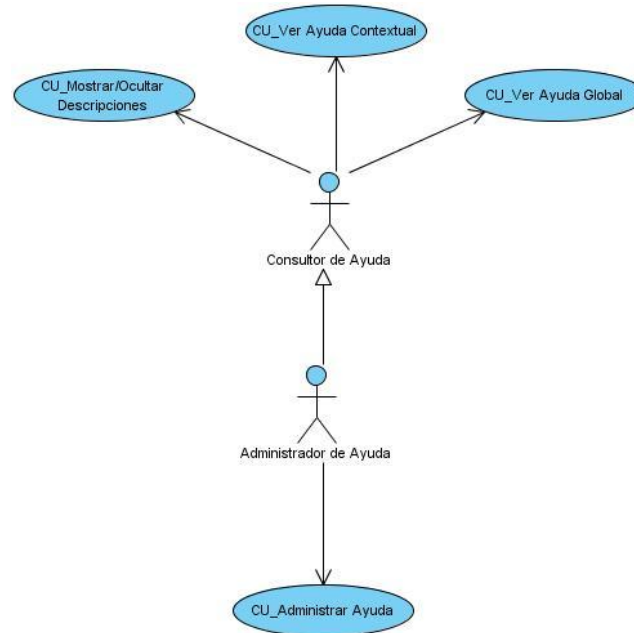


Figura 2. Diagrama de Casos de Uso del Subsistema Ayuda.

### 2.8.2. Casos de Uso

Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otros sistemas para conseguir un objetivo específico. A continuación se describe cómo ocurre el flujo de acciones de la interacción del usuario con el subsistema para el caso de uso Administrar Ayuda. La descripción textual de cada caso de uso se encuentra en el **Anexo 3** del documento.



**Tabla 2. Descripción textual del Caso de Uso Administrar Ayuda.**

Nombre	Administrar Ayuda
Objetivo	Gestionar las acciones que componen el Subsistema Ayuda y los contenidos asociados a las mismas.
Actor	Administrador de Ayuda
Complejidad	Alta
Nivel	Usuario
Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Para ver un contenido de Ayuda, debe estar seleccionado previamente, y el actor debe tener permiso para acceder al mismo, ya sea porque es el autor, un rol superior en jerarquía, o porque se le ha asignado temporalmente.</p> <p>Para modificar un contenido de Ayuda, debe estar seleccionado previamente y el actor debe ser el responsable temporal de la acción.</p> <p>Para incluir un contenido de Ayuda, el actor debe tener el permiso de incluir el elemento.</p> <p>Para eliminar un contenido de Ayuda, debe estar seleccionado previamente y el actor debe tener permiso para eliminar el elemento.</p>
Poscondiciones	Se incluyó, vio, modificó o eliminó el contenido de Ayuda.
Descripción	
<p>El caso de uso se inicia cuando el Administrador de Ayuda accede a la opción que le permite realizar una acción sobre los contenidos de Ayuda. El actor puede incluir, ver o modificar cualquiera de estos contenidos. En caso que el actor decida incluir un nuevo contenido el sistema dará la posibilidad de introducir los datos del mismo. En caso que seleccione la opción para ver un contenido, el sistema mostrará los datos del contenido seleccionado. En caso que seleccione la opción para modificar un contenido, el sistema mostrará los datos para su edición y los guardará al finalizar. En caso que seleccione la opción para eliminar un contenido, el sistema lo eliminará, terminando así el caso de uso</p>	
Requerimientos funcionales	RF 4, RF 4.1, RF 4.2, RF 4.3, RF 4.4, RF 4.5

**Tabla 3. Descripción del flujo básico de eventos de la sección Incluir Contenido del CU Administrar Ayuda.**

Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor accede a la	



opción para realizar una acción sobre el Contenido de Ayuda.	
	2. Muestra un listado de las acciones almacenadas en la base de datos. 3. Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Incluir un nuevo Contenido.</li> <li>• Ver los datos del Contenido de Ayuda. Ver Sección: “Ver Datos de Contenido”.</li> <li>• Modificar los datos del Contenido de Ayuda. Ver Sección: “Modificar Datos de Contenido”.</li> <li>• Eliminar los datos del Contenido de Ayuda. Ver Sección: “Eliminar Contenido”.</li> </ul>
4. Selecciona la opción de incluir un nuevo contenido de Ayuda.	
	5. Brinda la posibilidad de seleccionar o introducir o seleccionar los siguientes datos para crear el contenido de Ayuda: Nombre de la Acción. Apodo. Descripción de la Acción. Dirección URL. Y Permite: <ul style="list-style-type: none"> <li>• Guardar los datos.</li> <li>• Cancelar la operación.</li> <li>• Salir de la vista actual.</li> </ul>
6. Selecciona o introduce los datos para crear el contenido de Ayuda.	
7. Selecciona la opción para guardar los datos.	
	8. Valida los datos. 9. Crea el contenido de Ayuda. 10. Muestra un mensaje de información.
	11. El caso de uso termina.

**Tabla 4. Descripción del flujo alternativo de eventos de la sección Incluir Contenido del CU Administrar Ayuda.**

\*.a El actor selecciona la opción de Cancelar



Acciones del actor	Respuesta del sistema
	*.a.1 Elimina los datos creados.
	*.a.2 Muestra un mensaje de información.
	*.a.3 Regresa a la vista anterior.
	*.a.4 El caso de uso termina.
*.b El actor selecciona la opción de Salir de la vista actual.	
Acciones del actor	Respuesta del sistema
	*.b.1 Regresa a la vista anterior.
	*.b.2 El caso de uso termina.
7. a Existen datos incompletos.	
Acciones del actor	Respuesta del sistema
	7. a.1 Muestra un mensaje de información.
	7. a.2 Muestra un indicador sobre cada campo vacío.
	7. a.3 Regresa al paso 4 del Flujo Básico.
7. a Existen datos incorrectos.	
Acciones del actor	Respuesta del sistema
	7. a.1 Muestra un mensaje de información.
	7. a.2 Muestra un indicador sobre cada campo incorrecto.
	7. a.3 Regresa al paso 4 del Flujo Básico.

**Tabla 5. Descripción del flujo básico de eventos de la sección Ver Datos de Contenido del CU Administrar Ayuda.**

Acciones del actor	Respuesta del sistema
1. Selecciona la opción de ver los datos del contenido de Ayuda.	
	2. Muestra los datos del contenido de Ayuda: Nombre de la Acción. Apodo Descripción de la Acción. Dirección URL. 3. Y permite: <ul style="list-style-type: none"> <li>• Modificar los datos del contenido de Ayuda.</li> <li>• Eliminar los datos del contenido de Ayuda.</li> <li>• Salir de la vista actual.</li> </ul>



4. Selecciona la opción de salir de la vista actual.	
	5. Muestra la vista anterior.
	6. El caso de uso termina.

**Tabla 6. Descripción del flujo alterno de eventos de la sección Ver Datos de Contenido del CU Administrar Ayuda**

3. a El actor selecciona la opción para Modificar los datos del contenido de Ayuda.	
Acciones del actor	Respuesta del sistema
	3. a.1. Brinda la posibilidad de modificar los datos del contenido de Ayuda. Ver Sección: “Modificar Datos de Contenido”.
	3. a.2. El caso de uso termina.
3. a El actor selecciona la opción de Eliminar los datos del contenido de Ayuda.	
Acciones del actor	Respuesta del sistema
	3. a.1. Brinda la posibilidad de eliminar los datos del contenido de Ayuda. Ver Sección: “Eliminar Contenido”.
	3. a.2. El caso de uso termina.

**Tabla 7. Descripción del flujo básico de eventos de la sección Modificar Datos de Contenido del CU Administrar Ayuda.**

Acciones del actor	Respuesta del sistema
1. Selecciona la opción de modificar los datos del contenido de Ayuda.	
	2. Muestra los datos del contenido de Ayuda y brinda la posibilidad de cambiar sus valores. 3. Permite además: • Actualizar los datos. Cancelar la operación.
4. Modifica los datos que necesite.	
5. Selecciona la opción para actualizar los datos.	
	6. Valida los datos.
	7. Actualiza los datos del contenido de Ayuda.
	8. Muestra un mensaje de información.
	9. Muestra los datos del contenido de Ayuda.





	10. El caso de uso termina.
--	-----------------------------

**Tabla 8. Descripción del flujo alternativo de eventos de la sección Modificar Datos de Contenido del CU Administrar Ayuda.**

*a El actor selecciona la opción de Cancelar	
Acciones del actor	Respuesta del sistema
	*.a.1 Elimina los datos creados.
	*.a.2 Muestra un mensaje de información.
	*.a.3 Regresa a la vista anterior.
	*.a.4 El caso de uso termina.
5. a Existen datos incompletos.	
Acciones del actor	Respuesta del sistema
	5. a.1 Muestra un mensaje de información.
	5. a.2 Muestra un indicador sobre cada campo incompleto.
	5. a.3 Regresa al paso 2 del Flujo Básico.
5. b Existen datos incorrectos.	
Acciones del actor	Respuesta del sistema
	5. b.1 Muestra un mensaje de información.
	5. b.2 Muestra un indicador sobre cada campo incorrecto.
	5. b.3 Regresa al paso 2 del Flujo Básico.

**Tabla 9. Descripción del flujo básico de eventos de la sección Eliminar Contenido del CU Administrar Ayuda.**

Acciones del actor	Respuesta del sistema
1. Selecciona la opción de eliminar un contenido de Ayuda.	
	2. Muestra un mensaje de alerta. Y permite: <ul style="list-style-type: none"> <li>• Aceptar.</li> <li>• Cancelar.</li> </ul>
3. Selecciona la opción de Aceptar.	
	4. Elimina el contenido de Ayuda.
	5. Muestra un mensaje de información.
	6. El caso de uso termina.

**Tabla 10. Descripción del flujo alternativo de eventos de la sección Eliminar Contenido del CU Administrar**

**Ayuda.**

3.a El actor selecciona la opción de Cancelar	
Acciones del actor	Respuesta del sistema
	3. a.1 Regresa a la vista anterior.
	3. a.2 El caso de uso termina.

**2.9. Conclusiones**

En este capítulo se estudió el funcionamiento de la Plataforma ZERA para obtener un mejor entendimiento de los procesos a reflejar en el Subsistema Ayuda. Como resultado de esto se describió el proceso de creación de los contenidos de ayuda y se definieron las pautas a seguir para lograr su uniformidad en cuanto a formato, estilo de redacción y diseño. Se identificaron los requerimientos funcionales y no funcionales que rigen el comportamiento del subsistema. Seguidamente se detalló el modelo de Casos de Uso del Subsistema Ayuda, derivados de los requerimientos funcionales, en el que se incluyen las descripciones textuales de los mismos. Estos artefactos son una expresión de los requisitos del software enfocados hacia el usuario final y constituyen el punto de partida hacia el flujo de análisis y diseño del subsistema. Por tanto, se han puntualizado los procedimientos a realizar antes de comenzar a diseñar el Subsistema Ayuda, mediante la descripción de las principales tareas a realizar, sus responsables y objetivos.

## Análisis y Diseño del Subsistema Ayuda para la Plataforma ZERA



### 3.1. Introducción

La metodología RUP define el flujo de trabajo de Análisis y Diseño con el objetivo de transformar los requerimientos a un modelo de diseño, teniendo en cuenta una arquitectura que permita adaptar el sistema al entorno de implementación que se utilice. Dicho modelo debe ser lo suficientemente específico para que no existan ambigüedades.

### 3.2. Modelo de Datos.

Al utilizar una base de datos para gestionar la información, se refleja un modelo parcial de la realidad en un contexto informático. Para una mejor aproximación a este objetivo, se debe realizar un modelo de datos antes de crear físicamente la base de datos. Esto permite observar los patrones de datos y el uso potencial de los mismos.

Los Modelos de Datos Conceptuales tienen como objetivo describir las estructuras y restricciones para la integridad de los mismos. Con ellos se logra una visión general del negocio o institución que se analiza. Son fundamentalmente usados en la etapa de Análisis de los sistemas y representan tanto las entidades presentes en el dominio del problema a resolver como las relaciones entre ellas. El Modelo Entidad – Relación es el ejemplo más utilizado en esta clasificación. Estos modelos incorporan las entidades relevantes para el sistema a diseñar; así como sus relaciones y atributos.

Luego de crear el modelo se procede a crear físicamente las tablas de la base de datos; para lo que se hace una tabla para cada entidad del diagrama. Seguidamente se modelan las relaciones entre entidades, colocando los elementos identificadores según el tipo de conexión.

#### 3.2.1. Descripción de las entidades presentes en el Modelo Entidad – Relación

El modelo de datos del Subsistema Ayuda está compuesto de 1 entidad, las cuales modelan la arquitectura de la información de la plataforma ZERA. La entidad **tbHelpContext** representa las acciones que construyen las vistas de usuario y a las cuales se les asigna un contenido de Ayuda. Cada acción solo puede pertenecer a un único módulo y subsistema, aunque cada módulo puede tener tantas acciones asociadas

como necesite. Esta entidad almacena un título de bienvenida a esa acción, nombre de la acción, apodo (cuyo formato es *subsistema.módulo.acción*), descripción, réplica para garantizar el traslado de los datos entre los servidores de la aplicación y la dirección URL donde se encuentra el contenido de Ayuda asociado a la misma. Seguidamente se muestra el diagrama Entidad – Relación del Subsistema Ayuda:

tb_HelpContext		
<b>+id</b>	<b>smallint(5)</b>	<b>Nullable = false</b>
name	varchar(100)	Nullable = false
title	varchar(100)	Nullable = false
nickname	varchar(100)	Nullable = false
description	clob	Nullable = true
content_url	varchar(255)	Nullable = true
replication	varchar(5)	Nullable = false

Figura 3. Diagrama Entidad – Relación de la Base de Datos del Subsistema Ayuda.

### 3.3. Modelo de análisis

Con la realización del modelo de análisis se logra suavizar la transición al diseño del sistema y es más factible el cambio a una plataforma de programación diferente a la precisada. Además de brindar una visión general de la propuesta de sistema.

#### 3.3.1. Descripción de las clases del análisis

Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del mismo. Como característica tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. Para el análisis se definen las siguientes clases:

Clase Interfaz: Modelan la interacción entre el sistema y sus actores.

Clase Controladora: Coordina el trabajo de cada caso de uso, definen el flujo de control y las transacciones dentro de un caso de uso delegando el trabajo a otros objetos.

Clase Entidad: Estas clases modelan información, conceptos y sucesos que ocurren en el mundo real. La fuente principal de obtención son las clases entidades del negocio y el glosario de términos que se ha ido elaborando.

Los diagramas de clases del análisis para los casos de uso del Subsistema Ayuda pueden ser consultados en el **Anexo 4** del documento.



### 3.3.2. Diagramas de interacción del análisis

Los diagramas de interacción están clasificados como diagramas de comportamiento y muestran los objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos. Se dividen en diagramas de secuencia y diagramas de colaboración, siendo los primeros más recomendados para el diseño. Para elaborar estos diagramas se toman en consideración cada uno de los escenarios para cada caso de uso. Los diagramas de colaboración se podrán consultar en el **Anexo 5**.

### 3.4. Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a modelar. Al mismo tiempo constituye una abstracción del modelo de implementación y del código fuente, resultando en su evaluación como una entrada esencial a las actividades de implementación y prueba.

Al elaborar el modelo de diseño se desea adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación y componentes reutilizables. Con esto se crea un punto de partida para la implementación, siendo lo suficientemente claro para que el sistema pueda ser desarrollado sin ambigüedades.

#### 3.4.1. Patrones utilizados

Para el desarrollo de aplicaciones web es objetivo realizar un diseño que separe la presentación de la aplicación de su parte lógica, pues se minimiza el riesgo de un error al realizar cambios tanto en la interfaz como en las bases de datos. El patrón arquitectónico Modelo-Vista-Controlador propone la división del subsistema en 3 paquetes (capas) diferentes. Donde el Modelo representa las estructuras de datos y contiene funciones para gestionar la información de la base de datos. La Vista constituye la información presentada al usuario y que recibe del modelo. El Controlador se comporta como mediador entre el Modelo, la Vista y cualquier otro recurso necesario para crear una página. Con el uso de este patrón se logra separar la programación del servidor de la interfaz del cliente lo que permite reutilizar componentes con mayor facilidad.



### 3.4.2. Descripción de las clases del diseño

Para el diseño de aplicaciones web se definen clases, las cuales mayormente guardan estrecha relación con las anteriormente definidas durante el análisis. Una clase de diseño es similar a las que serán creadas en la implementación del sistema, pues para especificar una clase del diseño se utiliza el lenguaje de programación seleccionado. Al momento de confeccionar los diagramas que representaran la aplicación web se asocian estereotipos a las clases utilizadas y a las relaciones entre estas. Algunos de los más utilizados y que son empleados en el diseño del Subsistema Ayuda son:

Client Page: Una instancia de «Client Page» es una página Web con formato HTML. Son mostradas por un navegador de internet y pueden contener scripts que son interpretados por el navegador. Las páginas del cliente pueden estar asociadas con otras páginas del cliente o del servidor.

Server Page: Una página del servidor (server page en inglés) representa una página web que tiene scripts que son ejecutados por el servidor. Estos scripts actúan recíprocamente con recursos en el servidor (bases de datos, lógica de negocio o sistemas externos). Las operaciones que se definen en estas clases representan las funciones a implementar en el script. Así como los atributos representan variables visibles en el ámbito de la página y que serán accesibles para todas las funciones de la página.

Form: Una clase estereotipada como «form» es una colección de campos de entrada de datos que son parte de una página del cliente. Sus atributos representan los campos de la entrada del formulario de HTML y no tiene operaciones. Tiene una correspondencia directa con la etiqueta <FORM> de XHTML.

Frameset: Un juego de frames es un contenedor de múltiples páginas web y es un área visible dividida en rectángulos o frames.

Link: Esta relación corresponde directamente con una etiqueta <a> (ancla) de HTML. En un diagrama de clases un link es una asociación entre una página del cliente y cualquier otra página del cliente o una página del servidor.

Submit: Esta asociación es únicamente entre una clase «form» y una página del servidor, en la cual la página del servidor procesa los datos enviados por «form».

Build: Identifica qué página del servidor es responsable de la creación de una página del cliente. Una página del servidor puede construir múltiples páginas del cliente, pero una página del cliente sólo puede ser construida a través de una página del servidor. Esta relación siempre es unidireccional.

Redirect: Esta es también una relación unidireccional que indica que una página Web redirige hacia otra. Puede existir entre una página del servidor con otra página del servidor y de una página del cliente hacia



otra página del cliente denotando que la página de destino es solicitada automáticamente por el navegador.

Los diagramas de clases del diseño para cada Caso de Uso pueden ser consultados en el **Anexo 6** del documento.

### 3.4.3. Diagramas de interacción del diseño

Estos diagramas muestran un patrón de interacción entre los objetos que conforman un sistema. Tienen 2 clasificaciones, ambas basadas en la misma información, pero cada una enfatizando un aspecto en particular: diagramas de Secuencia y de Colaboración.

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin importar su orden. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El uso de estos diagramas da la posibilidad de representar los mensajes en función del tiempo y de añadir restricciones temporales. Los diagramas de secuencia están situados en el **Anexo 7**.

### 3.5. Conclusiones

En este capítulo se definió la estructura del modelo de datos para el Subsistema Ayuda, describiendo la entidad que lo compone y sus atributos. Se realizó un análisis a fondo de los requerimientos identificados para la elaboración del Subsistema Ayuda, describiendo las clases y el comportamiento de los objetos que lo componen. Luego los resultados de este análisis fueron transformados en el diseño del subsistema; especificando el patrón arquitectónico a utilizar, así como la estructura de las clases y su comportamiento en cada caso. El modelo de diseño obtenido como resultado, se considera la entrada principal para los siguientes flujos de trabajo Implementación y Prueba.

## Implementación y Pruebas del Subsistema Ayuda para la Plataforma ZERA



### 4.1. Introducción

En este capítulo se describe el procedimiento de desarrollo de la propuesta de solución del Subsistema Ayuda para la plataforma ZERA utilizando el framework de desarrollo Symfony. Se describe la estructura y el funcionamiento del plugin que se crea como respuesta a los requerimientos funcionales. Además de caracterizar el sistema creado para la administración del contenido de la Ayuda. Se definen los tipos de pruebas a realizar para asegurarse que el comportamiento requerido es correcto y que cumpla con todos los requerimientos. Además de realizar los casos de prueba que permitirán evaluar las funcionalidades y calidad del subsistema.

### 4.2. Flujo de Trabajo de Implementación

El objetivo principal de la fase de Construcción definida por RUP es especificar la organización del código y planificar las integraciones de sistema necesarias en cada iteración; además de implementar las clases y subsistemas detallados durante el diseño. Para conseguir estos objetivos se debe estructurar el Modelo de Implementación que incluye componentes, subsistemas y el producto final. Conjuntamente se crea el Plan de Integración, se realizan la implementación de componentes y la integración de subsistemas. Procediendo luego a la validación de componentes y subsistemas. Entre los artefactos generados en este flujo de trabajo se tiene el diagrama de componentes.

#### 4.2.1. Diagrama de componentes

Un componente representa una unidad de código (fuente, binario o ejecutable) que permite mostrar las dependencias en tiempo de compilación y ejecución. Las instancias de componentes de software muestran unidades de software en tiempo de ejecución y generalmente ayudan a identificar sus dependencias y su localización en nodos. Pueden mostrar también qué interfaces implementan y qué objetos contienen. Su representación es un rectángulo atravesado por una elipse y dos rectángulos más pequeños.

Un diagrama de componentes muestra las dependencias lógicas entre los componentes de un software y



representa la estructura del Modelo de Implementación. Incluye componentes (que representan al código fuente), subsistemas de implementación y sus relaciones.

Estos diagramas resultan muy útiles para organizar los subsistemas de implementación en capas o paquetes. Un paquete en un diagrama de componentes representa una división física del sistema y se organizan según la jerarquía de capas existente. A continuación se muestra el diagrama de componentes del Subsistema Ayuda donde se representa la división por capas siguiendo la solución descrita por el patrón Modelo-Vista-Controlador.

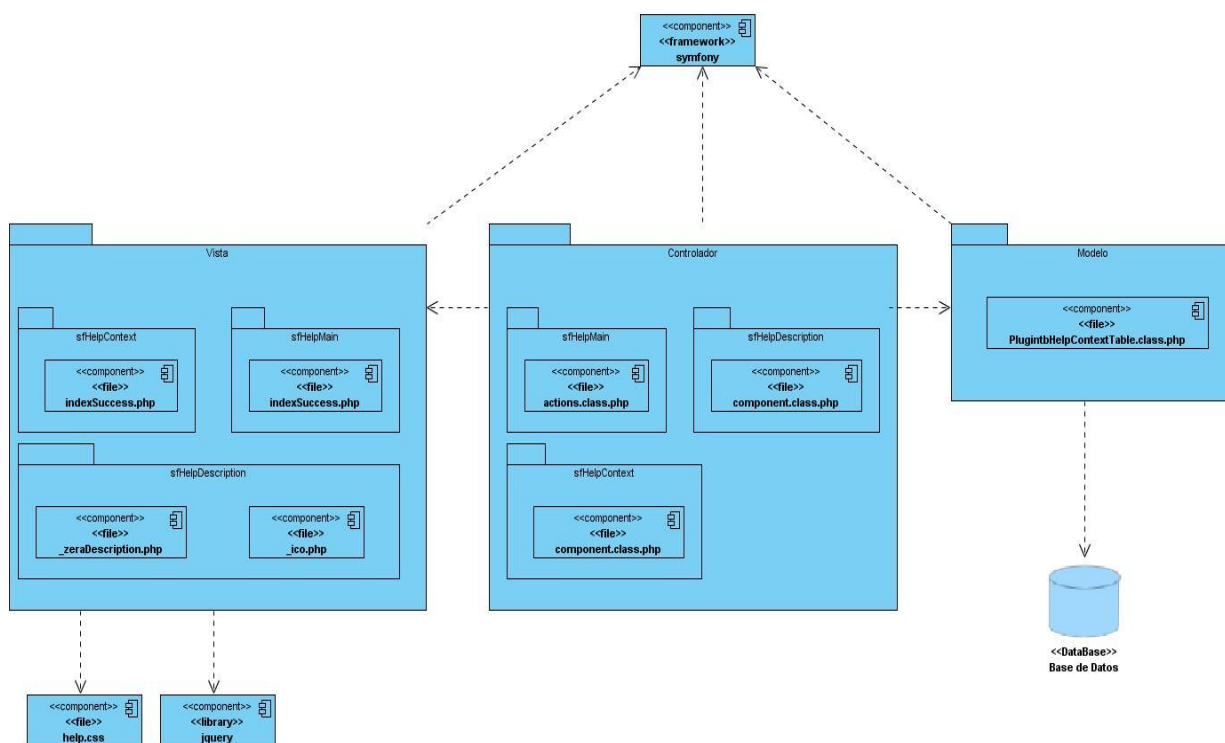


Figura 4. Diagrama de Componentes del Subsistema Ayuda.

#### 4.2.2. Diagrama de despliegue

Otro de los artefactos generados en el flujo de trabajo de implementación es el diagrama de despliegue, donde se representan los nodos que componen el sistema. Además de especificar los protocolos que utilizan para comunicarse. Como diagrama de despliegue para el Subsistema Ayuda se hace uso del diagrama definido en la arquitectura de la Plataforma ZERA. Debido a que el plugin y el subsistema de gestión de la Ayuda son parte de la misma y por tanto, están alojados en el mismo servidor y utilizan la

misma conexión a la base de datos.

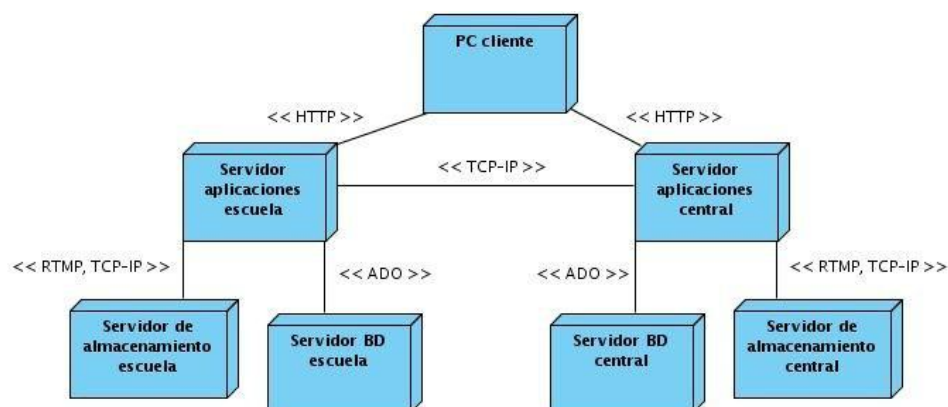


Figura 5. Diagrama de Despliegue del Subsistema Ayuda.

#### 4.2.3. Plugin sfAOHelpPlugin

En ocasiones, es necesario reutilizar código desarrollado para alguna aplicación. Los plugin de Symfony permiten agrupar código diseminado por diferentes archivos y reutilizarlo en otros proyectos. Además posibilitan encapsular clases, filtros, archivos de configuración, tareas, módulos, esquemas y extensiones para el modelo, los archivos estáticos, entre otros. La forma en la que Symfony carga los plugin permite que los proyectos puedan utilizarlos como si fueran parte del propio framework. Estos son fáciles de instalar, actualizar y desinstalar. Las aplicaciones Symfony sólo comparten el modelo, por tanto, los plugin proveen de una vía para compartir más componentes entre aplicaciones.

Para la solución se crea un plugin que encapsula las clases, archivos de configuración y hojas de estilos CSS que son de uso común para todos los subsistemas de la plataforma ZERA a los cuales se les incluye la Ayuda. El plugin se nombra *sfAOHelpPlugin* cumpliendo con las normas establecidas en la arquitectura definida para la plataforma ZERA. El directorio del plugin se organiza de forma muy similar al directorio de un proyecto. Los archivos del plugin se organizan de forma adecuada para que Symfony pueda cargarlos automáticamente cuando sea necesario.

En las aplicaciones Symfony, el acceso y la modificación de la información almacenada en la base de datos se realiza mediante objetos; de esta forma nunca se accede de forma explícita a la misma. Este comportamiento permite un alto nivel de abstracción y permite una fácil portabilidad.

Las bases de datos son estructuras relacionales, así como PHP 5 y Symfony están orientados a objetos. Para acceder de forma efectiva a la base de datos desde un contexto orientado a objetos, es necesaria



una interfaz que traduzca la lógica de los objetos a la lógica relacional. Esta interfaz se llama ORM (object-relational mapping) y está formada por objetos que permiten acceder a los datos; además de contener en sí mismos el código necesario para hacerlo. Symfony utiliza Doctrine como ORM. Para realizar esa traducción, el ORM necesita una descripción del modelo relacional, a la que se llama “*esquema*” (schema). En el esquema se definen las tablas, sus relaciones y las características de sus columnas. La sintaxis que utiliza Symfony para definir los esquemas hace uso del formato YAML<sup>12</sup>.

El modelo de datos del plugin del Subsistema Ayuda cuenta con 1 tabla: *tbHelpContext* para las acciones y la Ayuda según el contexto. Para construir las clases del modelo que necesita la capa del ORM se construye un esquema de la base de datos. Estas clases se generan mediante una tarea de línea de comandos llamada `symfony doctrine:build-model`, lo que reduce el tiempo de ejecución de la aplicación. Al ejecutar ese comando, se analiza el esquema y se generan automáticamente las clases base del modelo.

El **sfAOHelpPlugin** puede ser utilizado en cualquier proyecto symfony que requiera una ayuda contextual para sus usuarios finales. La forma en que se presenta la información a los usuarios se debe personalizar según el diseño y las características del producto donde esta se quiera incluir. Cuenta con 3 módulos:

**sfHelpDescription:** Cualquier interfaz de la plataforma puede hacer uso de este módulo si incluye el componente (`<?php include_component('sfHelpDescription', 'zeraDescription') ?>`). Este se encarga del proceso de búsqueda y muestra de las descripciones contextuales en la base de datos del plugin. Cada descripción es propia para cada acción.

**sfHelpMain:** Es el módulo de bienvenida a la ayuda de la plataforma. Brinda una serie de enlaces a los usuarios, en los que se obtiene información acerca de las características de ZERA.

**sfHelpContext:** Muestra la ayuda, según el contexto donde se encuentre el usuario, mediante la inclusión del componente (`<?php include_component('sfHelpDescription', 'ico') ?>`). Este permite a cada acción tener en su interfaz un ícono de ayuda, si en la base de datos del plugin existe un contenido para esa acción, entonces dicho ícono es visible. Al hacer clic sobre el mismo se hace una petición al plugin y la presentación de la ayuda contextual se hace a través de una ventana emergente. De esta manera se preserva el trabajo que el usuario viene realizando previo a la invocación de la ayuda.

---

<sup>12</sup> YAML es un formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python y Perl.



#### 4.2.4. Sistema para la Administración de la Ayuda

Para la gestión del contenido de la Ayuda, se define la creación de un sistema que implemente las operaciones a realizar con dicha información. En las aplicaciones web, las operaciones de acceso a los datos se dividen de la siguiente manera:

- Insertar un registro (creation).
- Obtener uno o varios registros (retrieval).
- Modificar un registro o alguna de sus columnas (modification).
- Borrar un registro (deletion).

La presentación de la interfaz es importante y por eso incluye opciones como el filtrado, la paginación y la ordenación de datos. La parte de administración generada automáticamente con Symfony tiene calidad suficiente como para entregarla al cliente formando parte de la aplicación que se ha desarrollado. No siendo este el caso en la administración del Subsistema Ayuda; dado que solo será permitida su gestión por los especialistas que forman parte del equipo de desarrollo.

#### 4.3. Pruebas de Software

Para determinar el nivel de calidad del producto desarrollado, este se sometió a un proceso de validación. Las pruebas de software combinaron las operaciones del sistema bajo condiciones controladas, junto a la evaluación de sus resultados. Fueron enfocadas principalmente a la valoración y determinación de la calidad del producto; teniendo como objetivo fundamental encontrar y exponer las debilidades en el mismo. Los niveles de prueba describen las mismas de menor a mayor escala, entre ellos se tiene:

**Pruebas de integración:** Se centraron en examinar la coherencia semántica (dinámica o estática) entre los diferentes módulos. Fueron realizadas por etapas, incluyendo progresivamente más módulos en cada prueba.

**Pruebas de sistema:** Comprobaron el sistema en su totalidad. Estuvieron dirigidas a demostrar qué objetivos no se cumplen, por lo que no fue posible realizarlas hasta que todos los objetivos estuvieron documentados.

**Pruebas de aceptación:** Fueron realizadas por el cliente, quien decidió qué prueba aplicarle al producto antes de aceptar su entrega. Se realizaron una vez pasadas las pruebas de integración.

Cada tipo de prueba tiene un objetivo específico y una técnica que lo soporte. Los tipos de pruebas empleados son:



**Integración incremental:** Se fueron probando las funcionalidades a medida que se añadieron a los módulos.

**Usabilidad:** Comprobó la facilidad de uso en las interfaces de usuario, así como el nivel de comprensión de los mensajes mostrados. A la vez que verificó que no existiera un número excesivo de opciones en pantalla.

**Seguridad:** Verificó la protección frente a entradas no deseadas.

**Resistencia:** Se probó la robustez del sistema frente al mal uso de la aplicación por parte de ciertos usuarios.

Es importante la planificación de las pruebas, definiendo la estrategia a seguir en cada una de ellas. La estrategia de pruebas describe el enfoque y los objetivos generales de las actividades de pruebas. En ella se incluyeron los niveles de pruebas a seguir, los tipos de pruebas a ser ejecutadas y los casos de pruebas diseñados para lograr los objetivos. Los casos de prueba reúnen un conjunto de entradas, condiciones de ejecución y resultados esperados durante la realización de cualquier función del software.

Para comprobar la calidad del Subsistema Ayuda se empleó el método de prueba de caja negra con la técnica de la Partición de Equivalencia. Con esta técnica se examinaron los valores válidos e inválidos de las entradas del sistema y el diseño de casos de prueba para la misma se basó en la evaluación de clases de equivalencia que representan dichos valores. Las salidas esperadas también fueron analizadas pues se deben generar resultados para todas las clases de equivalencia.

#### 4.3.1. Descripción de los Casos de Prueba del Subsistema Ayuda

El diseño de casos de prueba para la técnica de partición equivalente estuvo basado en una evaluación de las clases de equivalencia para una condición de entrada. “Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada.” (46) A continuación se muestran los diferentes escenarios que componen el caso de prueba Administrar Ayuda. El resto de los casos de prueba pueden ser consultados en el **Anexo 8** del presente documento.

**Tabla 11. Escenario Incluir Contenido del Caso de Prueba Administrar Ayuda.**

ID del escenario	Escenario	Nombre Acción.	Apo-do	Des-crip-ción	Direc-ción URL	Respuesta del sistema	Flujo cen-tral
EC 1.1:	Realizar una acción so-					Muestra un listado de las acciones almacenadas en la base de datos. Brinda la posibilidad de realizar las acciones:	Acciones



	bre un contenido de Ayuda.					<ul style="list-style-type: none"> <li>Incluir un nuevo Contenido de Ayuda.</li> <li>Ver los datos del Contenido de Ayuda. Ver Sección 1: "Ver datos del Contenido de Ayuda".</li> <li>Modificar los datos del Contenido de Ayuda. Ver Sección 2:"Modificar datos del Contenido de Ayuda".</li> <li>Eliminar los datos del Contenido de Ayuda. Ver Sección 3:"Eliminar datos del Contenido de Ayuda".</li> </ul>	
EC 1.2:	Selecciona la opción para incluir el contenido					<p>Brinda la posibilidad de introducir o seleccionar los siguientes datos para crear el contenido de Ayuda:</p> <ul style="list-style-type: none"> <li>Nombre de la Acción.</li> <li>Apodo.</li> <li>Descripción de la Acción.</li> <li>Dirección URL.</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>Guardar los datos.</li> <li>Salir de la vista actual.</li> </ul>	<p>Acciones</p> <p>Incluir</p>
EC 1.3:	Introduce los datos para crear el contenido.	V	V	V	V	<p>Valida los datos. Crea el contenido de Ayuda. Muestra un mensaje de información.</p>	<p>Acciones</p> <p>Incluir</p> <p>Guardar</p>
EC 1.4:	Selecciona la opción de Salir de la vista actual.					<p>Regresa a la vista anterior.</p>	<p>Acciones</p> <p>Incluir</p> <p>Vuelve a la vista anterior</p>
EC 1.5:	Existen datos incompletos.	I	V	V	V	<p>Muestra un mensaje de información. Muestra un indicador sobre cada campo vacío.</p> <p>Brinda la posibilidad de introducir o seleccionar los siguientes datos para crear el contenido de Ayuda:</p> <ul style="list-style-type: none"> <li>Nombre de la Acción.</li> <li>Apodo.</li> <li>Descripción de la Acción.</li> <li>Dirección URL.</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>Guardar los datos.</li> <li>Salir de la vista actual.</li> </ul>	<p>Acciones</p> <p>Incluir</p> <p>Guardar</p>
		V	I	V	V		
		V	V	I	V		
		V	V	V	I		
EC 1.6:	Existen datos incorrectos.	I	V	NA	V	<p>Muestra un mensaje de información. Muestra un indicador sobre cada campo incorrecto.</p> <p>Brinda la posibilidad de introducir o seleccionar los siguientes datos para crear el contenido de Ayuda:</p>	<p>Acciones</p> <p>Incluir</p>
		V	I	NA	V		



		V	V	NA	I	<ul style="list-style-type: none"> <li>Nombre de la Acción.</li> <li>Apodo.</li> <li>Descripción de la Acción.</li> <li>Dirección URL.</li> </ul> Y permite: <ul style="list-style-type: none"> <li>Guardar los datos.</li> <li>Salir de la vista actual.</li> </ul>	Guardar
--	--	---	---	----	---	--	---------

**Tabla 12. Escenario Ver Contenido del Caso de Prueba Administrar Ayuda.**

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1.1:	Selecciona la opción de ver los datos del contenido de Ayuda.	Muestra los datos del contenido de Ayuda: <ul style="list-style-type: none"> <li>Nombre de la Acción.</li> <li>Apodo</li> <li>Descripción de la Acción.</li> </ul> Y permite: <ul style="list-style-type: none"> <li>Modificar los datos del contenido de Ayuda.</li> <li>Eliminar los datos del contenido de Ayuda.</li> <li>Salir de la vista actual.</li> </ul>	Acciones Mostrar
EC 1.2:	Selecciona la opción de Modificar los datos del contenido de Ayuda.	Brinda la posibilidad de modificar los datos del contenido de Ayuda. Ver Sección: "Modificar datos del contenido de Ayuda".	Acciones Mostrar Editar
EC 1.3:	Selecciona la opción de Eliminar los datos del contenido de Ayuda.	Brinda la posibilidad de modificar los datos del contenido de Ayuda. Ver Sección: "Eliminar datos del contenido de Ayuda".	Acciones Mostrar Eliminar
EC 1.4:	Selecciona la opción de Salir de la vista actual.	Regresa a la vista anterior.	Acciones Mostrar Vuelve a la vista anterior

**Tabla 13. Escenario Modificar Contenido del Caso de Prueba Administrar Ayuda.**

ID del escenario	Escenario	Nombre Acción.	Apodo	Descripción	Dirección URL	Respuesta del sistema	Flujo central
EC 1.1:	Seleccio-					Muestra los datos del contenido de Ayuda:	Acciones



	na la opción de modificar los datos del Contenido de Ayuda.					<ul style="list-style-type: none"> <li>Nombre de la Acción.</li> <li>Apodo.</li> <li>Descripción de la Acción.</li> <li>Dirección URL.</li> </ul> Brinda la posibilidad de cambiar sus valores y permite además: <ul style="list-style-type: none"> <li>Actualizar los datos.</li> <li>Cancelar la operación en cualquier momento.</li> </ul>	<i>Editar</i>
EC 1.2:	Selecciona la opción de actualizar los datos.	V	V	V	V	Valida los datos. Actualiza los datos del contenido de Ayuda. Muestra un mensaje de información. Muestra los datos del contenido de Ayuda. Ver Sección: "Ver datos de contenido de Ayuda."	<i>Acciones</i> <i>Editar</i> <i>Actualizar</i>
EC 1.3:	Selecciona la opción de Cancelar					Elimina los datos creados. Muestra un mensaje de información.	<i>Acciones</i> <i>Editar</i> <i>Vuelve a la interfaz anterior</i>
EC 1.4:	Existen datos incompletos.	I	V	V	V	Muestra un mensaje de información. Muestra un indicador sobre cada campo vacío. Brinda la posibilidad de seleccionar o introducir o seleccionar los siguientes datos para actualizar el contenido de Ayuda: <ul style="list-style-type: none"> <li>Nombre de la Acción.</li> <li>Apodo.</li> <li>Descripción de la Acción.</li> <li>Dirección URL.</li> </ul> Y permite: <ul style="list-style-type: none"> <li>Actualizar los datos.</li> <li>Salir de la vista actual.</li> </ul>	<i>Acciones</i> <i>Editar</i> <i>Actualizar</i>
		V	V	V	V		
		V	I	V	V		
		V	V	I	V		
		V	V	V	I		
EC 1.5:	Existen datos incorrectos.	I	V	NA	V	Muestra un mensaje de información. Muestra un indicador sobre cada campo incorrecto. Brinda la posibilidad de seleccionar o introducir o seleccionar los siguientes datos para modificar el contenido de Ayuda: <ul style="list-style-type: none"> <li>Nombre de la Acción.</li> <li>Apodo.</li> <li>Descripción de la Acción.</li> <li>Dirección URL.</li> </ul> Y permite: <ul style="list-style-type: none"> <li>Actualizar los datos.</li> <li>Salir de la vista actual.</li> </ul>	<i>Acciones</i> <i>Editar</i> <i>Actualizar</i>
		V	I	NA	V		
		V	V	NA	I		

Tabla 14. Escenario Eliminar Contenido del Caso de Prueba Administrar Ayuda.

ID del escenario	Escenario	Respuesta del sistema	Flujo central
------------------	-----------	-----------------------	---------------





EC 1.1:	Selecciona la opción de Eliminar los datos del contenido de Ayuda.	Muestra un mensaje de alerta. Y permite: <ul style="list-style-type: none"> <li>• Aceptar.</li> <li>• Cancelar</li> </ul>	Acciones <i>Eliminar</i>
EC 1.2:	Selecciona la opción de Aceptar.	Elimina el contenido de Ayuda. Muestra un mensaje de información.	Acciones <i>Eliminar</i> <i>Aceptar</i>
EC 1.3:	Selecciona la opción de Cancelar.	Regresa a la vista anterior.	Acciones <i>Eliminar</i> <i>Cancelar</i>

Una vez ejecutadas las pruebas descritas, se procedió al análisis de los resultados obtenidos.

#### 4.3.2. Resultados de las validaciones realizadas

Las pruebas de integración fueron realizadas por el desarrollador a medida que se iba creando el subsistema. En total se llevaron a cabo 3 iteraciones, para cada módulo al que se integra la Ayuda, en las que el número de errores detectados fue disminuyendo.

Las pruebas funcionales se realizaron en 3 iteraciones, por un equipo calificado en revisiones de software. En la primera se detectaron 4 No Conformidades que fueron solucionadas por el equipo de desarrollo. Se llevó a cabo una prueba de regresión para comprobar que las No Conformidades identificadas anteriormente estuvieran corregidas. Seguidamente se procedió a la segunda iteración para la evaluación del subsistema, en la que se detectaron 3 No Conformidades. Luego de corregir los errores encontrados se comenzó la prueba de regresión con resultados satisfactorios y se dio paso a la tercera iteración de pruebas, en la que no se encontró ningún error. Al terminar la ejecución de las pruebas el equipo de revisión emitió un certificado de calidad para el Subsistema Ayuda de la Plataforma ZERA, que puede ser consultado en el **Anexo 9** del documento.

#### 4.4. Conclusiones

En este capítulo se concentró la documentación del proceso de implementación del Subsistema Ayuda para la plataforma ZERA. Se detallaron las clases y componentes del producto a entregar, y se definió la estructura del plugin que da solución a la problemática definida. Se puntualizaron las características del Sistema para la Administración de la Ayuda, así como las funcionalidades con que cuenta el mismo. Se



mencionaron los tipos de pruebas de software realizados para validar la solución implementada. Finalmente se expusieron los resultados de las pruebas realizadas al subsistema, determinando su nivel de calidad.



## Conclusiones

Se concluye que la metodología RUP permitió guiar adecuadamente el proceso de desarrollo del Subsistema Ayuda y generar los artefactos fundamentales para la construcción del mismo. Se definieron las pautas a seguir para la escritura del contenido a mostrar en la Ayuda, así como las disposiciones generales para su diseño y organización. Se realizó un análisis a fondo de los requerimientos obtenidos y se transformaron en el diseño del Subsistema Ayuda. Se implementó un plugin (compuesto por 3 módulos) que se integra con todos los subsistemas de la plataforma ZERA, dando cumplimiento a las funcionalidades definidas por el cliente. Las pruebas realizadas al subsistema indican que cuenta con un buen nivel de calidad. De manera general, se desarrolló un subsistema que permite brindar una asistencia personalizada e informativa a los usuarios de la plataforma ZERA durante su interacción con la misma, además de gestionar los contenidos de ayuda que lo componen.



## Recomendaciones

- ✓ Asociar videos tutoriales a los contenidos de Ayuda, para lograr una asistencia más completa a los usuarios de la plataforma ZERA.
- ✓ Documentar el código para lograr su mayor entendimiento por futuros desarrolladores del subsistema.



## Glosario de Términos

**Aprendizaje colaborativo:** Estrategias para propiciar el desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social) donde cada miembro del grupo es responsable tanto de su aprendizaje como el de sus compañeros.

**Educación a distancia:** Modalidad educativa en la que los estudiantes no necesitan asistir físicamente a ningún aula. Normalmente, se envía al estudiante por correo el material de estudio (textos escritos, videos, cintas de audio, CD-ROM) y él devuelve los ejercicios resueltos.

**Educación virtual:** Oportunidad de aprendizaje que se acomoda al tiempo y necesidad del estudiante. Proporciona herramientas de aprendizaje más estimulantes y motivadoras que las tradicionales.

**Hardware:** Componentes físicos, electrónicos y mecánicos de un sistema informático; normalmente la computadora misma, las unidades de disco externas, etc.

**Software:** Conjunto de programas, procedimientos y documentación asociado a un sistema informático. Específicamente hace referencia a los programas de la computadora que dirigen y controlan las actividades del sistema informático.

**HTML:** siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje con el que se definen las páginas Web. Básicamente se trata de un conjunto de etiquetas que sirven para definir la forma en la que presentar el texto y otros elementos de la página.

**Plugin:** Pequeño programa que añade alguna función a otro programa, habitualmente de mayor tamaño. Son muy utilizados en los navegadores web para ampliar sus funcionalidades.

**RSS:** Siglas de Really Simple Syndication, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos.

**Usabilidad:** Facilidad de uso, grado de eficacia, eficiencia y satisfacción con la que usuarios pueden lograr objetivos específicos, según el contexto de uso.

## Referencias Bibliográficas

1. **Muguía Álvarez, Lic. Dianelys y Castellanos Rodríguez, Lic. Kethicer** . monografias.com. [En línea] 23 de abril de 2006. [Citado el: 20 de octubre de 2010.] <http://www.monografias.com/trabajos31/software-educativo-cuba/software-educativo-cuba.shtml>.
2. **Sánchez, J.** *Construyendo y Aprendiendo con el Computador*. 1999.
3. **Olavarrieta Pérez, Noslén.** *Software Educativo Panorama Histórico Cultural Cubano y Universal*. Ciudad de la Habana : s.n., 2009.
4. **Jaurilaritza, Eusko.** Euskadi. [En línea] Gobierno Vasco, 2004. [Citado el: 15 de 2 de 2011.] <http://www.euskadi.net/o82/o82WebServlet?accion=10&id=24&ident=239&tag=PagDiccionario>.
5. **Capdet, Dolors.** e-Learning. [En línea] [Citado el: 15 de 1 de 2011.] [http://e-learningglobal.es/pageID\\_5295796.html](http://e-learningglobal.es/pageID_5295796.html).
6. Qué es un CMS? CMS-Hispano.org. [En línea] <http://cms-hispano.org/index.php?s=content&p=cms>.
7. EcuRed. [En línea] [http://www.ecured.cu/index.php/Plataformas\\_Educativas](http://www.ecured.cu/index.php/Plataformas_Educativas).
8. **Enríquez Vázquez, Larisa.** LCMS y objetos de aprendizaje. *Revista Digital Universitaria*. [En línea] 10 de 11 de 2004. <http://www.revista.unam.mx/vol.5/num10/art66/int66.htm>.
9. **Duffy, y otros.** *On line Help: Design and Evaluation*. Norwood, New Jersey : Ablex Publishing Corporation, 1992.
10. **Belen Gil, Ana y García, Francisco José.** *Sistema de apoyo en línea al usuario*. s.l. : Universidad de Salamanca, 2002.
11. **Sánchez Fernández, José Antonio** . *Diseño y evaluación de sistemas de ayuda interactivos*. s.l. : Universidad Politécnica de Madrid, 2001.
12. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. 2000.
13. **Fowler, Martin y Sierra, Alejandro.** La Nueva Metodología. [En línea] 4 de 2003. <http://www.programacionextrema.org/articulos/newMethodology.es.html>.
14. The Agile Unified Process (AUP). [En línea] [Citado el: 15 de diciembre de 2010.] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.
15. Grupo de Soluciones Innova. Rational Rose. [En línea] 2007.



- <http://www.rational.com.ar/herramientas/roseenterprise.html>.
16. Visual Paradigm. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
  17. Evolus. [En línea] Mozilla, 2010. <http://pencil.evolus.vn/en-US/Home.aspx>.
  18. RapiCapWin (official site). [En línea] <http://www.rapicapwin.com.ar/index-en.html>.
  19. Super Screen Capture/Recorder - Free Video Screen Recording, Screen Capture Software. [En línea] Zealsoft, 1997. <http://www.free-screen-capture.com/>.
  20. JShot. [En línea] <http://jshot.info/>.
  21. **Sessink, Olivier**. bluefish. [En línea] 2011. <http://bluefish.openoffice.nl/index.html>.
  22. **Quint, Vincent**. Welcome to Amaya. [En línea] INRIA and W3C, 2010. <http://www.w3.org/Amaya/>.
  23. Apache. *HTTP Server Project*. [En línea] The Apache Software Foundation. <http://httpd.apache.org/>.
  24. Cherokee. [En línea] <http://www.cherokee-project.com/>.
  25. **Pérez Valdéz, Damián**. Maestros del Web. [En línea] 2007. <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>.
  26. MySQL. The world's most popular open source database. [En línea] <http://www.mysql.com/products/enterprise/database/>.
  27. PostgreSQL-es. [En línea] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
  28. PostgreSQL. [En línea] PostgreSQL Global Development Group . <http://www.postgresql.org/about/>.
  29. Desarrollo WEB. [En línea] <http://www.desarrolloweb.com/articulos/243.php>.
  30. **The PHP Group**. PHP Manual. [En línea] Yahoo! Inc. <http://www.php.net/manual/en/preface.php>.
  31. **Suehring, Steve y Parra Pérez, Beatriz**. *Perl. Introducción al desarrollo Web*. s.l. : Anaya, 2006.
  32. **Eguíluz Pérez, Javier** . Introducción a JavaScript. *librosweb.es*. [En línea] <http://www.librosweb.es/javascript/>.
  33. **Musciano, Chuck y Kennedy, Bill**. *HTML & XHTML : The Definitive Guide*. s.l. : O'Reilly.
  34. **Adams, Cameron , y otros**. *The Art & Science Of CSS*. 2007.
  35. **Pérez Valdéz, Damián**. Los Frameworks de PHP agilizan tu trabajo. Maestros del Web. [En línea] 2007. <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>.
  36. Best Web Frameworks. [En línea] <http://www.bestwebframeworks.com/>.
  37. CakePHP Cookbook. [En línea] <http://book.cakephp.org/>.
  38. **Potencier, Fabien**. *Symfony: La guía definitiva*. 2007.
  39. **Newton, Aaron**. *MooTools Essentials: The Official MooTools Reference for JavaScript(TM) and Ajax*



(Firstpress). s.l. : Apress, 2008.

40. **Chaffer, Jonathan y Swedberg, Karl** . *Learning jQuery 1.3*. 2009.

41. The Official YAML Web Site. [En línea] <http://www.yaml.org/> .

42. **Álvarez, Miguel Ángel**. Manual de Blueprint CSS. *Desarrolloweb.com*. [En línea] 8 de 10 de 2010. [Citado el: 14 de 1 de 2011.] <http://www.desarrolloweb.com/manuales/manual-blueprint-css.html>..

43. **Boeck, Heiko** . *The Definitive Guide to NetBeans Platform*. s.l. : Apress, 2009.

44. eclipse. [En línea] <http://www.eclipse.org/home/categories/index.php?category=ide&tab=learn>.

45. zend: The php company. [En línea] <http://www.zend.com/en/products/studio/>.

46. **Pressman, Roger**. Ingeniería de software. Un enfoque práctico. Interamericana de España, 2002.