



Plataforma Web para la gestión de información científico-técnica de la red de investigadores de la Universidad de las Ciencias Informáticas

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.

Autores: Danaray Antigua Hopuy, Edisvel Melo Estevez

Tutores: Ing. Andy José Rivas Franco, Lic. Raynel Batista Tellez

Co-tutor: Ing. Damián Pérez Alfonso

Facultad 3

Junio de 2015

“En la tierra hace falta personas que trabajen más y critiquen menos, que construyan más y destruyan menos, que prometan menos y resuelvan más, que esperen recibir menos y dar más, que digan mejor ahora que mañana.”

Che



DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Edisvel Melo Estevez

Danaray Antigua Hopuy

Firma del Autor

Firma del Autor

Ing. Andy José Rivas Franco

Lic. Raynel Batista Tellez

Firma del Tutor

Firma del Tutor

Agradecimientos

A mi madre y padre Mercedes Melo Estevez por traerme al mundo por su constancia, respeto, sacrificio, por ser el hombro fiel donde siempre me he podido apoyar en todo momento, por su amor incondicional, por apoyarme, por su valor, por su excelente educación, por ser la razón que me levante cada día a esforzarme por el presente y el mañana. Gracias por darme la oportunidad de estudiar esta carrera, por promover mi desarrollo, debido a ti hoy me siento un hombre completo.

A mi hermana y madre Ambar Sarai Melo Estevez por su amor, dedicación, paciencia, en todo momento y horas de muchos consejos, ser parte importante de mí y representar la unidad familiar, por aconsejarme a todo momento, por ayudarme y apoyarme en los momentos malos y buenos a ti por ser una excelente hermana, amiga, mujer y madre.

A Sergio Alejandro Galvez Melo y a Cesar Angelo Galvez Melo por ser en todo momento mi ejemplo a seguir por sus momentos de inocencias, por ser uno de los más grandes tesoros que tengo en el mundo.

A Sergio Alberto Galvez Sardiñas y a la memoria de Frank David Galvez Fernández por ser como una familia y en todo momento querer y respetar a mi hermana, hecho suficiente para llevarlos siempre en mí y estar sumamente orgulloso de ustedes.

A mis Padrinos Angela y Cardena los cuales son un regalo para mí.

A toda mi familia gracias.

A mis tutores que se convirtieron en más que amigos, a Andy y Raynel.

A todos mis amigos y amistades de la universidad que he tenido el placer de conocer en especial a Iriannis, Juan, Nelson, Iliannis, Greter, Yanna, Ivanieta, Dayana, Yiselly, Mario, Tato, Oridalmis, Cesar, Alfredo, por estar ahí cuando los necesité.

A mis amigos de Facebook como Rosa, Mayelin Yohana, Milena, Eduardo, Sandra y, Sulay por sus aportes y buenos consejos, siempre los tendré presentes.

Como olvidar a una persona que apareció en mi vida cuando menos lo esperaba y gracias a su amor hoy representa más que aquellos que estuvieron cuando él no estaba, a mi amigo y más, a ti mil gracias Alejandro García Santiesteban.

Por último pero no menos importante a Danaray Antigua Hopuy mi compañera de tesis y mi amiga nunca dejes de ser como eres, en tan poco tiempo te convertiste en lo que pocos han logrado.

A todos muchas gracias De Edisvel Melo Estevez.

Le agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.

Le doy gracias a mis padres: Mayra Hopuy Díaz y Orlando Lázaro Antigua Bonacheas por apoyarme en todo momento, por los valores que me han inculcado, por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida, por ser una parte muy importante de mi vida, por haberme apoyado en las buenas y en las malas, sobre todo por su paciencia y amor incondicional, por ser la razón de que me levante cada día a esforzarme por el presente y el mañana, son mi principal motivación; gracias por haberme forjado a ser la persona que soy en la actualidad; muchos de mis logros se los debo a ustedes entre los que se incluye éste. Me formaron con reglas y con algunas libertades, pero al final de cuentas, me motivaron constantemente para alcanzar mis anhelos. Gracias por darme la oportunidad de estudiar esta carrera, por promover el desarrollo y la unión familiar en nuestra familia, en fin debido a la existencia de ustedes hoy me siento plena.

Me gustaría agradecer sinceramente a mis Tutores: Raynel Batista Tellez y Andy José Rivas Franco, por sus esfuerzos y dedicación. Sus conocimientos, orientaciones, manera de trabajar, sus persistencias, paciencia y motivación han sido fundamentales para mi formación como investigador. Han sido capaz de ganarse mi lealtad y admiración, así como sentirme en deuda con ustedes por todo lo recibido durante el período de tiempo que ha durado esta Tesis.

A mis dos hermanos por ser parte importante de mi vida y representar la unidad familiar, por aconsejarme a todo momento, por ayudarme y apoyarme en los momentos malos y buenos en el transcurso de la vida, por mostrarme lo bueno que es tener hermanos, por llenar mi vida de alegrías y amor cuando más lo he necesitado. ¡Gracias de Corazón!.

A mi tío Henry porque nunca dudaste de mí capacidad y siempre me incentivaste a seguir adelante. Además por tu forma tan especial de hacerme sentir bien en las peores circunstancias; gracias a todos

mis tíos, tías, primos y primas por su fortaleza en los momentos más difíciles, sin saberlo me ha enseñado mucho.

A todos mis amigos, sin excluir a ninguno, pero en especial a Carola, Duvergel, Fernando, Osmanis, que compartimos largas conversaciones, en las que muchos de nuestros planteamientos y convicciones se vieron reforzadas o reconfiguradas, gracias por confiar y creer en mí y haber hecho de mi etapa universitaria un trayecto de vivencias que nunca olvidaré. Por apoyarme en todos los momentos que los necesité, por compartir momentos inolvidables a su lado, por seguir disfrutando de nuestra amistad. Son personas incondicionales para mí, por lo que estaremos juntos en los buenos y malos momentos. Además de que siempre han creído en mí, y me han brindado su apoyo en cualquier situación.

A mi compañero y amigo Edisvel Melo Estevez por ser una persona muy especial, te he aprendido a querer en tan poco tiempo, espero que nunca cambies. Gracias por darme la oportunidad en todo este transcurso de la carrera de conocer a una persona realmente sincera, carismática, agradable, y sobre todo un buen amigo, te quiero y es de corazón.

Finalmente a todas aquellas personas que han y están marcado mi vida y que me han permitido ser parte de la suya, alumnos, compañeros de trabajo entrañables que han separado su camino y los que ahora caminan conmigo, y quienes confiaron en mi de una u otra forma para dejar huella por esta vida, a no pasar desapercibida, y a seguir siendo quien soy ahora, a todos ustedes de Danaray Antigua Hopuy.

Dedicatoria

Dedico esta tesis A. Mercedes Melo Estevez, A Ambar Sarai Melo Estevez y mis dos hijos Sergio Alejandro Galvez Melo y a Cesar Angelo Galvez Melo, quienes inspiraron mi espíritu para la conclusión de este trabajo de diploma. A todos aquellos que creyeron en mí. A todos ellos se los agradezco desde el fondo de mi alma. Para todos ellos hago esta dedicatoria.

Edisvel.

Dedico esta tesis A. Mayra Hopuy Díaz y Orlando Lázaro Antigua Bonacheas, a mis dos hermanos Carlos Rafael Hong Hopuy y Yoarneld Fernández Hopuy quienes son la razón de mí existir en este mundo. A todos aquellos que creyeron en mí. A todos ellos se los agradezco desde el fondo de mi alma.

Para todos ellos hago esta dedicatoria.

Danaray.

RESUMEN

El avance de las tecnologías de la información ha impactado el sector de la ciencia al proponer nuevos modos de intercambiar, compartir y socializar conocimientos en beneficio del desarrollo social. El contexto universitario provee el escenario ideal para potenciar el diálogo entre investigadores, especialistas, académicos, tecnólogos, decisores y empresarios en función de catalizar e introducir resultados. Sin embargo, en la Universidad de las Ciencias Informáticas la dispersión de los datos y fuentes de información ha estimulado el uso irracional de recursos y el desarrollo aislado de grupos, proyectos y resultados de investigación, que sin la requerida sinergia pudieran estar limitados en cuanto a impacto, visibilidad y competitividad.

La presente investigación persigue el objetivo de desarrollar una solución informática que eleve el trabajo colaborativo en la red de investigadores de la universidad y contribuya por consiguiente a la producción científica. Los autores del estudio proponen ir más allá de los enfoques tradicionales de las aplicaciones web y analizan cómo orientar la solución hacia nuevas tendencias del consumo de información. Los resultados obtenidos demuestran los beneficios de una aplicación capaz de elevar el trabajo colaborativo en el contexto universitario.

Palabras claves: contexto universitario, gestión de la información, trabajo colaborativo.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
INTRODUCCIÓN.....	5
1.1 <i>Gestión de información científico técnica</i>	5
1.2 <i>La evolución de la web</i>	6
1.3 <i>Soluciones informáticas de gestión de información científico técnica</i>	8
1.3.1 Portal Universidad de la Habana	9
1.3.2 Portal Instituto Superior Politécnico José Antonio Echeverría	9
1.3.3 Portal Universidad de Buenos Aires	10
1.3.4 Portal Universidad de Complutense de Madrid	10
1.3.5 Portal Universidad Nacional Autónoma de México.....	10
1.3.6 Resultados del estudio de las soluciones informáticas de gestión de información científico técnica	11
1.4 <i>Estudio de metodologías, tecnologías y herramientas</i>	12
1.4.1 Proceso de desarrollo de software	12
1.4.2 Lenguaje de Modelado	15
1.4.3 Herramientas de Ingeniería de Software Asistida por Computación.....	15
1.4.4 Lenguajes de programación.....	15
1.4.5 Servidor Web	17
1.4.6 Bibliotecas y Marco de trabajo para el desarrollo	18
1.4.7 Herramientas IDE.....	19
1.4.8 Gestor de base de datos	20
1.5 <i>Conclusiones</i>	21
CAPÍTULO 2: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN	23
INTRODUCCIÓN.....	23
2.1 <i>Descripción de la propuesta de solución</i>	23
2.2 <i>Modelo de dominio</i>	24
2.3 <i>Requisitos de la plataforma</i>	25
2.3.1 Requisitos funcionales	25
2.3.2 Requisitos no funcionales	27
2.4 <i>. Historias de usuario</i>	28

2.4.1	Estimación de tiempo por Historias de Usuario	28
2.4.2	Plan de iteraciones	30
2.4.3	Plan de entregas	38
2.4.4	Resumen de tareas	39
2.5	<i>Prototipos de interfaz de usuario</i>	41
2.5.1	Prototipos de baja fidelidad	42
2.5.2	Prototipos de alta fidelidad	42
2.6	<i>Modelo de diseño</i>	47
2.6.1	Estilo de arquitectura	48
2.6.2	Patrones arquitectónicos y de diseño	48
2.6.3	Diagrama de clases de la solución	52
2.6.4	Vista de componentes	53
2.6.5	Diagrama de Despliegue	54
2.7	<i>Conclusiones del capítulo</i>	55
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA		56
INTRODUCCIÓN		56
3.1	<i>Pruebas de software</i>	56
3.1.1	Niveles de prueba	56
3.2	<i>Placer de uso</i>	57
3.3	<i>Resultados de las pruebas funcionales</i>	58
3.4	<i>Resultados sobre las pruebas de caja negra</i>	59
3.5	<i>Resultados de las pruebas de resistencia y rendimiento</i>	59
3.6	<i>Resultados de las pruebas para el placer de uso</i>	61
3.6.1	Pensando en voz alta	61
3.6.2	Emocard	62
3.7	<i>Experimento</i>	63
3.8	<i>Conclusiones del capítulo</i>	66
CONCLUSIONES		67
REFERENCIAS		68

INTRODUCCIÓN

El creciente y vertiginoso avance de las nuevas Tecnologías de la Información y las Comunicaciones (TIC) ha traído consigo grandes cambios en el contexto social y laboral donde se desenvuelve el hombre moderno. El impacto de la TIC en la gestión de información científico-técnica ha propuesto nuevos modos de crear, distribuir y socializar conocimientos. Este fenómeno ha conducido al auge y desarrollo tecnológico y posicionado el proceso de creación, distribución y administración de la información científico-técnica en una pieza clave del tablero socio-económico mundial.

El aumento de la disponibilidad y accesibilidad de la información en el mundo contemporáneo, ha iniciado el camino a un conjunto de aplicaciones orientadas al manejo de la cantidad de datos generada en todos los rincones del mundo. La disponibilidad y accesibilidad responden a las necesidades propias del dominio o negocio del que fueron implementadas, razón para tener en cuenta la dispersión y descentralización de los datos en que se procesan especialmente en una sociedad que consume grandes volúmenes de información.

Cuba no está ajena a este fenómeno y diversos han sido los esfuerzos del gobierno enfocados a otorgar al país de la infraestructura propicia para la implementación de los procesos relacionados con la gestión de información científico-técnica (MSTI¹). Un importante papel en este proceso lo ha ocupado la Universidad de las Ciencias Informáticas (UCI), institución innovadora, de excelencia científica, académica y productiva que forma de manera continua profesionales integrales comprometidos con la patria; soporte de la informatización del país y la competitividad internacional de la industria cubana del software. (UCI, 2008).

En una universidad tan dinámica y numerosa la administración de la información productiva e investigativa suscita gran importancia en el desarrollo integral de la institución. Si se adiciona el papel protagónico de la UCI que toma en la informatización del país, la gestión de información científico-técnica debe estar orientada al uso de sistemas informáticos que garanticen la certificación, disponibilidad, actualización, accesibilidad de la información y trabajo colaborativo.

¹ Por sus siglas en inglés Management of Scientific and Technical Information.

A pesar de que la institución mantiene activos a diferentes sistemas informáticos dedicados a la gestión de la información científico-técnica, la dispersión de los datos y las fuentes de información limitan los resultados de grupos y proyectos de investigación.

Un estudio preliminar de la investigación evidenció que los investigadores de la Universidad de las Ciencias Informáticas (UCI) se identifican por la diversidad de afiliaciones, al compartir sus intereses e integrarse a grupos de investigación y líneas temáticas, resultando compleja la interacción y el intercambio de información. Sin la requerida sinergia los resultados pudieran estar limitados en cuanto a impacto, visibilidad y competitividad.

Teniendo en cuenta el contexto anterior puede afirmarse que:

El modo en que se gestiona la información científico-técnica en la red de investigadores de la UCI limita la accesibilidad, disponibilidad, trabajo colaborativo, la certificación y actualización de los datos.

La investigación se enmarca en el área objeto de las *aplicaciones web para sistemas de gestión de información*, con el propósito general de *desarrollar una plataforma para el proceso de MSTI que contribuya a elevar el trabajo colaborativo en la red de investigadores de la UCI garantizando la accesibilidad, disponibilidad, la certificación y actualización de los datos*. El estudio se concentró en realizar aportaciones en el área de *aplicaciones web para la gestión de información científico-técnica en la UCI*, presuponiendo que el desarrollo de una plataforma web para la gestión de información científico técnica en la Universidad, se elevará el trabajo colaborativo en la red de investigadores.

Para alcanzar los resultados esperados en la investigación se definieron los siguientes objetivos específicos y tareas investigativas:

Objetivos específicos

1. Analizar herramientas para la gestión de la información científico-técnica de manera que permita comparar sus características e identificar sus potencialidades.
2. Identificar tendencias y enfoques de diseño en aplicaciones web para la gestión de la información científico-técnica que permitan elevar los mecanismos de recuperación de la información.
3. Implementar una plataforma web que a partir de enfoques novedosos de adaptabilidad, accesibilidad y “social networking” permita elevar el trabajo colaborativo en la red de investigadores.

4. Validar la herramienta propuesta a partir de pruebas de software, técnicas de evaluación de usabilidad y placer de uso.

Tareas de investigación

- Captura de los requisitos del sistema.
- Diseño y validación de los prototipos.
- Caracterización de herramientas para la gestión de la información científico-técnica.
- Exploración de tecnologías y lenguajes a emplear en el desarrollo de la plataforma.
- Realización del análisis y diseño de la plataforma.
- Implementación de los requisitos del sistema.
- Evaluación de la plataforma a partir de pruebas de software y técnicas de evaluación de usabilidad.

Para lograr el objetivo de esta investigación se emplearon los siguientes métodos científicos:

Métodos teóricos

1. **Analítico – Sintético:** permitió analizar individualmente los principales conceptos relacionados con la gestión de la información científico-técnica, posibilitando un análisis diferenciado y la identificación de relaciones que se establecen entre ellos.
2. **Histórico - Lógico:** se empleó para investigar las aplicaciones existentes que permiten la detección y/o inferencia de redes de interacción social y la utilización de estas de los datos interpersonales.
3. **Hipotético – Deductivo:** se utilizó para guiar la investigación desde el planteamiento del problema hasta la verificación de la solución a partir de las validaciones, orientando la secuencia lógica de las tareas que se realizaron, arribando a conclusiones particulares.
4. **Modelación:** para la creación de una representación o modelo del proceso de negocio haciendo posible el diseño de los prototipos y la realización de los diagramas necesarios en el proceso de desarrollo de software.

Métodos empíricos:

1. **Entrevista:** propició comprobar la forma en que se gestiona la información científico-técnica en la UCI, propiciando el levantamiento de requisitos de la aplicación.

2. **Medición:** permitió la aplicación de métricas y estándares al desarrollo de la solución para evaluar su nivel de efectividad.

CAPÍTULO 1: Fundamentación teórica

Introducción

En este capítulo se expone la fundamentación teórica del trabajo, incluyéndose en el mismo el estudio de otras soluciones informáticas existentes en la actualidad y de las metodologías, modelos de desarrollo de software y herramientas a utilizar.

1.1 Gestión de información científico técnica

La MSTI implementa actividades que están relacionadas con la transferencia de conocimientos a la sociedad, originados por la investigación científica. Proporcionan respuesta a las solicitudes que emanan de la misma, las cuales pueden abrir nuevas vías de investigaciones para dar solución a interrogantes que surgen en cualquier campo del saber humano.

La MSTI constituye una pieza clave para el conocimiento que se descubre mediante la observación, la reflexión de manera fortuita o mediante un esfuerzo de estudio o de investigación, conociéndose que el proceso de transmisión de la información se denomina comunicación, cobrando vital importancia para el desarrollo económico y social. Precisamente esta actividad ha estado a la vanguardia de las transformaciones tecnológicas acometidas internacionalmente (Ciencias de la Información, 2008).

Un breve recorrido por la historia de la MSTI permite conocer que se refiere a la actividad de búsqueda que se caracteriza por ser reflexiva, sistemática y metódica; tiene por finalidad obtener conocimientos y solucionar problemas científicos, filosóficos o empírico-técnicos. Su empleo ha sido reflejado en la sociedad desde la antigüedad y ha sido orientado por la búsqueda de conocimiento para la solución de problemas a partir de métodos. El método científico indica el camino que se ha de transitar en esa indagación y las técnicas precisan la manera de recorrerlo (Pedroso, 2004).

La MSTI constituye un soporte de transmisión de conocimientos distinguiéndolo como un producto del entorno social y cultural. Para algunos el término MSTI, en particular la búsqueda, solo irradia en las ideas que se tiene de las cosas: *“el conocimiento se adquiere a través de las ideas”* (Del Ramo, 2012). Sin embargo, el hacerse de ideas requiere de gestionar la información, es decir, a partir de la búsqueda seleccionar la información, tener mecanismos de organización para analizar y procesar ese conjunto de “ideas”, además que esas ideas, deben tener conexión unas con otras y mantener una retroalimentación

entre el conocimiento previo y el nuevo. El modelo de la MSTI en realidad no es algo nuevo, es modificar e ingresar nuevos instrumentos que le permitan realizar su gestión.

1.2 La evolución de la web

Desde el surgimiento de Internet y la socialización de la información a través de las redes de datos interconectadas, la web ha experimentado estadios en su evolución: de la representación estática de la información (web expositiva), pasando por la construcción colectiva de los datos (web interactiva), a la extracción del conocimiento (la web inteligente o de los datos). La evolución que ha experimentado la web en la última década ha repercutido en el día a día de casi toda actividad moderna de nuestra sociedad. La comprensión por tanto de estos, ayudar a proyectar nuevos métodos de trabajo acorde a las nuevas tecnologías y potencialidades.

Desde la creación de la web y el sistema de hipertexto por Tim Berners Lee, la tecnología base apunta al lenguaje de marcas de hipertexto. El cual es interpretado por los navegadores y utiliza como medio de transporte el protocolo HTTP. En resumen la web 1.0 propone un modelo en el cual la tecnología HTML fue suficiente dado su carácter estático, unidireccional, no colaborativo y de mínimas interacciones.

En cambio la evolución a la web 2.0 supuso la necesidad de incorporar tecnologías como PHP, Java, ASP.NET, etc. Según Nafría en el 2007: *"El usuario es el nuevo rey de Internet"* (Nafria, Ismael, 2007), la Web 2.0 centran sus servicios en función de los usuarios, brindándoles la posibilidad de convertirse en los actores principales para crear, usar, distribuir y compartir libremente la información en la web. Esta posibilidad de mayor interactividad y conexión entre los usuarios, unido al surgimiento de los dispositivos móviles y de nuevas formas de interacción como la incorporación de pantallas táctiles en computadoras tradicionales, el aumento en la resolución de los monitores y cambios en la relación de aspecto, propiciaron una revolución en la interacción hombre-maquina. Según Alexa Rank en el 2014 más de la mitad de los accesos a internet se hacen desde dispositivos móviles (Kalt,David, 2014). En el corazón de esta nueva forma de interacción se encuentra el "Diseño Web Adaptable" el cual es un enfoque que sugiere que el diseño y el desarrollo web deben responder al comportamiento del usuario y el medio ambiente de este, basado en el tamaño de la pantalla, la plataforma y la orientación de los dispositivos usados para acceder a los sitios web (Kadlec, 2013). El diseño web adaptable permite crear aplicaciones web que se adaptan al ancho del dispositivo en que se esté navegando, lo que permite tener un único diseño para PC, móviles y tabletas (Marcotte, 2011).

El diálogo con el usuario constituye uno de los aspectos más importantes de cualquier sistema interactivo y es precisamente la interfaz del sistema la que facilita al usuario acceder a los recursos del ordenador. Producto de este nuevo escenario surgen nuevos patrones de diseños para la creación de interfaces de usuarios web (Pacholczyk, Dominic, 2014):

- **Actividades Recientes:** los usuarios quieren estar al día con lo que sucede alrededor de ellos y recibir información sobre las actividades recientes. Proveen a sus usuarios con una visión general de lo que hacen sus amigos o las personas que siguen.
- **Lista de amigos:** los usuarios quieren estar al tanto de quienes son sus amigos en el sitio. Para lograr esto generalmente se muestra una lista con los amigos. Esta lista es muy conveniente si se necesita saber quién compartió qué.
- **Vota para promover:** los usuarios quieren apoyar y compartir el contenido que les parezca relevante.
- **Mensajería directa:** los usuarios quieren enviar mensajes privados a través de la aplicación. Se debe permitir a los usuarios interactuar con otros usuarios a través de mensajes privados, como una parte integral de su experiencia de uso en la aplicación.
- **Like:** los usuarios quieren valorar el contenido de una manera simple, sin tener que preocuparse por el grado en que les gusta algo. Para esto se usan controles que permitan hacer elecciones de forma binaria (me gusta – no me gusta).
- **Encuentra e invita a amigos:** los usuarios quieren experimentar la aplicación junto con sus amigos. Proveer a los usuarios de un mecanismo para conectar y compartir la aplicación con sus amigos.
- **Estadísticas y Dashboards:** los usuarios quieren estar al tanto de sus actividades y estatus. Presentar información y estadísticas que resuman la actividad y el estatus en términos de números.
- **Grupos, amigos y contenidos:** los usuarios quieren organizar el contenido de acuerdo a sus propias recopilaciones.

La web interactiva, a pesar de brindar la oportunidad de estudiar la trazabilidad de los datos de comportamiento de los usuarios, supuso un desafío para el procesamiento de grandes volúmenes de información. Se impuso la necesidad de que las computadoras pudiesen “entender” el lenguaje de la gente, añadiéndole metadatos semánticos y ontológicos a la información. Este cambio tecnológico abrió el camino para el uso de la inteligencia artificial en una web denominada 3.0 o “Web de los Datos” (Cantillo, 2010).

Según Berners-lee (2009), la evolución de la web encaminada a la MSTI muestra como:

- La web ha favorecido el proceso científico técnico a través de tecnologías que proveen a profesores y estudiantes de un marco de trabajo cualitativamente superior a las formas tradicionales de investigación.
- Las tecnologías siguen su avance y se proyectan a investigaciones soportadas en la computación en la nube y el uso de tecnologías móviles (m-learning²).
- El uso de la tecnología web en la investigación científica posibilita la accesibilidad, disponibilidad, actualización y colaboración de información.

Las herramientas y recursos en la web son sin dudas un motor impulsor universal de la actividad científica. La web favorece la investigación científica en varias de sus etapas: búsqueda de información, uso de herramientas multipropósitos en línea y la divulgación de resultados, este último como uno de los propósitos esenciales de la ciencia (John ZIMAN, 2002) que es ante todo, un sistema de comunicación y la web es una oportunidad de fortalecer y optimizar dicho sistema.

1.3 Soluciones informáticas de gestión de información científico técnica

Los entornos basados en tecnologías de la Web 2.0 ofrecen un espacio ideal para la construcción compartida del conocimiento y se han convertido en una herramienta imprescindible para compartir recursos y materiales, para realizar colaborativamente tareas y experiencias de investigación, independientemente del espacio y tiempo (El Caparazón desde 2007, 2007).

La web favorece la investigación científica en varias de sus etapas: búsqueda de información, uso de herramientas multipropósitos en línea y la divulgación de resultados, este último como uno de los propósitos esenciales de la ciencia que es ante todo, un sistema de comunicación y la web es una oportunidad de fortalecer y optimizar dicho sistema (Ziman, 2002).

Las herramientas y recursos en la web y en especial las redes sociales académicas son sin dudas un motor impulsor universal de la actividad científica. A continuación se describen algunas de las soluciones para la MSTI, teniéndose como criterio de selección la visibilidad internacional por el ranking de las universidades.

² M-learning se denomina aprendizaje electrónico móvil.

1.3.1 Portal Universidad de la Habana

El portal Universidad de la Habana: es una solución integrada de la MSTI que ayuda a los estudiantes universitarios en formación a aumentar su rendimiento a través de su conocimiento y comunicación eficiente, comprometida y motivada. Ofrece las vías necesarias para la MSTI. El sistema proporciona el marco ideal para:

- Potenciar los nexos de la investigación y el postgrado, fortaleciendo su papel en la atención a las demandas y necesidades sociales, las relaciones internacionales.
- Preservar el patrimonio, mejorar la infraestructura existente e integrar plenamente las tecnologías de la información y las comunicaciones a la vida universitaria.

El portal cuenta con servicio de integración e implementa una Arquitectura Orientada a Servicios (SOA³) el cual provee un modelo para una arquitectura adaptable, flexible y abierta.

El portal incluye además grupos y líneas de investigación visibles, proyectos de investigación, perfil de investigadores, servicio de noticias tecnológicas, acuerdos y convenios entre universidades, relaciones con empresas y parques tecnológicos, casas editoriales (publicaciones propias), se muestran necesidades de investigación y observatorio tecnológico por mencionar algunas de las características que presenta (UH, 2012).

1.3.2 Portal Instituto Superior Politécnico José Antonio Echeverría

El portal Instituto Superior Politécnico José Antonio Echeverría: es una plataforma o portal de información que facilita la gestión de información científico técnica, proporciona una amplia gama de características para la gestión de la organización e información. Presenta además observatorios tecnológicos: un espacio de colaboración para el profesorado basado en la observación de la tecnología informática, tanto a nivel de hardware como de software.

El portal implementa por otro lado grupos y líneas de investigación visibles, proyectos de investigación, perfil de investigadores, servicio de noticias tecnológicas, relaciones con empresas y parques tecnológicos, casas editoriales (publicaciones propias), y se muestran necesidades de investigación por mencionar algunas de las características que predominan (Cujae, 1998).

³ SOA es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

1.3.3 Portal Universidad de Buenos Aires

El portal Universidad de Buenos Aires ofrece servicios de automatización de los procesos de MSTI, permitiendo automatizar las actividades y su cumplimiento por el personal de la organización. Muestra un sistema de autenticación utilizando redes sociales e integración con diversas plataformas similares, implementando SOA.

Cuenta con proyectos de investigación que responde a un total de 78 carreras de grado y 116 títulos derivados de aquellas (que constituyen un tercer nivel en la enseñanza), además de varias carreras de posgrado que son de cuarto nivel (especializaciones y máster), de quinto nivel (doctorados) y de sexto nivel (posdoctorados). Cerca del 30% de la investigación científica del país se realiza en esta institución. Cuatro de los cinco ganadores argentinos del Premio Nobel han sido estudiantes y profesores de esta Universidad y por ende miembros de este portal (UBA, 1995).

1.3.4 Portal Universidad de Complutense de Madrid

Es un sistema de gestión integral estándar que permite el control de los procesos que se llevan a cabo en la universidad con el objetivo de formar profesionales de la salud, con criterios de excelencia y robustos conocimientos científico técnicos, tratando de promulgar ser líderes en la enseñanza y la investigación en el ámbito de las ciencias de la salud.

El portal facilita el uso de la parametrización para adaptarse a las exigencias de cada entidad en particular, garantizando que sus reportes tengan la forma y el contenido que el usuario les defina. Como Sistema Integral, todos sus procesos trabajan en estrecha relación, tributando automáticamente al proceso central de la universidad. La plataforma y/o portal cuenta con un calendario de eventos guiando al usuario a encontrar eventos rápidamente que se ajustan a su programación (UCM, 2006).

1.3.5 Portal Universidad Nacional Autónoma de México

Es un sistema integrado de gestión de información universitaria, diseñado para ser utilizado por el sector educacional de México, con el objetivo de impartir educación superior para formar profesionales, investigadores, profesores universitarios y técnicos útiles a la sociedad; organizar y realizar investigaciones, principalmente acerca de las condiciones y los problemas nacionales, y extender con la

mayor amplitud posible los beneficios de la cultura. Sus funciones sustantivas son la docencia, la investigación y la difusión de la cultura.

Es un producto mexicano, donde están incorporados los principios de programación segura vigentes en el país, con la garantía de continuar desarrollándose nuevas opciones y brindar una respuesta rápida a las nuevas tecnologías y mecanismos que se establezcan por los organismos rectores de las actividades científicas y técnicas correspondientes. Apoyando el portal a la universidad, en ser la de mayor reconocimiento académico en América Latina y en Iberoamérica (UNAM, 2009)

1.3.6 Resultados del estudio de las soluciones informáticas de gestión de información científico técnica

Las soluciones informáticas orientadas a la MSTI que han sido abordadas en esta investigación, responden a determinadas necesidades del entorno educacional para el cual fueron creadas, razón para plantear que no son lo suficientemente flexibles como para adaptarse a las necesidades planteadas en la situación problemática de esta investigación. Por otro lado también deben tenerse en cuenta que presentan aspectos desfavorables que imposibilitan su uso en la Universidad de las Ciencias Informáticas.

El portal Universidad de la Habana y el portal Instituto Superior Politécnico José Antonio Echeverría son una de las mejores soluciones empresariales para la MSTI, a lo que cabe sumar el soporte que brinda a sus usuarios. Sin embargo esta solución es puramente informativa, de mínimas interacciones, no conecta a sus usuarios hacia intereses comunes, procesan los mismos datos, la información compartida no se encuentra correctamente clasificada. Los investigadores mayormente comparten y consumen la información desde sus ordenadores personales. Presentan dispersión de los datos y fuentes, y no cuenta con una apariencia adaptable al dispositivo.

En cuanto a las soluciones internacionales, provee gran cantidad de facilidades para una gestión personalizada de la información científico técnica, sin embargo estas soluciones son privativas y su adquisición y puesta en marcha requiere el desembolso de altas sumas de dinero, esta situación obviamente no está acorde con las políticas de independencia tecnológica adoptadas en el país y en la Universidad de las Ciencias Informáticas. Además sus grupos, proyectos y resultados de investigación se desarrollan de forma aislada dificultando la socialización del conocimiento.

De manera general, no resulta favorable la actualización de algunas de las soluciones antes mencionadas; debido a que las soluciones ya implementadas contienen funcionalidades que no son de interés para la propuesta de solución atentando contra la propia concepción.

Por lo anterior expuesto se concluye que:

- Las soluciones informáticas existentes no suplen las necesidades específicas del problema planteado.
- La mayoría de estas soluciones emplean tecnologías privativas.

Estas soluciones no responden a enfoques novedosos de los patrones de diseño para la creación de interfaces de usuarios web.

1.4 Estudio de metodologías, tecnologías y herramientas

En la actualidad, el uso intensivo de todo tipo de soluciones informáticas en múltiples sectores de la sociedad y sobre todo en los procesos productivos, ha provocado un notable crecimiento en la confiabilidad y calidad que estos deben poseer. Para desarrollar un producto que satisfaga estos indicadores se debe tener en cuenta una serie de tendencias, tecnologías, lenguajes, metodologías y herramientas indispensables para garantizar la entrega de un software competitivo y ajustado a las necesidades de los clientes.

A continuación se realiza un estudio de las metodologías con el propósito de determinar las más convenientes para la solución propuesta. Además se describen las tecnologías y herramientas seleccionadas para el desarrollo de la solución.

1.4.1 Proceso de desarrollo de software

Un proceso para el desarrollo de software, también denominado ciclo de vida del desarrollo de software es una estructura aplicada al desarrollo de un producto de software. Hay varios modelos a seguir para el establecimiento de un proceso para el desarrollo de software, cada uno de los cuales describe un enfoque diferente para diferentes actividades que tienen lugar durante el proceso. Algunos autores consideran un modelo de ciclo de vida un término más general que un determinado proceso para el desarrollo de software.

Para lograr la consecución exitosa de un proyecto de software, existen un conjunto de modelos que permiten perfilar el proceso de desarrollo, los cuales deben ser analizados cuidadosamente. Usualmente

la selección de uno de estos modelos o de la unión de varios responde a las necesidades y condiciones específicas del proyecto.

Estos modelos o metodologías se pueden clasificar en cinco grandes grupos (Classroom now, 2014).

- **Modelo de cascada:** es un proceso secuencial de desarrollo en el que los pasos de desarrollo son vistos hacia abajo (como en una cascada de agua).
- **Modelo iterativo incremental:** provee una estrategia para controlar la complejidad y los riesgos, desarrollando una parte del producto de software reservando el resto de los aspectos para el futuro.
- **Modelo espiral:** en cierto modo, es como una versión más grande e intensa del modelo incremental.
- **Modelos ágiles:** son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.
- **Modelo de prototipos:** en el método de creación de prototipos, el software es desarrollado en forma similar a una cebolla. Cuando el proyecto se inicia, el equipo se centra en la construcción de un prototipo con todos o la mayoría de las características y construye todo un programa para que el cliente lo utilice.

Las metodologías ágiles cumplen con el manifiesto ágil, una serie de principios que se agrupan en 4 valores (Raya, 2014):

- Los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, frente a la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

Para guiar el proceso de desarrollo de la solución planteada se empleará la metodología SXP adaptada al ciclo de vida definido para la actividad productiva, es la combinación de Scrum y XP, Scrum se enfoca en las prácticas de organización y gestión, mientras XP se centra en las prácticas de programación. Esa es la razón de que funcionen tan bien juntas: tratan de áreas diferentes y se complementan entre ellas. Para ello se ha comprobado que una de las principales ventajas de este modelo son: mejora la calidad del código, la concentración del equipo y la distribución de conocimiento entre el equipo.

Esta metodología ágil mejora la satisfacción del cliente dado que lo involucra y compromete a lo largo del proyecto, mejora la motivación e implicación del equipo de desarrollo, ahorra tanto tiempo como costes apoyando en mayor velocidad y eficiencia, elimina aquellas características innecesarias del producto. De esta forma mejorar la calidad del producto alertando rápidamente tanto de errores como de problemas

Muchos asocian las metodologías ágiles con falta de documentación o control sobre el proyecto, pero es falso. Lo que se desea es minimizar el impacto de las tareas que no son imprescindibles para conseguir el objetivo del proyecto. Se pretende aumentar la eficiencia de las personas involucradas y como resultado minimizar el coste.

Las metodologías de desarrollo, por tanto, no escapan de cualquier intento de comparación entre ágiles y tradicionales. Sin embargo, existe un consenso de que la elección depende más del tipo de proyecto en que se apliquen (Raya, 2014).

Entre las principales metodologías ágiles se puede encontrar (Pressman, 2011):

- **Scrum:** es una metodología de gestión de proyectos ágil, Scrum se enfoca en las prácticas de organización y gestión. Se trata de un esqueleto que incluye un pequeño conjunto de prácticas y roles predefinidos.
- **Extreme Programming (XP):** es una metodología de ingeniería de software ágil, XP se centra más en las prácticas de programación. Se trata de un conjunto de buenas prácticas de los cuales algunos son llevados a niveles “extremos”.
- **Agile Unified Process:** en español Proceso Unificado Ágil de Scott Ambler o (AUP) en inglés, es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.

A modo de conclusión, SXP pueden combinarse de forma fructífera, el uso de esta mejora considerablemente la eficiencia del trabajo al estar fuertemente basado en la experiencia y la fiabilidad de las personas involucradas en el proyecto.

1.4.2 Lenguaje de Modelado

Usualmente para desarrollar una solución informática con calidad y competitiva, el uso de lenguajes de modelado es prácticamente indispensable. Esta necesidad responde a que el uso de estos lenguajes permite que el desarrollo de software se formalice a través de estándares unificados. Se entiende por lenguaje de modelado cualquier lenguaje artificial que puede ser utilizado para expresar la información, el conocimiento o sistemas en una estructura que está definida por un conjunto coherente de reglas. Las reglas se utilizan para la interpretación del significado de los componentes en la estructura. Los lenguajes de modelado utilizan modelos visuales y diagramas que realizan esa representación de manera precisa, entendible y económica, lo que facilita su uso en las herramientas CASE⁴ convencionales (J, 2010).

1.4.3 Herramientas de Ingeniería de Software Asistida por Computación

1.4.3.1 Visual Paradigm

La herramienta Visual Paradigm propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Mora, Roberto Canales, 2004). Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas.

Entre sus principales características se pueden mencionar (Mora, Roberto Canales, 2004):

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Disponibilidad de integrarse en los principales IDEs.
- Generación de bases de datos.

Por las características antes mencionadas, así como las numerosas posibilidades que se brindan al usuario y a su condición de software libre, se selecciona Visual Paradigm como herramienta CASE.

1.4.4 Lenguajes de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina o para expresar algoritmos con precisión. A

⁴ CASE (Computer Aided Software Engineering, ingeniería asistida por computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

continuación se describen algunos de los lenguajes de programación más populares y acordes a las necesidades de la solución informática que se desea desarrollar.

1.4.4.1 CSS3

CSS3⁵ define la presentación de los documentos HTML. Está soportado por todos los navegadores en la actualidad. Algunos de sus beneficios son: control de la presentación de muchos documentos desde una única hoja de estilo; aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, entre otros).

Sus principales ventajas son:

- Separación del contenido y presentación.
- Dar estilo.
- Reutilización.
- Unificación del diseño de las páginas del sitio.

Accesibilidad y estructuración.

1.4.4.2 HTML5

HTML5⁶, es usado para estructurar y presentar el contenido para la web. Está relacionado también con la entrada en decadencia del estándar HTML4, que se combinaba con otros lenguajes para producir los sitios existentes hasta la fecha. Con HTML5, entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación.

Es un sistema para formatear las páginas y hacer ajustes a su aspecto. A través de este, los navegadores como Mozilla Firefox, Google Chrome, Internet Explorer y otros pueden saber cómo mostrar una página web determinada, ubicar los elementos, texto e imágenes. El nivel de sofisticación del código que se puede construir con HTML5 es la principal diferencia con su predecesor. Se puede reducir la dependencia de complementos y amplía el horizonte del desarrollo de aplicaciones que pueden ser usadas en múltiples dispositivos. Muestra la funcionalidad de «arrastrar y soltar» y nuevas etiquetas (Pemberton, 2010).

1.4.4.3 JavaScript

JavaScript es un lenguaje de programación del lado del cliente, es interpretado y por tanto no se necesita compilar los programas para ejecutarlo. Se utiliza principalmente para páginas web dinámicas. Gran parte

⁵ CSS son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada, es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación

⁶ HTML son siglas de HyperText Markup Language (lenguaje de marcas de hipertexto).

de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas, entre otros. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado (jquery, 2010). Como lenguaje de programación se seleccionó JavaScript, por el lado servidor se utilizó NodeJS y del cliente HTML5 y CSS3.

1.4.5 Servidor Web

Un servidor web es la tecnología que tiene implícito programas informáticos que procesan aplicaciones realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. Un servidor web es un programa que sirve datos en forma de Páginas Web, hipertextos o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo Http. Con esto, un servidor Web se mantiene a la espera de peticiones HTTP⁷, que son ejecutadas por un cliente HTTP; lo que se suele conocer como un navegador web (Fisteus, 2008).

1.4.5.1 NodeJs

Node es un intérprete Javascript del lado del servidor que cambia la noción de cómo debería trabajar un servidor. Su meta es permitir a un programador construir aplicaciones altamente escalables y escribir código que maneje decenas de miles de conexiones simultáneas en una sólo una máquina física.

Se seleccionó NodeJs del lado del servidor. La meta número uno declarada de Node es "proporcionar una manera fácil para construir programas de red escalables". ¿Cuál es el problema con los programas de servidor actuales? En lenguajes como Java y PHP, cada conexión genera un nuevo hilo que potencialmente viene acompañado de 2 MB de memoria. Un sistema operativo de 8 Gb de RAM, ofrece un número máximo teórico de conexiones concurrentes de cerca de 4.000 usuarios. Si se desea que la aplicación soporte más usuarios, a medida que crezca su base de clientes se requerirá incorporar otros servidores. La adquisición de tales equipamientos deberá ser añadida a los costos del servidor de negocio, servicios de tráfico y otros costos laborales. Por todas estas razones, el cuello de botella en toda la

⁷ HTTP en sus siglas en inglés **HyperText Transfer Protocol** es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

arquitectura de aplicación Web (incluyendo el rendimiento del tráfico, la velocidad de procesador y la velocidad de memoria) es el número máximo de conexiones concurrentes que podía manejar un servidor.

Node resuelve este problema cambiando la forma en que se realiza una conexión con el servidor. En lugar de generar un nuevo hilo de OS para cada conexión (y de asignarle la memoria acompañante), cada conexión dispara una ejecución de evento dentro del proceso del motor de Node. También afirma que nunca se quedará en punto muerto, porque no se permiten bloqueos y porque no se bloquea directamente para llamadas E/S. Node afirma que un servidor que lo ejecute puede soportar decenas de miles de conexiones concurrentes (Michael Abernethy, Freelancer, 2011).

1.4.6 Bibliotecas y Marco de trabajo para el desarrollo

Un marco de trabajo simplifica considerablemente el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Proporciona una estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener, además de facilitar la programación de aplicaciones pues encapsula operaciones complejas en instrucciones sencillas. Por otro lado las bibliotecas persiguen un objetivo similar, con la diferencia de que están orientadas a un uso más pasivo y usualmente no generan código fuente en la aplicación sino que el desarrollador hace uso de las funcionalidades que esta provee. A continuación se describen algunos de los principales marcos de trabajo de desarrollo para PHP y bibliotecas Javascript.

1.4.6.1 Bootstrap

Es una guía de estilos desarrollada por Twitter para simplificar el proceso de creación de diseños web y es compatible con la mayoría de los navegadores web.

Ofrece una serie de plantillas CSS y de ficheros JavaScript, que permiten conseguir interfaces que funcionen de manera adecuada en los navegadores actuales. Facilita la realización de interfaces con diseño adaptable para ser visualizadas de forma correcta en distintos dispositivos y a distintas escalas y resoluciones. Es integrable con bibliotecas como jQuery. Permite un diseño fijo basado en una matriz de 12 columnas (Guaita, Alvaro Martínez, 2012).

1.4.6.2 AngularJS

Es un marco de trabajo de JavaScript de código abierto, que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador

con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

AngularJS es un proyecto de código abierto, realizado en Javascript que contiene un conjunto de bibliotecas útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un marco de trabajo para el desarrollo, en este caso sobre el lenguaje Javascript con programación del lado del cliente.

1.4.6.3 jQuery

Es una biblioteca de JavaScript rápida y concisa que simplifica el uso del documento HTML, el manejo de eventos, la animación, y las interacciones Ajax para el desarrollo web rápido (jquery, 2010).

Anunciado en 2006 por su creador, John Resig, jQuery ganó rápidamente popularidad y apoyo al convertirse en una nueva forma de utilizar JavaScript para interactuar con HTML y CSS. Los selectores de jQuery son simples, es una biblioteca familiar y fácil de aprender para los diseñadores y desarrolladores por igual.

Maneja una gran cantidad de estándares para varios navegadores web. En la actualidad esta biblioteca es ampliamente utilizada en disímiles aplicaciones web. Su reputación ha crecido exponencialmente al igual que la comunidad de desarrolladores que comparten sus experiencias así como complementos que mejoran el trabajo con la biblioteca (Valdez, 2013).

1.4.7 Herramientas IDE

1.4.7.1 Netbeans 7.1

Netbeans es un entorno de desarrollo integrado de código abierto para desarrolladores de software. Cuenta con todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java (Oracle, Corporation, 2012).

NetBeans recibe el soporte integrado para lenguajes dinámicos, todo en una herramienta de gran alcance. El editor de PHP de NetBeans ofrece plantillas de código y generación (getters y setters), la refactorización, información sobre herramientas de parámetros, consejos y soluciones rápidas, y la

finalización de código inteligente. Provee de un resaltado de código sintáctico y semántico, documentación emergente, formateo de código y plegado, y el marcado de los sucesos y puntos de salida.

El IDE (Integrated Development Environment, Entorno Integrado de Desarrollo) también ofrece un completo editor de HTML, JavaScript y CSS, con la ventaja de proveer un resaltado de sintaxis completo, completamiento de código inteligente, documentación pop-up, y la comprobación de errores para HTML, CSS y JavaScript, incluyendo HTML5, JavaScript 1.7. El editor reconoce código HTML en los archivos de JavaScript y viceversa. El editor reconoce también HTML y JavaScript en XHTML, PHP y archivos JSP.

Teniendo en cuenta las características anteriormente planteadas, se selecciona el IDE Netbeans para desarrollar la solución propuesta, por todas las facilidades que brinda para el trabajo con proyectos PHP y al soporte a tecnologías de última generación como HTML5 y CSS3.

1.4.8 Gestor de base de datos

Un sistema de gestor de bases de datos (DBMS), algunas veces llamada simplemente un administrador de base de datos, es un programa que permite a uno o más usuarios de computadoras crear y acceder a los datos en una base de datos. El DBMS gestiona las peticiones del usuario (y las solicitudes de otros programas) para que estos sean libres de entender que los datos se encuentran físicamente en los medios de almacenamiento. En el manejo de solicitudes de los usuarios, el DBMS asegura:

- **La integridad de los datos:** es decir, asegurándose de que sigue siendo accesible y es constantemente organizada según lo previsto
- **Seguridad:** asegurándose de que sólo aquellos con privilegios de acceso pueden acceder a los datos (Christiansen, 2012).

1.4.8.1 MongoDB

MongoDB es un sistema de base de datos multiplataforma orientado a documentos, de esquema libre. Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. Los documentos de una misma colección pueden tener esquemas diferentes.

MongoDB es especialmente útil en entornos que requieran escalabilidad. Con sus opciones de replicación son muy sencillas de configurar, se puede conseguir un sistema que escale horizontalmente sin demasiados problemas (MongoDB, 2013).

MongoDB es seleccionado como gestor de base de datos debido a que funcionan con muy poco rendimiento y alta frecuencia de lecturas y escrituras, además posibilita los cambios en el esquema de datos frecuentes y datos a escala web.

1.5 Conclusiones

- El estudio de las principales terminologías, métodos, técnicas, definiciones y modelos conceptuales asociados a la gestión de la información científico-técnica y el trabajo colaborativo mediante aplicaciones web, permitió establecer el marco conceptual en el contexto de la problemática formulada, tomando además como referencia los principales estándares y tendencias internacionales recientes.
- El estudio de enfoques y tendencias de diseño en aplicaciones web para la gestión de la información científico-técnica permitió identificar mecanismos de recuperación de los datos, patrones y estilos que estimulan la cooperación entre los usuarios.
- El análisis de las principales fuentes bibliográficas permitió identificar los aspectos que caracterizan la gestión de la información científico-técnica, entre los que se señalan la adaptabilidad y accesibilidad de la información.
- Las soluciones observadas que gestionan información científico-técnica poseen un acentuado grado de localización, se definen por sus contextos de uso y se orientan por las necesidades y experiencias de sus usuarios, lo cual limita su despliegue en la UCI o haría costosa su adaptación. A pesar de su restringida utilidad para resolver el problema de la investigación, las aplicaciones estudiadas aportaron características esenciales para una solución propia al caso de la universidad.
- El análisis de las tecnologías para implementar la solución requerida hizo prevalecer el enfoque web, accesibilidad y adaptabilidad de la aplicación a los entornos y contextos de interacción de los usuarios, así como planteo el uso de herramientas libres acordes a las políticas de soberanía tecnológica de la UCI y el país.
- El proceso de desarrollo de la solución debe ser guiado por la metodología ágil SXP, empleando UML para estandarizar la descripción del proyecto y Visual Paradigm como herramienta de

Ingeniería de Software. Como lenguaje de programación se selecciona JavaScript del lado del servidor y del cliente, en este último se optó por el uso de HTML y CSS incluyendo diferentes framework como Bootstrap para las vistas y AngularJS y JQuery para el manejo del DOM[1]. Además se ha recomendado Netbeans en su versión 7.1 como servidor web y será el IDE empleado en el desarrollo de la solución. Como gestor de base de datos se utilizará MongoDB.

CAPÍTULO 2: Análisis, diseño e implementación

Introducción

En este capítulo se realiza la propuesta de solución al problema planteado con anterioridad, exponiendo los prototipos de interfaces, análisis y diseño de clases y los artefactos generados en la implementación.

2.1 Descripción de la propuesta de solución

Teniendo en cuenta los objetivos que persigue el presente trabajo de diploma y en aras de utilizar la información recopilada en el capítulo anterior, se propone como solución desarrollar una plataforma web que potencie el trabajo colaborativo en la red de investigadores de la UCI.

La solución propuesta debe ser capaz de brindar información relevante, actualizada y de interés del investigador. El sistema conforma la página de inicio personalizada con la información relacionada a los temas de interés del usuario. Esta característica del sistema pretende aportarle utilidad a la vez que de influir y motivar a los investigadores. Una de las metas de la plataforma es que sus usuarios participen de manera dinámica y proactiva en acciones que generalmente requieren un esfuerzo de la voluntad. Persiguiendo este objetivo se incluyeron características como:

- A través de la integración con el sitio de Caribbean Mind Forge (CMF), mostrar su ranking y proponer problemas de lógica.
- Brindar información actualizada a través de la integración con el canal de información tecnológica de Vigitec que tiene entre sus objetivos “Coordinar y dirigir la red de observatorios de la UCI, en función de las diferentes líneas de investigación de la universidad” además de “Anticipar de manera oportuna los cambios relevantes en el entorno científico, tecnológico y competitivo”. La plataforma filtra las noticias provenientes de Vigitec para mostrarlas a los investigadores atendiendo a los grupos, líneas y proyectos con los que se relaciona, posibilitando así que siempre exista información que resulte útil y atractiva.
- Posibilidad de realizar comentarios y dar seguimiento a temas a través de microblogging.
- Compartir información con otros usuarios de la plataforma así como realizar votación binaria. Esta característica es un patrón de diseño de interfaz de usuario que potencia la socialización de la información y es utilizada por la plataforma como criterio para jerarquizar la información.

- Atendiendo a las tendencias de uso de internet a nivel mundial así como el las proyecciones futuras de la Universidad de las Ciencias Informáticas, la plataforma contará con una apariencia adaptable desde el dispositivo desde el que se acceda, brindando una mejor experiencia de usuario para dispositivos móviles así como de pantallas tanto con relación de aspecto 16:9 como con pantallas con alta densidad de píxeles.

2.2 Modelo de dominio

El entorno en el que se circunscribe la solución que se desea desarrollar presenta flujos de información difusos que tienen múltiples orígenes, lo que conlleva al uso de un modelo de dominio para modelar el negocio a informatizar. El objetivo principal de este modelo es comprender y describir las clases más importantes dentro del contexto del sistema a través de diagramas UML.

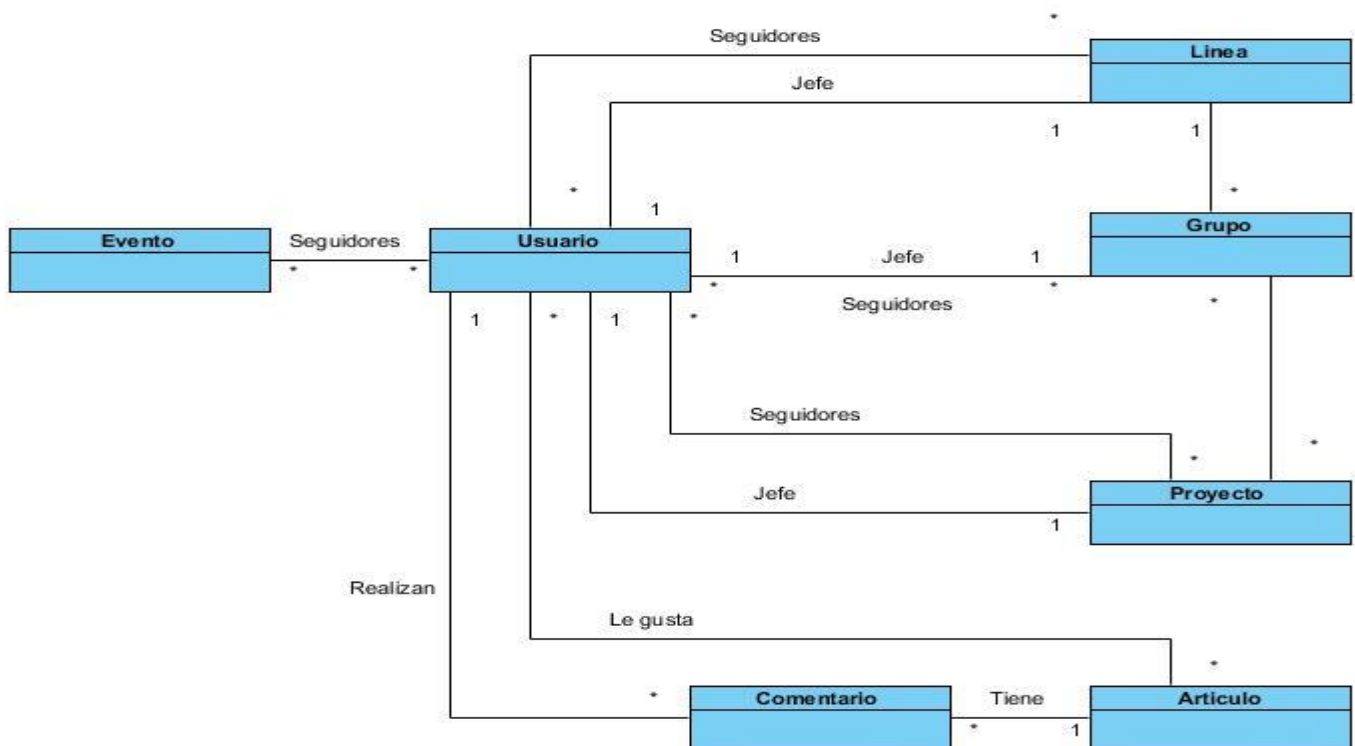


Figura 1: Diagrama modelo de dominio de la solución.

2.3 Requisitos de la plataforma

Un requisito es una circunstancia o condición necesaria. Un requisito en términos de software representa las condiciones o capacidades que debe tener un sistema, para cumplir un contrato, estándar o documento impuesto formalmente. Su propósito es formar un entendimiento común entre el usuario y el grupo de desarrollo del software respecto a los requisitos. Se dividen en dos categorías: requisitos funcionales y no funcionales (Caro, 2012).

La definición y especificación de los requisitos se basó en los resultados del estudio de herramientas homólogas y la aplicación de técnicas de la ingeniería de requisitos sobre una muestra confiable y representativa de usuarios reales o potenciales de la solución. El conjunto que reúne tales propiedades asciende a una población con tamaño determinado por casi la totalidad de los estudiantes, profesores y especialistas de la producción de la universidad. La selección de este grupo muestral descarta criterios estadísticos para basarse en el estudio “Número de Participantes en Pruebas con Usuarios” de (Nielsen, 2012), en que se proponen la selección de grupos entre 5 y 15 usuarios para la aplicación de las técnicas de captura de requisitos. Nielsen pudo demostrar con 83 casos de estudio que una cifra mayor a 15 usuarios en la muestra obtendría similares resultados cuando se aplican técnicas de captura de requisitos, y que por tanto, en ese rango se reportaban la mayoría de los hallazgos. Partiendo de este aporte se decide que los grupos de prueba de la muestra tengan un tamaño de 15 usuarios (ver anexo 1).

Sin embargo, en el estudio el valor fundamental de la muestra no descansa en su tamaño, sino en su representatividad. Para la eventual generalización de los resultados, la muestra deberá representar la mayoría de las propiedades de la población. Teniendo en cuenta su caracterización (ver Anexo 2) se decide que cada grupo de prueba se acerque al mismo cuadro de representación por estratos (ver cuadro en anexos). En esta selección muestral resulta significativo el enfoque de género, el balance etnográfico territorial y racial, el estatus, el grupo etario-generacional y en el caso de los estudiantes, la evolución cognitiva en la carrera. El método de selección recurrido fue el aleatorio simple.

A continuación se describen los requisitos definidos para la realización e implementación de la plataforma. Los gestionar no son requisitos funcionales, son una forma de agrupar los mismos.

2.3.1 Requisitos funcionales

1. Gestionar proyecto de investigación.

- 1.1. Adicionar un nuevo proyecto de investigación.
- 1.2. Modificar un proyecto de investigación existente.
- 1.3. Mostrar un proyecto de investigación existente.
- 1.4. Eliminar un proyecto de investigación existente.
2. Gestionar grupo de investigación.
 - 2.1. Adicionar un nuevo grupo de investigación.
 - 2.2. Modificar un grupo de investigación existente.
 - 2.3. Mostrar un grupo de investigación existente.
 - 2.4. Eliminar un grupo de investigación existente.
3. Gestionar línea de investigación.
 - 3.1. Adicionar una nueva línea de investigación.
 - 3.2. Modificar una línea de investigación existente.
 - 3.3. Mostrar una línea de investigación existente.
 - 3.4. Eliminar una línea de investigación existente.
4. Gestionar perfiles de investigación.
 - 4.1. Adicionar un nuevo perfil de investigación.
 - 4.2. Modificar un perfil de investigación existente.
 - 4.3. Mostrar un perfil de investigación existente.
 - 4.4. Eliminar un perfil de investigación existente.
5. Gestionar evento de investigación.
 - 5.1. Adicionar un nuevo evento de investigación.
 - 5.2. Modificar un evento de investigación existente.
 - 5.3. Mostrar un evento de investigación existente.
 - 5.4. Eliminar un evento de investigación existente.
6. Autenticar usuario
7. Mostrar agenda de eventos recientes.
8. Notificar seguimiento de evento por correo electrónico.
9. Generar lista de recomendaciones de artículos según el perfil del usuario.
10. Integrar servicios externos.
11. Configurar los servicios de consumo de artículos.

2.3.2 Requisitos no funcionales

Especifica criterios que pueden usarse para calificar el trabajo de un sistema en lugar de sus comportamientos específicos, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar. Describen aspectos visibles del sistema para el usuario y no influyen de forma directa en el comportamiento funcional del sistema. Son propiedades o cualidades que el producto debe cumplir (Caro, 2012).

Software

Sistema Operativo en los servidores:

1. Multiplataforma siempre que soporte el lenguaje PHP.

Sistema operativo en los clientes:

1. Sistemas UNIX o Windows, así como Google Android, Apple iOS y Windows Phone.
2. Navegadores a partir de las versiones: Google Chrome 30, Mozilla Firefox 29 o Zafari 5.1 y Opera a partir del 24.

Hardware

1. Procesador Core i3 con 2 GB de RAM como mínimo.

Seguridad

1. Permitir la autenticación en el sistema a través de una cuenta de usuario única por autenticación LDAP (Lightweight Directory Access Protocol).
2. Garantizar que el acceso por las cuentas de administración al servidor se realice con contraseñas cifradas por la técnica de md5 y protocolo HTTPS.

Accesibilidad

1. Soportar la accesibilidad desde móviles y tabletas a través de un diseño adaptable.

Apariencia o Interfaz

1. Utilizar colores neutrales para no incitar cansancio visual ni entretener la atención de los usuarios.
2. Utilización de iconografía y elementos de la estructura básica de la interfaz en formato SVG para evitar su distorsión al cambiar de resoluciones de pantalla o dispositivos.
3. Diseño Web Adaptable desde el dispositivo desde el que se acceda.

4. Permitir aumentar la tipografía para proporcionar la visualización de las interfaces.
5. Garantizar la visualización del logotipo en el mismo sitio de la interfaz y durante toda la navegación.
6. Nombrar los contenidos de las etiquetas de forma explícita y/o intuitiva.

Rendimiento

1. Admitir el paso concurrente de 100 usuarios.
2. Mostrar al usuario una advertencia de afectación del rendimiento cuando el sistema no admita más concurrencia (más de 100 usuarios conectados).

Usabilidad

1. Defender la disponibilidad de la interfaz de usuario 24 horas al día.

2.4 . Historias de usuario

Las historias de usuario (HU) son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Permiten responder rápidamente a los requisitos cambiantes. Además constituyen una manera simple para describir una tarea concisa aportando valor al usuario o al negocio. Obtienen ser creadas durante las conversaciones con los usuarios interesados sobre nuevas funcionalidades o mejoras del proyecto (Q-Vision Technologies, 2014).

2.4.1 Estimación de tiempo por Historias de Usuario

La programación extrema basa sus procesos de planificación en estimaciones temporales de las historias de usuario, las cuáles deben ser realizadas por los desarrolladores durante las diversas reuniones de planificación (Juan Quijano, 2012).

El objetivo principal es estimar cuánto tiempo llevará implementar las historias; todas las estimaciones que maneja la programación extrema se cuantifican en semana de desarrollo ideal, lo cual no es más que la cantidad de trabajo que se puede realizar durante una semana sin distracciones y suponiendo que el desarrollador sabe exactamente lo que hay que hacer (Juan Quijano, 2012).

Es de esperar que una historia sea lo suficientemente pequeña como para que el equipo la desarrolle durante una entrega, es decir, de una a tres semanas, más de tres semanas implica que se debe señalar

al cliente que debe dividir una historia de usuario y menos de una semana implica que la historia es demasiado sencilla y será necesario decirle al cliente que hay que unir dos o más de ellas para su mejor interpretación.

Historia de Usuario	Puntos de estimación
Gestionar proyecto de investigación.	2
Gestionar grupo de investigación.	2
Gestionar línea de investigación.	2
Gestionar perfiles de investigación.	2
Gestionar evento de investigación.	2
Autenticar usuario	3
Mostrar agenda de eventos recientes	2
Integrar servicios externos	2
Notificar seguimiento de evento por correo electrónico	3
Generar recomendaciones de artículos según el perfil del usuario	3
Configurar los servicios de consumo de artículos	3

Tabla 1. Estimación del tiempo de las HU.

2.4.2 Plan de iteraciones

Un plan de iteración está constituido por un conjunto secuencial de actividades y tareas, cada una tiene recursos asignados y pueden depender a su vez de otras tareas. Es un plan detallado, que se realiza una vez por iteración.

El contenido de la iteración puede variar, dependiendo de la posición dentro del ciclo de vida y de la naturaleza del proyecto. El plan de iteración incluye porciones relevantes de todas las disciplinas para cada iteración en particular, la prioridad de implementación se evalúa en base al cliente y el equipo de desarrollo, y el impacto del riesgo en base al juicio de experto. Los planes de iteración por lo general muestran un planeamiento detallado de quien va a realizar una tarea/actividad de acuerdo en conformidad a que criterios de evaluación. La implementación de los requisitos del sistema será en dos iteraciones, las cuales se describen a continuación:

Iteración No 1

La iteración tiene como objetivo implementar las historias de usuario HU-01, HU-02, HU-03, HU-04, HU-05. Representan una muestra de los requisitos críticos de la herramienta.

Historia de Usuario			
Código	HU-01	Nombre	Gestionar proyecto de investigación.
Usuario	Administrador	Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	1	Puntos estimados	2

Descripción	La funcionalidad permitirá realizar operaciones como adicionar, modificar, mostrar y eliminar los proyectos de investigación existentes, así como asignar grupos de investigación a los proyectos de investigación.
--------------------	---

Tabla 2. Historia de usuario HU-01.

Historia de Usuario			
Código	HU-02	Nombre	Gestionar grupo de investigación.
Usuario	Administrador	Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	1	Puntos estimados	2
Descripción	La funcionalidad permitirá realizar operaciones como adicionar, modificar, mostrar y eliminar los grupos de investigación existentes, así como asignar la línea de investigación al grupo de investigación.		

Tabla 3. Historia de usuario HU-02.

Historia de Usuario			
Código	HU-03	Nombre	Gestionar línea de investigación.

Usuario	Administrador	Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	1	Puntos estimados	2
Descripción	La funcionalidad permitirá realizar operaciones como adicionar, modificar, mostrar y eliminar las líneas de investigación existentes.		

Tabla 4. Historia de usuario HU-03.

Historia de Usuario			
Código	HU-04	Nombre	Gestionar perfiles de investigación.
Usuario	Administrador	Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	1	Puntos estimados	2
Descripción	La funcionalidad permitirá realizar operaciones como adicionar, modificar, mostrar y eliminar los perfiles de investigación existentes.		

Tabla 5. Historia de usuario HU-04.

Historia de Usuario			
Código	HU-05	Nombre	Gestionar eventos de investigación.
Usuario	Administrador	Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	1	Puntos estimados	2
Descripción	La funcionalidad permitirá realizar operaciones como adicionar, modificar, mostrar y eliminar los eventos de investigación existentes.		

Tabla 6. Historia de usuario HU-05.

Iteración No2

La iteración tiene como objetivo implementar las historias de usuario HU-06, HU-07, HU-08, HU-09. HU-10, HU-11. Representan otros requisitos funcionales de la herramienta.

Historia de Usuario

Código	HU-06	Nombre	Autenticar usuario
Usuario	Investigador	Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	1	Puntos estimados	3
Descripción	La funcionalidad permitirá al sistema contar con servicio de autenticación bajo el estándar LDAP.		

Tabla 7. Historia de usuario HU-06.

Historia de Usuario			
Código	HU-07	Nombre	Mostrar agenda de eventos recientes
Usuario		Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	2	Puntos estimados	2

Descripción	La funcionalidad permitirá al sistema mostrar una agenda de los eventos más recientes.
--------------------	--

Tabla 8. Historia de usuario HU-07.

Historia de Usuario			
Código	HU-08	Nombre	Integrar servicios externos
Usuario		Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	2	Puntos estimados	2
Descripción	La funcionalidad permitirá al sistema beneficiarse del servicio de integración con el CMF, Firefoxmanía y ofrecer a los usuarios la integración con el canal de información tecnológica de Vigitec.		

Tabla 9. Historia de usuario HU-08.

Historia de Usuario			
Código	HU-9	Nombre	Notificar seguimiento de evento por correo electrónico
Usuario		Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta

Iteración	2	Puntos estimados	3
Descripción	La funcionalidad permitirá al sistema notificar al usuario el evento a través del correo electrónico.		

Tabla 10. Historia de usuario HU-9.

Historia de Usuario			
Código	HU-10	Nombre	Generar recomendaciones de artículos según el perfil del usuario
Usuario		Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	2	Puntos estimados	3
Descripción	La funcionalidad permitirá generar una lista de recomendaciones de artículos dependiendo de las características de cada perfil de usuario.		

Tabla 11. Historia de usuario HU-10.

Historia de Usuario

Código	HU-11	Nombre	Configurar los servicios de consumo de artículos
Usuario		Tipo de actividad	Nueva
Riesgo	Alto	Prioridad	Alta
Iteración	2	Puntos estimados	3
Descripción	La funcionalidad permitirá al sistema adicionar servicios de consumo de artículos de otros canales de información.		

Tabla 12. Historia de usuario HU-11.

Descritas las HU a implementar, se genera el plan de iteraciones como se muestra en la Tabla 12.

Iteración	Número de Historia	Historia de usuario	Duración (semanas)	
Iteración No1	HU-01	Gestionar proyecto de investigación	2	10
	HU-02	Gestionar grupo de investigación.	2	
	HU-03	Gestionar perfiles de investigación	2	
	HU-04	Gestionar línea de investigación	2	

	HU-05	Gestionar evento de investigación	2	
Iteración No2	HU-06	Autenticar usuario	3	16
	HU-07	Mostrar agenda de eventos recientes	2	
	HU-08	Integrar servicios externos	2	
	HU-09	Notificar seguimiento de evento por correo electrónico	3	
	HU-10	Generar recomendaciones de artículos según el perfil del usuario	3	
	HU-11	Configurar los servicios de consumo de artículos	3	

Tabla 13. Plan de iteraciones.

2.4.3 Plan de entregas

Determinada la duración de cada iteración se presenta el plan de entrega elaborado para la fase de implementación.

Iteración	Iteración No1	Iteración No2
------------------	----------------------	----------------------

Cantidad de HU	5	6
Fecha de entrega	3/03/2015	5/06/2015

Tabla 14. Plan de entrega.

2.4.4 Resumen de tareas

A continuación se muestra un resumen de las tareas de ingeniería a desarrollar por cada una de las Historias de Usuario en las iteraciones.

Iteración	Historia de Usuario	Número de tarea	Tarea de Ingeniería
Iteración No 1	Gestionar Proyecto de investigación	HU-01T01	Adicionar nuevo proyecto
		HU-01T02	Modificar proyecto
		HU-01T03	Mostrar proyecto
		HU-01T04	Eliminar proyecto
	Gestionar grupo de investigación	HU-02T01	Adicionar nuevo grupo
		HU-02T02	Modificar grupo
		HU-02T03	Mostrar grupo

		HU-02T04	Eliminar grupo
	Gestionar línea de investigación	HU-03T01	Adicionar nueva línea
		HU-03T02	Modificar línea
		HU-03T03	Mostrar línea
		HU-03T04	Eliminar línea
	Gestionar perfil de investigación	HU-04T01	Adicionar nuevo perfil
		HU-04T02	Modificar perfil
		HU-04T03	Mostrar perfil
		HU-04T04	Eliminar perfil
	Gestionar evento de investigación	HU-05T01	Adicionar nuevo evento
		HU-05T02	Modificar evento
		HU-05T03	Mostrar evento
		HU-05T04	Eliminar evento

Iteración No 2	Autenticar usuario	HU-06T01	Integrar con servicio de autenticación LDAP/UCI
	Mostrar agenda de eventos recientes	HU-07T01	Mostrar eventos
	Integrar servicios externos	HU-08T01	Consumir servicio web Vigitec
		HU-08T02	Consumir servicio web CMF
	Notificar seguimiento de evento por correo electrónico	HU-9T01	Notificar evento
	Generar recomendaciones de artículos según el perfil del usuario	HU-10T01	Mostrar artículos según el interés del investigador
	Configurar los servicios de consumo de artículos	HU-11T01	Adicionar servicio de consumo

Tabla 15. Resumen de tareas de ingeniería por HU.

2.5 Prototipos de interfaz de usuario

Los prototipos ayudan a identificar, comunicar y probar un producto antes de crearlo. Al tratarse de un espacio de información científico técnica, se ha de conseguir que la atención del usuario que entra en la

web se centre única y exclusivamente en las noticias que se exponen para no causar en este una frustración que puede hacer que no vuelva a visitar la web (Blanco, 2014).

2.5.1 Prototipos de baja fidelidad

Su objetivo es verificar si los usuarios son capaces de realizar sus tareas con la interfaz propuesta. La utilización de esta técnica de prototipado no precisa incorporar avances tecnológicos; solo es necesario que capture la funcionalidad del sistema y que comunique la información y sus interacciones adecuadamente. Es una manera de bajo costo de proporcionar prototipos, significa que los prototipos a utilizar no tienen el aspecto real de la interfaz que se está testeando, aun cuando operan de la misma forma. La idea es conseguir una gran cantidad de información de la interacción entre la interfaz y el usuario mediante la evaluación de este prototipo. Para la confección de los prototipos de interfaz de usuario (IU) de baja fidelidad (BF) se empleó el software Balsamiq Mockups el cual ofrece la misma velocidad y la sensación áspera como dibujar con lápiz, con la ventaja del medio digital: arrastrar y soltar para redimensionar y reorganizar elementos, permitiendo realizar cambios en cualquier momento.

Menú principal

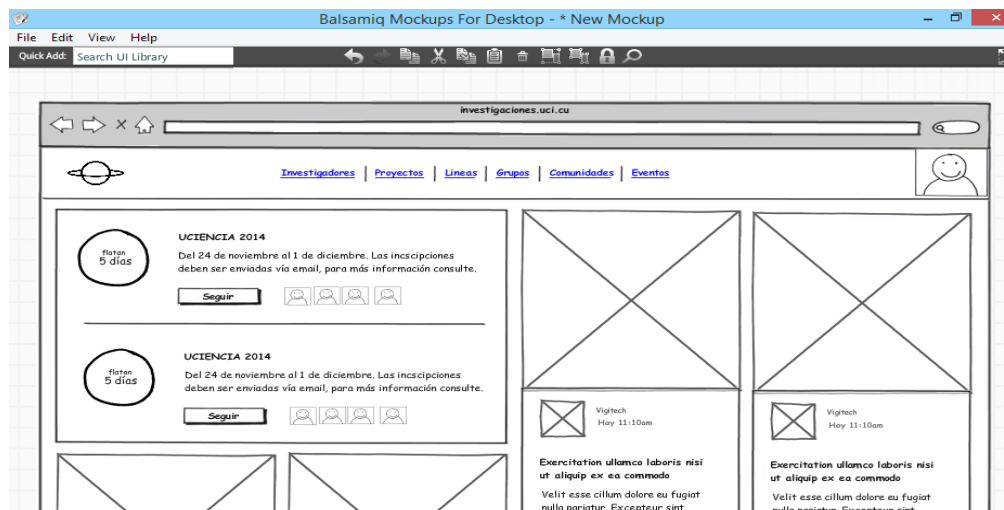


Figura 2: Prototipo de IU de BF Menú principal.

2.5.2 Prototipos de alta fidelidad

En la elaboración de estos prototipos de alta fidelidad (AF) se tuvieron en cuenta los patrones de diseño para la creación de interfaces de usuario descritos con anterioridad en el Capítulo 1. Por ejemplo el patrón

“Actividades Recientes” se tradujo en características como la visualización de eventos recientes en la página principal (Figura 3), o la opción de “Seguir” a otros usuarios.



Figura 3: Prototipo de IU de AF patrón “Actividades Recientes”.

Otros patrones como “Vota para promover”, “Like” (Figura 4), y “Encuentra e invita a amigos” (Figura 5) también se convirtieron en características de la plataforma que motivan e invitan a los usuarios a utilizar el sistema por su utilidad, en la forma en que los acopla hacia intereses comunes.



Figura 4: Prototipo de IU de AF patrones “Vota para Promover” y “Like”.

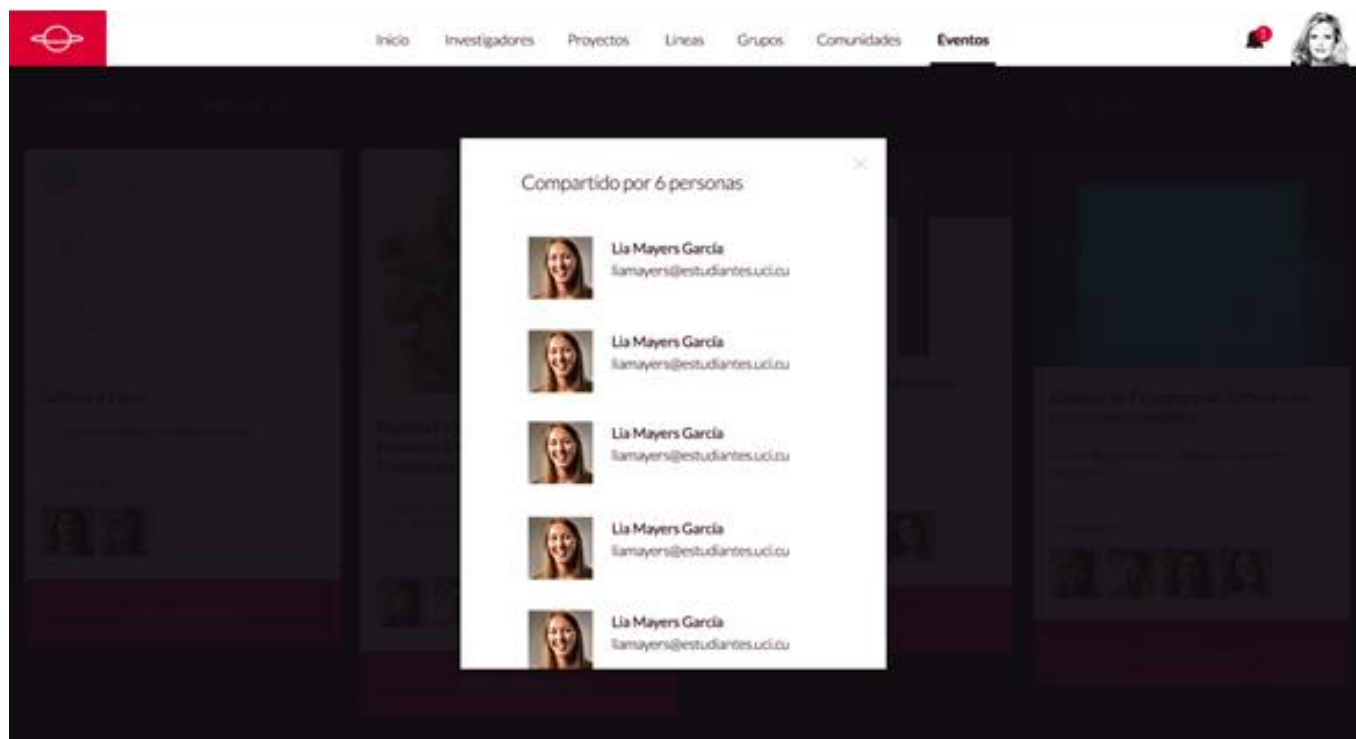


Figura 5: Prototipo de IU de AF patrón “Encuentra e invita a amigos”.

El empleo de estos patrones le añade a la solución una interfaz amena sobre la base de atributos de calidad como la simplicidad y la coherencia visual; además le permite mantener la consistencia en el diseño durante la navegación; establecer jerarquías visuales entre los componentes de mayor y menor significación entre otros aspectos para mejorar el diseño de interacción de la herramienta y posibilitar un óptimo desempeño del usuario.

A continuación se muestran prototipos de alta fidelidad de las principales pantallas claves para la plataforma, así como una breve descripción de los elementos contenidos en la interfaz.

Menú principal

La interfaz muestra el menú principal (Figura 6). Correspondiente a las principales funcionalidades ordenadas por niveles de prioridad según la progresión visual del usuario.

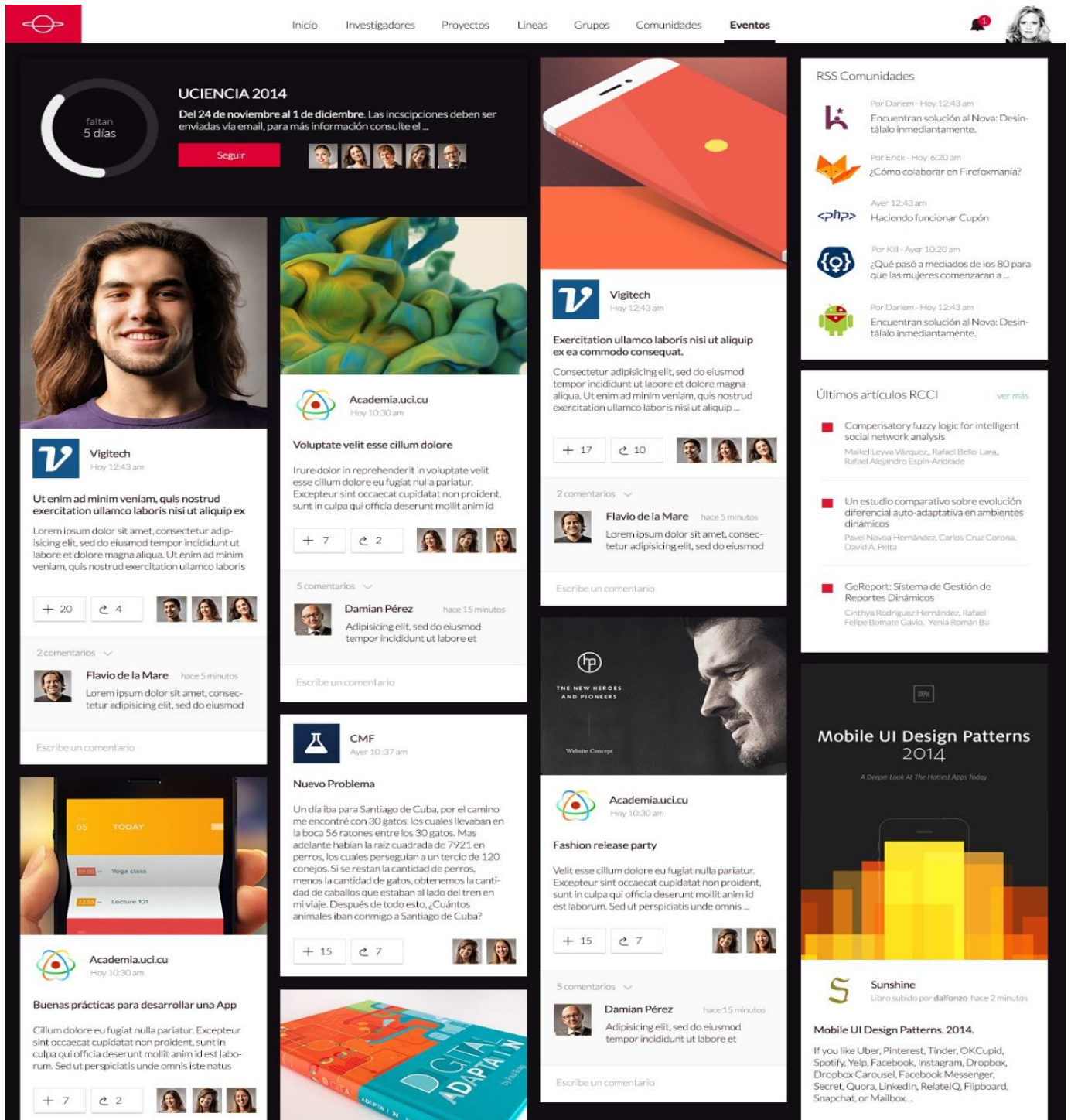


Figura 6: Prototipo de IU de AF “Menú principal”.

Gestión de eventos

Esta pantalla muestra los eventos próximos a realizarse en forma de una línea de tiempo y con una barra de progreso circular que indica los días restantes para la celebración de los mismos. Cada evento brinda la opción de seguirlos y de esta forma en la página principal se muestra al usuario interesado la información relacionada a este.

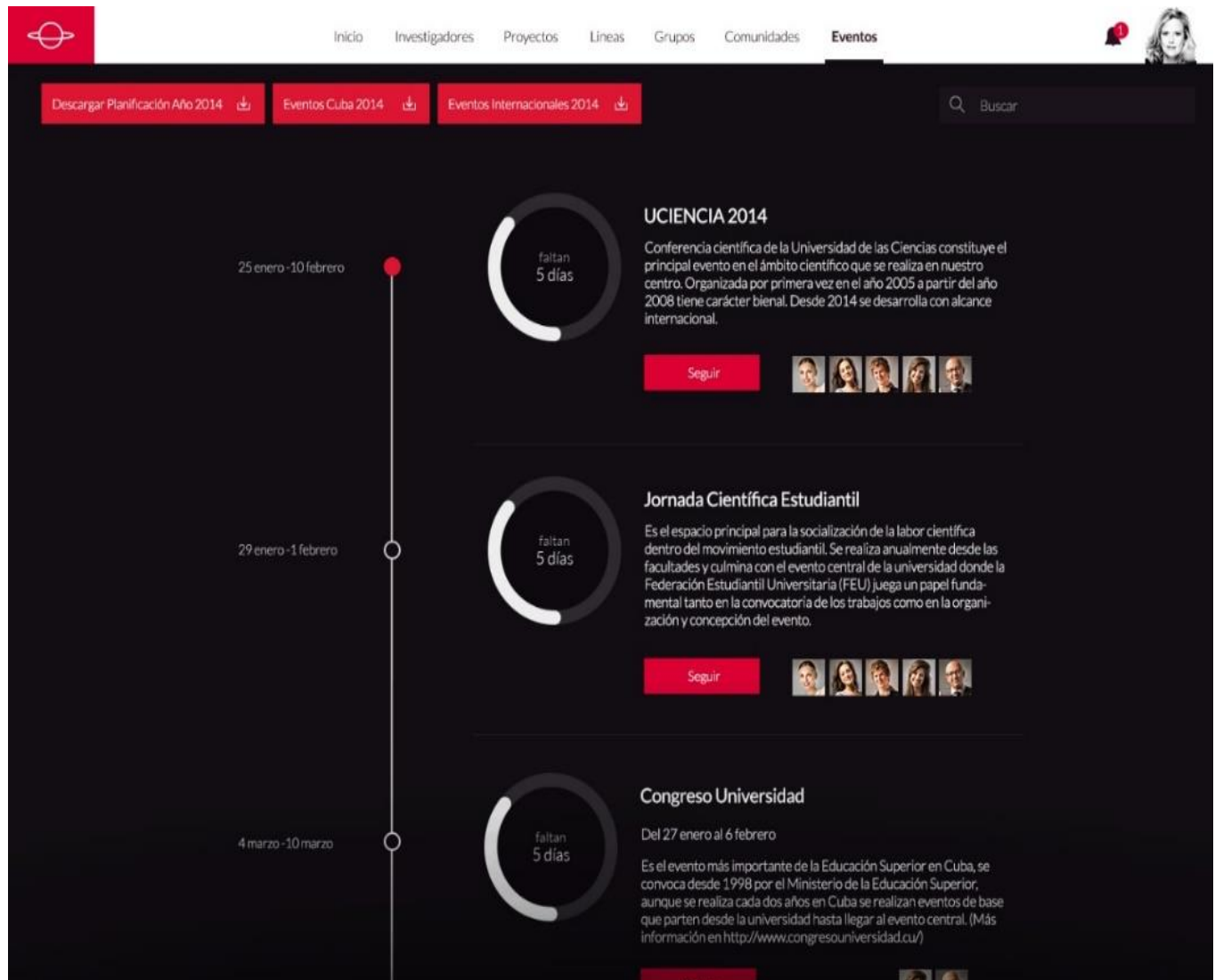


Figura 7: Prototipo de IU de AF “Calendario de eventos”.

Perfil de investigador

Muestra el perfil de cada investigador así como a los grupos, líneas y proyectos que está asociado y la cantidad de publicaciones que ha realizado.

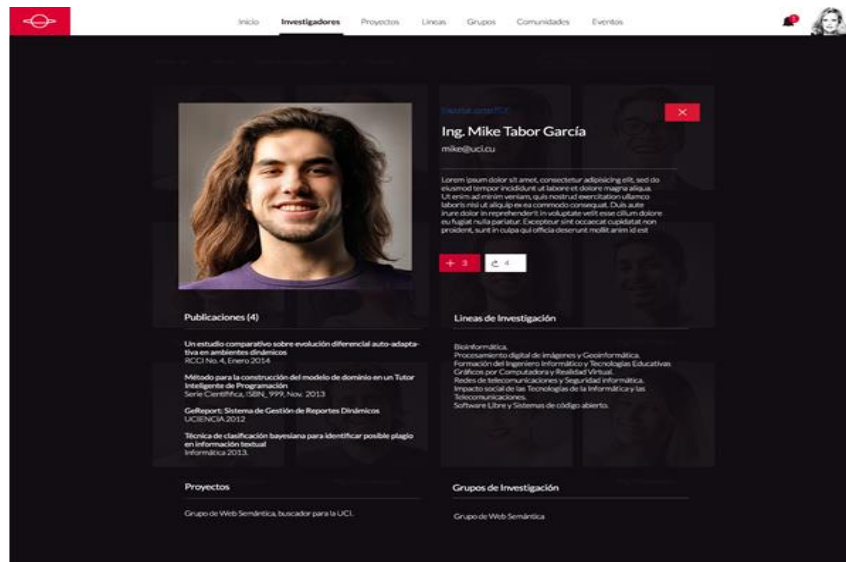


Figura 8: Prototipo de IU de AF “Perfil de investigador”.



Figura 9: Prototipo de IU de AF “Perfil de investigador” vista desde un Tablet.

2.6 Modelo de diseño

El objetivo fundamental de este modelo es alcanzar una solución lógicamente basada en el paradigma orientado a objetos. Su base es la confección de diagramas que revelan gráficamente cómo los objetos se comunicarán entre ellos con el fin de llegar a satisfacer los requisitos. Para preparar un buen diseño de

clases se hace necesaria la aplicación de los principios de la asignación de responsabilidades y utilizar los patrones de diseño (Larman, 1999).

2.6.1 Estilo de arquitectura

Proporciona una visión general del sistema a desarrollar. Describe la estructura y la organización de los componentes del software, sus propiedades y las conexiones entre ellos. Los componentes del software incluyen módulos de programas y varias representaciones de datos que son manipulados por el programa (Pantoja, 2010).

Un estilo arquitectónico es un conjunto coordinado de restricciones que establecen los roles/rasgos de los elementos de la arquitectura y las relaciones permitidas entre estos.

2.6.1.1 Estilo de llamada y retorno

Este estilo enfatiza y resalta la escalabilidad dándoles a la solución estrategias para combatir el crecimiento del contenido y crecimiento de la audiencia. Se encuentra entre los más generalizados en sistemas de gran escala y fue seleccionado para el desarrollo de la propuesta de solución.

2.6.2 Patrones arquitectónicos y de diseño

Los patrones arquitectónicos ofrecen soluciones a problemas de arquitectura de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones (Pressman, 2011).

La elaboración de un buen diseño del software contribuye directamente a la calidad del producto final, gran parte de esta nace en la utilización adecuada de los patrones de diseño y arquitectónicos existentes en la actualidad.

Con empleo del marco de trabajo MongoDB se garantiza la utilización de varios de estos patrones, permitiendo a los desarrolladores utilizar buenas prácticas de programación y ahorrar tiempo y recursos. A continuación se describen los principales patrones empleados en el diseño e implementación de la solución.

2.6.2.1 Patrón Modelo Vista Controlador (MVC)

El patrón MVC obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el marco de trabajo (ZANINOTTO, 2009), la interfaz de usuario se guarda en la vista, la manipulación de datos se guarda en el modelo y el procesamiento de las peticiones el controlador.

Con NodeJS el MVC se trabaja con controladores (el archivo routes/index.php) y vistas (alojadas en la carpeta views), y el modelo es el que se encarga del acceso a bases de datos a través de la clase (Mongoose), debe resolver las peticiones que le hace el controlador, y el controlador simplemente es un puente entre las vistas y los modelos.

AngularJS utiliza de forma no convencional en el cliente el MVC, mientras que el MVC ha adquirido diferentes matrices de significado a través de los años desde que apareció por primera vez, este marco de trabajo incorpora los principios básicos a su manera de construir páginas web actuales, separando las aplicaciones en componentes distintos: presentación, datos y lógica, fomenta el bajo acoplamiento entre los componentes y agrega una capa superior administrando la comunicación con el servidor.

2.6.2.2 Patrones GRASP

Los Patrones Generales de Software para Asignar Responsabilidades por sus siglas en inglés GRASP (General Responsibility Assignment Software Patterns), describen los principios fundamentales de asignación de responsabilidades a objetos, expresado en formas de patrones (Bustos, 2011).

Estos patrones hacen más fácil reutilizar con éxito los diseños y arquitecturas, ayudan a elegir entre diseños alternativos, hacen a un sistema reutilizable y evitan alternativas que comprometen a la reutilización. Además facilitan el aprendizaje al programador inexperto y ayudan a reutilizar código. Para el diseño de la plataforma se tuvieron en cuenta una serie de patrones de gran utilidad que proponen una forma de reutilizar la experiencia de los desarrolladores.

Experto: este patrón plantea que se debe asignar una responsabilidad al experto en información, en otras palabras, a la clase que cuenta con los datos necesarios para cumplir la responsabilidad. De esta forma se conserva el encapsulamiento de la información, puesto que los objetos ejecutan las tareas que le corresponden de acuerdo a la información que poseen, lo que da lugar a sistemas más robustos y fáciles de mantener.

Este patrón se evidencia en cada clase Modelo, cada una contiene la información referente a la entidad que representa y es responsable de realizar la labor que tiene encomendada.

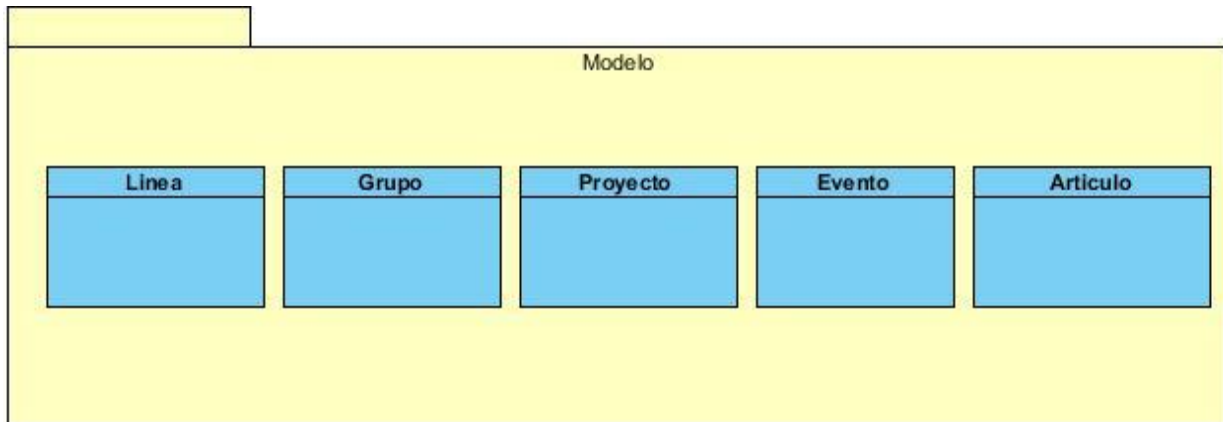


Figura 10: Patrón experto

Creador: este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, con esto se logra menos dependencia y mayores oportunidades de reutilización de código.

El uso de este patrón se evidencia en la clase Proyecto que crea instancias de la clase Grupo.

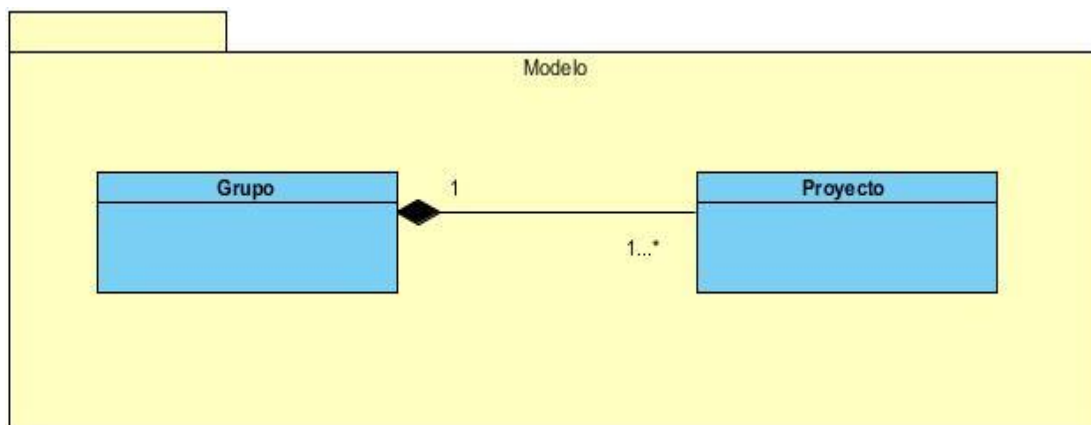


Figura 10: Patrón creador

Controlador: asigna a clases específicas la responsabilidad de controlar el flujo de eventos del sistema. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no las realiza, sino las delega en otras clases con las que mantiene un modelo de alta cohesión.

Este patrón se ve reflejado en todas las clases controladoras. La arquitectura MVC brinda una capa específicamente para los controladores, que son el núcleo de este, y especifica la presencia de este patrón.

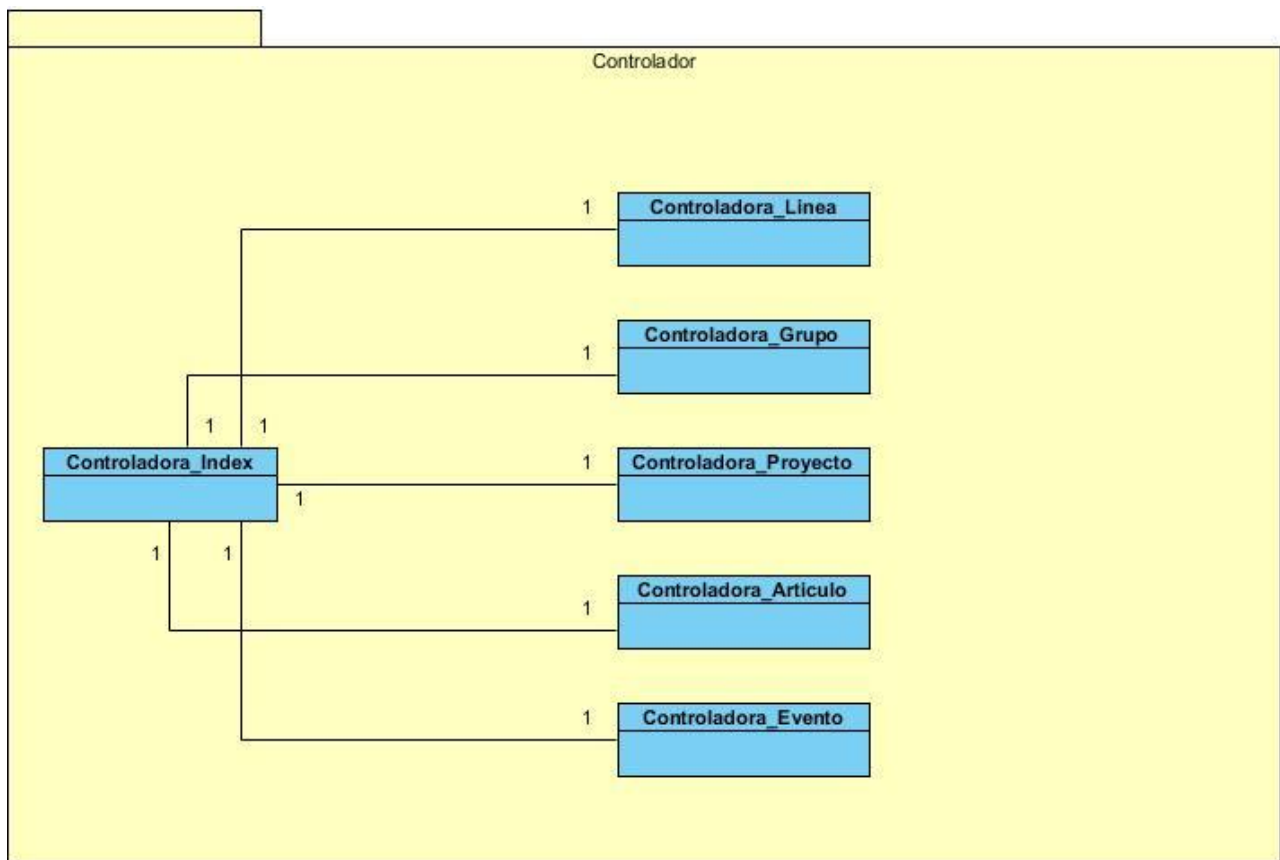


Figura 11: Patrón controlador

2.6.2.3 Patrones GoF

Los patrones GoF (Gang of Four, en español Pandilla de los Cuatro), se clasifican en 3 categorías basadas en su propósito: creacionales, estructurales y de comportamiento (ZANINOTTO, 2009). Para el diseño de la herramienta se tuvieron en cuenta los siguientes patrones creacionales:

Prototipo/Prototype: crea nuevos objetos clonándolos de una instancia ya existente, se evidencia cuando la clase Controladora_Proyecto crea un objeto clonándolo de una instancia de la clase Proyecto.

Este patrón se ve reflejado en la siguiente figura.

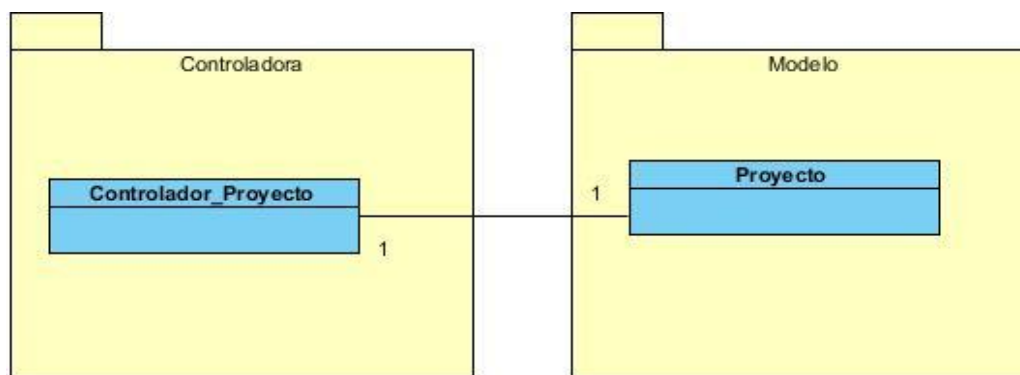


Figura 12: Patrón prototipo

Solitario/Singleton: garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Se crea una clase que gestiona una sola instancia de ella misma y cada vez que se necesite, simplemente se consulta la clase y esta devolverá siempre, la misma instancia.

Este patrón se evidencia en la clase Mongoose, encargada de realizar la conexión con la base de datos, se observa cuando la clase intenta acceder para realizar la conexión y necesita una instancia, y en ese momento se consulta la entidad que devuelve los datos que se necesita.

2.6.3 Diagrama de clases de la solución

Un diagrama de clases es una descripción de las clases en un sistema y sus relaciones. No describe el comportamiento dinámico del sistema sino la estructura del mismo mostrando sus clases, atributos y relaciones entre ellos (Pressman, 2004).

A continuación se muestran las clases identificadas durante el diseño de la solución. Dicha representación sirve como una aproximación al diseño definitivo del modelo de datos.

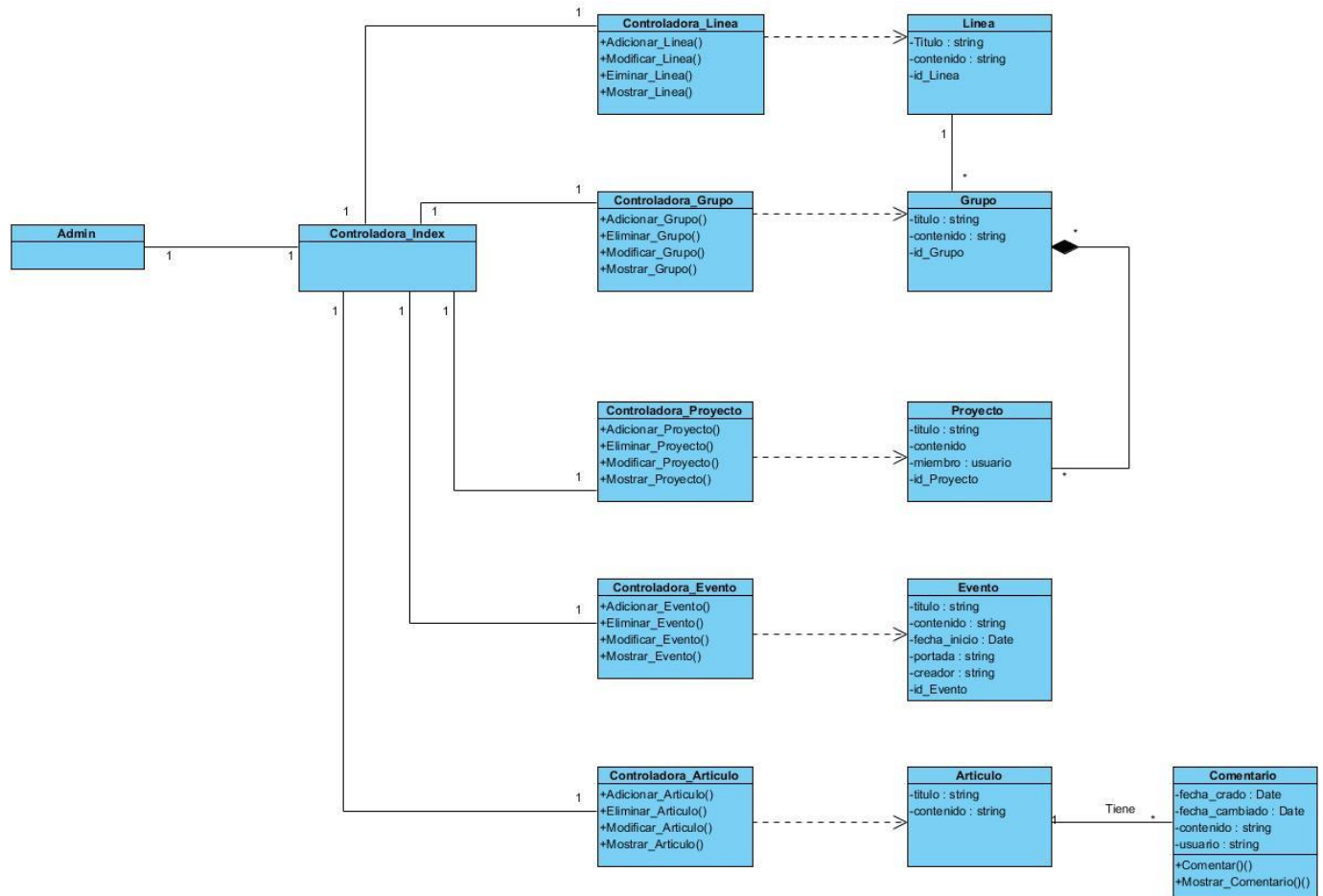


Figura 13: Diagrama de clases de la solución.

2.6.4 Vista de componentes

El diagrama de componentes muestra cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos y sus relaciones (Pressman, 2004). Permite además, describir la vista de implementación estática de un sistema. Un componente se corresponde con una o más clases, interfaces o colaboraciones. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. A continuación se muestra el diagrama de componentes de la aplicación para la acción de crear un nuevo proyecto.

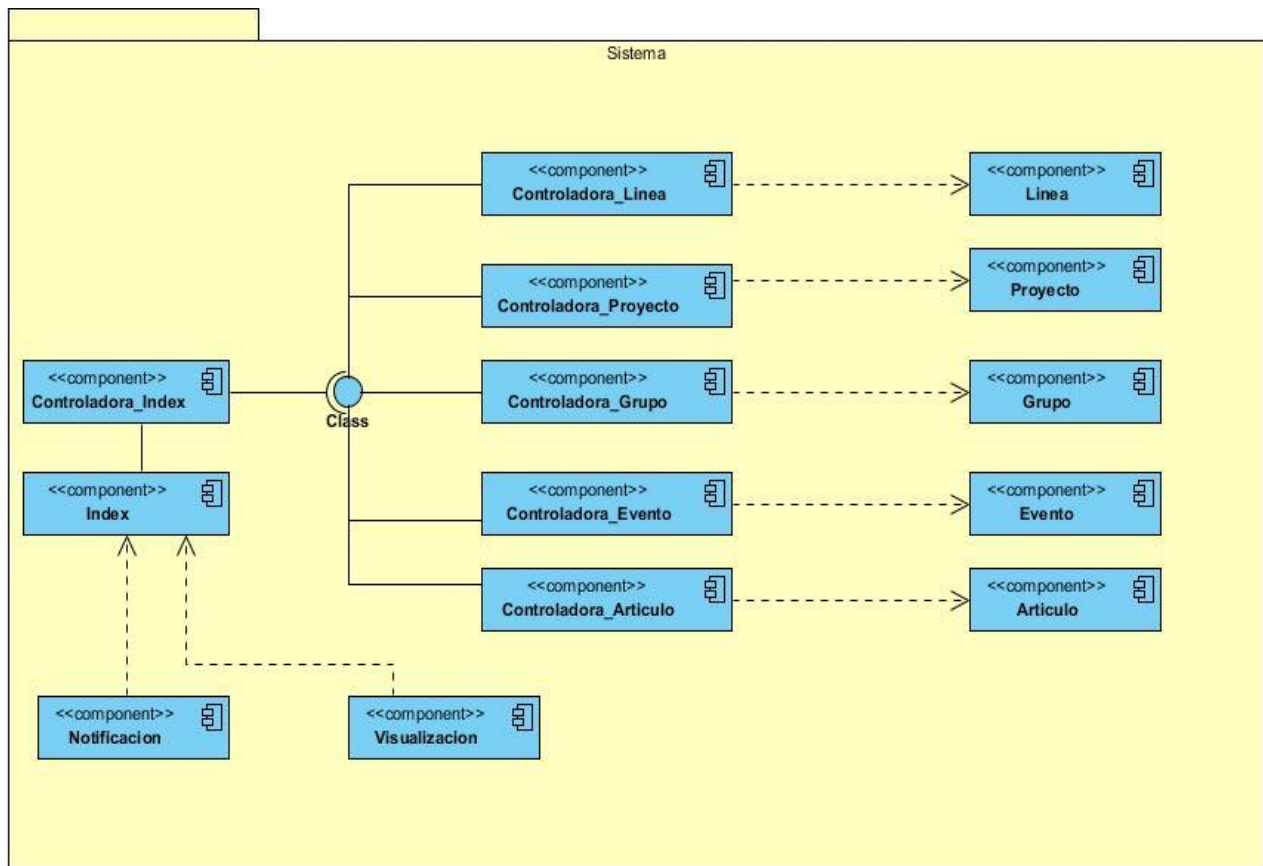


Figura 14: Diagrama componente de la solución

2.6.5 Diagrama de Despliegue

El modelo de despliegue o diagrama de despliegue es un tipo de diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista del despliegue (distribución) de los artefactos del software en los destinos de despliegue, se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Este tipo de diagramas muestran la configuración de los componentes de hardware, los procesos así como los elementos de procesamiento en tiempo de ejecución. En estos esquemas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes.

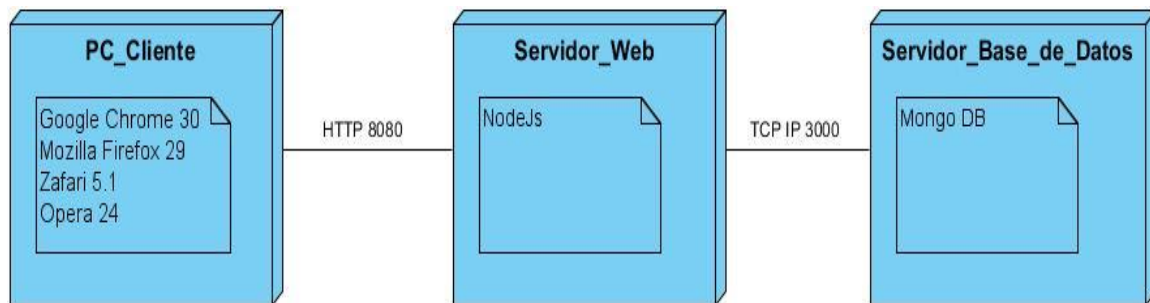


Figura 15: Muestra del diagrama despliegue de la solución

2.7 Conclusiones del capítulo

- La generación de los artefactos en la etapa de implementación de la plataforma siguiendo la metodología SXP y la descripción del diseño del sistema orientados por las definiciones abordadas en el Capítulo 1, facilitaron las tareas de implementación y la reducción de errores y riesgos.
- La aplicación de técnicas de la ingeniería de requisitos para la captura de los principales requerimientos contribuyó a la confiabilidad de los datos recolectados y a satisfacer las expectativas de los usuarios mediante las características de la aplicación.
- El empleo de una muestra confiable y representativa de usuarios reales o potenciales de la solución sentó las bases para la generalización de los resultados.
- La especificación de requisitos, el estudio de herramientas homólogas y los patrones de diseño permitieron poner énfasis en los prototipos de interfaz de usuarios con el objetivo de aportar utilidad a la plataforma para influir y motivar a los investigadores a interactuar.

CAPÍTULO 3: Validación de la solución propuesta

Introducción

En este capítulo se plasman los resultados de los casos de prueba y otras técnicas de medición que permitieron obtener una valoración integral de la solución propuesta y determinar las limitaciones del sistema.

3.1 Pruebas de software

Por lo general, las pruebas involucran operaciones del sistema evaluando los resultados bajo condiciones controladas, lo que hace que la realización de pruebas al software sea un factor de vital importancia.

3.1.1 Niveles de prueba

En la actualidad existen varios niveles de prueba, cada uno de ellos se relaciona con determinado momento del ciclo de vida del software. A continuación se describen cada nivel (Pressman, Roger S.2004, 2004).

3.1.1.1 Pruebas Unitarias

En este nivel de pruebas se ejecutan fundamentalmente casos de prueba de caja blanca diseñados para:

- Probar las estructuras de datos locales para asegurar que los datos conserven su integridad de manera que no puedan ser alterados o modificados durante los pasos de ejecución del algoritmo.
- Probar las condiciones límites que aseguren el correcto funcionamiento del componente en los límites establecidos como restricciones de procesamiento.
- Ejercitar todos los caminos independientes de la estructura de control con el fin de asegurar que todas las sentencias del módulo o componente se ejecutan por lo menos una vez. Los casos de prueba diseñados para ejecutar a este nivel debe descubrir errores como:
 - Comparaciones entre tipos de datos distintos.
 - Operadores lógicos o de procedencia incorrectos.
 - Igualdad esperada cuando los errores de precisión la hacen poco probable.
 - Variables o comparaciones incorrectas.
 - Terminación de bucles inexistentes.
 - Fallo de salida en iteraciones divergentes.
 - Variables de bucles modificadas inapropiadamente.

En este contexto el uso de las pruebas de límites son de extrema utilidad, puesto que el software tiende a fallar frecuentemente al enfrentarse a situaciones extremas. La selección correcta de los juegos de datos con los que se realizará la prueba constituye un paso primordial en esta etapa (Pressman, Roger S.2004, 2004).

3.1.1.2 Pruebas de sistema

Las pruebas de sistemas caen fuera del ámbito del proceso de software y no las realiza únicamente el desarrollador del software (Gerardo, 2013). Sin embargo los pasos durante el diseño del software y durante la prueba pueden mejorar la posibilidad de éxito de las pruebas de sistema. Este tipo de pruebas se centran fundamentalmente en verificar la interacción de los actores con el sistema, por lo que a menudo los casos de pruebas se obtienen a partir de las descripciones de los casos de uso, aunque también se le aplican pruebas al sistema como un todo (Pressman, Roger S.2004, 2004).

3.1.1.3 Pruebas de aceptación

La mayoría de los desarrolladores de software llevan a cabo un proceso denominado Alfa o Beta para descubrir errores que solo el usuario final puede descubrir (Pressman, Roger S.2004, 2004). Las Pruebas Alfa consisten en llevar al cliente al entorno de desarrollo para probar el sistema. De esta forma el cliente trabaja en un entorno controlado y siempre tiene un experto a mano para ayudarlo a usar el sistema y para analizar los resultados. Las Pruebas Beta por su parte, se realizan luego de culminadas satisfactoriamente las Pruebas Alfa. Se desarrollan en el entorno del cliente, es decir, en el lugar de trabajo para el que fue diseñado el sistema. De esta manera el cliente puede probar el producto a su manera, tratando siempre de encontrarle fallos. Las Pruebas de Aceptación del Cliente incluyen dos tipos de pruebas a niveles diferentes; las pruebas de aceptación y las pruebas piloto. Ambas se corresponden con los niveles de Pruebas Alfa y Pruebas Beta respectivamente (Cabrera González, 2011).

3.2 Placer de uso

Estudios demuestran que los seres humanos son más eficientes y creativos en la solución de problemas cuando son felices (Norman, Donald A. 2002, 2002). Algunas investigaciones también indican que la emoción está estrechamente vinculada al placer de uso (Agarwal, 2009). Las pruebas que se realizarán están encaminadas a determinar la respuesta emocional, el sentimiento que se obtiene de la interacción con la IU para medir la aceptación y el placer que producen en los usuarios de la plataforma. Existen

herramientas que permiten la medición de la respuesta emocional tanto de forma verbal (pensando de en voz alta) como no verbal (emocard).

3.3 Resultados de las pruebas funcionales

Para la realización de estas pruebas se ejecutaron las pruebas funcionales en tres iteraciones. La primera iteración con un total de 61 no conformidades, se corrigieron y se procedió a la segunda iteración.

Para la ejecución de la segunda iteración se realizó una regresión sobre las no conformidades detectadas durante la primera iteración de las pruebas encontrándose nuevas no conformidades un total de 13. Para la ejecución de la tercera iteración se realizó una regresión sobre las no conformidades detectadas durante la segunda iteración de las pruebas encontrándose 5 no conformidades. En la última iteración Los resultados obtenidos reflejaron la solución en su totalidad de las no conformidades anteriores y no se encontraron nuevas, por lo que no fue necesaria una cuarta iteración de las pruebas funcionales.

El departamento de calidad del Centro de Informatización de Entidades (CEGEL), luego de culminado el proceso de pruebas decidió emitir un acta de liberación para la herramienta. A continuación se presenta el resumen de no conformidades detectadas en las dos iteraciones de pruebas funcionales realizadas a la aplicación.

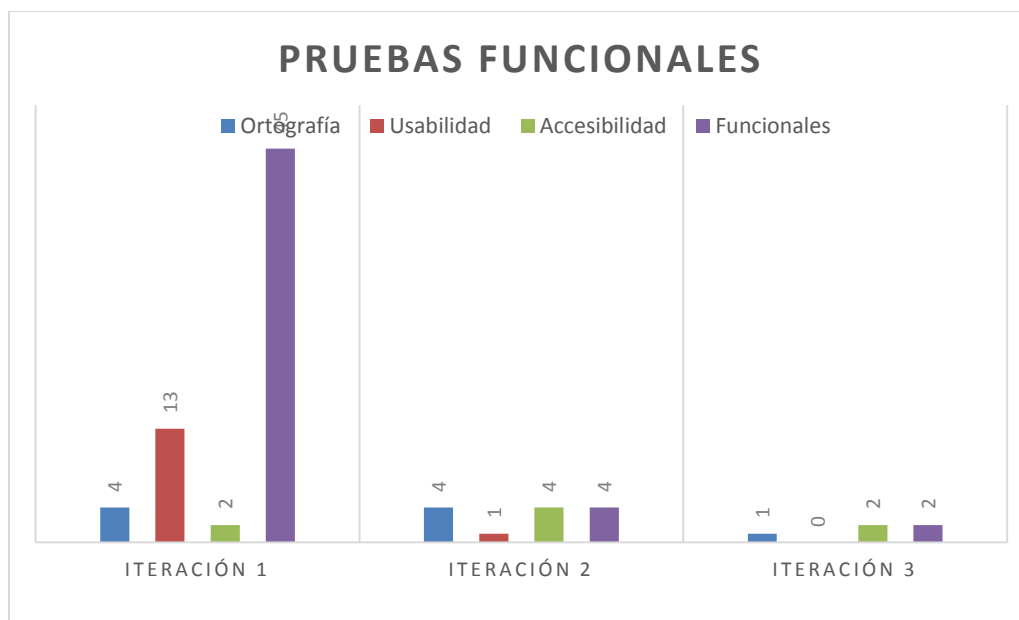


Figura 13: No Conformidades encontradas en el sistema durante las pruebas funcionales

Las no conformidades se encuentran detalladas en el anexo 3.

3.4 Resultados sobre las pruebas de caja negra

A partir de la realización de las pruebas de caja negra, como parte de las pruebas funcionales, se demostró el correcto funcionamiento de la solución a través del cumplimiento de los requisitos funcionales planteados por los clientes. La realización de estas pruebas permitió también encontrar clases no válidas que no se correspondían con las esperadas. Dichas clases no válidas fueron analizadas y revisadas para garantizar el cumplimiento con los requerimientos del cliente.

3.5 Resultados de las pruebas de resistencia y rendimiento

Las pruebas de resistencia y rendimiento por lo general son realizadas por programas encargados de simular condiciones extremas para el software a probar, aunque también pueden ser realizadas manualmente por miembros del equipo u otras personas seleccionadas mediante diferentes métodos.

La solución implementada, por las características que posee y el negocio al que está enfocado, recibirá una notable cantidad de peticiones por parte de los usuarios o sistemas que usen los servicios de la misma. Por esta razón se hace necesario realizar un estudio de la resistencia y rendimiento de las diferentes funcionalidades del sistema.

Inicialmente se efectuaron pruebas con los miembros del equipo de desarrollo, pero a pesar de obtener resultados satisfactorios, se hizo necesaria la realización de un estudio más profundo en aras de reconocer las debilidades de la solución ante un flujo mayor de peticiones.

Se decidió utilizar para las pruebas de carga y estrés la herramienta JMeter por las funcionalidades que presenta ya que las mismas permiten realizar los test que serán empleados y aplicar los mismos, garantizando que los resultados arrojados sean de fácil entendimiento para la persona que ejecuta la prueba, además de ser una herramienta libre. También se pueden utilizar muchas de sus funcionalidades para probar aspectos de las pruebas de Disponibilidad y Red.

JMeter se clasifica como una herramienta dinámica de medida por las funcionalidades que incorpora, como la manipulación de datos de entrada para la ejecución de las pruebas y la comparación de los resultados para la evaluación realizados aplicando el método de caja negra. Además se clasifica por su especialización en una herramienta de grabación y reproducción.

La utilización del JMeter supone un 94 % de tiempo menos para la realización de estas pruebas. La mayor inversión de tiempo que se necesita para la realización de las pruebas es la fase de estudio de los casos de uso críticos en la aplicación Web y la elaboración del plan de pruebas en la herramienta. Permite almacenar los resultados de la prueba y generar gráficos que representan los aspectos que se han probado.

Los resultados obtenidos en estas pruebas son satisfactorios. El 100% de las funcionalidades dio respuesta a la petición en menos de 1.5 segundos lo cual evidencia la optimización realizada en algunas funcionalidades y el correcto funcionamiento en general de la aplicación.

Al efectuarse la prueba se obtuvo como resultado un 100% de respuestas satisfactorias con tiempo de respuesta promedio de 1901 ms, el cual se puede calificar de satisfactorio. A continuación se muestran tablas y gráficos que complementan los datos obtenidos en las pruebas.

Funcionalidad	Peticiones	Promedio(ms)	Media(ms)	Mínima(ms)	Máximo(ms)
Buscador de líneas de investigación	100	2660	2110	2075	12306
Modificar líneas de investigación	100	1570	1260	1220	5845
Adicionar proyecto de investigación	100	1105	1085	1070	1400
Eliminar grupo de investigación	100	2270	2235	2225	2985
Total	100	1901	2090	1070	12306

Tabla 16: Resultados de las pruebas de carga y estrés a funcionalidades críticas con 100 peticiones.

No obstante estos resultados se decidieron aumentar la carga de usuarios y así verificar el desenvolvimiento del sistema en condiciones de estrés. Para ello se rediseñó el plan de pruebas para

trabajar con 150 peticiones arrojando como resultado un 100% de éxito en las llamadas así como un tiempo de respuesta promedio de 3616 ms. A continuación se muestran tablas y gráficas que complementan los datos obtenidos en las pruebas.

3.6 Resultados de las pruebas para el placer de uso

3.6.1 Pensando en voz alta

La prueba “pensando en voz alta”, es uno de los métodos más sencillos y longevos (Nielsen, 2012) en el campo de la UE. Esta prueba consiste en pedir a los usuarios que, mientras interactúan con las IU, expresen sus opiniones sobre el sistema se tomó como muestra 6 usuarios determinado según el informe “Beyond Usability” (Agarwal, 2009), según su género (masculino o femenino). Estas características hacen de ella un método de evaluación que puede ser utilizado como una herramienta para medir la respuesta emocional de los usuarios de forma verbal y para realizar mejoras en función de la usabilidad.

Con la aplicación de la técnica se obtuvieron los resultados siguientes.

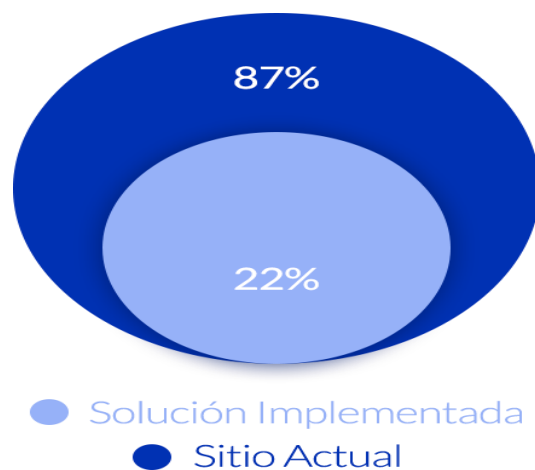


Figura 16: Pensando en voz alta.

El 85% de los usuarios solicitaron cambios decisivos o simplemente emitieron criterios poco favorables sobre la solución implementada mientras que solo un 20% manifestó su desagrado sobre la Solución Implementada. Como plantea y demuestra Norman (2004) ningún producto recibe una aceptación total, por lo que los resultados de las pruebas responden a una tendencia natural

3.6.2 Emocard

Es una técnica efectiva para la mediación de la respuesta emocional de forma no verbal. Utiliza tarjetas que les permita a los usuarios expresar lo que sienten en el instante de forma sencilla, eliminando las limitaciones en el lenguaje. Para los usuarios supone una dificultad expresar con palabras sus emociones, y es común que los evaluadores no logren encontrar realmente la verdadera sensación que provoca la aplicación. Esta técnica permite al usuario identificar (a través de rostros que emulan estados de ánimos) como se siente al interactuar con el sistema.

Se evaluó una muestra de 6 usuarios (el tamaño de la muestra se determinó según el informe “Beyond Usability” (Agarwal, 2009) a los que se entregaron, según su género (masculino o femenino) un juego de 8 tarjetas con rostros de hombre o mujer, en donde estaban registrados diversos estados desde “excitación” hasta “calma” y con inclinaciones negativas o positivas (ver anexo 2).

Para el completamiento de esta prueba a los usuarios se les pidió reflejar su estado de ánimo en dos momentos diferentes, al entrar y al salir de la plataforma. Tomando los dos grupos de resultados quedaría de la siguiente manera:

- Primera etapa: Respuesta emocional del usuario al comenzar a utilizar la plataforma.



Figura 17: Aplicación de la técnica del Emocard primera etapa.

Una vez que los usuarios ya se habían registrado en la aplicación se les pidió que reflejaran sus emociones mediante la técnica de Emocard.

Los usuarios coinciden entre la categoría de agradable y neutral. No sucede lo mismo con la versión antigua pues esta logra una menor aceptación, los resultados muestran una mayor incidencia en la categoría neutral con una tendencia hacia la no aceptación.

- Segunda etapa: Estado emocional del usuario una vez que termina el trabajo con la plataforma.



Figura 18: Aplicación de la técnica del Emocard segunda etapa.

La nueva versión provoca placer en los usuarios, en contraste con la versión antigua que produce una reacción emocional neutra con tendencias al desagrado, todo esto producto de la utilización de las técnicas para la medición de las emociones aplicadas.

3.7 Experimento

Se tuvo en cuenta dos grupos de personas basadas en el estudio de Nielsen expuesto anteriormente en el capítulo 2 en el epígrafe 2.4, con el fin de realizar pruebas y acciones para medir el trabajo colaborativo evaluando sus indicadores en cuanto al por ciento de aceptación en un periodo de 15 días. Estas se

realizaron teniendo en cuenta la solución implementada. A continuación se muestra los resultados alcanzados.

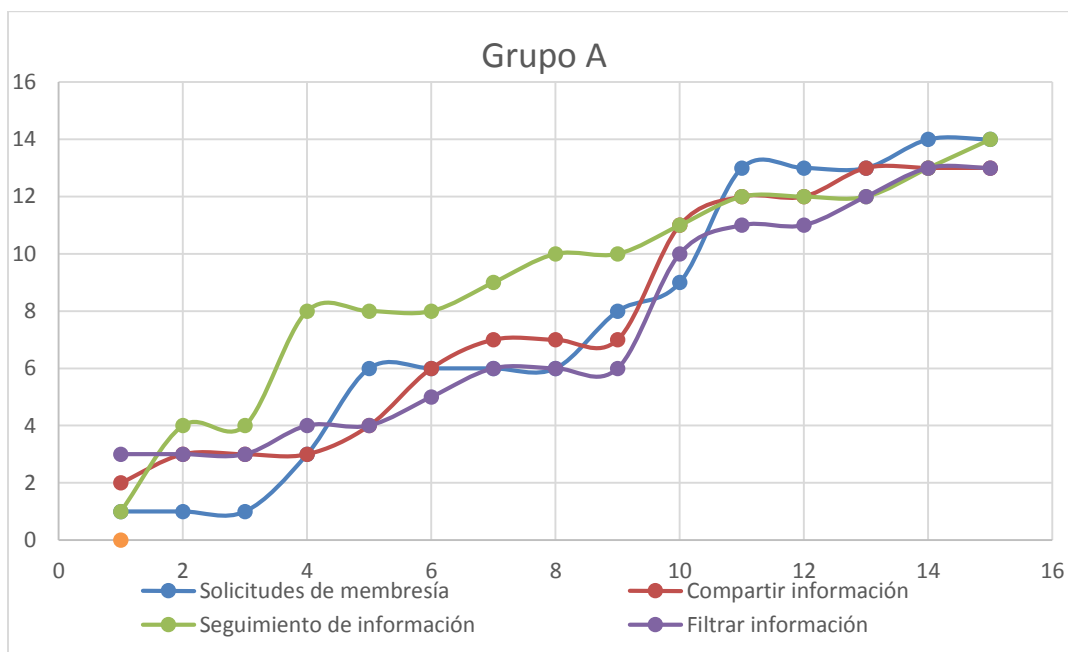


Figura 19: Resultados del experimento sobre el primer grupo de personas.

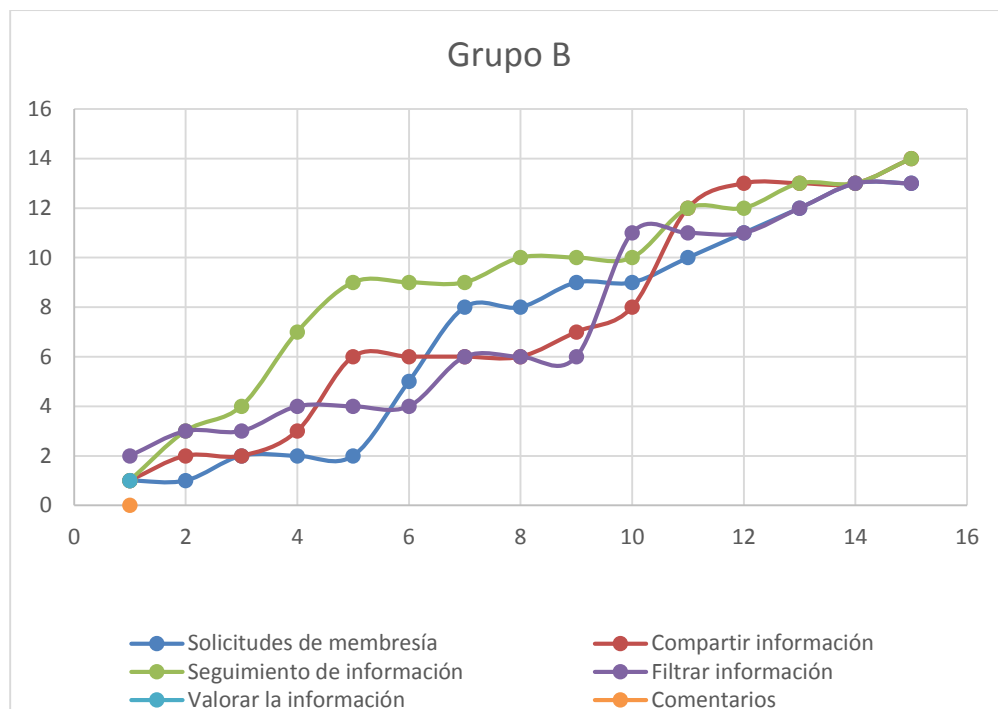


Figura 20: Resultados del experimento sobre el segundo grupo de personas.

Luego de concluida la prueba se puede afirmar que los resultados fueron similares. Esta potencia eleva el trabajo colaborativo de múltiples usuarios en la red de investigadores de la universidad de manera óptima, con patrones de diseño para la creación de interfaces web que permiten que este sea dominante sobre el existente en cuando a los indicadores representado en la tabla. A continuación se muestra una tabla con los resultados combinados.

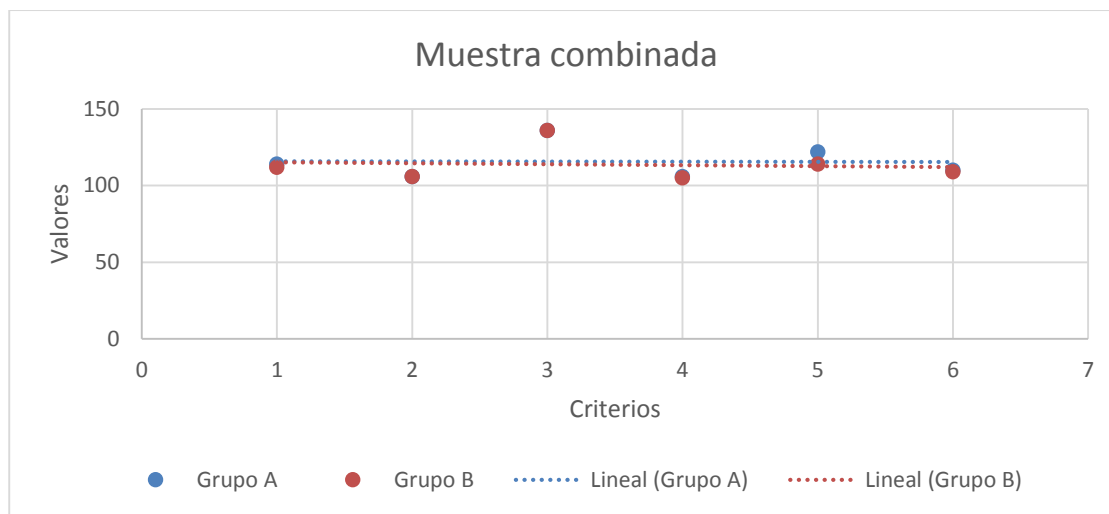


Figura 21: Resultados del experimento combinado sobre los dos grupos de personas.

3.8 Conclusiones del capítulo

- El estudio de los principales tipos y métodos de pruebas que se llevaron a cabo durante el ciclo de vida del software permitieron aplicar las mediciones y comprobar la calidad de los resultados.
- Se describieron las pruebas de caja negra efectuadas. Dichas pruebas corroboraron el cumplimiento de los requerimientos planteados por los clientes así como la obtención de los datos que se esperaban como respuesta de la aplicación.
- Para realizar las pruebas de resistencia y rendimiento fue necesario estudiar las principales herramientas existentes en el mercado, seleccionándose JMeter como software para medir los niveles de carga y estrés que soporta la aplicación.
- Se identificaron las funcionalidades críticas de la solución con el objetivo de chequear su funcionamiento bajo condiciones de estrés, llegándose a obtener resultados satisfactorios.
- Además otras funcionalidades también fueron analizadas, siempre teniendo en cuenta las características del servidor donde fueron probadas las funcionalidades, lo que permitió establecer estimados para un servidor de producción y llegar a la conclusión de que el sistema funciona adecuadamente.

Conclusiones

En el presente trabajo se realizó un análisis de los enfoques existentes en la gestión de la información científico-técnico. Se creó una solución informática que posee el potencial de elevar el trabajo colaborativo de los individuos en sus contextos de interacción científica. En función de los objetivos de la investigación los resultados permitieron arribar a las siguientes conclusiones:

- El estudio de enfoques y tendencias de diseño en aplicaciones web para la gestión de la información científico-técnica permitió identificar mecanismos de recuperación de los datos, patrones y estilos que estimulan la cooperación entre los usuarios.
- El análisis de las principales fuentes bibliográficas permitió identificar los aspectos que caracterizan la gestión de la información científico-técnica, entre los que se señalan la adaptabilidad y accesibilidad de la información.
- El análisis de las tecnologías para implementar la solución requerida hizo prevalecer el enfoque web, accesibilidad y adaptabilidad de la aplicación a los entornos y contextos de interacción de los usuarios, así como planteo el uso de herramientas libres acordes a las políticas de soberanía tecnológica de la UCI y el país.
- La especificación de requisitos, el estudio de herramientas homólogas y los patrones de diseño permitieron poner énfasis en los prototipos de interfaz de usuarios con el objetivo de aportar utilidad a la plataforma para influir y motivar a los investigadores a interactuar. La implementación de estas características permitieron potenciar el trabajo colaborativo en la red de investigadores.

Referencias

Agarwal, Anshu y Meyer, Andrew. 2009. *Evaluating Emotional Response as an Integral Part of the User Experience*. Boston, MA, USA. : issn 3265-6936, 2009.

—. **2009.** *Evaluating Emotional Response as an Integral Part of the User Experience*. Boston, MA, USA : imms 6995-5842, 2009. 9568.

Berners-lee. 2009. The Semantic Web en línea. *The Semantic Web en línea*. [En línea] Scientific American, 2 de noviembre de 2009. [Citado el: 20 de mayo de 2015.] <http://www.scientificamerican.com/2001/0501issue/0501bernersee.html>. 3265.

Blanco, Iván Durango. 2014. ENSEÑAR ES APRENDER DOS VECES. *ENSEÑAR ES APRENDER DOS VECES*. [En línea] Licencia Creative Commons, 3 de junio de 2014. [Citado el: 8 de mayo de 2015.] <http://aprendiendo2veces.blogspot.com/2012/05/prototipado-de-interfaces-de-usuario.html>. 3546.

Bustos, Juan Pablo. 2011. Slideshare. *Slideshare*. [En línea] Technology and Software, 21 de mayo de 2011. [Citado el: 1 de mayo de 2015.] <http://es.slideshare.net/jpbthames/patrones-para-asignar-responsabilidades-grasp>. 9564.

Cabrera González, Miguel P. 2011. *Sistema económico integrado Versat*. Sarasola. Santa Clara : issn 2218-3620, 2011.

Cantillo, Ricardo. 2010. *Evaluación del Sitio Web del Grupo Empresarial*. La Habana : Facultad de Turismo, Universidad de La Habana, 2010. Trabajo de Diploma.

Caro, Paola. 2012. Metodología Gestión de Requerimientos. *Metodología Gestión de Requerimientos*. [En línea] Producciones Internacionales, 9 de mayo de 2012. [Citado el: 7 de mayo de 2015.] <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>. 4205.

Christiansen, Simon. 2012. Database Management System (DBMS). *Database Management System (DBMS)*. [En línea] 11 de Enero de 2012. [Citado el: 6 de diciembre de 2014.] <http://searchsqlserver.techtarget.com/definition/database-management-system>.

Ciencias de la Información. **Dante, Gloria Ponjuán. 2008.** La Habana : issn 3024-3840, abril de 2008, *Ciencias de la Información*, Vol. 46, pág. <http://www.redalyc.org/revista.oa?id=1814>.

Classroom now. 2014. ProcesosdeSoftware. *ProcesosdeSoftware*. [En línea] Creative Commons Attribution Share, 11 de agosto de 2014. [Citado el: 17 de abril de 2015.] <http://procesosdesoftware.wikispaces.com/METODOLOGIAS+PARA+DESARROLLO+DE+SOFTWARE.BY/SA>.

Cujae. 1998. Cujae. *Instituto Superior Politécnico José Antonio Echeverría*. [En línea] 26 de octubre de 1998. [Citado el: 10 de marzo de 2015.] <http://www.uh.cu/>.

Del Ramo. 2012. beginning of scientific research ,. *Artículo científico*. Valencia. : s.n., 2012. Vol. 48.

El Caparazón desde 2007. 2007. El Caparazón desde 2007. *Tendencias de la web en la Web 2.0 expo 2009*. [En línea] 2007. [Citado el: 3 de mayo de 2015.] <http://www.dreig.eu/caparazon/2009/04/02/tendencias-de-la-web-en-la-web-20-expo-2009/>.

Fisteus, Jesus Arias. 2008. *Tecnologías Web de cliente y servidor*. Madrid : GCI, 2008. DTD/693.

Gerardo, Adner. 2013. Slideshare. *Slideshare*. [En línea] Technology and Software, 29 de junio de 2013. [Citado el: 10 de mayo de 2015.] <http://es.slideshare.net/abnergerardo/pruebas-de-sistemas-y-aceptacion-23663195>. 2548.

Guaita, Alvaro Martínez. 2012. desarrolloweb.com. *desarrolloweb.com*. [En línea] 2012. [Citado el: 7 de diciembre de 2014.] <http://www.desarrolloweb.com/actualidad/nuevo-bootstrap-2-2-7608.html>.

J, Carlos Mario Zapata. 2010. *UN MODELO DE DIÁLOGO PARA LA GENERACIÓN AUTOMÁTICA DE ESPECIFICACIONES*. Colombia : s.n., 2010. ISBN 978-958-44-6419-4.

John ZIMAN. 2002. Cambridge University Press. *Cambridge University Press*. [En línea] 26 de mayo de 2002. [Citado el: 13 de febrero de 2015.] <http://www.cambridge.org/sa/academic/subjects/general-science/popular-science/real-science-what-it-and-what-it-means>.

jquery. 2010. The jQuery Project. *The Write Less, Do More, JavaScript Library*. [En línea] 2010. [Citado el: 6 de diciembre de 2014.] <http://jquery.com/>.

Juan Quijano. 2012. Genbeta:dev. *Genbeta:dev*. [En línea] 28 de febrero de 2012. [Citado el: 21 de mayo de 2015.] <http://www.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional>.

Kadlec, Tim. 2013. *Implementing Responsive Desing. Building sites for an anywhere, everywhere.* Los Ángeles : New Riders : 978-0-321-82168-3, 2013.

Kalt,David. 2014. Alexa. *Alexa.* [En línea] Davil Kalt, 2014. [Citado el: 7 de diciembre de 2014.] <http://www.alexacom/>.

Larman, Craig. 1999. Applying UML and Pattern. *An Introduction to Object-Oriented Analysis and Design.* s.l. : Upper Saddle River: Prentice Hall, Inc., 1999. ISBN: 970, 1999.

Marcotte, Ethan. 2011. *Responsive Web Desing.* New York :A book Apart : 978-09844425-7-7, 2011.

Michael Abernethy, Freelancer. 2011. IBM Bluemix. *Freelance Programmer.* [En línea] 14 de junio de 2011. [Citado el: 4 de febrero de 2015.] <http://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/>.

MongoDB. 2013. MongoDB_UpRunning. *MongoDB_UpRunning.* [En línea] 13 de octubre de 2013. [Citado el: 4 de febrero de 2015.] http://www.stratebi.es/todobi/Oct13/MongoDB_UpRunning.pdf.

Mora, Roberto Canales. 2004. Visual Paradigm. UML, BPMN and Database Tool for Software Development.[Online]. *Visual Paradigm. UML.* [En línea] 1 de Febrero de 2004. [Citado el: 5 de febrero de 2015.] <http://www.visual-paradigm.com..>

Nafria, Ismael. 2007. Web 2.0. El usuario, el nuevo rey de Internet. *Web 2.0. El usuario, el nuevo rey de Internet.* [En línea] 19 de noviembre de 2007. [Citado el: 6 de febrero de 2015.] <http://ismaelnafria.blogspot.com/2007/10/web-20-el-usuario-el-nuevo-rey-de.html>.

Nielsen. 2012. Number of Participants in iser testing. *Nielsen Norman Group.* [En línea] junio de 2012. [Citado el: 13 de marzo de 2015.] <http://www.useit.com/alertbox/number-of-test-user.html>.

Norman, Donald A. 2002. 2002. Emotion and Design: Attractive things work better.[En línea]. *Don Norman: Designing For People.* [En línea] 22 de febrero de 2002. [Citado el: 13 de marzo de 2015.] http://www.jnd.org/dn.mss/emotion_design_at.html..

Novas, Pablo. 2012. fernetjs. *fernetjs.* [En línea] technologies, 28 de febrero de 2012. [Citado el: 17 de mayo de 2015.] <http://fernetjs.com/2012/02/estructura-de-un-sitio-web-mvc-en-nodejs/>. 1545.

Oracle, Corporation. 2012. FOSS License Exception. *MySQL.* [En línea] 11 de Enero de 2012. [Citado el: 17 de marzo de 2015.] <http://www.mysql.com/about/legal/licensing/foss-exception/>.

- Pacholczyk, Dominic. 2014.** Web UI Design Patterns. *A Deeper Look At the Hottest Websites and Web Apps Today*. [En línea] 28 de enero de 2014. [Citado el: 17 de marzo de 2015.] <http://www.uxpin.com/web-design-patterns.html>.
- Pantoja, E.B. 2010.** 2010, El patrón de diseño Modelo-Vista-Controlador (MVC). .
- Pedroso, Evelin. 2004.** *Historia del desarrollo de la Ciencia de la Información*. habana : Ciencias Médicas, 2004. ISSN 1024-9435.
- Pemberton, Steven. 2010.** *XHTML™ Modularization 1.1*. Leon : W3C, 2010. DTD/256.
- Pressman, Roger S.2004. 2004.** *Ingeniería del Software. Un enfoque práctico*. s.l. : Quinta Edición. s.l. : McGraw Hill,, 2004.
- Pressman, Roger S.2004. 2004.** *Ingeniería del Software. Un enfoque práctico*. *FreeLibros.org*. [En línea] 18 de febrero de 2004. [Citado el: 20 de marzo de 2015.] <http://www.freelibros.org/ingenieria/ingenieria-del-software-un-enfoque-practico-7ma-edicion-roger-s-pressman.html>.
- Pressman, Roger. 2011.** *Software Engineering, a practitioner's approach. (7ª edición)*. Buenos Aires : ISBN 9780071267823, 2011.
- Q-Vision Technologies. 2014.** q-vision. [En línea] 18 de febrero de 2014. [Citado el: 9 de abril de 2015.] <http://www.qvision.us/es/servicios/gestion-de-requerimientos-de-software/generacion-y-verificacion-de-historia-de-usuario-scrum>.
- Raya, Raúl. 2014.** LeanMonitorB. *LeanMonitorB*. [En línea] Corporation Methodologies, 24 de mayo de 2014. [Citado el: 02 de mayo de 2015.] <http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles/>. 2569.
- UBA. 1995.** UBA. *Universidad de buenos Aires*. [En línea] 2 de abril de 1995. [Citado el: 3 de abril de 2015.] <http://www.uba.ar/>.
- UCM. 2006.** UCM. *Universidad Computense de Madrid*. [En línea] 25 de agosto de 2006. [Citado el: 5 de abril de 2015.] <https://www.ucm.es/>.
- UH. 2012.** UH. *Universidad de la Habana*. [En línea] 2012. [Citado el: 6 de diciembre de 2015.] <http://www.uh.cu/>.

UNAM. 2009. UNAM. *Portal Universidad Nacional Autónoma de México*. [En línea] 2009. [Citado el: 6 de Diciembre de 2015.] <http://www.unam.mx/>.

Valdez, Joel. 2013. 2013.

Yusef, Hassan Montero y Sergio Ortega. 2009. Informe APEI sobre usabilidad. [aut. libro] Raquel Lavandera Fernández. *Informe APEI sobre usabilidad*. s.l. : Vol. 3. 978-84-692-3782-3, 2009, págs. págs. 9-11.

ZANINOTTO, F. y POTENCIER, F. 2009. librosweb.es. Symfony 1.1, la guía definitiva. [En línea] 2009. http://www.librosweb.es/symfony_1_1. 2009. librosweb.es. *librosweb.es*. [En línea] 12 de septiembre de 2009. [Citado el: 12 de mayo de 2015.] http://www.librosweb.es/symfony_1_1..

ZANINOTTO, F. y POTENCIER, F. 2009. Patrones, la guía definitiva. [En línea]. *Patrones, la guía definitiva*. [En línea] 2 de agosto de 2009. [Citado el: 13 de mayo de 2015.] http://www.librosweb.es/symfony_1_1..

Ziman, Mario. 2002. An analysis of information transfer in system-reservoir interactions. *An analysis of information transfer in system-reservoir interactions*. [En línea] 5 de marzo de 2002. [Citado el: 8 de marzo de 2015.] <http://www.muni.cz/research/publications/484392>.