

Universidad de las Ciencias Informáticas

Facultad 2



**Título: HERRAMIENTA INFORMÁTICA PARA LA
ELABORACIÓN DE MAPAS CONCEPTUALES
COLABORATIVOS BASADOS EN LA WEB 2.0**

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

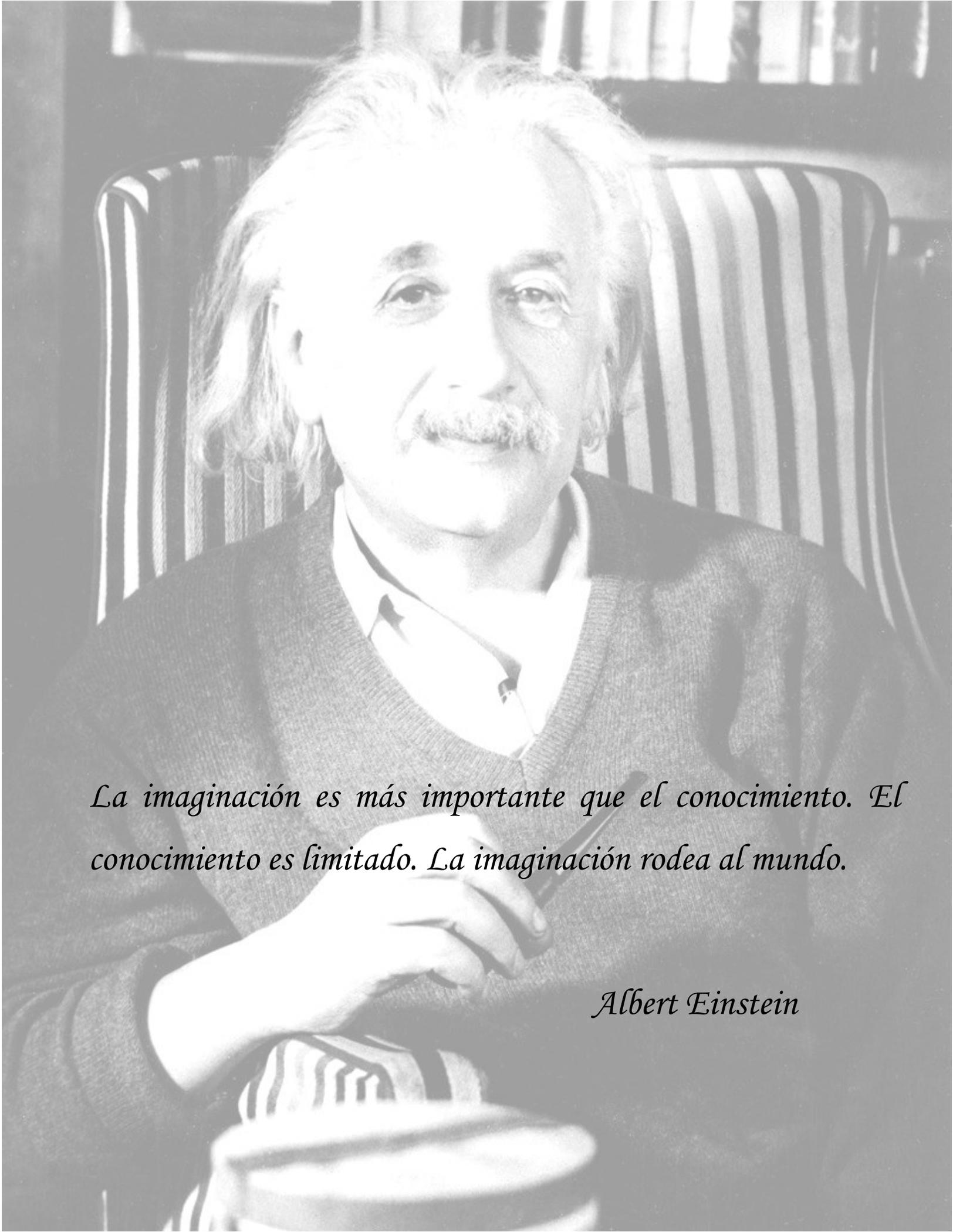
Autor (es): Yordanka Pupo Gómez

Grabiél Montero Coello

Tutor (es): Ing. Deivis Ricardo Álvarez Mendoza

Ing. Aliennis Mercedes González Hurtado

La Habana, Junio de 2013

A black and white photograph of Albert Einstein. He is seated in a chair with vertical stripes, looking directly at the camera with a slight smile. He has his characteristic wild, white hair and a mustache. He is wearing a dark, V-neck sweater over a light-colored collared shirt. His right hand is resting on his lap, holding a pipe. The background is slightly out of focus, showing what appears to be a bookshelf.

La imaginación es más importante que el conocimiento. El conocimiento es limitado. La imaginación rodea al mundo.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes _____ del año _____.

Autores

Yordanka Pupo Gómez

Grabiél Montero Coello

Tutores

Ing. Deivis Ricardo Álvarez Mendoza

Ing. Aliennis Mercedes González Hurtado

AGRADECIMIENTOS

Grabiél: Agradecer a toda mi familia que siempre ha estado cuando más la he necesitado, en especial a mis padres Gabriel Montero Ferrer y Odalis Vaillant Coello, a mis abuelas Ana América y Agustina las cuales las considero como mis otras 2 madres, a mis tías Maritsa, Analeidis, Siomara y Onidia, un agradecimiento muy grande a Estrellita que siempre me ha tratado como su hijo, y le doy las gracias por siempre estar a mi lado y brindarme todo su cariño. Agradecer a todas las personas que conocí en estos 5 años, en especial a Dagoberto, Moreu, Andy, Robert, Lidiagnis, Boris, Henry, Travieso, Daniel, Alain, Jorge Carlos, Reynel y a mi compañera de tesis Yordanka Pupo Gómez, que aunque no compartamos las mismas creencias hemos sabido trabajar juntos centrados en un mismo objetivo.

Yordanka: Agradezco a Dios por sus palabras de amor, sabiduría y corrección; por su fidelidad y hacer posible lo imposible para mí. A mi esposo Yoelvis por todo su amor, ayuda y dedicación todos estos años sin lo cual quizás no hubiese sido posible vencer todos los duros momentos en la universidad. A mi mamá Gladys por su entrega total y dedicación hacia mí; por ser mi amiga y confiar en mí siempre, lo cual me hace andar segura; por nunca decir no a mis sueños y dejarme volar en pos de ellos. A mi hermano Yordán por ser mi compañía desde la barriguita de nuestra madre; por todo su amor, preocupación, atención y cuidado. A toda mi gran familia; a mis abuelos, especialmente mi abuela Dulce; a mis tíos y tías; a mis primos en especial Yaneisi a quien quiero como hermana por la constante comunicación y apoyo en estos años. Agradezco también a toda la familia de mi esposo que inserto como mi nueva familia. A todos mis hermanos cristianos, gracias por compartir todas las alegrías y tristezas; por las comidas, viajes a la playa, paseos, actividades, compañía en los ciclones y enfermedades, en las bodas que celebramos, etc.; gracias porque me hice adicta de tanto cariño y amor por lo que sé que nos buscaremos y estaremos siempre unidos. A todos mis amigos y amigas de mi bella ciudad de Holguín, en especial Olguita por su apoyo constante; y a aquellos que aún en la distancia se preocupan y están pendiente de mí. A mi país por esta oportunidad gratuita que les brinda a todos los cubanos de estudiar. A todos los profesores y amigos que han interactuado conmigo hasta hoy en la universidad. Especialmente a esas amigas que han convivido conmigo en el apartamento, quienes logran formar lazos estrechos de amistad, llegando a ser como familia (Kenia, Aliuska, Isis, Leybis, Lisandra, Adys, Rosalia, Aliannis, Yeni, entre otras). A mi compañero de tesis Grabiél Montero Coello por enfrentar conmigo esta difícil tarea, agradezco que fuese un buen compañero y que a través de este trabajo nos hubiésemos conocido y aprendiéramos a trabajar en equipo.

DEDICATORIA

Grabiél

Dedicado a mi familia y a todas las personas que por una razón u otra brindaron su granito de arena.

Yordanka

Dedico este trabajo a mi querido Dios, por ser el sustento y la mayor satisfacción de mi vida.

A mi esposo Yoelvis por ser mi compañero, hermano y amigo todos estos años en la universidad (UCI).

A mi mamita linda Gladys por ser la mujer más especial, digna de todo mi cariño, respeto y admiración.

A mi hermano Yordán por ser esa gran parte de mi vida, sin la cual no soy feliz.

A mi papá quien hubiese estado feliz de verme culminar esta carrera.

RESUMEN

Actualmente a través de las Tecnologías de la Información y las Comunicaciones se obtienen un conjunto de herramientas útiles para apoyar tanto trabajos docentes como de investigación. Estas con el transcurso del tiempo han hecho notar que son necesarias en el proceso de enseñanza-aprendizaje. Podemos mencionar entre muchas que existen, aquellas que permiten elaborar mapas conceptuales, debido a que han demostrado ser importantes para el desarrollo de este proceso. Constituyen una estrategia para desarrollar el aprendizaje significativo y habilidades como el análisis y la síntesis.

En internet se pueden encontrar varios software basados en la Web 2.0 que permiten elaborar dichos mapas, los cuales se pueden utilizar sin necesidad de descarga o instalación. Estos posibilitan la interacción social, dando lugar a que otras personas puedan visualizar, descargar, reutilizar y comentar los mapas conceptuales o utilizarlos como referencia, logrando así una estrecha colaboración entre los usuarios. Estos software facilitan al usuario proveer conocimiento por medio de la información que plasma en sus mapas conceptuales y además recibir nuevos conocimientos aportados por otros usuarios.

Debido a las características particulares de la Universidad de las Ciencias Informáticas en cuanto al restringido acceso a Internet, el fácil acceso a la red local, y con el objetivo que los estudiantes puedan desarrollar el proceso de análisis y síntesis, se crea una aplicación útil para la realización de mapas conceptuales basados en la Web 2.0 que facilita la colaboración entre sus usuarios. Los estudiantes podrán crear sus mapas conceptuales y seleccionar como colaboradores suyos a otros estudiantes y profesores. Los mapas conceptuales pueden ser comentados y editados por sus colaboradores.

Palabras claves: mapas conceptuales, Web 2.0, aprendizaje significativo.

ÍNDICE DE CONTENIDOS

AGRADECIMIENTOS	3
DEDICATORIA	4
RESUMEN	5
ÍNDICE DE CONTENIDOS	6
ÍNDICE DE TABLAS	9
ÍNDICE DE ILUSTRACIONES	11
INTRODUCCIÓN	12
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	16
Introducción.....	16
1.1. Aprendizaje significativo	16
1.2. Mapas Conceptuales	17
1.3. Los Mapas Conceptuales en el proceso docente-educativo.....	20
1.4. Herramientas para la Construcción de Mapas Conceptuales.....	22
1.5. Metodología de desarrollo de software	24
1.6. Herramienta y lenguaje de modelado	28
1.7. Lenguajes de Programación	28
1.8. Plataforma de desarrollo.....	30
1.9. Web 2.0	31
1.10. Servidor Web	32
1.11. Sistemas Gestores de Bases de Datos	32
1.12. Conclusiones parciales	33
CAPÍTULO 2: ANÁLISIS DE LA APLICACIÓN	35

Introducción.....	35
2.1. Propuesta de solución	35
2.3. Características no funcionales del sistema	35
2.4. Personal relacionados con el sistema.....	36
2.5. Historia de usuarios.....	37
2.6. Planificación	42
2.7. Planificación de duración de las Iteraciones	44
2.8. Planificación de entrega	45
2.9. Conclusiones parciales.....	46
CAPÍTULO 3: DISEÑO DE LA APLICACIÓN	47
Introducción.....	47
3.1. Diseño	47
3.2. Arquitectura	47
3.3. Patrones de Arquitectura	48
3.4. Patrones de Diseño	50
3.5. Modelo Entidad-Relación.....	53
3.6. Tarjetas CRC (Clase-Responsabilidad-Colaboración)	53
3.7. Prototipo de Interfaz de Usuario	54
3.8. Conclusiones parciales.....	55
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	56
Introducción.....	56
4.1. Implementación	56
4.2. Validación de la Propuesta	64

4.3 Conclusiones parciales.....	73
CONCLUSIONES GENERALES.....	74
RECOMENDACIONES.....	75
REFERENCIAS BIBLIOGRÁFICAS.....	76
GLOSARIO DE TÉRMINOS.....	81
ANEXOS.....	82

ÍNDICE DE TABLAS

Tabla 1. Personal relacionado con el sistema	37
Tabla 2. HU Crear Mapa	38
Tabla 3. HU Salvar Mapa.....	39
Tabla 4. HU Eliminar Mapa	39
Tabla 5. HU Buscar Mapa.....	39
Tabla 6. HU Insertar Comentario	40
Tabla 7. HU Autenticar Usuario.....	40
Tabla 8. HU Buscar Usuario	40
Tabla 9. HU Añadir Colaborador	40
Tabla 10. HU Eliminar Colaborador.....	41
Tabla 11. HU Importar Mapa.....	41
Tabla 12. HU Editar Mapa.....	41
Tabla 13. HU Exportar Mapa	42
Tabla 14. HU Insertar Imagen	42
Tabla 15. HU Trazas del Mapa	42
Tabla 16. Estimación de esfuerzo por Historia de Usuario	43
Tabla 17. Planificación de duración de iteraciones.....	45
Tabla 18. Planificación de entregas.	46
Tabla 19. Tarjeta CRC clase UsuarioBundle\DefaultController	54
Tabla 20. Tarjeta CRC clase userMap	54
Tabla 21. HU para la iteración 1.....	57
Tabla 22. Tarea de Ingeniería Implementar funcionalidad Crear Mapa	57
Tabla 23. Tarea de Ingeniería Implementar funcionalidad Salvar Mapa	58
Tabla 24. Tarea de Ingeniería Implementar funcionalidad Eliminar Mapa	58
Tabla 25. Tarea de Ingeniería Implementar funcionalidad Buscar Mapa.....	59
Tabla 26. HU para la iteración 2.....	59
Tabla 27. Tarea de Ingeniería Implementar funcionalidad Insertar Comentario	59
Tabla 28. Tarea de Ingeniería Implementar funcionalidad Autenticar Usuario.....	60
Tabla 29. Tarea de Ingeniería Implementar funcionalidad Buscar Usuario	60

Tabla 30. HU para la iteración 3.....	61
Tabla 31. Tarea de Ingeniería Implementar funcionalidad Añadir Colaborador	61
Tabla 32. Tarea de Ingeniería Implementar funcionalidad Eliminar Colaborador	62
Tabla 33. Tarea de Ingeniería Implementar funcionalidad Importar Mapa.....	62
Tabla 34. Tarea de Ingeniería Implementar funcionalidad Editar Mapa.....	62
Tabla 35. HU para la iteración 4.....	63
Tabla 36. Tarea de Ingeniería Implementar funcionalidad Exportar Mapa	63
Tabla 37. Tarea de Ingeniería Implementar funcionalidad Insertar Imagen.....	64
Tabla 38. Tarea de Ingeniería Implementar funcionalidad Trazas del Mapa	64
Tabla 39. Prueba de aceptación No 1	66
Tabla 40. Prueba de aceptación No 2.....	66
Tabla 41. Prueba de aceptación No 3.....	67
Tabla 42. Prueba de aceptación No 4	67
Tabla 43. Prueba de aceptación No 5.....	68
Tabla 44. Prueba de aceptación No 6.....	68
Tabla 45. Prueba de aceptación No 7.....	69
Tabla 46. Prueba de aceptación No 8.....	69
Tabla 47. Prueba de aceptación No 9.....	70
Tabla 48. Prueba de aceptación No 10.....	70
Tabla 49. Prueba de aceptación No 11.....	71
Tabla 50. Prueba de aceptación No 12.....	71
Tabla 51. Prueba de aceptación No 13.....	71
Tabla 52. Prueba de aceptación No 14.....	72

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Mapa Conceptual.....	19
Ilustración 2. Tipos de Mapas Conceptuales.....	20
Ilustración 3. MVC.....	50
Ilustración 4. Modelo Entidad Relación de la BD.....	53
Ilustración 5. Prototipo de Interfaz de Usuario de la herramienta	55
Ilustración 6. Relación de NC.....	73

INTRODUCCIÓN

La humanidad está sufriendo significativas transformaciones. La globalización ha llegado a los rincones más inimaginables del planeta y la educación se ha visto impactada en todos sus niveles. Con la introducción de las nuevas Tecnologías de la Información y las Comunicaciones (TIC) al sector educativo, se pretende conseguir que las personas estén más preparadas, capacitadas profesionalmente para jugar un rol protagónico en el desarrollo de la sociedad.

En correspondencia con lo antes expresado la Universidad de Ciencias Informáticas (UCI) ha adoptado un modelo pedagógico centrado en el aprendizaje (1). Este modelo se expresa en términos de la necesidad de formar integralmente al joven y prepararlo para que se desempeñe social y profesionalmente en la sociedad. Posee como aspecto central de sus objetivos que los estudiantes adquieran competencias necesarias que les permitan desarrollarse a plenitud, sobre la base entre otras formas del aprender a aprender. De modo que, cuando el estudiante aprende a aprender aprovechando la información que tiene disponible y las herramientas que brindan las TIC, puede construir su propio conocimiento (1).

Por tanto urge la necesidad de buscar y aplicar vías de acción de enseñanzas que ayuden al logro de aprendizajes significativos en los estudiantes, así como lograr un proceso de enseñanza-aprendizaje capaz de tributar al cambio educativo que la sociedad demanda en estos tiempos utilizando de forma eficiente las TIC.

Los estudiantes necesitan adquirir habilidades para realizar el análisis y síntesis de los contenidos. El modelo educativo centrado en el aprendizaje conlleva a que los profesores encarguen a sus alumnos a realizar tareas y trabajos independientes. Por esta razón el estudiante al enfrentar estas tareas debe aplicar el análisis cuando investiga y la síntesis cuando va a redactar sobre los aspectos más relevantes de su investigación.

En la UCI debido al uso de Internet el proceso de enseñanza-aprendizaje no se centra en una simple y pasiva recepción de los conocimientos adquiridos en clase, sino que se dirige en la constante búsqueda, análisis y síntesis de información.

Como una importante herramienta metodológica en el proceso de enseñanza-aprendizaje se insertan los Mapas Conceptuales (MC). Los MC se encuentran entre los procedimientos que han sido utilizados para favorecer el desarrollo de estrategias de aprendizaje en los estudiantes universitarios. Constituyen un

mecanismo para desarrollar el aprendizaje significativo, así como su utilidad para relacionar conceptos y organizar ideas. Con la creación de los MC se practican las habilidades de análisis y síntesis.

El uso de los MC en el ámbito educativo, permite su utilización mediante dos maneras. Se pueden diseñar MC de forma autónoma en un procedimiento cualquiera de aprendizaje y el que implica la colaboración a través del intercambio de ideas o conocimientos con otras personas.

Actualmente en Internet, se encuentran disponibles diversos software que permiten elaborar mapas conceptuales como por ejemplo: CmapTools, Creately, Gliffy, entre otros; algunos de ellos son descargables y pueden ser instalados, otros con características Web 2.0 pueden utilizarse en línea sin necesidad de descargarlos o de instalar software alguno. Estos últimos permiten a cualquier usuario elaborar los mapas conceptuales y llegar a ellos desde cualquier lugar con acceso a Internet. Además, posibilitan la interacción social mediante la cual otras personas pueden visualizar, descargar, reutilizar y comentar los MC o simplemente utilizarlos como referencia. De esta manera es la Web 2.0 una excelente opción para la realización de una herramienta informática basada en el diseño de mapas conceptuales colaborativos. Diseñar mapas conceptuales con la facilidad que otros usuarios puedan verlos y comentarlos hace realidad la colaboración, lo cual permite que el conocimiento sea compartido.

A partir de entrevistas realizadas a un grupo de profesores (Ver Anexo 1) se ha logrado constatar que muchos de los estudiantes en la UCI actualmente poseen una estructura de apropiación del conocimiento a partir de un aprendizaje memorístico y reproductivo. Presentan dificultades para comprender e interpretar un texto y para sintetizar los aspectos más relevantes, por lo que no logran asociar el contenido que reciben o estudian con el conocimiento previo acumulado o con experiencias anteriores. Demuestran deficiencias para aprender significativamente y el conocimiento que llegan adquirir tiende a ser temporal afectando su desempeño docente y como futuro profesional. Con la perspectiva de contribuir con el desarrollo de las habilidades del análisis y la síntesis en los estudiantes, se hace necesario utilizar herramientas de apoyo al proceso de enseñanza-aprendizaje como son los MC.

En la UCI se les hace difícil a los estudiantes trabajar con los diferentes software existentes para elaborar MC a través de Internet porque se necesita acceso pleno a Internet para interactuar con estos. Por otro lado aquellos que se descargan de Internet tienen la desventaja que no están conectados a la red, por lo que no se hace posible la interacción dinámica con otros usuarios.

Considerando la situación problemática descrita se identifica como **problema a resolver**: ¿Cómo diseñar mapas conceptuales colaborativos para contribuir al desarrollo del proceso de análisis y síntesis del contenido en los estudiantes de la Universidad de las Ciencias Informáticas?

El presente trabajo centra su **objeto de estudio** en el diseño de mapas conceptuales colaborativos. Se enmarca el **campo de acción** en un software que permita el diseño de mapas conceptuales colaborativos en la Universidad de las Ciencias Informáticas.

Objetivo general: Desarrollar un software que permita el diseño de mapas conceptuales colaborativos basado en la Web 2.0.

La **idea a defender** plantea que la elaboración de mapas conceptuales colaborativos contribuirá al desarrollo del proceso de análisis y síntesis del contenido en los estudiantes de la Universidad de las Ciencias Informáticas.

Para el cumplimiento organizado y bien distribuido del objetivo general se plantean las siguientes **tareas de investigación**:

- Estudio acerca de los mapas conceptuales, el aprendizaje significativo y su aplicación en el proceso docente-educativo en el nivel superior.
- Caracterización de las herramientas computacionales para el diseño de mapas conceptuales.
- Caracterización de la Web 2.0 para el desarrollo de herramientas computacionales.
- Realización del modelo de análisis y diseño de un software para la creación de mapas conceptuales colaborativos.
- Elaboración del modelo de implementación de un software para la creación de mapas conceptuales colaborativos.
- Realización de la validación a la propuesta desarrollada.

Para el desarrollo de las tareas de investigación se hace uso de los **Métodos de investigación**. Estos se dividen en teóricos y empíricos.

Métodos Teóricos:

Analítico – Sintético: Este método permite estudiar los mapas conceptuales basados en el aprendizaje significativo, buscar información y hacer un análisis sobre el uso de estos para el trabajo tanto individual como colaborativo, así como las distintas herramientas existentes que permiten la creación de los mismos,

obteniendo los elementos fundamentales y las características esenciales para la elaboración del software final.

Histórico-lógico: En esta investigación dicho método se usa con el objetivo de estudiar todo lo referente a los mapas conceptuales basados en el aprendizaje significativo y cómo ha evolucionado durante el transcurso de los años el desarrollo de estos mapas de forma colaborativa.

Métodos Empíricos:

Entrevista: Se realizaron entrevistas a 10 profesores del colectivo de tercer año de la facultad 2 en la UCI a través del método no probabilístico, específicamente el muestreo intencional. La entrevista se caracterizó por ser individual y estructurada. Se realiza con el fin de obtener información acerca del trabajo con los mapas conceptuales por parte de los estudiantes, la necesidad de fomentar el aprendizaje significativo y habilidades como el análisis y la síntesis.

El presente documento se encuentra estructurado por cuatro capítulos que agruparán los contenidos de la siguiente manera:

Capítulo 1. “Fundamentación Teórica”: Muestra el resultado de la investigación bibliográfica sobre el objeto de estudio, metodología, lenguajes de programación y herramientas de desarrollo que serán utilizadas para el desarrollo del software.

Capítulo 2. “Análisis de la Aplicación”: En este capítulo se alude a la propuesta de solución; se realizan las especificaciones de las funcionalidades del sistema. Describe características de la aplicación a desarrollar.

Capítulo 3. “Diseño de la Aplicación”: Se describen los patrones de diseño y arquitectura, la estructura que presentará la base de datos a utilizar y las Tarjetas de Cargo o Clase, Responsabilidad y Colaboración (CRC).

Capítulo 4. “Implementación y Pruebas: Realiza las tareas de Ingeniería. Se efectúan las Pruebas Unitarias y de Aceptación para validar la propuesta de solución.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

Para comprender en qué consiste la presente investigación, se hace preciso conocer aspectos relevantes acerca de los mapas conceptuales así como algunas herramientas existentes para la elaboración de los mismos. Este capítulo conforma la base teórica de la solución propuesta. En este tópico se abordan varios conceptos, herramientas y tecnologías, que son objeto de estudio y análisis con el propósito de fundamentar su uso en la solución que se propone.

1.1. Aprendizaje significativo

Existen diferentes tipos de aprendizajes entre los cuales podemos mencionar, el memorístico y el significativo; estos integran nuevos conocimientos en la estructura cognitiva a pesar de obtener resultados diferentes.

La teoría del aprendizaje significativo fue creada por David Ausubel. Esta se establece cuando un conocimiento nuevo se incorpora o asimila a una estructura cognitiva previa, construyendo una nueva organización. De esta forma, los conceptos inclusos adquieren un significado personal para el aprendizaje. El conocimiento incluido permite la incorporación de nuevos conceptos y proposiciones a la estructura cognitiva, la cual sufre una reestructuración continua en este tipo de aprendizaje creando un proceso dinámico (2). Ausubel afirma que la estructura cognitiva de una persona es el factor que determina que el material nuevo se torne significativo permitiendo lograr su adquisición y retención (3).

En el aprendizaje significativo, los conceptos aprendidos se retienen durante más tiempo; algunos toda la vida (4). Esto no significa que el olvido no pueda darse también en este tipo de aprendizaje, pero al estar el conocimiento incrustado en la estructura cognitiva, éste no se pierde. Así aparece el fenómeno que Ausubel denomina “inclusión obliterativa”, donde el alumno olvida los mensajes específicos aprendidos, pero en la estructura cognitiva permanecen ideas mejoradas. Estas ideas son las que facilitaran el aprendizaje futuro (5).

El término significativo se refiere tanto a un contenido con estructuración lógica propia, como a aquel

material que potencialmente puede ser aprendido de forma significativa; se utiliza en oposición al aprendizaje de contenido "sin sentido". La condición para que un aprendizaje sea significativo es que sea incorporado de forma sustancial a la estructura mental del sujeto, y para esto es necesario relacionar el material nuevo que se intenta incorporar, con el que el sujeto ya posee (6).

El proceso de unión de los nuevos conocimientos con los ya existentes es un proceso activo y personal. Activo, porque depende de la asimilación deliberada de la tarea de aprendizaje por parte del alumno. Personal, porque la significación de toda la tarea de aprendizaje, depende de los recursos cognitivos que utilice cada alumno (6).

Para los estudiantes universitarios el aprendizaje significativo es muy necesario debido a que reciben un gran número de nuevos conceptos que deben ser relacionados y si no son adquiridos eficazmente pueden tener problemas de asimilación.

1.2. Mapas Conceptuales

Orígenes de los mapas conceptuales

El mapa conceptual fue creado por el psicólogo Joseph Novak en los años setenta y desarrollado en el departamento de educación de la Universidad de Cornell en los Estados Unidos. Se encuentra basado en la teoría del aprendizaje significativo.

En la Universidad de Cornell, el grupo de trabajo de Novak, Gowin y Mussonda estaba inmerso en un estudio alargado de 12 años en el que utilizaba entrevistas como elementos de registro. Este estudio se pretendía evaluar cómo determinados métodos de formación en estados iniciales influían en el aprendizaje posterior a largo plazo. El método utilizado como el más válido y fiable para evaluar el conocimiento del alumno fue la entrevista personal, que se grababa para su posterior análisis. Entonces, surgió la problemática de cómo extraer y sintetizar la información proporcionada por los estudiantes en los cientos de registros de audio realizados. Así se ideó una técnica de transcripción de los datos que se recogían en las entrevistas (7).

Esta técnica consistía en unos diagramas de conceptos extraídos de las mismas. Estos mapas iniciales fueron una representación bidimensional de conceptos enlazados, pero sin caracterizar el significado del enlace mediante palabras. El grupo de trabajo observó que los meros enlaces no reflejaban el conocimiento real de los alumnos tal como se deducía de los datos recogidos. En consecuencia introdujeron pequeñas

expresiones o palabras de enlace a modo de etiquetas sobre las líneas. Así, en esencia, el mapa conceptual adoptó su forma actual. La eficiencia de esta técnica era altísima pues una entrevista transcrita en 15 o 20 páginas, se convertía en un mapa de una sola hoja sin perder conceptos esenciales o proposiciones significativas del entrevistado (7).

Conceptos de Mapa Conceptual

Se han publicado diferentes criterios sobre el concepto de los MC, uno de ellos precisamente definido por Novak, su creador, publicado en su texto "Aprendiendo a aprender", define el MC como "una técnica que representa, simultáneamente, una estrategia de aprendizaje, un método para captar lo más significativo de un tema y un recurso esquemático para representar un conjunto de significados conceptuales, incluidos en una estructura de proposiciones" (8).

Un mapa conceptual es sólo un diagrama que tiene por objeto representar relaciones significativas entre conceptos y que lo hace en forma de proposiciones, siendo por tanto una técnica para exponer el entendimiento conceptual y proposicional que el sujeto tiene sobre un determinado conocimiento (7).

Elementos que conforman un Mapa Conceptual

El MC está compuesto por varios elementos como son: Conceptos, Palabras de Enlace y Propositiones. Los conceptos se definen como palabras que determinan un significado, aquellas que transmiten algo representativo. Un ejemplo de un concepto es la palabra casa, la cual hace imaginar algo de forma abstracta. Las palabras de enlace son aquellas que van a unir a los conceptos; se usan verbos, artículos, preposiciones o conjunciones. Las proposiciones son ideas que se forman por la unión de dos o más conceptos a través de palabra de enlace.

En la confección del mapa conceptual el concepto puede ser representado dentro de en un círculo, elipse o cuadrado. Los conceptos se unen por las palabras de enlace que van acompañadas por una línea la cual puede tener un símbolo de flecha indicando el sentido.

A continuación se muestra una imagen de un mapa conceptual (Ver Ilustración 1):

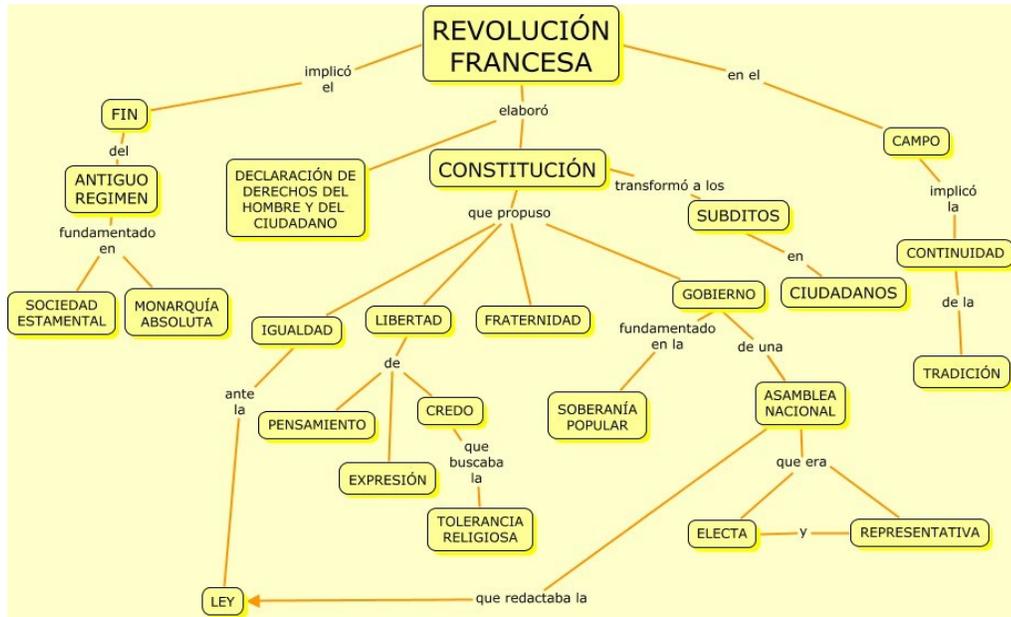


Ilustración 1. Mapa Conceptual

Tipos de diseño de Mapas Conceptuales

Los mapas conceptuales pueden ser diseñados de varias maneras pero existen cuatro categorías en las que se pueden clasificar (Ver Figura 2):

- ✓ El Mapa conceptual en araña: El concepto central o tema principal se ubica en el centro del mapa y los demás conceptos se insertan hacia el mediante las palabras de enlace.
- ✓ El Mapa conceptual jerarquizado: La información se presenta siguiendo un orden descendente y los conceptos más importantes ocupan los lugares superiores por lo que el nodo superior será el concepto principal del mapa.
- ✓ El mapa conceptual en diagrama de flujos: La información se expande en un formato lineal. Los conceptos son ubicados uno detrás del otro.
- ✓ El mapa conceptual sistémico: La información se establece de manera secuencial pero además se le añaden entradas y salidas.

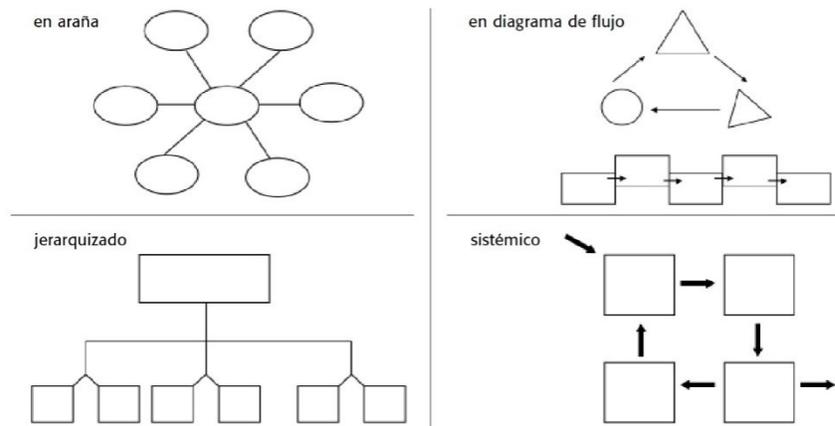


Ilustración 2. Tipos de Mapas Conceptuales

Pasos para la Construcción de un Mapa Conceptual

Para la construcción de los mapas conceptuales se deben seguir cuatro pasos que propone su autor Novak.

1. Identificar la pregunta, tema o campo de conocimiento que se quiere representar.
2. Elaborar una lista de 10 a 20 conceptos relacionados con el tema a desarrollar.
3. Ordenar los conceptos jerárquicamente del más amplio y general, al más detallado y concreto. Situar los primeros en la parte superior y los últimos en la inferior. Dado que algunos conceptos tendrán el mismo nivel jerárquico aparece un orden bidimensional.
4. Construir el mapa: disponer los conceptos sobre un plano según el orden anterior, unir mediante una línea aquellos conceptos que estén relacionados, y etiquetar dichos enlaces con unas pocas palabras que pongan de relieve la naturaleza de la relación (9).

1.3. Los Mapas Conceptuales en el proceso docente-educativo

La educación se centra en desarrollar una enseñanza con calidad, formar integralmente al joven y prepararlo para que se desempeñe como profesional en la sociedad. Tiene como aspecto fundamental que los estudiantes adquieran las competencias que les permitan desarrollarse a plenitud sobre la base entre otras formas de aprender a aprender.

Los mapas conceptuales se insertan en el programa educativo de aprender a aprender como recursos

metodológicos destinados especialmente a la enseñanza, pero también pueden utilizarse como estrategias de comunicación en otros muchos campos de la actividad humana. En el ámbito de la educación, debe destacarse que los mapas conceptuales dirigen la atención, tanto del estudiante como del profesor, sobre el número de ideas importantes en las que deben concentrarse en cualquier tarea específica de aprendizaje (10).

Pueden servir como estrategias destinadas a mejorar los procesos de enseñanza y aprendizaje, favoreciendo la reflexión y la comprensión. Además de las destrezas cognitivas individuales, los mapas conceptuales también contribuyen, desde el punto de vista colectivo, a lograr una mayor uniformidad en el nivel de conocimiento de los integrantes de un grupo o de una clase (10).

Novak ha descrito el acto de hacer mapas como una actividad creativa, en la cual el estudiante debe hacer un esfuerzo para aclarar significados, identificando los conceptos importantes, relaciones y estructura del contenido tratado. La creación de conocimiento requiere un alto nivel de aprendizaje significativo, los mapas conceptuales facilitan este proceso, por lo que resultan importantes en el aprendizaje, principalmente debido a que (11):

- Proporcionan una rápida visualización de los contenidos de aprendizaje.
- Favorecen el recuerdo y el aprendizaje de manera jerárquica organizada.
- Permiten una rápida detección de los conceptos claves de un tema, así como de las relaciones entre los mismos.
- Favorecen el desarrollo del pensamiento lógico.
- Los materiales elaborados utilizando Mapas Conceptuales facilitan el estudio independiente.

Permiten que el alumno pueda explorar su conocimiento previo acerca de un nuevo tema, así como la integración de la nueva información que ha aprendido (11).

Necesidad del Mapa Conceptual para la Programación

Una de las asignaturas de la carrera de Ingeniería Informática se encuentra Programación 1. Esta disciplina resulta compleja en su proceso de aprendizaje para los estudiantes en su primer año de la carrera. A partir de este momento el alumno empieza a especializarse con técnicas de programación más novedosas. En este ambiente en el cual se desarrolla tendrá que integrar todos los conocimientos recibidos en la asignatura

de Introducción a la Informática. En esta etapa, los estudiantes solo han adquirido una panorámica general del objeto de estudio de la Informática y sus aristas fundamentales. Estos conocimientos son indispensables para un buen entendimiento de los restantes contenidos que recibirán en el transcurso de toda la carrera; pero independientemente de esta realidad, es inevitable que se produzca en ellos un choque cuando se enfrenten a su primera asignatura de programación y al empleo de sus técnicas; por tanto, surgirán como consecuencia, problemas en la captación de los contenidos propios de la referida asignatura. El constante cambio de los paradigmas de programación introduce nuevas particularidades al desarrollo del proceso de enseñanza de esta disciplina (12).

¿Cómo debe enseñarse la programación?

Algunos autores para responder a esta interrogante han realizado proposiciones sobre reflexiones didácticas que incluyen diferentes aspectos. Sin embargo, se debe considerar lo siguiente (12):

Por los propios cambios que se presentan en la actualidad, el estudiante debe dominar una base conceptual sólida, además de un grupo de habilidades y capacidades, todo lo cual facilite manipular el conjunto de abstracciones que conlleva a la elaboración de una aplicación para la solución de un problema, todo esto bajo una sintaxis y una semántica.

El alumno va a manipular un gran número de conceptos, por lo cual debe tener alguna forma de organizarlos. Algunos, por su complejidad, resulta convenientes expresarlos en términos de conceptos más sencillos. Las habilidades y capacidades que debe desarrollar, dependen en gran medida de la base conceptual que tenga.

La posibilidad que en un mapa conceptual se pueda incluir recursos que permitan el estudio organizado de los estudiantes, evidentemente constituye una herramienta que les resulta provechosa dado que facilita una visualización integradora de un grupo de conceptos (12).

1.4. Herramientas para la Construcción de Mapas Conceptuales

Actualmente se puede contar con varias herramientas de software las cuales pueden encontrarse en Internet y posibilitan la elaboración de MC. Estas tienen la posibilidad de ser descargadas o de ser utilizadas desde Internet. A continuación se mencionan algunas de estas herramientas con una breve descripción de las mismas.

Cmap Toolkit

Herramienta de software abierto para construir, compartir, navegar y debatir modelos de conocimiento representados en forma de MC. Está habilitada para el trabajo en red, permite a los usuarios construir y colaborar con sus colegas durante la construcción del MC, por medio de Internet. Es muy intuitiva y fácil de utilizar (13).

Inspiration

El programa Inspiration es una herramienta informática de aprendizaje visual, utilizada por los docentes de todo el mundo. Especialmente diseñada para la creación de diagramas en forma de telaraña, mapas de ideas y MC. Permite exportar los mapas creados a formatos gráficos como jpg, gif y bmp. Permite a sus usuarios crear y editar mapas conceptuales contando con diferentes colores, imágenes y textos que se pueden incluir. Posee modelos prediseñados para familiarizar al usuario con la herramienta (14).

Smart Ideas

Facilita la elaboración de MC entre otros diagramas. Ofrece un entorno de trabajo que se configura de acuerdo con el tipo de diagrama que se elabore; es un programa sencillo, claro e intuitivo. Permite exportar los diagramas creados a formatos como jpg, gif, png y bmp. Ofrece librerías, plantillas y ejemplos; los diagramas se pueden realizar partiendo de cero o basándose en una plantilla o un ejemplo (15).

Creately

Es una de las herramientas que encontramos en la red basada en la Web 2.0, se utiliza para elaborar mapas conceptuales. Permite realizar trabajos de forma colaborativa (16). El trabajo final se puede exportar como PDF o como imagen. Es sencilla de usar y entre las opciones que posee permite incorporar imágenes prediseñadas y realizar comentarios en tiempo real (17).

Gliffy

Es una aplicación web que permite crear MC. Se encuentra disponible en Internet y su acceso es de forma gratuita. Para utilizarla no se necesita un software instalado en la computadora. Es una herramienta que facilita el trabajo colaborativo. Permite guardar los mapas realizados en formato JPG y el idioma en que está publicado es en inglés (18).

CmapTools

Este software fue desarrollado por el “Institute for Human and Machine Cognition” (IHMC), de la Universidad de West Florida (Estados Unidos), se diseñó con el objetivo de apoyar la construcción de modelos de conocimiento representados en forma de “Mapas Conceptuales”. Es una herramienta gratuita que permite a los usuarios navegar, compartir y debatir modelos representados como MC. Cada usuario puede construir sus propios mapas en su computador personal, compartirlos en servidores disponibles en Internet, enlazarlos a otros servidores de Cmpas, crear páginas web en los servidores y editarlos de forma sincrónica a través de Internet (19).

OpenOffice Draw

Este programa gratuito forma parte de la suite de oficina de OpenOffice.org, se diseñó especialmente para elaborar gráficos y diagramas en general. Es apropiado para que los estudiantes realicen organigramas, telarañas, mapas de ideas, MC y diagramas causa-efecto. Su instalación es sencilla, pero es necesario instalar OpenOffice.org. Es compatible con los SO Windows, Linux y Solaris (15).

1.5. Metodología de desarrollo de software

Una metodología de desarrollo de software es un conjunto de procedimientos y técnicas que especifican el modo en que debe hacerse el software con el fin de orientar a los desarrolladores. Existen diversas metodologías las cuales se clasifican en dos grupos: las tradicionales y las ágiles.

Metodologías tradicionales

Las metodologías tradicionales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo; ofrecen por lo general cierta resistencia a los cambios que puedan producirse durante el ciclo de desarrollo del producto, y es un proceso muy bien controlado con muchas políticas y normas (20). Imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada (21).

Requieren de una serie de controles, procesos y mecanismos estrictos, tediosos, lentos y rígidos, con un resultado poco claro a la vista del cliente y/o usuario final. En las primeras etapas de estos procesos de desarrollos se tiene como resultado un contrato con una extensa documentación, respecto a los

requerimientos, necesidades, alcances y limitaciones del proyecto; adicionalmente a esto, los planteamientos resultantes de las etapas iniciales de estas metodologías resultan poco eficientes debido a que tratan de anticipar todos los posibles escenarios que se podrían presentar durante el desarrollo del proyecto (22).

Existen varias metodologías tradicionales y entre ellas se encuentra RUP (Proceso Unificado de Desarrollo) la cual es la más utilizada para el desarrollo de sistemas orientados a objetos. RUP es guiado por casos de uso y centrado en la arquitectura, utilizando UML como lenguaje de modelado. Define cuatro fases esenciales (Inicio, Elaboración, Construcción y Transición); nueve flujos de trabajos, seis de Ingeniería y tres de apoyo los cuales se mencionan a continuación (23):

- Modelado del Negocio
- Requerimientos
- Análisis y Diseño
- Implementación
- Prueba
- Despliegue
- Administración del proyecto,
- Administración de configuración y cambios
- Ambiente.

Metodologías ágiles

Las metodologías ágiles permiten la integración del usuario final o usuario experto dentro del proceso de desarrollo, y no tan solo como un espectador del mismo; además el hecho de acercarse físicamente a todos los miembros del equipo de desarrollo, no solo reduce costos de transferencia de información, sino que mejora el proceso de comunicación entre expertos y desarrolladores (22).

Principales ideas de la metodología ágil:

- Se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los

procesos utilizados.

- Se hace mucho más importante crear un producto software que funcione que escribir mucha documentación.
- El cliente está en todo momento colaborando en el proyecto.
- Es más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan (24).

Entre todas las metodologías ágiles que existen se hace referencia a las que a continuación se mencionan (25):

SCRUM:

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos con un rápido cambio de requisitos.

Crystal Methodologies:

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos.

Dynamic Systems Development Method (DSDM):

Define el marco para desarrollar un proceso de producción de software. Es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos.

Adaptive Software Development (ASD):

Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios.

Feature-Driven Development (FDD):

Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.

Lean Development (LD):

Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil

en los años 80. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente.

Programación Extrema (XP, en inglés *Extreme Programming*):

La XP fue concebida por Kent Beck, es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Además se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios (25). Consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo (24).

Roles en XP

El concepto de rol se utiliza para definir quienes realizan las diferentes actividades. Según propone el creador de esta metodología podemos encontrar los siguientes roles (25):

- Programador: Escribe las pruebas unitarias y produce el código del sistema.
- Cliente: Es quien redacta las historias de los usuarios y las pruebas funcionales (pruebas de aceptación del cliente) para validar su implementación. El cliente da una gran prioridad a las historias de usuarios y decide cual implementar en cada iteración centrándose en aportar mayor valor al negocio.
- Encargado de Pruebas (Tester): Ayuda al cliente a escribir las pruebas funcionales (pruebas de aceptación del cliente). Se encarga de ejecutar las pruebas con regularidad, difunde los resultados obtenidos al equipo y es el responsable de las herramientas que dan soporte a las pruebas.
- Encargado de Seguimiento (Tracker): Es el que proporciona la realimentación al equipo. Realiza el seguimiento del proceso de cada iteración y verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado en ello para la mejora de futuras estimaciones.
- Entrenador (Coach): Es el responsable del proceso global. Se encarga de proveer guías al equipo de forma que se apliquen las practicas XP y se vaya siguiendo el proceso correctamente.
- Consultor: Es un miembro externo del equipo con un conocimiento específico en algún tema que es necesario para el proyecto, en el que surjan problemas.

- Gestor (Big boss): Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es la de coordinación.

Debido a que el software a desarrollar no es un proyecto extenso y requiere realizarse en el menor tiempo posible y la necesidad de tener al cliente como parte del equipo de desarrollo, se determinó emplear la metodología XP. Independientemente de las otras metodologías ágiles anteriormente mencionadas, XP es una de las más usadas, posee amplia documentación y sus características se ajustan a las del sistema a desarrollar.

1.6. Herramienta y lenguaje de modelado

Como herramienta de modelado se seleccionó Visual Paradigm 8.0 el cual ayuda a construir más rápido aplicaciones de calidad y a un menor coste; cuenta con el uso del lenguaje de Modelamiento Unificado UML. Este lenguaje permite visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar elementos conceptuales como lo son procesos de negocio y funciones de sistema, además de elementos concretos como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (26).

1.7. Lenguajes de Programación

PHP 5.3

PHP (PHP acrónimo de "Hypertext Preprocessor") es un lenguaje de script interpretado en el lado del servidor, la mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. El propósito de este lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas web (27).

PHP se escribe dentro del código HTML, es gratuito e independiente de plataforma. Está desarrollado en política de código abierto por lo cual a lo largo de su historia ha tenido muchas contribuciones de otros

desarrolladores. Posee compatibilidad con las bases de datos más comunes como MySQL (27).

JavaScript

JavaScript debido a su compatibilidad con la mayoría de los navegadores modernos es el lenguaje de programación del lado del cliente más utilizado, además se utiliza para añadir interactividad a las páginas. Una página web dinámica es aquella que incorpora contenidos dinámicos y elementos que tengan movimiento, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario (28).

Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (29).

Este lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas, posee un conjunto de características de las cuales podemos mencionar que su código se integra en las páginas HTML y no es necesario declarar los tipos de variables que van a utilizarse. No emplea clases ni herencia típicas de la programación orientada a objetos.

JSON

Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. Es un subconjunto de la notación literal de objetos de *JavaScript*, un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidas por los programadores de la familia de lenguajes C, incluyendo C++, C#, Java, JavaScript y PHP entre otros. Estas propiedades hacen que *JSON* sea un lenguaje ideal para el intercambio de datos (13). Se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia cuando la fuente de datos es explícitamente de fiar (30).

Jquery como framework para JavaScript

Jquery es un framework JavaScript; se utiliza para la creación de aplicaciones complejas del lado del cliente. Brinda ayuda en la creación de interfaces de usuario y efectos dinámicos. Cuando se programa Javascript con jQuery se obtiene una interfaz para programación (31).

Implementa una serie de clases (de programación orientada a objetos) que permiten programar sin

preocuparse del navegador con el que se visita el usuario. Ofrece una infraestructura para la creación de aplicaciones complejas del lado del cliente. Con Jquery existe una ayuda en la creación de interfaces de usuario y efectos dinámicos. Es un producto bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización (32).

Convierte la programación del lado del cliente en un problema más sencillo, simplifica muchos procedimientos JavaScript que comúnmente se usan para programar la interacción de las páginas Web. Es de gran ayuda en la creación de interfaces de usuario y efectos dinámicos. Al trabajar JavaScript con Jquery se obtiene una interfaz que funciona correctamente en los navegadores. Teniendo conocimiento de las librerías del framework y programar utilizando las clases, sus propiedades y métodos para la creación de las aplicaciones se pueden obtener los resultados deseados (33).

Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE, en inglés *Integrated Development Environment*) es un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software (34). Se considera una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Tanto Eclipse como NetBeans son herramientas IDE entre otras que existen. Como selección del IDE a utilizar se decidió el NetBeans en su versión 7.3 porque permite desarrollar aplicaciones web utilizando el marco de trabajo Symfony, es muy utilizado por programadores de PHP y es distribuido bajo la licencia (GPL) (35).

1.8. Plataforma de desarrollo

Symfony2 2.1

Symfony2 proporciona un conjunto de herramientas para desarrollar aplicaciones web. Es una colección de más de veinte bibliotecas independientes que se pueden utilizar dentro de cualquier proyecto PHP. Estas bibliotecas, llamadas componentes de Symfony2, contienen algo útil para casi cualquier situación, independientemente de cómo se desarrolle el proyecto. La plataforma Symfony2 proporciona una selección de componentes y bibliotecas de terceros; y proporciona configuración sensible. El objetivo de la plataforma es integrar muchas herramientas independientes; es un paquete que se puede configurar o sustituir completamente (36).

En Symfony2, con el objetivo de proporcionar ayuda al crear instancias, organizar y recuperar objetos en una aplicación, se implementó un contenedor de servicios que permite la estandarización y centralización de la forma en que se construyen los objetos en una aplicación. Proporciona facilidades para el trabajo, y acentúa una arquitectura que promueve el código reutilizable (37).

Una de las características de Simfony2 que lo hacen una excelente herramienta es su flexibilidad. Para el formato de los archivos de configuración usa PHP, Yaml y XML y con el fin de tener un buen rendimiento, tanto Yaml como XML se transforman en PHP; las anotaciones del código se compilan y se ejecutan a PHP.

1.9. Web 2.0

La transición que se ha dado de aplicaciones web tradicionales hacia aquellas que están enfocadas al usuario final, puede ser concebida con la Web 2.0, por esta razón constituye una actitud y no precisamente una tecnología. Se trata de aplicaciones que generen colaboración (38).

La Web 2.0 es un conjunto de aplicaciones y herramientas, que permiten marcar una nueva tendencia en cuanto al uso de los diferentes servicios que se ofrecen en la red, puesto que permiten a los usuarios navegar e interactuar de manera dinámica con la información, intercambiar contenidos, socializar opiniones y aportar en la construcción de aprendizajes colectivos (39).

Es un concepto de un nuevo software que permite convertir a la persona que navega en internet no sólo en un consumidor de información elaborada por otros, sino también en un emisor o productor de información. Se caracteriza por tener acceso libre a la información y compartir el conocimiento (40).

Para el cumplimiento del objetivo general de este trabajo se decide tener presente las características de la Web 2.0 debido a que mediante estas se cumple el propósito de la colaboración entre usuarios. Posee un diseño centrado en el usuario facilitando el intercambio interactivo de información. Constituye una vía para aportar conocimiento y a su vez adquirirlos de otras personas. Permite construir un nuevo conocimiento abarcando contenido anteriormente desconocido. Es una alternativa que puede usarse como vía de enseñanza en el ámbito educativo. Una buena aplicación de la Web 2.0 como método de enseñanza facilitará el desarrollo del aprendizaje en los estudiantes.

1.10. Servidor Web

Un servidor web es conocido como programa que atiende y responde a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP (por sus siglas en inglés de: Protocolo de Transferencia de Hipertexto) (41). Suministra páginas web a los navegadores que lo solicitan.

Cuando un usuario hace clic sobre un enlace a una página web, se envía una solicitud al servidor web para localizar los datos nombrados por ese enlace. El servidor web recibe esta solicitud y suministra los datos que le han sido solicitados o bien devuelve un mensaje de error (42).

Apache 2.2

Apache es un servidor web flexible, rápido, eficiente y posee continua actualización. Es un servidor altamente configurable de diseño modular. Resulta muy sencillo ampliar las capacidades de este servidor debido a que se pueden escribir módulos para realizar determinadas funciones, lo que implica que haya gran cantidad de ellos disponibles para su utilización (43).

Proporciona un servidor seguro y eficiente que provee servicios HTTP en sincronía con los estándares HTTP actuales. Es una tecnología gratuita de código abierto (44). Posee una licencia descendiente de las licencias BSD (Distribución de *Software* Berkeley), la cual permite modificar el código fuente. Tiene como características fundamentales que corre sobre todos los Sistemas Operativos (SO), permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor y es posible configurarlo para que ejecute una respuesta al usuario cuando ocurra un error en concreto (43).

Por las razones anteriormente expuestas se decide utilizar el servidor Apache en su versión 2.2 para el desarrollo de la presente investigación.

1.11. Sistemas Gestores de Bases de Datos

Un sistema gestor de bases de datos (SGBD) es un software que sirve de interfaz entre la base de datos, el usuario, y las aplicaciones que lo manejan. Debido a que permite manipular de forma clara, ordenada y precisa los datos, este sistema es importante para una apropiada manipulación de los mismos. Existen varios gestores de bases de datos como son Oracle, Microsoft SQL Server, PostgreSQL y MySQL.

MySQL es un SGBD relacional multiplataforma, fácil de usar y rápido en lecturas de datos. Algunas de las ventajas que posee es que consume muy pocos recursos y no suele perder información ni corromper los datos. Además de que es posible integrarlo con el lenguaje PHP es muy sencillo de integrar con frameworks de desarrollo como Zend, Symfony o CodeIgniter. Las características principales de MySQL son (45):

- El principal objetivo es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.
- El servidor soporta mensajes de error en distintas lenguas.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

Se decide utilizar como gestor de base de datos la versión de MySQL 5.5. Este aporta importantes mejoras, especialmente en las áreas de rendimiento y escalabilidad de las aplicaciones web a través de múltiples entornos operativos, entre ellos Windows, Linux, Oracle Solaris y Mac OS X. Permite mayor disponibilidad; debido a sus nuevos sistemas de réplica semi-síncrona se han mejorado notablemente la velocidad de recuperación y la fiabilidad de la base de datos. Gracias a las mejoras en los índices y las particiones de tablas, y capacidades de diagnóstico avanzado, su usabilidad es más sencilla. Posee el nuevo motor de almacenamiento InnoDB el cual soporta transacciones de tipo ACID (en inglés: Atomicity, Consistency, Isolation and Durability; en español: Atomicidad, Consistencia, Aislamiento y Durabilidad) garantizando la integridad de las tablas, la integridad referencial y la recuperación ante problemas (46).

1.12. Conclusiones parciales

En este capítulo se abordaron los elementos concernientes a la fundamentación teórica de la investigación. La teoría del aprendizaje significativo fue uno de los aspectos estudiados; a través de este se dio a conocer que su función es propiciar una mejor asimilación de los conocimientos adquiridos. Gracias a este tipo de aprendizaje, los estudiantes pueden mantener por más largo tiempo en su estructura cognitiva las

enseñanzas impartidas en clase.

Con el estudio acerca de los MC se pudo conocer que estos sirven como herramienta capaz de mejorar la capacidad de analizar y sintetizar sobre algún tema a tratar en su mapa. Demostraron ser necesarios en el proceso docente debido a que ellos favorecen el desarrollo del pensamiento lógico, la reflexión y comprensión.

Se reflejaron algunas herramientas existentes para la elaboración de MC y como aquellas que están basadas en las características de la Web 2.0 proporcionan interacción, comunicación y colaboración entre los usuarios.

Luego del análisis de esta investigación se concluye que la metodología a utilizar es XP y como plataforma de desarrollo Symfony2 2.1 y el sistema gestor de base de datos MySQL 5.5. Se hará uso de los lenguajes de programación PHP 5.3 y JavaScript y el IDE a utilizar será NetBeans 7.3.

CAPÍTULO 2: ANÁLISIS DE LA APLICACIÓN

Introducción

El capítulo actual tiene como propósito exponer la propuesta de solución y los requisitos no funcionales que el sistema debe cumplir para satisfacer las necesidades del cliente. A partir de la metodología de desarrollo de software seleccionada en el capítulo anterior para la elaboración de la aplicación propuesta, se recogerán los aspectos significativos relacionados con el análisis del sistema a codificar.

2.1. Propuesta de solución

Como propuesta de solución que fundamenta esta investigación se plantea la creación de una aplicación web para el diseño de mapas conceptuales que sirva de herramienta para apoyar el proceso de análisis y síntesis del contenido en los estudiantes. El sistema permitirá gestionar mapas conceptuales y gestionar colaboradores lo cual se le permitirá al usuario previamente autenticado. El usuario que es autor de algún MC podrá definir a otros usuarios como colaboradores; estos podrán aportar ideas y conocimiento permitiendo el intercambio entre ellos. Estos colaboradores pueden modificar el mapa o simplemente emitir algún comentario.

2.3. Características no funcionales del sistema

Los requisitos no funcionales son cualidades o propiedades que el producto debe tener, características que hacen al producto atractivo, usable, rápido y confiable. Estos son aspectos importantes que se deben cumplir para lograr un aprovechamiento óptimo de las funcionalidades del sistema teniendo en cuenta el entorno en el que será utilizado.

Confidencialidad

La información manejada en el sistema estará protegida contra acceso no autorizado utilizando para esto mecanismos de autenticación.

Integridad

La información manejada por el sistema será objeto de cuidadosa protección contra estados inconsistentes

y la corrupción, por lo que será de vital importancia el uso de políticas de salvadas de la base de datos.

Disponibilidad

Los usuarios luego de autenticarse correctamente en el sistema, tendrán acceso a la aplicación siempre que la conexión con el servidor esté disponible.

Apariencia o Interfaz Externa

Este tipo de requisito establece que la interfaz debe ser sencilla, amigable e intuitiva. Debe poseer una apariencia profesional, legible y simple de usar. Se utilizan colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del propósito del uso de la herramienta.

Requisitos de software

Se debe contar con la instalación del servidor web Apache 2.2 o una versión superior en la máquina servidor. En las computadoras que serán usadas por los usuarios sólo se requiere de un navegador web.

Requisitos de hardware

Es necesario que todas las computadoras estén conectadas a la red para que puedan utilizar la herramienta y tener 256 MB de RAM mínimo o superior y un micro Intel Celeron como mínimo. El sistema requiere de una computadora que haga función de servidor para la que se requiere que tenga una memoria RAM de 2 GB y un micro Intel Pentium 4 como mínimo.

2.4. Personal relacionados con el sistema

Al iniciar el desarrollo de un sistema informático es importante definir el personal al cual va dirigido el mismo. El personal o usuario relacionado con el sistema serán todas las personas que interactúan con la aplicación obteniendo un resultado de uno o varios procesos que se ejecutan.

Personal/Usuario	Descripción
Estudiante	Es la persona que gestiona sus mapas conceptuales y luego de ser autor de un mapa puede seleccionar a otros usuarios que sean sus colaboradores.
Profesores	Podrán gestionar sus mapas conceptuales y ser colaboradores de los estudiantes con el propósito de evaluar el trabajo que ellos realizan con la

	herramienta. Al ser colaboradores podrán ver y editar el mapa de aquel estudiante que lo añadió como colaborador.
--	---

Tabla 1. Personal relacionado con el sistema

2.5. Historia de usuarios

Las Historias de Usuario (HU) son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (25).

El usuario cliente es quien decide que tareas realizará la aplicación por tanto describirá la actividad a través de pequeños textos escritos bajo la terminología del cliente, no del desarrollador, de manera que sea sencilla y clara sin profundizar en detalles (47).

En un principio si no se identifican todas las HU no existe problema alguno puesto que el proceso de elaboración de las iteraciones permite una retroalimentación en sí mismo. Al inicio de cada iteración estarán registrados los cambios en las Historias de Usuario y es a partir de ello que se planificará la siguiente.

Cada HU es una simple descripción de una funcionalidad del software por la que el cliente va a pagar. En cualquier momento estas historias pueden romperse, remplazarse, unirse o dividirse. Entre las principales características que debe tener una buena historia de usuario se encuentran: la historia debe ser entendida por el cliente (representa un concepto y no una especificación detallada), debe devolver algún valor para el cliente, el tamaño de las Historias de Usuario debe ser tal que se puedan construir varias de ellas en una iteración, deben ser independientes unas de otras, cada historia debe permitir ser probada (48).

Clasificación de las Historias de Usuario

La prioridad en el negocio se clasifica de la siguiente manera:

Alto: Se le asigna a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.

Medio: Se le asigna a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que

estas tengan una afectación sobre el sistema que se esté desarrollando.

Bajo: Se le asigna a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

Las HU serán representadas mediante tablas las cuales están compuestas por los siguientes campos:

- **Número:** contiene el identificador de la HU, número incremental en el tiempo.
- **Nombre:** contiene el nombre que identifica a la HU para identificarlas mejor entre el cliente y los desarrolladores.
- **Usuario:** personal que puede acceder a la HU.
- **Iteración Asignada:** número de la iteración en que se va a implementar la HU.
- **Prioridad del Negocio:** tipo de prioridad de la HU (Alta, Media o Baja).
- **Puntos Estimados:** estimación hecha por el equipo de desarrollo del tiempo de duración. Cada punto es considerado como una semana de trabajo.
- **Puntos Reales:** tiempo real en el que se realizó la HU.
- **Descripción:** breve descripción del proceso que define la HU.

Se muestran a continuación las Historias de Usuarios del sistema a realizarse.

Historia de Usuario		
Número: 1	Nombre: Crear Mapa	
Usuario: Estudiante/Profesor	Iteración Asignada: 1	
Prioridad del Negocio: Alta	Puntos Estimados: 1	Puntos Reales: 1
Descripción: Se selecciona la opción crear mapa y el usuario podrá diseñar su mapa.		

Tabla 2. HU Crear Mapa

Historia de Usuario	
Número: 2	Nombre: Salvar Mapa

Usuario: Estudiante/Profesor	Iteración Asignada: 1	
Prioridad del Negocio: Alta	Puntos Estimados: 1	Puntos Reales: 1
Descripción: Se selecciona la opción salvar mapa y se guardarán los cambios realizados en el mapa.		

Tabla 3. HU Salvar Mapa

Historia de Usuario		
Número: 3	Nombre: Eliminar Mapa	
Usuario: Estudiante/Profesor	Iteración Asignada: 1	
Prioridad del Negocio: Alta	Puntos Estimados: 1/2	Puntos Reales: 1/2
Descripción: Se selecciona la opción eliminar mapa y el mapa quedará eliminado.		

Tabla 4. HU Eliminar Mapa

Historia de Usuario		
Número: 4	Nombre: Buscar Mapa	
Usuario: Estudiante/Profesor	Iteración Asignada: 1	
Prioridad del Negocio: Alta	Puntos Estimados: 1/2	Puntos Reales: 1/2
Descripción: El usuario selecciona la opción Buscar Mapa y encuentra los mapas que se han realizado.		

Tabla 5. HU Buscar Mapa

Historia de Usuario		
Número: 5	Nombre: Insertar Comentario	
Usuario: Estudiante/Profesor	Iteración Asignada: 2	
Prioridad del Negocio: Alta	Puntos Estimados: 1	Puntos Reales: 1

Descripción: Se escribirá un comentario acerca del mapa en el que se está trabajando.

Tabla 6. HU Insertar Comentario

Historia de Usuario		
Número: 6	Nombre: Autenticar Usuario	
Usuario: Estudiante/Profesor	Iteración Asignada: 2	
Prioridad del Negocio: Alta	Puntos Estimados: 1	Puntos Reales: 1
Descripción: Se brindará la posibilidad al usuario que inicie sesión especificando sus datos (usuario y contraseña).		

Tabla 7. HU Autenticar Usuario

Historia de Usuario		
Número: 7	Nombre: Buscar Usuario	
Usuario: Estudiante/Profesor	Iteración Asignada: 2	
Prioridad del Negocio: Alta	Puntos Estimados: 1	Puntos Reales: 1
Descripción: Se brindará la posibilidad de realizar una búsqueda especificando el usuario de este.		

Tabla 8. HU Buscar Usuario

Historia de Usuario		
Número: 8	Nombre: Añadir Colaborador	
Usuario: Estudiante/Profesor	Iteración Asignada: 3	
Prioridad del Negocio: Alta	Puntos Estimados: 1	Puntos Reales: 1
Descripción: El usuario seleccionará a otro usuario como colaborador suyo.		

Tabla 9. HU Añadir Colaborador

Historia de Usuario		
Número: 9	Nombre: Eliminar Colaborador	
Usuario: Estudiante/Profesor	Iteración Asignada: 3	
Prioridad del Negocio: Alta	Puntos Estimados: 1/2	Puntos Reales: 1/2
Descripción: El usuario seleccionará a un colaborador de su lista de colaboradores y lo eliminará.		

Tabla 10. HU Eliminar Colaborador

Historia de Usuario		
Número: 10	Nombre: Importar Mapa	
Usuario: Estudiante/Profesor	Iteración Asignada: 3	
Prioridad del Negocio: Alta	Puntos Estimados: 1/2	Puntos Reales: 1/2
Descripción: El usuario podrá importar algún mapa conceptual que haya sido salvado.		

Tabla 11. HU Importar Mapa

Historia de Usuario		
Número: 11	Nombre: Editar Mapa	
Usuario: Estudiante/Profesor	Iteración Asignada: 3	
Prioridad del Negocio: Alta	Puntos Estimados: 1	Puntos Reales: 1
Descripción: El usuario editará el MC de otro usuario, modificándolo.		

Tabla 12. HU Editar Mapa

Historia de Usuario	
Número: 12	Nombre: Exportar Mapa
Usuario: Estudiante/Profesor	Iteración Asignada: 4

Prioridad del Negocio: Medio	Puntos Estimados: 1	Puntos Reales:
Descripción: Luego que el usuario haya realizado su mapa conceptual podrá exportarlo.		

Tabla 13. HU Exportar Mapa

Historia de Usuario		
Número: 13	Nombre: Insertar Imagen	
Usuario: Estudiante/Profesor	Iteración Asignada: 4	
Prioridad del Negocio: Medio	Puntos Estimados: 1	Puntos Reales:
Descripción: Se debe seleccionar una imagen y esta podrá ser insertada dentro del área del MC.		

Tabla 14. HU Insertar Imagen

Historia de Usuario		
Número: 14	Nombre: Trazas del Mapa	
Usuario: Estudiante/Profesor	Iteración Asignada: 4	
Prioridad del Negocio: Medio	Puntos Estimados: 1	Puntos Reales:
Descripción: El usuario debe seleccionar una opción y el sistema mostrará las actividades que se han realizado sobre el mapa conceptual.		

Tabla 15. HU Trazas del Mapa

2.6. Planificación

En la planificación el cliente y el grupo de desarrolladores especifican el orden de prioridad que tendrá cada historia de usuario para implementarse. Se realiza una estimación del esfuerzo que costará implementar cada Historia de Usuario, la cual se expresa utilizando como medida el punto, el cual es considerado como

una semana de trabajo.

Estimación de esfuerzo por Historia de Usuario

Los desarrolladores hacen una planificación del tiempo que necesitan para implementar las HU y así determinar el tiempo total para el desarrollo de la aplicación. Para el logro del sistema propuesto se ha realizado la estimación de esfuerzo por cada historia de usuario, las que se muestran a continuación:

No. HU	Historia de Usuario	Punto de Estimación (semanas)
1	Crear Mapa	1
2	Salvar Mapa	1
3	Eliminar Mapa	1/2
4	Buscar Mapa	1/2
5	Insertar Comentario	1
6	Autenticar Usuario	1
7	Buscar Usuario	1
8	Añadir Colaborador	1
9	Eliminar Colaborador	1/2
10	Importar Mapa	1/2
11	Editar Mapa	1
12	Exportar Mapa	1
13	Insertar Imagen	1
14	Trazas del Mapa	1

Tabla 16. Estimación de esfuerzo por Historia de Usuario

2.7. Planificación de duración de las Iteraciones

Luego de ser identificadas y descritas las Historias de Usuarios, también de estimar su esfuerzo, se procede a la planificación de duración de las iteraciones, donde se realizarán cuatro iteraciones, las cuales se describen a continuación:

Iteración 1:

Durante la primera iteración se implementarán las cuatros primeras historias de usuarios. Todas las HU planificadas para esta iteración tienen prioridad alta debido a que son muy necesarias para cumplir el propósito de la aplicación. Se obtendrá la versión 0.1 del producto propiciando probar estas funciones.

Iteración 2:

El objetivo de esta iteración es la implementación de las HU 5,6 y 7 las cuales tienen alta prioridad y propician fundamentales acciones que ayudan a complementar el desarrollo del producto. Al finalizar se contará con una versión de prueba 0.2 con las funcionalidades concernientes a insertar comentario, autenticar usuario y buscar usuario.

Iteración 3:

En el transcurso de esta iteración se estará realizando la implementación de la HU 8, 9, 10 y 11. Se realizan actividades de prioridad alta las cuales facilitarán añadir un colaborador, eliminar un colaborador, importar el mapa y editar el mapa. Concluida esta iteración se obtendrá la versión 0.3 del producto.

Iteración 4:

La cuarta iteración se encargará de realizar las HU 12,13, y 14; estas HU poseen prioridad media. Estas facilitarán exportar el mapa, insertar imagen y mostrar las trazas del mapa. Concluida esta iteración se obtendrá la versión 1.0 la cual estará preparada para ponerla en práctica en un ambiente de trabajo real.

Iteraciones	Historia de Usuario	Duración (Semanas)
	Crear Mapa	1
	Salvar Mapa	1

Iteración 1	Eliminar Mapa	1/2
	Buscar Mapa	1/2
	Total de semanas en la iteración 1:	3
Iteración 2	Insertar Comentario	1
	Autenticar Usuario	1
	Buscar Usuario	1
	Total de semanas en la iteración 2:	3
Iteración 3	Añadir Colaborador	1
	Eliminar Colaborador	1/2
	Importar Mapa	1/2
	Editar Mapa	1
	Total de semanas en la iteración 3:	3
Iteración 4	Exportar Mapa	1
	Insertar Imagen	1
	Trazas del Mapa	1
	Total de semanas en la iteración 4:	3
Total		12

Tabla 17. Planificación de duración de iteraciones.

2.8. Planificación de entrega

La planificación de entrega establece la fecha fin de cada iteración, mostrando una versión desarrollada del producto en ese momento hasta lograr el producto final en la fecha establecida.

Sistema	Final 1 Iteración	Final 2da Iteración	3ra Iteración	4ta Iteración
Herramienta informática para la elaboración de mapas conceptuales colaborativos basados en la Web 2.0.	18/3/2013 Versión 0.1	8/4/2013 Versión 0.2	29/4/2013 Versión 0.3	20/5/2013 Versión 1.0

Tabla 18. Planificación de entregas.

2.9. Conclusiones parciales

En este capítulo quedó expuesta como propuesta de solución a realizar, la elaboración un software que permita elaborar mapas conceptuales y la colaboración entre los usuarios. Fueron identificados los requisitos que el sistema debe cumplir para un buen funcionamiento del mismo. Se definió que el personal a quien va dirigido el desarrollo del sistema son los estudiantes y profesores. Se elaboró la descripción de 14 Historias de Usuarios que serán objetos de automatización. Las HU fueron divididas en 4 iteraciones demostrando el esfuerzo planificado al desarrollo de cada una de ellas por parte del equipo de desarrollo; se estimó que el tiempo de realización de cada HU oscila entre 1 semana o la mitad de una semana. El tiempo de duración planificado para cada iteración es de tres semanas como lo propone la metodología XP. De esta manera el escenario queda preparado para comenzar a realizar el diseño de la aplicación.

CAPÍTULO 3: DISEÑO DE LA APLICACIÓN

Introducción

El presente capítulo aborda los temas de diseño, pues para llevar a cabo la implementación la metodología XP sugiere que hay que conseguir diseños simples y sencillos para lograr un fácil entendimiento que conlleve menos costo de tiempo y esfuerzo.

Se proponen temas como patrón de arquitectura a seguir y los patrones de diseño a utilizar para la implementación del sistema, además quedarán definidas las tarjetas CRC (Colaborador - Responsabilidad - Clase) como técnica de diseño.

3.1. Diseño

XP establece prácticas especializadas que inciden directamente en la realización del diseño para lograr un sistema robusto y reutilizable tratando de mantener su simplicidad. Permite hacer entregas pequeñas y frecuentes de valor para el cliente. A la hora de darle cumplimiento a la actividad de diseñar, XP no especifica ninguna técnica de modelado, puede utilizarse indistintamente sencillos esquemas en una pizarra, diagramas de clases utilizando UML o tarjetas CRC (Clase, Responsabilidad y Colaboración) siempre que sean útiles, tributen a la comprensión y no requieran mucho tiempo en su creación (49).

3.2. Arquitectura

La arquitectura de software es la organización fundamental de un sistema, consiste en el diseño de componentes de una aplicación, generalmente utilizando patrones arquitectónicos. Es una vista estructural de alto nivel, ocurre muy tempranamente en el ciclo de vida y define los estilos o grupos de estilos adecuados para cumplir con los requisitos no funcionales.

Constituye la panorámica conceptual de la estructura de la arquitectura. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código (50).

Otra definición contribuida por el autor Paul Clements, plantea que es a grandes rasgos, una vista general del sistema que incluye los componentes principales, la conducta de estos componentes con el resto del sistema y las formas en que deben interactuar y coordinarse para alcanzar la misión del sistema (51).

Representa un modelo relativamente pequeño, intelectualmente tratable, de la forma en que un sistema se estructura y sus componentes se entienden entre sí; este modelo es transferible a través de sistemas; en particular, se puede aplicar a otros sistemas que presentan requerimientos parecidos y puede promover reutilización en gran escala. El diseño arquitectónico soporta reutilización de grandes componentes o incluso de framework en los que se pueden integrar componentes (52).

Según Roger Pressman es la representación que capacita al ingeniero del software para: analizar la efectividad del diseño para la consecución de los requisitos fijados, considerar las alternativas arquitectónicas en una etapa en la cual hacer cambios en el diseño es relativamente fácil, y reducir los riesgos asociados a la construcción del software (53).

3.3. Patrones de Arquitectura

Los patrones de arquitectura expresan un paradigma fundamental para estructurar u organizar un sistema software. Proporcionan un conjunto de subsistemas o módulos predefinidos, con reglas y guías para organizar las relaciones entre ellos.

Estos se enfocan en dar solución a un problema en específico. Son patrones del software, que se encargan de definir la estructura de un sistema. Estos a su vez se componen de subsistemas con sus responsabilidades, también poseen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño (54).

Patrón Arquitectónico Modelo - Vista – Controlador (MVC)

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el controlador es el Sistema de Gestión de Base de Datos y el modelo es el modelo de datos (55).

Los autores coinciden con que: *“El MVC tiene como características principales que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas”* (56).

Una de sus ventajas es el soporte de vistas múltiples. Debido a que la vista se halla separada del modelo y no hay dependencia directa de este con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los datos de manera simultánea (57).

El framework seleccionado para el desarrollo del sistema a implementar, está desarrollado bajo las restricciones y recomendaciones de la arquitectura MVC.

Modelo: Esta es la representación específica de la información con la cual el sistema opera. Representa la información con la que trabaja el sistema, es decir, su lógica de negocio. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos. Contiene funciones para insertar, consultar y actualizar información de la base de datos.

Vista: Esta presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. Presenta información dinámica al usuario en una página web.

Controlador: Es el que responde a los eventos, usualmente acciones del usuario e invoca cambios al modelo y probablemente en la vista.

El ciclo de vida de este patrón comienza por una solicitud realizada al controlador por parte del usuario, luego el controlador decide a quién debe delegar la tarea solicitada y es aquí donde empieza a trabajar el modelo. El modelo es el encargado de realizar operaciones sobre la información que maneja para cumplir con lo que le solicita el controlador y al terminar su trabajo, le envía al controlador la información resultante de sus operaciones. El controlador le pasa a la vista lo recibido del modelo la cual se encarga de convertir los datos en una información visual capaz de ser entendida por el usuario. A continuación se muestra una imagen representando el MVC (Ver: Ilustración 3):

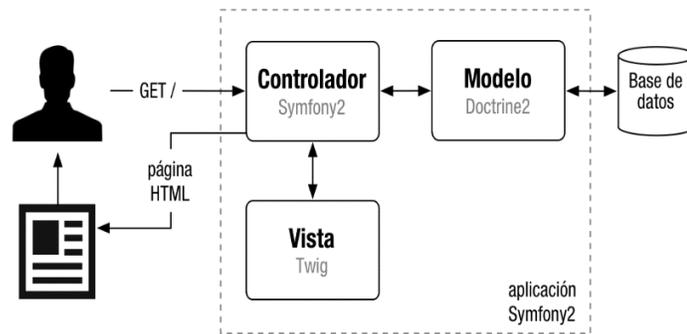


Ilustración 3. MVC

Cuando el usuario solicita ver la portada del sitio, internamente sucede lo siguiente (58) :

1. El sistema de enrutamiento determina qué Controlador está asociado con la página de la portada.
2. Symfony2 ejecuta el Controlador asociado a la portada. Un controlador no es más que una clase PHP en la que puedes ejecutar cualquier código que quieras.
3. El Controlador solicita al Modelo los datos. El modelo no es más que una clase PHP especializada en obtener información, normalmente de una base de datos.
4. Con los datos devueltos por el Modelo, el Controlador solicita a la Vista que cree una página mediante una plantilla y que inserte los datos del Modelo.
5. El Controlador entrega al servidor la página creada por la Vista.

3.4. Patrones de Diseño

Los patrones de diseño “*son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y se ha demostrado que funcionan*” (59).

El patrón de diseño identifica las clases e instancias participantes, sus roles y colaboraciones y la distribución de responsabilidades. Identifica los aspectos clave de una estructura de diseño común, lo que hace que sea útil para crear un diseño orientado a objetos reutilizable (60).

Los Patrones Generales de Asignación de Responsabilidades de Software (GRASP, por sus siglas en

inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos, de forma tal que se pueda diseñar software orientado a objetos (61). Dentro de los GRASP podemos mencionar: Creador, Experto, Alta Cohesión, Bajo Acoplamiento y Controlador, los cuales se utilizaron para la realización de la solución.

Creador: Tiene como propósito fundamental hallar un creador que se conecte con el objeto producido en cualquier evento. Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos.

En el framework Symfony2 el controlador frontal es el encargado de instanciar las clases controladoras y éstas, a su vez, instancian objetos del modelo.

Experto: Permite asignar una responsabilidad al experto en información, es decir, la clase que cuenta con la información necesaria para cumplir la responsabilidad.

En el caso de Symfony2, las clases controladoras operan las peticiones del cliente, y las clases llamadas entidades del negocio son las que se emplean para el acceso a los datos, debido a que contienen y representan los datos que manejará el sistema.

Alta Cohesión: La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Este patrón mejora la calidad y facilidad del diseño, genera un bajo acoplamiento y promueve la reutilización (62).

En Symfony2 se encuentran un grupo módulos independientes llamados bundles, lo que demuestra una alta cohesión; se pueden reutilizar dichos módulos en otro proyecto sin tener que utilizar el framework completo.

Bajo acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases. Acoplamiento alto significa que una clase recurre a muchas otras clases. El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios y que aumentan la oportunidad de una mayor productividad (62).

Con el uso de Symfony2 el bajo acoplamiento es concebido debido a que las clases que implementan el acceso a datos y la lógica del negocio se encuentran en el modelo, las cuales no se encuentran

estrechamente relacionadas con las clases de la vista o el controlador; esto propicia que exista baja dependencia entre ellas.

Controlador: La mayor parte de los sistemas reciben eventos de entrada externa. En estos casos hay que elegir controladores que manejen esos eventos de entrada. Este patrón ofrece una guía para tomar decisiones apropiadas en la elección de los controladores de eventos. Su utilización propicia que las operaciones del sistema se manejen en la capa de dominio de los objetos, y no en la de presentación (62). Symfony2 favorece a la utilización de este patrón debido a que las peticiones son manejadas por el controlador frontal, implica que todas las solicitudes son dirigidas a un script PHP que se encarga de instanciar al controlador frontal y este determinará qué controlador se debe ejecutar.

Patrones GOF

Decorator (Decorador):

El objetivo de este patrón es extender la funcionalidad de un objeto dinámicamente (63).

El archivo `base.html.twig`, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página.

Abstract Factory:

También conocido como Fábrica abstracta, permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí haciendo transparente el tipo de familia concreta que se esté usando (63).

Cuando el framework necesita crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea. Como la definición por defecto de la factoría para las peticiones es `getRequest()`, Symfony2 crea un objeto de esta clase para tratar con las peticiones.

Dependency Injection (Inyección de dependencias). Una de las características más sobresaliente de Symfony2 es el uso intensivo que hace de la Inyección de Dependencias, un potente patrón de diseño mediante el que se facilita la creación y configuración de objetos que prestan servicios en una aplicación gracias a la gestión automática de sus dependencias. Contribuye a crear un código más desacoplado y coherente (36).

Este patrón es utilizado al acceder a la base de datos. En las clases controladoras se realiza la instancia

del modelo a través de un servicio que se encarga de cargar la configuración y sus dependencias de manera automática.

3.5. Modelo Entidad-Relación

Una tarea importante es el diseño de la Base de Datos (BD) debido a que el sistema necesitará trabajar con la información que se almacena en dicha bases de datos. El modelo Entidad-Relación proporciona un fácil entendimiento de la estructura de datos, mediante el diseño. Se pueden describir los elementos existentes y la manera en que ellos se relacionan.

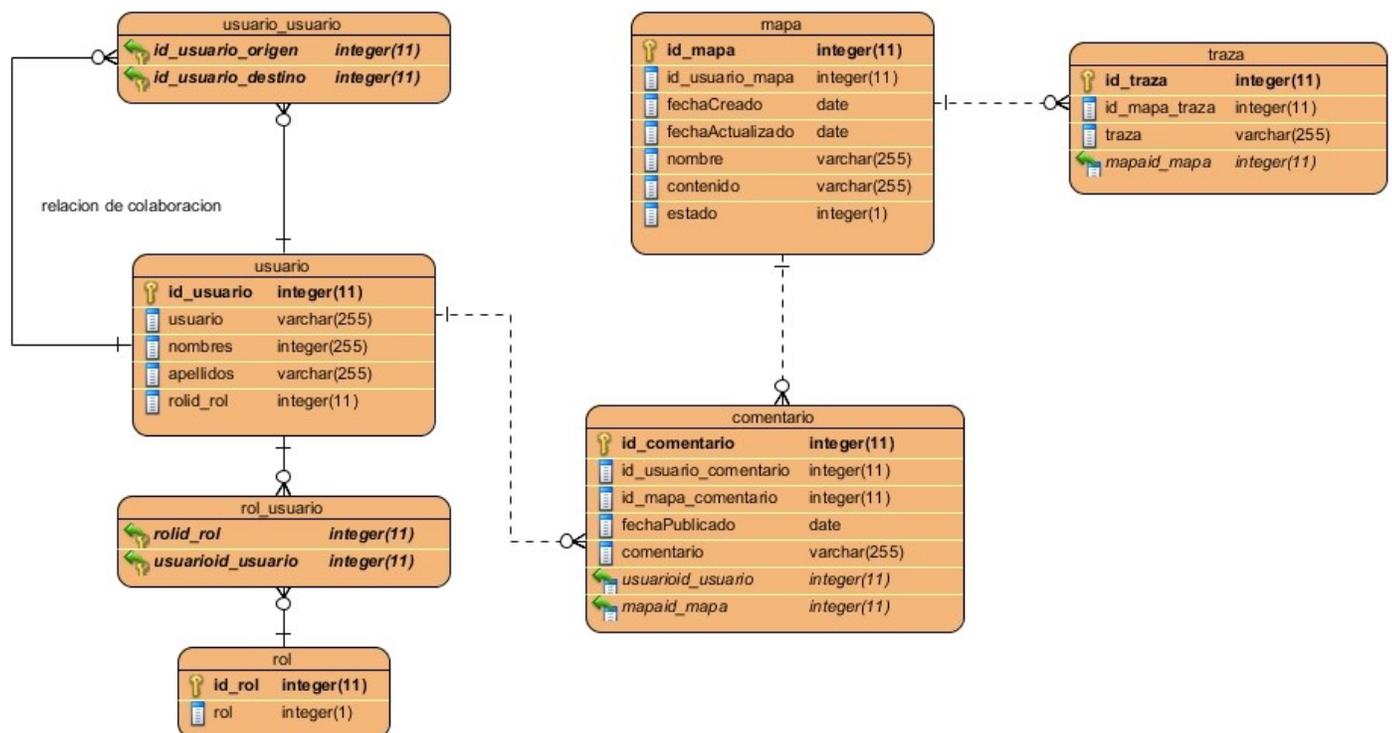


Ilustración 4. Modelo Entidad Relación de la BD

3.6. Tarjetas CRC (Clase-Responsabilidad-Colaboración)

La metodología XP propone el uso de tarjetas CRC (Clase-Responsabilidad-Colaboración) como técnica de modelado para crear diseños de clases orientados a responsabilidades; no especifica que se utilice

obligatoriamente el Lenguaje de Modelado Unificado (UML). El uso de las tarjetas CRC permiten a los desarrolladores centrarse en el desarrollo orientado a objetos. Se definen con la finalidad de obtener un diseño simple y no incurrir en la implementación de características que no son necesarias.

El título de la tarjeta corresponde al nombre de la clase y es ubicado en la parte superior, las Responsabilidades son colocadas en la izquierda y las Colaboraciones en la parte derecha. Con la finalidad de obtener un diseño simple y entendible se elaboraron las tarjetas CRC que se muestran a continuación:

Clase: UsuarioBundle\DefaultController	
Responsabilidades:	Colaboraciones:
Verificar la veracidad de los datos de autenticación del usuario.	Ldap

Tabla 19. Tarjeta CRC clase UsuarioBundle\DefaultController

Clase: userMap	
Responsabilidades:	Colaboraciones:
Guardar el mapa en la Base de Datos.	

Tabla 20. Tarjeta CRC clase userMap

3.7. Prototipo de Interfaz de Usuario

El prototipo de interfaz de usuario (IU) se utiliza para representar el sistema. Se expone con el fin de dar a conocer al cliente la perspectiva que se tiene a desarrollar como interfaz y de esta manera se acuerden algunos cambios y se corrijan errores hasta que el cliente esté satisfecho. La Ilustración 6 presenta el prototipo de IU de la herramienta propuesta.



Ilustración 5. Prototipo de Interfaz de Usuario de la herramienta

3.8. Conclusiones parciales

La construcción de un sistema informático debe estar sustentado sobre un buen diseño arquitectónico. Es por esta razón que en este capítulo se definieron los patrones de diseño y el patrón de arquitectura, apropiados para la construcción del software propuesto. A manera de tener una idea de cómo fue elaborada la persistencia de la información manejada por la aplicación, se presentó el diagrama entidad relación de la base de datos utilizada. Según la metodología seleccionada se realizó la técnica de las tarjetas CRC para detallar las responsabilidades de cada clase y se presentó el prototipo de interfaz de usuario. Es de esta manera como concluyen las actividades relacionadas con el diseño, dando paso a la implementación.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

Introducción

En el presente capítulo se expondrán aspectos relacionados con la implementación del software. Se describen Tareas de Ingeniería generadas en cada una de las iteraciones realizadas y las pruebas que validan el software; dichas pruebas permiten a los desarrolladores conseguir que el producto tenga un alto grado de fiabilidad que se caracterice por la ausencia mínima de errores.

4.1. Implementación

La implementación de los requisitos planteados por el cliente y que están plasmados en las Historias de Usuario, se realiza en el transcurso de las iteraciones a las que pertenecen. La codificación es guiada por las Tareas de Ingeniería (TI) y según propone la metodología seleccionada se realizará a través de parejas de programadores.

Tareas de Ingeniería

Las Tareas de Ingeniería permiten que se realice una correcta implementación de los requisitos y se describen utilizando un lenguaje técnico debido a que serán distribuidas a los desarrolladores. Según la planificación de las iteraciones y por cada HU que pertenece a esta, se realizan una o varias TI. Estas tareas son fichas que se confeccionan introduciendo los siguientes datos:

Número de tarea: Representa el identificador de la tarea.

Número de Historia de usuario: Representa la historia de usuario asociada a esta tarea.

Tipo de Tarea: Se especifica si la tarea es de Desarrollo, Corrección o Mejora.

Puntos Estimados: Duración estimada de la tarea.

Fecha Inicio y Fecha Fin: Se especifica el día, mes y año en que comienza y termina la tarea.

Programador Responsable: Persona encargada de su realización.

Descripción: Breve descripción de la tarea.

A continuación se muestra por cada iteración las HU implementadas y las TI asociadas a estas.

Iteración 1:

Iteración	No. HU	Historia de Usuario	Duración (Semanas)
1	1	Crear Mapa	1
	2	Salvar Mapa	1
	3	Eliminar Mapa	1/2
	4	Buscar Mapa	1/2
	Total		3

Tabla 21. HU para la iteración 1

Tareas de Ingeniería para la Iteración #1:

Tarea de Ingeniería	
Número de Tarea: 1	Número de la HU: 1
Nombre de Tarea: Implementar funcionalidad Crear Mapa	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 26/2/2013	Fecha Fin: 4/3/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Crear Mapa, el sistema muestra esta opción que permite al usuario diseñar su mapa conceptual del modo que desee.	

Tabla 22. Tarea de Ingeniería Implementar funcionalidad Crear Mapa

Tarea de Ingeniería	
Número de Tarea: 2	Número de la HU: 2

Nombre de Tarea: Implementar funcionalidad Salvar Mapa	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 5/3/2013	Fecha Fin: 11/3/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Salvar Mapa, el sistema muestra esta opción que permite al usuario guardar su mapa conceptual.	

Tabla 23. Tarea de Ingeniería Implementar funcionalidad Salvar Mapa

Tarea de Ingeniería	
Número de Tarea: 3	Número de la HU: 3
Nombre de Tarea: Implementar funcionalidad Eliminar Mapa	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 12/3/2013	Fecha Fin: 15/3/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Eliminar Mapa, el sistema muestra esta opción que permite al usuario eliminar su mapa conceptual anteriormente elaborado.	

Tabla 24. Tarea de Ingeniería Implementar funcionalidad Eliminar Mapa

Tarea de Ingeniería	
Número de Tarea: 4	Número de la HU: 4 Buscar Mapa
Nombre de Tarea: Implementar funcionalidad Buscar Mapa	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 16/3/2013	Fecha Fin: 18/3/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	

Descripción: Se implementa la funcionalidad Buscar Mapa, el sistema muestra esta opción que permite al usuario encontrar los mapas que se hayan realizados y guardados en la base de datos del sistema.

Tabla 25. Tarea de Ingeniería Implementar funcionalidad Buscar Mapa

Iteración 2:

Iteración	No. HU	Historia de Usuario	Duración (Semanas)
2	5	Insertar Comentario	1
	6	Autenticar Usuario	1
	7	Buscar Usuario	1
	Total		3

Tabla 26. HU para la iteración 2

Tareas de Ingeniería para la Iteración #2:

Tarea de Ingeniería	
Número de Tarea: 5	Número de la HU: 5
Nombre de Tarea: Implementar funcionalidad Insertar Comentario	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 19/3/2013	Fecha Fin: 25/3/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Insertar Comentario, el sistema muestra esta opción que permite al usuario escribir un comentario acerca del mapa.	

Tabla 27. Tarea de Ingeniería Implementar funcionalidad Insertar Comentario

Tarea de Ingeniería	
Número de Tarea: 6	Número de la HU: 6
Nombre de Tarea: Implementar funcionalidad Autenticar Usuario	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 26/3/2013	Fecha Fin: 1/4/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Autenticar Usuario, el sistema muestra esta opción que permite al usuario entrar los datos usuario y contraseña para tener acceso a la aplicación.	

Tabla 28. Tarea de Ingeniería Implementar funcionalidad Autenticar Usuario

Tarea de Ingeniería	
Número de Tarea: 7	Número de la HU: 7
Nombre de Tarea: Implementar funcionalidad Buscar Usuario	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 2/4/2013	Fecha Fin: 8/4/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Buscar Usuario, el sistema muestra esta opción que permite entrar el usuario que desea buscar.	

Tabla 29. Tarea de Ingeniería Implementar funcionalidad Buscar Usuario

Iteración 3:

Iteración	No. HU	Historia de Usuario	Duración (Semanas)
------------------	-------------------	----------------------------	-------------------------------

3	8	Añadir Colaborador	1
	9	Eliminar Colaborador	1/2
	10	Importar Mapa	1/2
	11	Editar Mapa	1
	Total		3

Tabla 30. HU para la iteración 3

Tareas de Ingeniería para la Iteración #3:

Tarea de Ingeniería	
Número de Tarea: 8	Número de la HU: 8
Nombre de Tarea: Implementar funcionalidad Añadir Colaborador	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 9/4/2013	Fecha Fin: 15/4/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Añadir Colaborador; el sistema permite añadir a un usuario como colaborador.	

Tabla 31. Tarea de Ingeniería Implementar funcionalidad Añadir Colaborador

Tarea de Ingeniería	
Número de Tarea: 9	Número de la HU: 9
Nombre de Tarea: Implementar funcionalidad Eliminar Colaborador	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 16/4/2013	Fecha Fin: 19/4/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	

Descripción: Se implementa la funcionalidad Eliminar Colaborador, el sistema muestra al usuario su lista de colaboradores permitiéndole eliminar el que desee.

Tabla 32. Tarea de Ingeniería Implementar funcionalidad Eliminar Colaborador

Tarea de Ingeniería	
Número de Tarea: 10	Número de la HU: 10
Nombre de Tarea: Implementar funcionalidad Importar Mapa	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 20/4/2013	Fecha Fin: 22/4/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Importar Mapa, el sistema muestra esta opción que permite al usuario importar cualquier mapa que haya sido previamente salvado.	

Tabla 33. Tarea de Ingeniería Implementar funcionalidad Importar Mapa

Tarea de Ingeniería	
Número de Tarea: 11	Número de la HU: 11
Nombre de Tarea: Implementar funcionalidad Editar Mapa	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 23/4/2013	Fecha Fin: 29/4/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Editar Mapa, el sistema permite realizar cambios sobre el mapa bajo la condición que este usuario haya sido seleccionado como colaborador por el usuario autor de ese mapa.	

Tabla 34. Tarea de Ingeniería Implementar funcionalidad Editar Mapa

Iteración 4:

Iteración	No. HU	Historia de Usuario	Duración (Semanas)
4	12	Exportar Mapa	1
	13	Insertar Imagen	1
	14	Trazas del Mapa	1
	Total		3

Tabla 35. HU para la iteración 4

Tareas de Ingeniería para la Iteración #4:

Tarea de Ingeniería	
Número de Tarea: 12	Número de la HU: 12
Nombre de Tarea: Implementar funcionalidad Exportar Mapa	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 30/4/2013	Fecha Fin: 6/5/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Exportar Mapa, el sistema muestra esta opción que permite al usuario exportar el mapa que haya elaborado.	

Tabla 36. Tarea de Ingeniería Implementar funcionalidad Exportar Mapa

Tarea de Ingeniería	
Número de Tarea: 13	Número de la HU: 13
Nombre de Tarea: Implementar funcionalidad Insertar Imagen	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 7/5/2013	Fecha Fin: 13/5/2013

Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez
Descripción: Se implementa la funcionalidad Insertar Imagen, el sistema muestra esta opción que permite al usuario escoger la imagen que desee e insertarla en su mapa.

Tabla 37. Tarea de Ingeniería Implementar funcionalidad Insertar Imagen

Tarea de Ingeniería	
Número de Tarea: 14	Número de la HU: 14
Nombre de Tarea: Implementar funcionalidad Trazas del Mapa	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 14/5/2013	Fecha Fin: 20/5/2013
Programador Responsable: Grabiél Montero Coello Yordanka Pupo Gómez	
Descripción: Se implementa la funcionalidad Trazas del Mapa, el sistema muestra al usuario los cambios realizados sobre el mapa.	

Tabla 38. Tarea de Ingeniería Implementar funcionalidad Trazas del Mapa

4.2. Validación de la Propuesta

La validación consiste en la realización de pruebas durante el desarrollo de un software; es una labor de mucha importancia debido a que dichas pruebas propician que el producto se desarrolle con calidad y tenga buena aceptación por el cliente. Las pruebas de software consisten en un conjunto de actividades en la que un sistema o componente es ejecutado bajo unas condiciones o requisitos especificados, los resultados de estas son registrados y analizados con el fin de corregir cualquier deficiencia encontrada.

Pruebas unitarias

Las pruebas unitarias aseguran que un único componente de la aplicación produce una salida correcta para una determinada entrada. Este tipo de pruebas validan la forma en la que las funciones y métodos trabajan en cada caso particular (64).

Deben ser construidas por los programadores con el empleo de algún mecanismo que permita

automatizarlas de modo tal que tanto su implementación y ejecución consuman el menor tiempo posible permitiendo sacarles el mejor provecho (47).

Estas pruebas son ejecutadas constantemente a todos los métodos significativos de las clases del proyecto y se realizan nuevamente ante cada modificación del sistema. Ayudan a asegurar que las partes individuales que conforman el sistema funcionen correctamente.

Para la realización de las pruebas unitarias al software realizado, la plataforma de desarrollo Symfony2 integra una librería independiente llamada PHPUnit como framework para desarrollar tests y cuenta con el directorio Tests para gestionarlas.

Pruebas de aceptación

Las pruebas de aceptación son creadas a partir de las historias de usuario. Durante una iteración la historia de usuario seleccionada se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a testear cuando una historia de usuario ha sido correctamente implementada. Una HU puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento (65).

Las pruebas de aceptación poseen gran importancia ya que permiten confirmar que la historia de usuario ha sido implementada correctamente. Su principal objetivo es validar que el sistema cumple con el funcionamiento esperado y permitir al cliente determinar su aceptación.

A continuación se muestran los datos que recogen las pruebas de aceptación:

Código: Identificador de la prueba.

Número Historia de Usuario: Número de la Historia de Usuario asociada a la prueba.

Nombre: Nombre que se le da a la prueba a realizar.

Descripción: Se describe la funcionalidad que se desea probar.

Condiciones de Ejecución: Se explican las condiciones que deben cumplirse para poder llevar a cabo la prueba.

Entradas / Pasos de Ejecución: Se detallan los pasos a seguir durante el desarrollo de la prueba.

Resultado esperado: Breve descripción del resultado que se espera obtener con la prueba realizada.

Evaluación de la prueba: Se emite una evaluación sobre la misma que puede ser Satisfactoria, Parcialmente Bien o Mal.

Prueba de Aceptación	
Código de prueba: P1.	Número Historia de Usuario: 1.
Nombre: Crear Mapa	
Descripción: Prueba para verificar la funcionalidad Crear Mapa.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. 	
Entrada/ Pasos de Ejecución:	
<ul style="list-style-type: none"> • El usuario accede a la interfaz inicial, selecciona la opción Crear Mapa. Selecciona los elementos (concepto y enlace) necesarios para diseñar su mapa y trabaja con ellos. 	
Resultado Esperado: Se crea el mapa conceptual.	
Evaluación de la prueba: Satisfactoria	

Tabla 39. Prueba de aceptación No 1

Prueba de Aceptación	
Código de prueba: P2.	Número Historia de Usuario: 2.
Nombre: Salvar Mapa	
Descripción: Prueba para verificar la funcionalidad Salvar Mapa.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. • El usuario debe estar editando algún mapa conceptual. 	
Entrada/ Pasos de Ejecución:	
<ul style="list-style-type: none"> • El usuario selecciona la opción Salvar. 	
Resultado Esperado: El sistema guarda el mapa y muestra un mensaje notificando que se ha salvado correctamente.	
Evaluación de la prueba: Satisfactoria	

Tabla 40. Prueba de aceptación No 2

Prueba de Aceptación	
Código de prueba: P3.	Número Historia de Usuario: 3.
Nombre: Eliminar Mapa	
Descripción: Prueba para verificar la funcionalidad Eliminar Mapa.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. • El usuario debe haber creado algún mapa conceptual y luego lo ha salvado. 	
Entrada/ Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la opción Eliminar. 	
Resultado Esperado: El sistema elimina el mapa conceptual.	
Evaluación de la prueba: Satisfactoria	

Tabla 41. Prueba de aceptación No 3

Prueba de Aceptación	
Código de prueba: P4.	Número Historia de Usuario: 4.
Nombre: Buscar Mapa	
Descripción: Prueba para verificar la funcionalidad Buscar Mapa.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. • El usuario debe haber creado algún mapa conceptual y luego lo ha salvado. 	
Entrada/ Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la opción Buscar Mapa. 	
Resultado Esperado: El sistema muestra el mapa conceptual.	
Evaluación de la prueba: Satisfactoria	

Tabla 42. Prueba de aceptación No 4

Prueba de Aceptación	
Código de prueba: P5.	Número Historia de Usuario: 5.

Nombre: Insertar Comentario
Descripción: Prueba para verificar la funcionalidad Insertar Comentario.
Condiciones de Ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. • El usuario debe haber creado algún mapa conceptual y luego lo ha salvado o el usuario debe ser colaborador de algún mapa.
Entrada/ Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la opción Insertar Comentario.
Resultado Esperado: El sistema permite al usuario escribir su comentario.
Evaluación de la prueba: Satisfactoria

Tabla 43. Prueba de aceptación No 5

Prueba de Aceptación	
Código de prueba: P6.	Número Historia de Usuario: 6.
Nombre: Autenticar Usuario	
Descripción: Prueba para verificar la funcionalidad Autenticar Usuario.	
Condiciones de Ejecución:	
Entrada/ Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario introduce los datos usuario y contraseña. 	
Resultado Esperado: El sistema muestra la interfaz principal.	
Evaluación de la prueba: Satisfactoria	

Tabla 44. Prueba de aceptación No 6

Prueba de Aceptación	
Código de prueba: P7.	Número Historia de Usuario: 7.
Nombre: Buscar Usuario	
Descripción: Prueba para verificar la funcionalidad Buscar Usuario.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. 	
Entrada/ Pasos de Ejecución:	

<ul style="list-style-type: none"> El usuario introduce otro usuario que desea buscar y presiona la opción Buscar Usuario.
Resultado Esperado: El sistema muestra el usuario encontrado.
Evaluación de la prueba: Satisfactoria

Tabla 45. Prueba de aceptación No 7

Prueba de Aceptación	
Código de prueba: P8.	Número Historia de Usuario: 8.
Nombre: Añadir Colaborador	
Descripción: Prueba para verificar la funcionalidad Añadir Colaborador.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> El usuario debe estar autenticado en el sistema. Selecciona la opción "Buscar Usuario". 	
Entrada/ Pasos de Ejecución:	
<ul style="list-style-type: none"> El usuario selecciona un usuario como colaborador. 	
Resultado Esperado: El sistema añade el usuario como colaborador.	
Evaluación de la prueba: Parcialmente Bien	

Tabla 46. Prueba de aceptación No 8

Prueba de Aceptación	
Código de prueba: P9.	Número Historia de Usuario: 9.
Nombre: Eliminar Colaborador	
Descripción: Prueba para verificar la funcionalidad Eliminar Colaborador.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> El usuario debe estar autenticado en el sistema. 	
Entrada/ Pasos de Ejecución:	
<ul style="list-style-type: none"> El usuario selecciona en la lista de colaboradores a un usuario y lo desmarca como colaborador. 	
Resultado Esperado: El sistema elimina al usuario seleccionado de la lista de colaboradores.	

Evaluación de la prueba: Satisfactoria

Tabla 47. Prueba de aceptación No 9

Prueba de Aceptación	
Código de prueba: P10.	Número Historia de Usuario: 10.
Nombre: Importar Mapa	
Descripción: Prueba para verificar la funcionalidad Importar Mapa.	
Condiciones de Ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado en el sistema.• El usuario debe tener un mapa creado.	
Entrada/ Pasos de Ejecución: <ul style="list-style-type: none">• El usuario selecciona la opción Importar Mapa.	
Resultado Esperado: El sistema muestra el mapa que se importa.	
Evaluación de la prueba: Parcialmente Bien	

Tabla 48. Prueba de aceptación No 10

Prueba de Aceptación	
Código de prueba: P11.	Número Historia de Usuario: 11.
Nombre: Editar Mapa	
Descripción: Prueba para verificar la funcionalidad Editar Mapa.	
Condiciones de Ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado en el sistema.• El usuario debe ser colaborador de otro usuario.	
Entrada/ Pasos de Ejecución: <ul style="list-style-type: none">• El usuario selecciona la opción Editar Mapa	
Resultado Esperado: El sistema muestra el mapa conceptual y permite que se modifique el mapa.	

Evaluación de la prueba: Parcialmente Bien

Tabla 49. Prueba de aceptación No 11

Prueba de Aceptación	
Código de prueba: P12.	Número Historia de Usuario: 12.
Nombre: Exportar Mapa	
Descripción: Prueba para verificar la funcionalidad Exportar Mapa.	
Condiciones de Ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado en el sistema.• El usuario debe tener crear su mapa conceptual.	
Entrada/ Pasos de Ejecución: <ul style="list-style-type: none">• El usuario selecciona la opción Exportar Mapa.	
Resultado Esperado: El sistema exporta el mapa conceptual.	
Evaluación de la prueba: Mal	

Tabla 50. Prueba de aceptación No 12

Prueba de Aceptación	
Código de prueba: P13.	Número Historia de Usuario: 13.
Nombre: Insertar Imagen	
Descripción: Prueba para verificar la funcionalidad Insertar Imagen.	
Condiciones de Ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado en el sistema.• El usuario debe abrir un mapa conceptual o debe estar creando uno.	
Entrada/ Pasos de Ejecución: <ul style="list-style-type: none">• El usuario selecciona la opción Insertar Imagen.	
Resultado Esperado: El sistema permite al usuario seleccionar una imagen e insertarla en el diseño del mapa conceptual.	
Evaluación de la prueba: Mal	

Tabla 51. Prueba de aceptación No 13

Prueba de Aceptación	
Código de prueba: P14.	Número Historia de Usuario: 14.
Nombre: Trazas del Mapa	
Descripción: Prueba para verificar la funcionalidad Trazas del Mapa.	
Condiciones de Ejecución: <ul style="list-style-type: none"> El usuario debe estar autenticado en el sistema. 	
Entrada/ Pasos de Ejecución: <ul style="list-style-type: none"> El usuario selecciona la opción Trazas del mapa. 	
Resultado Esperado: El sistema muestra todas las actividades realizadas sobre los mapas conceptuales del usuario.	
Evaluación de la prueba: Mal	

Tabla 52. Prueba de aceptación No 14

Registro de no conformidades

Cuando se realizan las pruebas se detectan algunas no conformidades en cada una de las iteraciones. Las no conformidades (NC) son errores encontrados y funcionalidades no esperados por el cliente. Al final de cada iteración se le muestra al cliente una versión del software de manera que pueda detectar las no conformidades que serán corregidas al inicio de la subsiguiente iteración. A continuación se muestra una gráfica representando la cantidad de NC en cada iteración (Ver Ilustración 6):

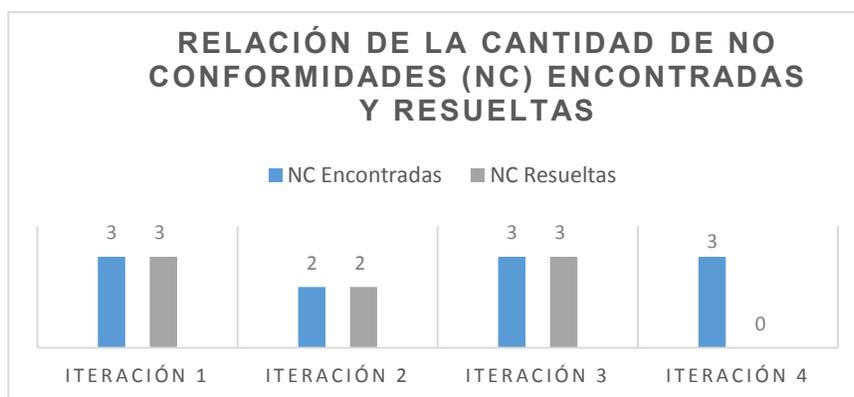


Ilustración 6. Relación de NC

4.3 Conclusiones parciales

En este capítulo se mostraron las HU que fueron implementadas en cada iteración. Se describieron las Tareas de Ingeniería correspondiente a dichas HU. Además se realizó la validación del sistema mediante las pruebas unitarias y pruebas de aceptación. Las mismas arrojaron diversas no conformidades identificadas en cada iteración; estas ayudaron a mejorar el producto elaborado. A pesar de no ser corregidas las NC encontradas en la iteración 4, estas no afectan el funcionamiento básico de la aplicación desarrollada por lo cual se obtiene un producto listo para ser utilizado.

CONCLUSIONES GENERALES

- ❖ Se realizó un estudio acerca del aprendizaje significativo, los mapas conceptuales y su aplicación en el proceso de enseñanza-aprendizaje; demostrando que la creación de mapas conceptuales, es una buena práctica que pueden usar los estudiantes universitarios para desarrollar un aprendizaje significativo que les permita analizar y sintetizar los conocimientos que reciben.
- ❖ Se realizó un análisis sobre algunas herramientas informáticas que se utilizan para la realización de mapas conceptuales. Se demuestra que algunas pueden ser trabajadas desde la computadora sin estar conectadas a la red; y otras que están basadas en la Web 2.0 pueden utilizarse a través del acceso pleno a internet facilitando la colaboración entre usuarios.
- ❖ Se realizó el análisis del software para un corto período de tiempo contando con una planificación; y se elaboró el diseño siguiendo las instrucciones que propone la metodología XP.
- ❖ Se modeló las tareas de implementación de la herramienta para la elaboración de mapas conceptuales colaborativos basados en la Web 2.0; estas lograron abarcar todas las funcionalidades necesarias para la obtención de un software aplicativo.
- ❖ Se realizó la validación de la solución del software propuesto a través de las pruebas unitarias y las pruebas funcionales.

RECOMENDACIONES

- ❖ Hacer uso del presente trabajo como herramienta de apoyo al proceso de enseñanza- aprendizaje en todas las facultades de la UCI.
- ❖ Elaborar nuevas versiones de la aplicación para realizar las funcionalidades que quedaron pendientes con la finalidad de ampliar las operaciones que brinda el sistema.

REFERENCIAS BIBLIOGRÁFICAS

1. **Bases y principios del proceso de enseñanza-aprendizaje centrado en el aprendizaje en la UCI.**
2. **Moreira, M.A. Mapas conceptuales y aprendizaje significativo en ciencias.** Porto Alegre : s.n., 1988.
3. **Fernández, Fatima. A. Didáctica: Teoría y Práctica.** La Habana : Pueblo y Educación, 2004.
4. **Novak, J.D. Learning Science and the Science of Learning.** *Studies in Science Education.* 1988.
5. **Ausubel, D.P. Educational Psychology: A cognitive view.** New York: Holt, Rinehart and Winston : s.n., 1968.
6. **Hernandez, A. S. Tesis Doctoral: Mapas Conceptuales y condiciones instruccionales.**
7. **Abella , Fernando R. Utilización de los Mapas Conceptuales como herramienta evaluadora del aprendizaje significativo de alumno universitario en ciencias con independencia de su conocimiento de la metodología .** Facultad de Ciencias de la Educación, Universidad Ramon Llull. 2008.
8. **Pérez de Prado, Antonio. Tesis de Maestría Uso de Mapas Conceptuales para la enseñanza de la física.** Universidad de Matanzas. Matanzas : s.n., 1999.
9. **Novak, J.D. y Gowin, D.B. Aprendiendo a aprender. .** Barcelona: Martinez Roca. 1988.
10. **González, Anabel. La importancia de los mapas conceptuales en el proceso de enseñanza-aprendizaje. Ejemplos ilustrativos de ello en temas claves de la educación.** 2009. ISSN 1988-6047.
11. **Novak, J. Reporte. Ayudar a los alumnos a aprender cómo aprender. La opinión de un profesor-investigador en Enseñanza de las Ciencias.** 1991.
12. **Colectivo, autores. Preparación Pedagógica Integral.** Ciudad de la Habana : Félix Varela, 2003. ISBN 959-258-535-0.
13. **Pérez, Javier Eguíluz. LibrosWeb: Introducción a JavaScript. [En línea] 2007.** http://librosweb.es/javascript/capitulo_1/breve_historia.html..
14. **Inspiration es una herramienta para el diseño de Mapas Conceptuales.** <http://www.dataprix.com/empresa/recursos/inspiration-herramienta-para-diseno-mapas-conceptuales>. [En línea]

15. Ojeda, A. D. C., Fe E., y otros. *Los mapas conceptuales: una poderosa herramienta para el aprendizaje significativo*. 2007. ACIMED.
16. Elabora esquemas y mapas conceptuales colaborativos con Creately. [En línea] <http://laboratorio.educacontic.es/Elabora+esquemas+y+mapas+conceptuales+colaborativos+con+Creately>.
17. ineveryCREA Argentina. [En línea] <http://www.ineverycrea.com.ar/comunidad/ineverycreaargentina/recurso/Creately-para-elaborar-mapas-conceptuales-y-esque/7b132316-eca3-441d-a94c-5bd5620b97fa>.
18. Gliffy. [En línea] <http://invete.wikispaces.com/Gliffy>.
19. Regalado, O.L. *Mapas Conceptuales con CmapTools*.
20. Quispe Carita, Vilma, Huamantuco Solorzano, Dante Harry y Vargas Yupanqui, José Luis. *METODOLOGIA RUP (RATIONAL UNIFIED PROCESS)*. Perú : s.n., 2011.
21. Acuña, K.B. *Selección de metodologías de desarrollo para aplicaciones web en la facultad de informática de la Universidad de Cienfuegos*. Universidad de Cienfuegos. Cienfuegos : s.n., 2009.
22. Solis, R.T. *XP Programación Extrema para Desarrollo de Sistema Basados en Web*. 2010.
23. Ivar Jacobson, Grady Booch y James Rumbaugh. *El Proceso Unificado de Desarrollo. La guía completa del Proceso Unificado*. Madrid : s.n., 2000.
24. Carrillo, P.I., Pérez, G.R. *Metodología del desarrollo del software*. 2008.
25. Canós, J.H., Letelier, P., Penadés, M.C. *Métodologías Ágiles en el Desarrollo de Software*. Universidad Politécnica de Valencia.
26. Visual Paradigm for UML. [En línea] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p).
27. Desarrollo Web. [En línea] [Citado el: 20 de enero de 2013.] <http://www.desarrolloweb.com/articulos/392.php>.
28. desarrolloweb.com. *Qué es JavaScript*. [En línea]

<http://www.prograweb.com.mx/pweb/0202ladoCliente.html>.

29. LibrosWeb. *Capítulo 1. Introducción. ¿Qué es JavaScript?* [En línea] http://librosweb.es/javascript/capitulo_1.html.

30. JSON. [En línea] [Citado el: 16 de enero de 2013.] <http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=1090..>

31. Introducción a jQuery. [En línea] <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>.

32. Alvarez, M. A. Manual de JQuery. [En línea] 2010. DESARROLLOWEB.COM.

33. Bustamante, Y.Ch.,Martínez,E.R. *Sistema Web Inteligente apoyado en Mapas Conceptuales para la asignatura de Álgebra Lineal*. 2012. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas..

34. González Barahona, J.S.y.G.R. *Introducción al software libre*. 2007.

35. NetBeansIDE. [En línea] [Citado el: 12 de abril de 2013.] www.netbeans.org.

36. Pacheco, Nacho. *Manual de Symfony2 Release 2.0.1*. 2011.

37. Potencier, Fabien. *Guía de inicio rápido. Symfony 2*. 2011.

38. Maestros del web. [En línea] 28 de mayo de 2013. <http://www.maestrosdelweb.com/editorial/web2/>.

39. Web 2.0 y educación. [En línea] [Citado el: 28 de mayo de 2013.] <http://carmenalicialopez.wordpress.com/>.

40. Area Moreira, Manuel. *Introducción a la Tecnología*. Universidad de la Laguna. España : s.n., 2009. Manual electrónico. licencia Creative Commons.

41. Valdés, Damián Pérez. Maestros de la web . [En línea] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web..>

42. Ricote, A. y Figueredo, M. *SGIF: Sistema de Gestión de la Información para la facultad*. Ciudad de La Habana : s.n., 2006.

43. Conceptos básicos del servidor web. [En línea] http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php..
44. The Apache Software Foundation. [En línea] 2011. <http://httpd.apache.org/>..
45. Las principales características de MYSQL. [En línea] <http://dev.mysql.com/doc/refman/5.0/es/features.html>.
46. Soy de Mac. [En línea] <http://www.soydemac.com/2010/12/16/oracle-lanza-mysql-5-5-con-nuevas-caracteristicas-y-beneficios-review/>.
47. Echeverry, L.M y Delgado, L.E. *Caso práctico de la metodología ágil XP al desarrollo del software*. 2007.
48. Kent Beck, Martin Fowler. *Planificando la programación extrema*. Traducido de: Planning Extreme Programming.
49. Martín Morales, Humberto y Abreu Lugo, Marlon. *Herramienta para la gestión de contenido y configuración de ejercicios y clases reutilizables*. Ciudad de La Habana : s.n., 2009.
50. González Cornejo, J. E. *Arquitectura en Capas. Un camino hacia los procesos distribuidos*. 2001.
51. Clements, Paul C. *A survey of Architecture Description Languages*. 1996. Software Engineering Institute Carnegie Mellon University.
52. Reynoso, Carlos Billy. *Introducción a la Arquitectura de Software*. Buenos Aires : s.n., 2004. Universidad de Buenos Aires.
53. Pressman, Roger S. *Ingeniería del software*. 2002. McGrawHill.
54. Velásquez, Key. Patrones Arquitectónicos. [En línea] [Citado el: 21 de marzo de 2013.] <http://www.buenastareas.com/ensayos/Patrones-Arquitectonicos/1069013.html>.
55. DesarrolloWeb.com. [En línea] <http://www.desarrolloweb.com/wiki/mvc-modelo-vista-controlador.html>.
56. Patrón MVC. [En línea] 2008. [Citado el: 13 de enero de 2013.] <http://www.proactiva-calidad.com/java/patrones/mvc.htm>..
57. Cárdenas, Zuzel Díaz Pérez Adony González. *Sistema Gestor de Trabajos de Diploma*. Habana :

- s.n., 2009. TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN INFORMÁTICA.
58. Eguiluz, Javier . Desarrollo web ágil con Symfony2. 2012.
59. García, Joaquín. IngenieroSoftware. [En línea] 2005.
<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php..>
60. Durante Lerate, Rosa M, González, Leandro Pastrana y Montes, Noelia Sales. *Sistemas para el Control de Versiones*. 2008.
61. Grosso, Andrés. Prácticas de Software: Patrones Grasp. [En línea] 2011.
<http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
62. Larman, Craig. *UML y Patrones Introducción al análisis y diseño orientado a objetos*. La Habana : Félix Varela, 2004.
63. Craig, Larman. *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos*. 1999.
64. librosweb.es. [En línea] mayo de 2013.
65. Fernández, Escribano Gerardo. *Introducción a Extreme Programming*. 2002.

GLOSARIO DE TÉRMINOS

Aprendizaje: Proceso mediante el cual un individuo adquiere conocimiento y experiencia, de forma que le permita modificar su comportamiento y adaptarlo a nuevas condiciones de su entorno.

GPL: Es una licencia pública general de GNU o más conocida por su nombre en inglés *General Public License*. Está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

GRASP: General Responsibility Assignment Software Patterns o Patrones de Principios Generales para Asignar Responsabilidades. Son una serie de buenas prácticas de aplicación recomendable en el diseño de software.

IDE: Integrated Development Environment o Entorno de Desarrollo Integrado. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de gráficas GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de otras aplicaciones existentes.

Interfaz: Es uno de los componentes más importantes de cualquier sistema computacional, funciona como el vínculo o comunicación entre el humano y la máquina.

JavaScript: Lenguaje de programación interpretado que permite mejoras en la interfaz de usuario y páginas web dinámicas.

Multiplataforma: es un término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

Software: Término en inglés para describir a los programas de computación. Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. Soporte lógico de un computador digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica.

TIC: Tecnologías de la Información y las Comunicaciones. Son todos los servicios, software y hardware que interconectados contribuirán a mejorar las condiciones de vida de las personas.

ANEXOS

Anexo1:

Se realizó una entrevista de forma individual y estructurada. A continuación se muestran las preguntas realizadas:

- ❖ ¿Cómo usted considera que el estudiante se aprende una materia?
- ❖ ¿Cree usted que los estudiantes utilizan un aprendizaje memorístico y reproductivo? ¿Por qué?
- ❖ ¿Cree usted como profesor que los estudiantes logran vincular un contenido anteriormente recibido con el contenido nuevo que reciben?
- ❖ ¿Cree usted que muchos estudiantes carecen de las habilidades análisis y síntesis de los contenidos?
- ❖ ¿Ha trabajado usted alguna vez con los mapas conceptuales en función del aprendizaje significativo en los estudiantes?
- ❖ ¿Ha podido constatar la utilización de los mapas conceptuales por parte de los estudiantes en trabajos independientes, talleres, seminarios u otras tareas expuestas en clases?