



**SISTEMA PARA LA GESTIÓN DE LA AGENDA DE
TRABAJO DEL CONSEJO DE DIRECCIÓN DE LA
FACULTAD 4**

**Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor: Arleén García Jiménez

Tutor: MSc. Mailin Carballosa Infante

La Habana, Cuba

Junio de 2015

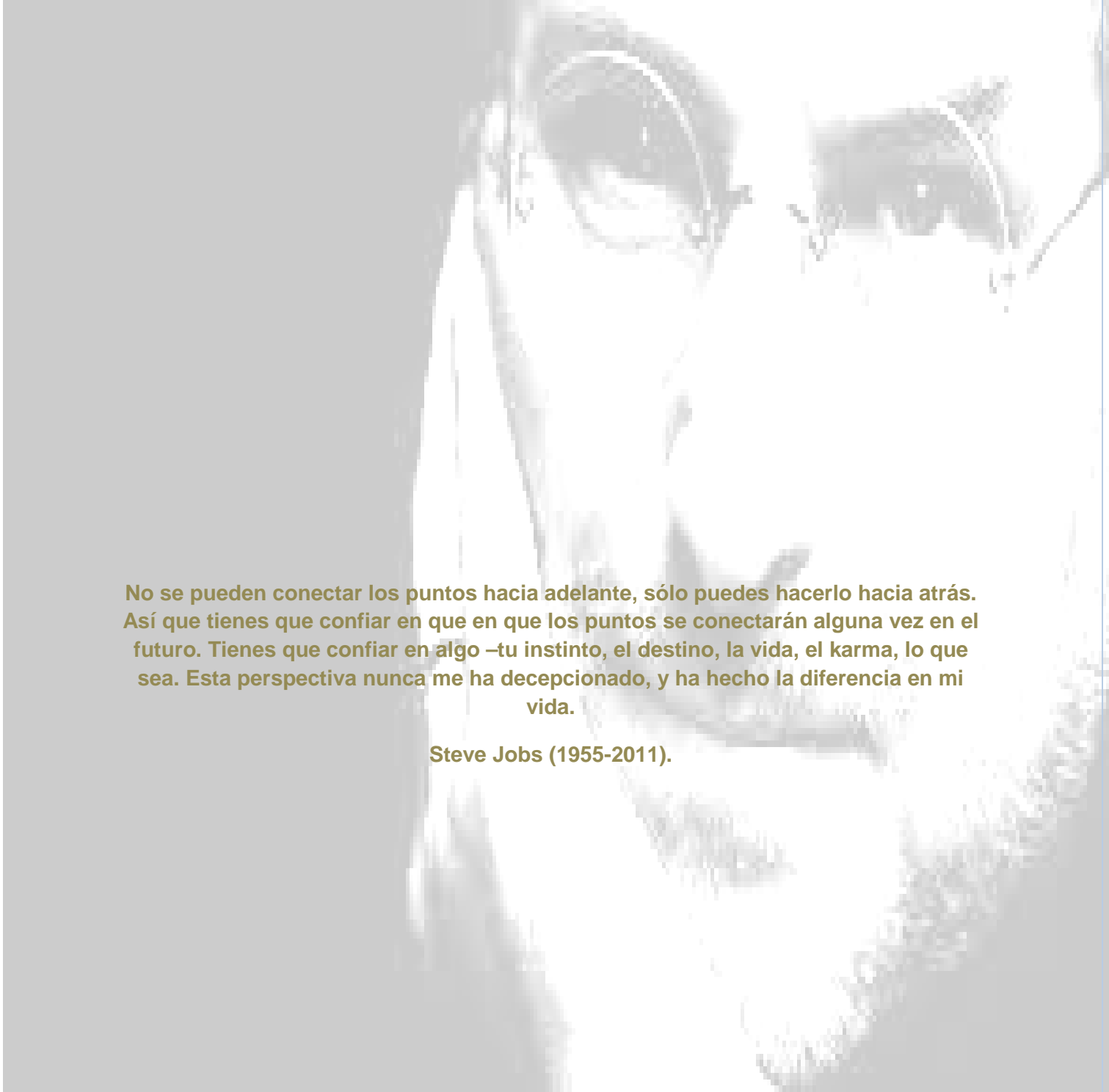
Declaración de Autoría

Declaración de Autoría

Declaro ser el autor del presente trabajo de diploma y otorgo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Firma del autor

Firma del tutor



No se pueden conectar los puntos hacia adelante, sólo puedes hacerlo hacia atrás. Así que tienes que confiar en que en que los puntos se conectarán alguna vez en el futuro. Tienes que confiar en algo –tu instinto, el destino, la vida, el karma, lo que sea. Esta perspectiva nunca me ha decepcionado, y ha hecho la diferencia en mi vida.

Steve Jobs (1955-2011).

*D*EDICATORIA

A mi madre por ser mi motor impulsor en esta vida, por su apoyo, su amor incondicional, su sacrificio incansable, por confiar siempre en mí, por no darme la espalda, por lo que hemos pasado y por lo que vendrá.

A mi abuela Georgina por estar ahí a mi lado desde siempre y ser mi apoyo en todo y ser guía para mi madre.

A Miriam mi otra madre por creer en cada momento en mí, por estar al pie del cañón cuando las cosas estuvieron duras, por la confianza, el apoyo, el amor incondicional, sin exigir nada a cambio.

A GRADECIMIENTOS

A la más ilustre de las mujeres, mi MADRE: Xiomara, mi abuela Georgina, por estar siempre apoyándome, a mis abuelos Dinorah y Tata que ya no están y no pudieron verme dar este importante paso en mi vida.

A Miriam - Veva unidas así para que sepan que están en el mismo espacio de mi corazón, a ellas por apoyarme estando lejos de mi familia, por regañarme y aconsejarme cuando no hacía las cosas bien.

A mi familia en pleno, mis primos, mi tía Ernestina, mi tía Hortensia que hoy no está entre nosotros, a José Ramón por ser un ejemplo para mí en este paso, a Mary y Alayn que en la distancia han estado a mi lado, a los amigos de mi madre que han estado en todo momento.

A mi tutora Mailin, por ser tan entregada en su labor de guía durante estos meses de trabajo y brindarme su ayuda y apoyo a toda costa.

A mi nueva tía Esperanza por hacerme parte de su vida, a María Elena por su alegría, cariño, por el apoyo, por los consejos, a Tamara (mi cliente preferido) que nunca me puso un reparo todas las veces que la necesite.

A mis amigos de Villa Clara, especialmente a Maivis, Yunier, Laura, Leidy, Leandry, Lelo, Regina, Emelina y sus familias.

Agradecimientos

A todos mis profesores que han aportado en mi preparación profesional y personal, a Rosita Tomas, Nieves, Emilia, Gladys, Miguel Ángel, los profes de la universidad, Rogelio Acosta y la profe Lombillo que me enseñaron a ser Alumno Ayudante de Matemática, a la profe Maritza por la confianza, a Dania mi profe guía por la dedicación, al profe Basulto por las enseñanzas, a Francisca Mercedes por enseñarme a hacer una mejor FEU y a ser responsable de mis actos, a mi oponente por sus comentarios atinados y su ayuda en todo momento, a mi tribunal.

A mis amigos de la universidad, Yusdel, Yuniór, Yordanis, Yidian con los que aprendí que la FEU es una gran familia; a Adrián, Diego, Yamile, Roxana, Ángel Manuel y Orlay, por la confianza; a mis nuevos amigos Yeni, Neysa, Niurka, Malidia, Dennys, a todos gracias por la incondicionalidad.

A quien se ganó el título de mi mejor amiga de la universidad “Bell”, por el apoyo, la confianza, por estar cuando todos se fueron, por los momentos compartidos y por darme la posibilidad de conocerla.

A todos los que en mi vida han aportado sus enseñanzas y me han hecho mejor persona.

El desarrollo científico y tecnológico alcanzado por la humanidad en la rama de la informática y las comunicaciones, ha permitido que las condiciones estén creadas para agilizar los procesos de gestión de información y documental. El presente trabajo aborda los principales aspectos que se desarrollaron con el fin de realizar un sistema de gestión para la agenda de trabajo del Consejo de dirección de la Facultad 4. En la investigación realizada se presenta un estudio de las principales herramientas y tecnologías utilizadas en el proceso de creación de la solución, así como los principales módulos que se implementaron para lograr las funcionalidades requeridas. Por otro lado se especifican las pruebas a las que fue sometida la solución elaborada, quedando demostrado que la misma cumple con las funcionalidades requeridas por el cliente. El resultado de esta investigación deja constancia documental de la metodología empleada, que permitió cumplir el objetivo general propuesto y la satisfacción de las necesidades del cliente, relacionadas con las insuficiencias expresadas por el mismo.

Palabras clave: agenda de trabajo, gestión de información, gestión documental, unidad documental.

Contenido

Introducción	1
Capítulo 1: Fundamentación Teórica	6
1.1. Introducción	6
1.1.1. Principales conceptos asociados al dominio del problema.....	6
1.1.2. Gestión de la Información	9
1.1.3. Gestión documental	9
1.1.4. Unidades documentales.....	10
1.2 Sistemas de gestión documental	11
1.2.1 Aplicaciones de los sistemas de gestión documental.....	11
1.2.2 Sistemas de gestión documental en el ámbito internacional.....	13
1.3 Tendencias y tecnologías actuales a considerar	15
1.3.1 Metodologías de desarrollo de software	16
1.3.2 Metodologías ágiles	17
1.4 Herramienta para el modelado de procesos de negocio.....	19
1.5 Sistemas de Gestión de Contenidos (CMS).....	20
1.5.1 Joomla.....	21
1.5.2 Drupal.....	22
1.5.3 Selección de CMS a emplear	23
1.6 Lenguaje de programación	23
1.6.1 Lenguajes y tecnologías del lado del cliente.....	24
1.6.2 Lenguaje del lado del servidor.....	26
1.7 Sistemas Gestores de Bases de Datos (SGBD).....	27
1.7.1 MySQL	28
1.7.2 PostgreSQL.....	28
1.7.3 SGBD seleccionado	29
1.8 Entorno de Desarrollo Integrado (IDE).....	30
1.8.1 Zend Studio	30
1.8.2 NetBeans 7.14	31
1.8.3 IDE seleccionado.....	31

1.9 Herramientas para el diseño de prototipos de interfaz de usuario	32
1.10 Servidor web.....	33
1.10.1 Internet Information Server	33
1.10.2 Apache 2.2.22.....	33
1.10.3 Servidor web seleccionado.....	34
1.11 Conclusiones parciales del capítulo	34
Capítulo 2: Propuesta de solución	35
2.1. Introducción	35
2.2. Descripción de los procesos vinculados al campo de acción.....	35
2.3. Objetos de informatización.....	36
2.4. Propuesta de solución	37
2.5. Personal relacionado con el sistema.....	37
2.6. Definición de contenidos y servicios.....	38
2.7. Modelo de dominio	39
2.7.1. Diagramas de modelado de negocio.....	39
2.8. Exploración	40
2.8.1 Historias de usuario.....	41
2.9. Planificación.....	42
2.9.1 Estimación de esfuerzo por HU.....	43
2.9.2 Plan de iteraciones.....	43
2.9.3. Plan de duración de las iteraciones.....	44
2.9.4. Plan de entregas.....	44
2.10. Diseño del sistema	44
2.10.1. Arquitectura de información.....	45
2.11. Tarjetas CRC.....	46
2.12. Conclusiones parciales del capítulo	47
Capítulo 3. Implementación y Pruebas.....	49
3.1. Introducción	49
3.1. Patrón arquitectónico de Drupal.....	49
3.2. Estructura de Drupal	50
3.2.1. Temas en Drupal.....	51

3.3. Creación de las tablas adicionales en la base de datos.....	53
3.4. Fase de implementación.....	53
3.5. Módulos en Drupal	53
3.5.1. Módulos empleados y desarrollados en la solución	54
3.5.2. Patrones de diseño	54
3.6. Pruebas de software	55
3.7. Resultados obtenidos.....	56
3.8. Conclusiones parciales del capítulo	57
Conclusiones generales.....	58
Recomendaciones	59
Glosario de términos	60
Referencias bibliográficas	63
Anexos.....	70
Anexo 1: Flujo de trabajo "Plan de temas" del proceso Agenda de trabajo del Consejo de dirección.....	70
Anexo 2: Historias de usuario del sistema GESTA.....	70
Anexo 3: Estimación de esfuerzo por HU.....	78
Anexo 4: Plan de duración de las iteraciones.	79
Anexo 5: Plan de entregas.....	81
Anexo 6: Tareas de implementación.....	82
Anexo 7: Pruebas unitarias.....	94
Anexo 8: Diseño de casos de prueba	95

Índice de ilustraciones

Índice de ilustraciones

Ilustración 1: Diagrama del flujo de trabajo Citación al Consejo de dirección.....	40
Ilustración 2: Prototipo de interfaz de usuario de Agenda de trabajo.....	45
Ilustración 3: Representación de la Arquitectura de la Información del sistema	46
Ilustración 4: MVC en Drupal (Butcher et. al 2013)	50
Ilustración 5: Estructura de Drupal (Butcher et. al 2013).....	51
Ilustración 6: Tema del sistema.	52
Ilustración 7: Flujo de trabajo "Plan de temas" del proceso Agenda de trabajo.....	70
Ilustración 8: Menú de acceso a la interfaz de prueba.	94
Ilustración 9: Pruebas implementadas en el módulo Rules.	94
Ilustración 10: Proceso de avance de la prueba.....	95

Índice de tablas

Tabla 1: Usuarios relacionados con el sistema y su descripción.....	37
Tabla 2. HU Registrar cuenta en el sistema.....	42
Tabla 3. Tarjeta CRC Módulo Citación.....	47
Tabla 4. HU Autenticar usuario en el sistema.	70
Tabla 5. HU Gestionar citación.	71
Tabla 6. HU Visualizar citación.	71
Tabla 7. HU Publicar informe y acta del Consejo de dirección.	72
Tabla 8. HU Mostrar informe y acta del Consejo de dirección.....	72
Tabla 9. HU Mostrar listado de citaciones al Consejo de dirección.	72
Tabla 10. HU Gestionar acta del Consejo de dirección.	73
Tabla 11. HU Eliminar citaciones.	73
Tabla 12. HU Filtrar búsqueda de usuarios y documentos por atributo.	74
Tabla 13. HU Mostrar listado de actas del Consejo de dirección.	74
Tabla 14. HU Eliminar informe y acta del Consejo de dirección.	75
Tabla 15. HU Ver listado de miembros del Consejo de dirección.....	75
Tabla 16. HU Administración de usuarios.	76
Tabla 17. HU Exportar a PDF acta e informe del Consejo de dirección.	76
Tabla 18. HU Imprimir acta e informe del Consejo de dirección.....	77
Tabla 19.HU Apariencia.....	77
Tabla 20. HU Seguridad.	77
Tabla 21. HU Condiciones tecnológicas.....	78
Tabla 22. HU Condiciones legales.	78
Tabla 23. Estimación de esfuerzo por HU.....	78
Tabla 24. Plan de duración de las iteraciones.....	79

Índice de tablas

Tabla 25. Plan de entregas.....	81
Tabla 26. Tareas de implementación (1era iteración).	82
Tabla 27. Tareas de implementación (2da iteración).....	87
Tabla 28. Tareas de implementación (3era iteración).	91
Tabla 29. Casos de prueba de aceptación (1era iteración).....	95
Tabla 30. Casos de prueba de aceptación (2da iteración).....	101
Tabla 31. Casos de prueba de aceptación (3era iteración).....	106

Introducción

La actual sociedad de la información y el conocimiento se caracteriza por la necesidad de identificar y utilizar el conocimiento y la información existente en las entidades para ponerlas en función de su misión, objetivos y desarrollo en general. Con el surgimiento de las Tecnologías de la Información y las Comunicaciones (TIC) y el acceso de las personas a estas, se comienza a sustituir los documentos en formato duro por los documentos electrónicos. La continua evolución y la rápida adopción por la sociedad de estas tecnologías, principalmente de las organizaciones, ofrece nuevas posibilidades de comunicación.

Este nuevo soporte trajo consigo el intercambio de información y que cada día se generen, reciban y almacenen documentos en formato digital en un número cada vez mayor de organizaciones. El crecimiento exponencial ha colocado a la Gestión Documental (GD) en un lugar privilegiado, volviéndose imprescindible para garantizar el uso adecuado y oportuno de la información.

La autora Ponjuán, luego de haber consultado al autor Nonaka define la GD, *“como un proceso que permite analizar y controlar sistemáticamente a lo largo del ciclo de vida la información registrada que crea, recibe, mantiene y utiliza una organización en correspondencia con su misión, objetivos y operaciones.”* (Ponjuán, 2005). Por lo que la GD, puede verse como un proceso para mantener la información en un formato que permita su acceso oportuno. Para ello, requiere tareas y procedimientos para cada fase y la explotación de esta información registrada, que es evidencia de las actividades y transacciones de las organizaciones, posibilitando lograr una mayor eficacia.

La mayoría de las organizaciones necesitan acceder y consultar de forma frecuente información archivada. En otros casos es la importancia de los documentos o el volumen de información lo que estimula a buscar nuevas soluciones innovadoras que ofrezcan ventajas y valor añadido sobre los sistemas tradicionales de archivo y almacenamiento. Por lo que cobra auge en la industria informática el paradigma de los sistemas para la gestión de documentos electrónicos cuyo progreso ha sido notable en el desarrollo de las tecnologías en la última década. Específicamente en Cuba las TIC han tenido un gran impacto en todas las esferas y una de ellas es el ámbito empresarial y de las

organizaciones en la cual ha tomado un papel protagónico en el uso de estas nuevas tecnologías.

Existe un grupo de instituciones encargadas de informatizar los diferentes procesos llevados a cabo en una empresa; un ejemplo de esta tendencia es la Universidad de las Ciencias Informáticas (UCI), que se ha integrado a la informatización de procesos tanto dentro como fuera de su campus universitario. Asociada a la universidad, se encuentran siete facultades, una de ellas la Facultad 4. Esta facultad está estructurada en un Consejo de dirección (C/D) formado por: un decano, cuatro vicedecanatos¹, un jefe de centro de desarrollo², una secretaría docente, cuatro jefes de departamento³. Lo integran además, los secretarios de la UJC⁴, de la CTC⁵, Núcleo del PCC⁶ y el presidente de la FEU⁷. Este órgano se encarga de gestionar los procesos docentes, investigativos y productivos de la facultad, teniendo en cuenta que está vinculada a un centro de educación superior con un modelo de formación que fomenta la investigación y la producción.

Con una frecuencia quincenal se efectúan los Consejos de dirección en la Facultad 4, por lo que se requiere llevar el control de los temas abordados en estas reuniones. Los temas a tratar, son aprobados, a principios de cada curso lectivo, generando un documento con las propuestas de temas para cada reunión.

La citación para un CD se envía por correo electrónico que contiene el orden del día, esta no siempre contiene las actividades que fueron planificadas, sino que pueden adicionarse otras que urgen por la necesidad y la vorágine con que se trabaja en la UCI y en la Facultad 4. Además, se debe llevar, por parte de la secretaria del Consejo de dirección y de forma manual, el control de las opiniones por cada tópico del Consejo así como el listado de los acuerdos tomados, para luego transcribir a formato digital, pudiéndose omitir opiniones y criterios de peso. A su vez, los miembros que deben rendir cuentas al CD deben previamente circular los informes por lo que cada vez que deseen consultarlos deben acceder a su correo electrónico, en ocasiones quedan pendientes algunos puntos

¹ Vicedecanato de Investigación y Postgrado, Vicedecanato de Extensión Universitaria, Vicedecanato de Formación y Vicedecanato de Administración.

² Centro de Tecnologías para la Formación (FORTES).

³ Departamento de Ciencias Básicas, Departamento de Humanidades, Departamento de Programación y Departamento de Ingeniería de Software.

⁴ Unión de Jóvenes Comunistas.

⁵ Central de Trabajadores de Cuba.

⁶ Partido Comunista de Cuba.

⁷ Federación de Estudiantes Universitarios.

de las Agendas de meses anteriores que pueden ser olvidados por los miembros del CD para la próxima reunión.

Teniendo en cuenta lo antes expuesto, se plantea el siguiente **problema a resolver**: ¿Cómo contribuir a elevar la gestión de las Agendas de trabajo del Consejo de dirección de la Facultad 4 y la información generada?

El **objetivo general**: Desarrollar un sistema para la gestión de Agendas de trabajo del Consejo de dirección de la Facultad 4.

Como **objeto de estudio** se definen los sistemas de gestión documental.

A partir de lo cual se define como **campo de acción**: sistemas de gestión documental para el control de la Agenda de trabajo del Consejo de dirección de la Facultad 4.

En correspondencia con el objetivo general de la investigación que se presenta se articulan los siguientes **objetivos específicos**:

- ✓ Formular el marco teórico de la investigación.
- ✓ Modelar los procesos relacionados con la gestión de la Agenda de trabajo del Consejo de dirección de la Facultad 4.
- ✓ Describir la propuesta del sistema.
- ✓ Implementar funcionalidades teniendo en cuenta la estructura de diseño y los requisitos definidos.
- ✓ Validar el funcionamiento del sistema mediante pruebas de software.

A partir de los presupuestos anteriores, se sostiene como **Hipótesis**:

Si se desarrolla un sistema de gestión para la agenda del Consejo de dirección de la Facultad 4 (GESTA), entonces se contribuirá a elevar la gestión de las agendas de trabajo del Consejo de dirección y la información generada.

Para la realización de esta investigación se utiliza la combinación de métodos teóricos y empíricos. Entre los primeros se emplean:

El método **Histórico- lógico** se utiliza para identificar los aspectos que definen la gestión documental, los sistemas de gestión de documentos electrónicos y los sistemas de gestión para elevar el control en las organizaciones, posibilitó la realización de un análisis bibliográfico del objeto de estudio y seleccionar los conceptos necesarios de la temática en cuestión que caracterizan el estado actual del fenómeno de investigación.

El método **Analítico- sintético** se utiliza para hacer un análisis y estudio del objeto de investigación, mediante la determinación de sus componentes o partes que lo integran, relacionarlos de manera que se pueda ver el comportamiento del objeto de estudio al integrar sus partes como un sistema, en torno a la gestión y el control de la Agenda de trabajo del Consejo de dirección de la Facultad 4.

El método **Hipotético- Deductivo** se utiliza para partiendo del problema descrito, plantear el objetivo e hipótesis y verificar los elementos planteados teóricamente.

El método de **Modelación** se emplea para analizar la realidad mediante diversos modelos y diagramas que permitan reflejar las necesidades del cliente logrando un mejor entendimiento y comprensión del problema a resolver, para ello se utiliza una herramienta de modelado que facilita el entendimiento de los procesos a implementar.

Dentro de los métodos empíricos se emplean:

El método **Observación participante** posibilita realizar el software de gestión, permitiendo ampliar y corroborar los elementos que se identificaron en la problemática y la factibilidad de la propuesta de solución.

El método **Análisis documental** posibilita analizar, clasificar, seleccionar los contenidos en la bibliografía de la investigación, referentes al proceso de gestión documental y cómo este puede ayudar a elevar el control de las actividades de las organizaciones, por lo que se recurrió a la revisión de documentos, la búsqueda bibliográfica, teorías y análisis de criterios de autores.

El presente documento está estructurado en tres capítulos:

Capítulo 1: Fundamentación teórica.

En este capítulo se hace referencia a los elementos teóricos que soportan la investigación. Además se realiza un estudio del arte de la evolución y situación actual de las herramientas cuyo objetivo se centra en la gestión documental para las organizaciones. Se describen soluciones similares existentes y se realizan comparaciones para llegar a la propuesta de solución que sustenta la investigación. Se presentan los lenguajes de programación y modelado, así como los estándares y tecnologías que se utilizarán para desarrollar la propuesta planteada fundamentando su selección en base al estudio realizado.

Capítulo II: Propuesta de solución.

En este capítulo se realiza una descripción de las características del sistema a implementar. Se especifican los requisitos funcionales y no funcionales que debe cumplir la aplicación, descritos en Historias de usuario. También se muestra el modelo de dominio y la descripción de los principales objetos de informatización durante la construcción del sistema. Se brinda una propuesta de los contenidos y servicios por los que estará compuesto el sistema una vez concluido el proceso de implementación.

Capítulo III: Implementación y pruebas.

Este capítulo presenta lo relacionado con el proceso de implementación de las funcionalidades identificadas, sobre la base del lenguaje de programación y metodología seleccionadas. Se especifica la arquitectura seleccionada, así como los patrones de diseño e implementación aplicados. Se definen los tipos de pruebas que se aplicarán al sistema y los resultados que arrojan las mismas luego de su aplicación.

Capítulo 1: Fundamentación Teórica

Capítulo 1: Fundamentación Teórica

1.1. Introducción

En este capítulo se exponen los elementos teóricos y conceptuales necesarios para la implementación de un sistema de gestión que permita elevar el control de las actividades de la Agenda de trabajo del Consejo de dirección de la Facultad 4. Con el objetivo de favorecer una mejor comprensión del proceso de gestión y de cada uno de los elementos que lo componen se realiza un estudio de estos referentes, lo que conlleva a identificar las principales fuentes bibliográficas y áreas del conocimiento que engloba al objeto de estudio y campo de acción.

Además, se realiza la fundamentación y concepción que debe guiar la implementación de un sistema de gestión para elevar el control de las actividades del Consejo de dirección de la Facultad 4.

1.1.1. Principales conceptos asociados al dominio del problema

Para el desarrollo y mayor entendimiento de la presente investigación, es necesario puntualizar algunos conceptos fundamentales que son referidos al marco metodológico.

El autor Rodríguez refiere que proceso aplicado a una empresa u organización, "*es un conjunto de actividades interrelacionadas, que persiguen la creación de valor y que su salida final es la conformación de un bien o servicio para un cliente que puede ser interno o externo a la organización*" (Rodríguez, 2005).

Independientemente del tipo de proceso, controlar la calidad es uno de los aspectos a tener en cuenta, puesto que un error que se cometa en cualquiera de los procesos o subprocesos puede comprometer el resultado final y con ello la calidad de este y a su vez el nivel de satisfacción del cliente. En el desarrollo de la presente investigación se constató la existencia de una amplia gama de criterios emitidos sobre la calidad del proceso, antes esta divergencia se asume la valoración del autor Latino al definir la calidad del proceso como "*la acción de medir variables clave en el proceso para detectar cualquier variación inaceptable*" (Latino, 2011). A su vez, cada proceso concluye con la adquisición de conocimiento por parte de los implicados.

No se puede hablar de ciencia y tecnología sin mencionar el conocimiento. El autor

Capítulo 1: Fundamentación Teórica

Vygotsky, considera que “*el conocimiento no es un objeto que se pasa de uno a otro, sino que es algo que se construye por medio de operaciones y habilidades cognoscitivas que se inducen en la interacción social*” (Vygotsky, 1985). Sin embargo, el investigador Díaz Balart, refirió que “*el conocimiento es imprescindible para el desarrollo de una organización; el escenario actual de las organizaciones que hacen ciencia se marca por un ciclo de vida corto de los productos, mercados globales, competidores desconocidos, necesidades cambiantes de los clientes, entornos dinámicos, tecnología compleja y con compromisos competitivos*” (Díaz Balart, 2002). Por otra parte, investigadores como Alavi y Sánchez coinciden en cuanto al conocimiento, “*como la capacidad humana para gestionar información, para solucionar problemas y llegar a resultados*” (Alavi, et al, 2001; Sánchez, et al. 2006). El autor de la presente investigación coincide con los autores antes citados y considera que el concepto que guiará este trabajo es el referido por Díaz Balart. De lo anterior, se puede concluir que el conocimiento es un recurso que tiene cada persona, que se puede transferir y tiene como base el uso de la información para solucionar problemas y estimular la obtención de resultados. El empleo del conocimiento en las organizaciones logra transformarlas y aportarles valor agregado.

Algunos investigadores suelen clasificar el conocimiento. Autores como Ruesta, Ortega, Welsh, Lanzillotta, toman de base el modelo matricial de conversión o transferencia del conocimiento (SECI⁸) de Nonaka confluendo en la teoría “*que el conocimiento pasa por un proceso que lo transforma de tácito (contenidos en los sistemas de información, en las bases de datos y en las personas) a explícitos (capturados y almacenados en un formato reutilizable que permite realizar búsquedas) y otra vez en tácitos, lo cual permite que otras personas de la organización puedan aprenderlos y utilizarlos*” (Ruesta et al. 2008). Entonces se puede concluir que el conocimiento se clasifica en: tácito y explícito. A su vez, cuenta de dos elementos importantes: la información y los datos. De las clasificaciones del conocimiento conocidas, en este trabajo sólo se definirá el conocimiento explícito, puesto que el autor considera que existe una estrecha relación entre este tipo de conocimiento y la información, un grupo de investigadores encabezado por Nonaka define el conocimiento explícito, como “*conocimiento que está codificado y que es transmisible a través de algún sistema de lenguaje formal. Dentro de esta*

⁸**Socialización:** De tácito a tácito; **Combinación:** De explícito a explícito;
Exteriorización: De tácito a explícito; **Interiorización:** De explícito a tácito.

Capítulo 1: Fundamentación Teórica

categoría se encuentran los documentos, reportes, memos, mensajes, presentaciones, diseños, especificaciones, experiencias, simulaciones, entre otras” (Nonaka et al. 1999).

De lo antes expuesto, se puede inferir que el conocimiento no es algo estático sino dinámico, que se desarrolla y se transforma, cada vez que las personas tienen un nuevo intercambio entre ellos y los sujetos con los que interactúan. Con frecuencia, en las organizaciones el conocimiento no solo está disponible en documentos o bases de datos, sino también en las rutinas, procesos, prácticas y normas de la organización. De ahí que existan posiciones en relación a plantear un nuevo enfoque que se refiere a la Gestión del Conocimiento (GC). Para el autor González, la GC es *“un proceso sistemático de evaluar y diagnosticar la organización, planear, adquirir, socializar, almacenar, generalizar y aplicar los conocimientos a todos los niveles de la organización, basado en las tecnologías” (González, et al. 2009).*

En el contexto de las organizaciones un grupo de especialistas del CITMA⁹, reconoce que la GC *“tiene un nuevo enfoque gerencial que reconoce y utiliza el recurso humano y el conocimiento que este posee y aporta a la organización con el uso apropiado e intensivo de las tecnologías. Que se vincula eficientemente a la gestión de la información, es el conjunto de procesos y herramientas que permiten la integración sistémica de acciones para el aprovechamiento y utilización del conocimiento, la información y la experiencia acumulada en el desarrollo cualitativo de una organización” (CITMA, 2002).*

El autor de la investigación que se presenta teniendo en cuenta lo antes expuesto, coincide en que la información con que se cuenta por parte de los miembros de las organizaciones, es el principal elemento de la gestión del conocimiento. La gestión del conocimiento es un proceso organizado, dinámico, continuo, encaminado a aumentar el aprendizaje de las personas, comprende una transformación continua de datos en informaciones, esta información en conocimientos que generan más conocimiento. El objetivo de la gestión del conocimiento es que el conocimiento se pueda utilizar, compartir y desarrollar en una organización y por parte de los individuos que en ella trabajan, con el fin de poder cumplir sus objetivos, se produzcan mejores resultados y exista mejor eficiencia.

De la literatura consultada **(Stair, 2000)** y **(Martinrey, 2004)**, la mayoría de los autores confluyen en que la información se define como: *“...un conjunto de datos que poseen un*

⁹ Ministerio de Ciencia, Tecnología y Medio Ambiente de la República de Cuba.

Capítulo 1: Fundamentación Teórica

significado, de modo tal que reducen la incertidumbre y aumentan el conocimiento de quién se acerca a contemplarlos. Estos datos se encuentran disponibles para su uso inmediato y están estrictamente ligados a la toma de decisiones” (Ferrell, 2004).

Agrupando estos conceptos, se puede concluir que la información es un conocimiento explícito extraído por las personas o sistemas expertos como resultado de la interacción con el entorno con el que se relacionan. Esta información a diferencia de los datos, tiene una estructura útil que modifica las interacciones que puedan ocurrir entre quien posee dicha información, cómo la aplique y su entorno.

1.1.2. Gestión de la Información

La Gestión de la Información (GI) surge cuando los sistemas informáticos empezaron a ser comunes en las organizaciones. Actualmente, la manera en que se gestiona la información y los sistemas que de esta tarea se encargan, han ido variando con el tiempo. Varios autores han definido la gestión de la información, para esta investigación se referencia la ofrecida por los autores Emma Rodríguez, Belina Capote y Diego González cuando afirman que, *“la gestión de la información es el proceso que se encarga de gestionar la información necesaria para la toma de decisiones y un mejor funcionamiento de los procesos, productos y servicios de la organización” (Capote et al. 2003).*

La Gestión de la Información y su aplicación tienen en cuenta la obtención de la información, su transformación, el costo y que sea empleada por las personas indicadas dentro de una organización. Por lo que a modo de resumen, puede entenderse la gestión de información, como un proceso continuo y permanente que reúne, organiza, coordina y controla entre otros los procesos y servicios de información de una organización, para responder de manera pertinente a las necesidades de los usuarios. Esta gestión involucra la captura, organización, recuperación, publicación y distribución de la información para percibir y generar el conocimiento. A su vez se debe cuidar de la información su integridad, su disponibilidad y la confidencialidad.

1.1.3. Gestión documental

Herrera refirió que la Gestión Documental (GD), *“es un compuesto de normas, técnicas y prácticas usadas para administrar el flujo de documentos que se genera en una organización, permitir recuperar dicha información desde ellos, determinar el tiempo que los documentos deben permanecer archivados o eliminar los que de cierta forma ya han perdido validez asegurando así mismo la conservación indefinida de los documentos más*

Capítulo 1: Fundamentación Teórica

valiosos” (Herrera, 2008).

En la bibliografía consultada se coincide en el uso de la tecnología y procedimientos que permiten la gestión y el acceso unificado a la información generada en las organizaciones, por parte de los miembros o clientes que la conforman. Dentro de los beneficios que son de interés para la presente investigación se evidencian:

- Se establece un nuevo espacio de trabajo compartido entre los miembros de la organización donde la información es compartida y aprovechada de manera más eficiente y como un recurso colectivo.
- Se accede de inmediato a la totalidad de la información necesaria para alguna actividad específica, con la ventaja de reducción del tiempo de consultas, tareas de archivo y ahorro de espacio físico.
- Se aumenta el valor de la información y se evita la duplicación de tareas, así como los tiempos de búsqueda de información interna.
- Se reducen drásticamente situaciones como la duplicidad de documentos archivados, dobles grabaciones, el abuso de tecnologías como el correo electrónico, entre otros.

1.1.4. Unidades documentales

El autor Galende define las unidades documentales como, *“una unidad de análisis en los procesos de identificación y caracterización documental. Pueden ser simples, cuando está constituida por un solo tipo documental, o compleja, cuando la constituyen varios, formando un expediente (Galende et al. 2003).*

Galende Díaz presenta estudios de cómo los documentos surgieron para plasmar las funciones y actividades de las organizaciones y las personas, y de cómo dan origen a las unidades documentales simples (carta, un informe, memoria, registro) y complejas (expedientes). En esta investigación se profundiza en las unidades documentales complejas que, según Galende Díaz *“son un conjunto ordenado de documentos (...) que materializan actuaciones secuenciales y procedimientos de la organización encaminadas a la resolución de un asunto, estos a su vez cuentan con una autonomía” (Galende et al. 2003).*

Como consecuencia al auge de las TIC, la definición de unidades documentales y de los documentos adquirió una nueva dimensión. Los documentos actualmente suelen ser

Capítulo 1: Fundamentación Teórica

almacenados y procesados en formato digital; no como objetos físicos. De ahí que emerge la definición de documento electrónico, para el autor BJÖRK “*es un contenedor de información en formato digital, que recoge información simultáneamente de varias fuentes, en torno a específicos temas para encontrar las necesidades de un individuo en particular*” (Bjork, 2002). Es sabido que la evolución tecnológica ha dado lugar a que se cuenten con herramientas informáticas que ayuden a la optimización de la gestión de los documentos.

1.2 Sistemas de gestión documental

Para el autor Freire los Sistemas de gestión documental (SGD) son, “*programas de gestión de bases de datos que disponen de una tecnología idónea para el tratamiento de documentos*” (Freire, 2014). Entre sus ventajas un software, debe lograr la reducción del costo de una organización para almacenar y manipular cierta documentación. Al implementarlos se ahorra un tiempo valioso debido a que los miembros de la organización ya no tienen que valerse de esa misma cantidad de tiempo en la clasificación de archivos y recuperación de documentos. Para Rendón un SGD “*es una parte del sistema de información de la empresa desarrollado para almacenar y recuperar documentos, que debe estar diseñado para coordinar y controlar todas aquellas funciones y actividades específicas que afectan a la creación, recepción, almacenamiento, acceso y preservación de los documentos, salvaguardando sus características estructurales, y contextuales, y garantizando su autenticidad y veracidad*” (Rendón, 2015). A partir del análisis anterior, el autor del estudio que se presenta coincide con los autores Freire y Bustelo y a modo de conclusiones expone que, un sistema de gestión documental es un programa informático creado con el fin de optimizar los procesos de creación, modificación y eliminación de documentos procurando que esta gestión esté guiada a mantener el control de los mismos.

1.2.1 Aplicaciones de los sistemas de gestión documental

Dado que la presente investigación hace referencia a los sistemas de gestión documental, se mencionarán algunos ejemplos reales de los mismos, lo que demuestra la existencia de multitud de dominios de aplicación. Cabe destacar que los más significativos a escala nacional e internacional se estudian en este trabajo para el diseño del sistema de gestión de esta investigación.

Se pueden citar: *Windows Personal Librarian (CSI)*, orientado al uso de bases de datos de

Capítulo 1: Fundamentación Teórica

texto completo y permite el acceso a imágenes; *InvesDoc*, que se integra con bases de datos relacionales y documentales lo que hace que sea adaptable a cualquier necesidad en el campo de la Gestión documental; *Documentum* (*Enterprise Content Management*, ECM por sus siglas en inglés, gestor de contenido empresarial), es una plataforma para la gestión de contenidos de aplicaciones empresariales empleado fundamentalmente en procesos de facturas, gestión de expedientes, gestión de contenidos web y gestión de patrimonio histórico. Por su parte, NUXEO, es propietario y permite gestionar documentos realizando versiones, flujos de trabajo asociados a documentos, publicación remota o búsqueda avanzada y suele ser empleado para la gestión documental de páginas web, registros, imágenes y contenidos.

En Cuba el desarrollo de los sistemas de gestión documental, se ha extendido. Los sistemas de gestión documental se han empleado en las más disímiles tareas, el ejemplo más conocido es Alfresco, que permite la creación de un repositorio de archivos y contenidos, almacenando toda la información en un mismo repositorio. Es una alternativa de código abierto para la gestión de contenidos empresariales (*Enterprise Content Management*, ECM por sus siglas en inglés). En la UCI se utiliza Alfresco en su versión *Community*¹⁰. Está desarrollado en lenguaje de programación Java, basado en estándares abiertos y de escala empresarial para sistemas operativos Windows, Unix Solaris y algunas versiones de Linux. Al consultarse el portal web Nuevas Tecnologías para PyMes el autor Shariff, destaca algunas de las características de Alfresco, “*cuenta con una interfaz familiar para la gestión de documentos lo que garantiza que el nivel de aprendizaje y facilidad de interacción del usuario sea fácil y posee un eficiente control de versiones de los documentos manipulados, por solo citar dos elementos importantes*” (Shariff, 2013).

Babel, al decir de los autores Baéz, Rodríguez y Cepero es un sistema automatizado para el control y gestión de las solicitudes de los servicios propios de la Unidad de Traducción del Centro de Información de ETECSA¹¹, desarrollado por la Facultad de Ingeniería de la Universidad Central “Marta Abreu” de las Villas. Los autores refieren que el sistema, “*permite organizar, clasificar y almacenar todos los documentos traducidos, posibilitando la gestión de los mismos, con el objetivo de facilitar el acceso a la información de los usuarios, así como la utilización eficiente y segura de estos documentos*” (Baez et al.

¹⁰ Es software libre, con licencia GPL de código abierto y estándares abiertos.

¹¹ Empresa de Telecomunicaciones de Cuba.

Capítulo 1: Fundamentación Teórica

2005).

AvilaDoc es otro sistema de gestión de documentos electrónicos, desarrollado sobre una plataforma de software libre, *“con una base de datos centralizada, destinado a la gestión, tramitación y resguardo de archivos electrónicos”* (Badillo et al. 2006). Es una herramienta que gestiona *“el tramitado interno de los documentos sin necesidad de copias, o envíos físicos, permite el control de la entrada y salida de documentos, el manejo de las anotaciones a los documentos por usuarios, así como la recuperación de la información a través del contenido de la ficha del documento”* (Badillo et al. 2006).

En la bibliografía consultada otro número de trabajos e investigaciones que se han estudiado proceden de la UCI. Uno de ellos se desarrolló en el año 2007 y consistía en un sistema de Gestión documental del Sistema de Gestión Penitenciaria. Según sus autores tuvo como objetivo, *“el desarrollo de procedimientos y herramientas para sustentar la gestión documental penitenciaria brindándole al cliente del proyecto SIGEP de la UCI un sistema con información bien organizada y disponible que respondiera a sus necesidades de la forma más rápida y eficiente posible”* (Viera et al. 2007). En el año 2011, se implementa un sistema para la gestión de documentos electrónicos al área de Calidad del Centro de Tecnologías para la Formación (FORTES) de la Facultad 4, que tuvo como objetivo, *“gestionar los documentos que se manejan en dicha área del Departamento de Producción Audiovisual, producto a que en aquel momento se dificultaba la interacción y el control de dichos documentos”* (Navarro et al. 2011).

Más reciente en el tiempo se implementó eXcriba, un software para la gestión de los documentos administrativos, que teniendo en cuenta las funcionalidades de Alfresco desarrolla su núcleo. El software eXcriba, permite la automatización de los flujos documentales de una entidad, la gestión de documentos, carpetas, control de versiones, control de acceso, permisos y notificaciones.

1.2.2 Sistemas de gestión documental en el ámbito internacional

Luego del estudio de los sistemas de gestión documental disponibles en la web, se llega a la conclusión que internacionalmente existen sistemas de gestión similares al que se pretende alcanzar y se obtuvo una visión de cómo manejar algunos de los procesos que se encuentran inmersos dentro de la estructura de los sistemas de gestión documental, lo que facilita el trabajo a la hora de tomar ideas para la construcción de la propuesta de solución. Algunos ejemplos existentes que han servido de base para estos estudios que

Capítulo 1: Fundamentación Teórica

se presentan son:

- ✓ **Orfeo** (disponible en: <http://orfeogpl.org/ata/>), fue desarrollado por la Superintendencia de Servicios Públicos Domiciliarios (SSPD) en Colombia y licenciado como software bajo licencia GNU's hoy *Unix* (GNU, en español GNU no es Unix)/ *General Public License*(GPL, en español Licencia Pública General) para compartir el conocimiento y mantener la creación colectiva libre. Este sistema se caracteriza por una interfaz gráfica fácil de usar; la gestión sobre el documento comprende la reasignación, modificación y archivo; agiliza las búsquedas de documentos por diversos campos de búsqueda y el manejo de formatos varios.
- ✓ **Quixpux** (disponible en: <http://www.gestiondocumental.gob.ec/>), este sistema fue modificado a partir del sistema de gestión documental Orfeo, por lo que emplea tecnologías y estándares abiertos. Dentro de las funcionalidades que lo distinguen están: permite la creación, envío, recepción, almacenamiento y clasificación de documentos, circulares y anexos; realiza búsquedas, recuperación y presentación de documentos; organiza y clasifica los documentos digitales en carpetas o expedientes virtuales; permite la creación compartida de documentos, entre otras.

Atendiendo a los sistemas de gestión documental estudiados, Orfeo y Quixpux no presentan en sus funcionalidades el envío de notificaciones cuando ocurren modificaciones a los usuarios afectados por la misma, no logran administrar los permisos de acceso y modificación de documentos. *Documentum*, es un gestor que aunque permite disímiles funcionalidades en la gestión de documentos es una herramienta propietaria. Por su parte, NUXEO, aunque soporta el protocolo *WebDAV* para acceder a los repositorios, de contar con una arquitectura basada en estándares y que garantiza la seguridad gracias a la implementación de estándares SSL¹² y *Single Sign On* (SSO), no reconoce el concepto de rol y emplea usuarios virtuales con escasos permisos. Con respecto a los sistemas existentes en Cuba y que son citados en la presente tesis, Babel no se emplea puesto que solo se encarga de la gestión de documentos y no realiza otras funcionalidades que pueden aportarle valor agregado al software que se desea para darle cumplimiento al objetivo general propuesto. AvilaDoc implementa algunas funcionalidades necesarias para la solución que se propone, sin embargo, carece de notificaciones por correo electrónico y de un motor que incluya diversos criterios de búsqueda. Por otra

¹²Son protocolos criptográficos que se proporcionan comunicaciones seguras a través de una red.

Capítulo 1: Fundamentación Teórica

parte, aunque es ampliamente conocido en la UCI, extendido su uso y posee una serie de características que cumplen con los requisitos básicos del proceso en cuestión, no se selecciona Alfresco ya que tiene doble licencia: *Alfresco Community* y *Alfresco Enterprise*, ambas son versiones de código abierto, aunque la versión *Enterprise*, diseñada para importantes volúmenes de trabajo, es de pago. La versión *Community* no tiene soporte del fabricante, tampoco sobre el código fuente ya que es más una versión de pruebas y nuevas características que la comunidad de usuarios ayuda a probar y depurar. A su vez, Alfresco está más orientado a ser un administrador de documentos, aunque por medio de programación se pueden extender sus funcionalidades.

Como conclusión del estudio de las aplicaciones anteriores se hace necesario implementar una herramienta informática que contenga entre otras, las siguientes características: centralización de la información de importancia para los miembros del Consejo de dirección de la Facultad 4 referente a los documentos que se generan en su ámbito, facilidad para el manejo de documentos y datos independientemente del formato que le da el creador a los documentos, llevar un control de acceso, distribución y modificación de documentos, que los documentos sean accesibles por los usuarios autorizados, así como la edición de los documentos, definición de roles y permisos de usuarios, uso de notificaciones cuando se cree una citación para el Consejo de dirección, ampliar los criterios de búsqueda de documentos o usuarios en el sistema ampliándose los patrones de búsqueda. A su vez, todos los sistemas analizados tienen en común que son desarrollados como plataformas web.

1.3 Tendencias y tecnologías actuales a considerar

Al no contar con un sistema capaz de solucionar la situación problemática existente, es preciso desarrollar una nueva solución y para garantizar un proceso de desarrollo de software satisfactorio se procede a efectuar un estudio de las metodologías de desarrollo a emplear, herramientas y tecnologías actuales, para identificar las que serán empleadas en la solución. En el caso de una aplicación web, para completar una implementación eficiente existen básicamente, según el autor Morales, dos estrategias: “1) *la utilización de un framework server-side scripting*, y 2) *la utilización de un Sistema de Gestión de Contenidos (en lo adelante CMS, por sus siglas en inglés Content Management System)*” (Morales et al. 2005). El estudio que se muestra a continuación está sujeto a la segunda estrategia, puesto que un CMS es factible para el manejo de permisos y usuarios,

Capítulo 1: Fundamentación Teórica

definiéndose los niveles de acceso, reúnen un grupo de funcionalidades que fueron previamente desarrolladas y solo deben ser personalizadas a los requerimientos del cliente y cuentan con otras funcionalidades que pueden emplearse como base arquitectónica para desarrollar otras. Al emplear un CMS, el manejo de presentación se controla mediante plantillas fácilmente personalizables. Mientras que con un *framework* se hace necesario construir componentes para tales funciones.

1.3.1 Metodologías de desarrollo de software

Todo producto de software ha sido diseñado e implementado siguiendo una metodología de desarrollo de software. Avison y Fitzgerald presentan una definición de metodología de desarrollo vista como “una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información”. (Avison et al. 1995). En la presente investigación, para seleccionar qué metodología es la más adecuada en el desarrollo del proceso de construcción de software se realizó un estudio enfocado en los dos grandes grupos de metodologías existentes: ágiles y tradicionales (Joskowicz, 2008) y (Letelier et al. 2008).

Entre las metodologías tradicionales, también denominadas clásicas, pesadas o robustas se encuentran: MSF (por sus siglas en inglés, *Microsoft Solution Framework*), RUP (por sus siglas en inglés, *Rational Unified Process*), y sus híbridos. Por otra parte, el investigador (Suárez, 2009) propone metodologías de desarrollo de software centradas específicamente en el desarrollo *web*, entre las que se encuentran: *Navigational Development Techniques*(NDT), *Ubiquitous Web Applications* (UWA), *Object Oriented Hypermedia Design Model* (OOHDM) y otras actualmente consideradas obsoletas.

Teniendo en cuenta los planteamientos anteriores, el autor de la presente tesis coincide en la robustez de las metodologías tradicionales. Sin embargo, estas metodologías no resultan ser el enfoque más adecuado para proyectos donde el entorno es muy cambiante, y en los que se exige reducir drásticamente los tiempos de desarrollo pero manteniendo la calidad del software y en los que se tiene constante interacción con el cliente. Por otra parte las metodologías ágiles cumplen el rol de ser adaptables a los cambios que puedan suceder a lo largo del proceso de producción de software, en lugar de ser predecibles, ya que se encuentran especialmente orientadas a proyectos pequeños y constituyen una solución a medida para ese entorno, aportando una elevada

Capítulo 1: Fundamentación Teórica

simplificación, que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto.

Las tradicionales se caracterizan por ser muy rígidas y dirigidas por la documentación que generan en cada una de las actividades que desarrollan; mientras que las ágiles intentan ser procesos que se adaptan y crecen con el cambio. Estas últimas se encuentran, en gran medida, orientadas a las personas y no tanto al proceso; y por su parte, las tradicionales, tienden a intentar planear gran parte del proceso de software en gran detalle para un plazo de largo tiempo, con el fin de definir un proceso que debe funcionar bien independientemente de quien lo utilice.

Luego de realizado el estudio anterior, el autor de la presente investigación selecciona las metodologías ágiles como las que se adaptan para el desarrollo de la solución de la investigación, puesto que el ambiente bajo el cual se trabajará se adecúa a las características de las mencionadas metodologías. Para ello se pueden consultar los estudios comparativos presentado por los autores **(Letelier et al. 2008)** y **(Khrantchenko, 2004)**. A continuación se selecciona, dentro de las metodologías ágiles, la que mejor se adapta al desarrollo del presente trabajo.

1.3.2 Metodologías ágiles

Las metodologías ágiles están especialmente definidas para desarrollar proyectos con requisitos poco definidos o cambiantes, aplicables a equipos pequeños de trabajo. Dentro de las metodologías ágiles se incluyen, según el autor Letelier, “*Extreme Programming (XP), SCRUM, Crystal Methodologies, Xbreed, Adaptive Software Development(ASD), Dynamic Systems Development Method(DSDM), entre otras*” **(Letelier et al. 2008)**.

Las metodologías ágiles cambian significativamente algunos de los énfasis de los métodos ingenieriles. La diferencia inmediata es, refiere el autor Canós, que son menos orientadas al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada. De muchas maneras son más bien orientadas al código, siguiendo un camino que dice que la parte más importante de la documentación es el código fuente. Estas metodologías ofrecen una buena solución para proyectos donde el entorno es inestable y los requisitos no se conocen con exactitud, ya que están pensados para trabajar con incertidumbre.

Para el desarrollo de la propuesta de solución se tuvo en cuenta que el período de desarrollo es corto, puesto que la implementación de la solución se limita solamente a

Capítulo 1: Fundamentación Teórica

cuatro meses de trabajo continuo y las dimensiones del proyecto son pequeñas. Además se consultó el libro *Agile software development ecosystems* publicado en el 2002 por el autor Highsmith, donde se muestran indicadores en correspondencia con el nivel de agilidad que presentan cada una de las metodologías ágiles estudiadas. En el texto se le asigna un valor entre 1 y 5 de acuerdo al nivel de satisfacción para cada indicador donde el valor 5 funge como máxima calificación. De este estudio se evidencian las metodologías ASD y XP como las que más alto tienen el nivel de agilidad, la consulta hecha fue otro de los criterios seguidos para entre ambas seleccionar a XP debido que las características que define se pueden aplicar en concordancia al ambiente en que se implementará el producto final.

1.3.2.1 Extreme Programming (XP)

La Programación Extrema o *Extreme Programming* (XP) es una metodología ágil formulada por *Kent Beck* en 1996 a partir de un “conjunto de principios, prácticas y técnicas que facilitan de manera exitosa la finalización de proyectos, que surgen como respuesta y posible solución a los problemas derivados del constante cambio en los requerimientos” (**Beck, 1999**). En una investigación conjunta los autores Shore y Warden afirman que, “XP no es un conjunto de reglas a seguir, sino más bien un gran abanico de principios, técnicas y prácticas, con suficientemente eficiencia probada, que definen una forma de trabajar en armonía con los valores personales y organizacionales, y teniendo su punto de partida en cinco valores fundamentales que debe cumplir el equipo de desarrollo” (**Shore et al. 2008**).

XP cuenta con un ciclo de vida estructurado, resumiéndose en cuatro fases fundamentales según (**Joskowicz, 2008**). Este ciclo de vida de la metodología se organiza en: fase de exploración, fase de planificación, fase de producción y fase de mantenimiento (**Riola, 2008**). La fase de exploración, en la que el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Además se elaboran las Historias de usuario (HU) que no son más que las especificaciones del cliente de lo que el sistema debe realizar. La segunda es la fase de planificación donde se aprueban las Historias de usuario y se acuerda el alcance del proyecto. Los desarrolladores estiman el esfuerzo a dedicar a cada HU y se define el cronograma de ejecución. Se precisan las iteraciones de desarrollo y las pruebas para iteración por parte del cliente. La tercera es la fase de producción, mediante la cual se

Capítulo 1: Fundamentación Teórica

ejecutan las iteraciones elaboradas en la fase anterior, presentando al cliente avances del desarrollo de forma constante, logrando la retroalimentación entre cliente y desarrolladores. Una de las condiciones para ejecutar esta fase es la disponibilidad total del cliente. La cuarta es la fase de mantenimiento, la cual requiere un mayor esfuerzo para satisfacer las tareas del cliente por parte de los desarrolladores. Esta fase puede requerir cambios en la arquitectura del sistema.

Luego del análisis realizado se puede agregar que dado que el presente proyecto es pequeño, se cuenta con un solo desarrollador, frecuente comunicación entre el cliente y el desarrollador y la disponibilidad del cliente cada vez que se le requiere. Además, la producción de pequeñas entregas funcionales para mostrar un resultado rápido, integración continua, empleo de estándares de codificación y horario de trabajo de 40 horas por semanas sin trabajar horas extras; son características que se ajustan perfectamente a las condiciones reales del presente proyecto y hacen de XP la metodología de desarrollo de software a emplear.

1.4 Herramienta para el modelado de procesos de negocio

Además de la aproximación metodológica, uno de los factores determinantes en el éxito de un proyecto de software, lo constituye la acertada selección de las herramientas de trabajo que se emplearán durante su desarrollo, es por ello que en la actualidad la administración de proyectos atiende cuidadosamente la designación de herramientas como una tarea de vital importancia.

De los antes expuestos existen una amplia gama de herramientas para el modelado de los procesos del negocio como: *Magic Draw*, es una herramienta compatible con el estándar UML 2.3, multiplataforma, con desarrollo de código para varios lenguajes (Java, C++, entre otros) y su licencia es propietaria; *Eclipse Process Framework (EPF Composer)*, es una herramienta usada por ingenieros de procesos, líderes y administradores de negocios, se distribuye de forma gratuita por la licencia *Eclipse EPF Composer*, como herramienta se provee capacidades de ingeniería de procesos para la selección, adecuación y rápido ensamblado de procesos, sin embargo su conocimiento y uso no está extendido en la UCI; *STAR UML* y *Visual Paradigm* son herramientas de modelado de procesos de software basadas en el Lenguaje de Modelado Unificado (UML) al no emplearse este lenguaje en la propuesta de solución no son las más indicada para modelar los procesos de negocio de la solución y *Bigazi Process Modeler* que es la

Capítulo 1: Fundamentación Teórica

seleccionada.

Bigazi Process Modeler 2.6

Bigazi Process Modeler es un software gratis utilizado para diagramar, documentar y simular procesos usando la notación estándar BPMN (*Business Process Modeling Notation*). Los diagramas que generan se pueden exportar como archivos de imagen, o archivos de Visio, aunque se pueden publicar en Word, PDF, Web y Wiki. Como herramienta para el modelado de procesos de negocio permite esquematizar las actividades y decisiones que se toman en el negocio.

En la bibliografía consultada (**Bigazi.BPMN, 2011**) y (**Gjoni, 2015**), se refiere que, una vez concluida la representación del flujo de trabajo, la aplicación puede documentar los proyectos de forma automática partiendo de la información que se haya incluido en los esquemas.

1.5 Sistemas de Gestión de Contenidos (CMS)

El análisis de la bibliografía refiere a los sistemas de gestión de contenidos (*Content Management System* en inglés, abreviado CMS) según el autor Serrano, como: “...un software que se utiliza principalmente para facilitar la gestión de la web” (**Serrano, 2007**). Aunque esta definición ha evolucionado en los últimos años, el autor Gil agrega que, “el CMS controla también quién puede editar y visualizar los contenidos, convirtiéndose en una herramienta de gestión integral para la publicación de aplicaciones web” (**Gil, 2011**). Generalmente un CMS es una aplicación con una base de datos asociada en la que se almacenan contenidos, separados de los estilos o diseño. El autor de la presente investigación coincide con ambos investigadores y concluye que son herramientas que permiten crear y mantener una web con facilidad, encargándose de los trabajos más complejos que hasta ahora ocupaban el tiempo de los administradores.

Se selecciona un CMS para el desarrollo de las funcionalidades del sistema, puesto que provee una arquitectura con funcionalidades tales como: autenticación, manejo de roles, reutilización de objetos o componentes, páginas interactivas, consistencia de la web, módulos y/o extensiones para ampliar funcionalidades, en dependencia del CMS que se emplee, lo cual ahorra tiempo de desarrollo.

Los diferentes CMS existentes se pueden agrupar según varias categorías pero las más usadas, según el tipo de aplicación que permita gestionar son: genéricos, blogs, wikis,

Capítulo 1: Fundamentación Teórica

foros, galerías de imágenes, comercio electrónico. Cada una de estas clasificaciones con características propias responde a necesidades específicas de los usuarios.

Para el autor Rodríguez, en Internet se encuentran distintos tipos de CMS, según “*el tipo de plataforma que se desee implementar, pero si se necesita montar una plataforma web con múltiples funcionalidades, se debe seleccionar uno de entre los CMS genéricos disponibles, como Drupal, TYPO3, Joomla o OpenCMS*” (Rodríguez, 2011). De los CMS genéricos, los que no fueron diseñados específicamente para ser usados como blogs, sino que se adaptan a gran tipo de aplicaciones web, como una web personal, de una organización, etc., algunos de los más utilizados en la actualidad son Joomla y Drupal. Según el autor Rodríguez, “ *fueron publicados a principios de este siglo y tienen en común, que están desarrollados en PHP, pueden ejecutarse en un servidor Apache o ISS y se distribuyen como software libre y gratuito*” (Rodríguez, 2011). Cada uno de ellos cuenta con su propia comunidad de usuarios y desarrolladores que contribuyen al desarrollo del proyecto, ya sea trabajando en la mejora del software aportando nuevos módulos para incrementar o mejorar sus funcionalidades. En la presente investigación se profundiza en cada uno de estos CMS.

1.5.1 Joomla

Joomla es un CMS que permite crear aplicaciones *web* dinámicas e interactivas de forma simple. Este gestor de contenidos surge en el 2005 y es resultado de una división del proyecto Mambo¹³. Joomla, es uno de los CMS genéricos más usados y conocidos del mercado, con respecto a las características técnicas, se encuentra liberado bajo una licencia *GPL*¹⁴ (Licencia Pública General). Emplea PHP como lenguaje de programación, requiere para su funcionamiento una base de datos creada con un gestor de bases de datos *MySQL* y servidor HTTP *Apache*. Aunque las versiones más actuales incorporan *PostgreSQL*, como gestor de bases de datos y *Microsoft IIS (Internet Information Server)* como servidor *web*. Aunque Joomla permite la creación de aplicaciones web de forma rápida, pueden existir problemas con la categorización de contenido y opciones de configuración y de diseño limitadas.

¹³ Sistema de portales CMS basado en lenguaje de programación PHP y base de datos SQL de código abierto. Basa todo su aspecto en plantillas o temas.

¹⁴ El software cubierto por esta licencia es *software* libre y protegido de intentos de apropiación.

Capítulo 1: Fundamentación Teórica

1.5.2 Drupal

Para el autor Gil, “*Drupal es un CMS de código abierto, considerado como un framework modular, que aplica el patrón arquitectónico Modelo-Vista-Controlador, multipropósito y muy configurable*” (Gil, 2011), que permite publicar artículos, imágenes, gestionar documentación, imágenes u otros archivos, y servicios añadidos como foros, encuestas, el envío de correos electrónico con notificaciones y administración de usuarios y permisos. Es de libre distribución, con licencia GNU/GPL, desarrollado y mantenido por una activa comunidad de usuarios. Gil refiere que, “*Drupal está desarrollado con el lenguaje de programación PHP y utiliza maquetación con hojas de estilo CSS para la presentación de los contenidos, con lo que es posible construir aplicaciones web totalmente accesibles*” (Gil, 2011).

Dentro de las principales características del CMS Drupal que se pueden consultar en (Drupal, 2015) se encuentran:

- ✓ Ayuda en línea: posee un robusto sistema de ayuda en línea y páginas de ayuda para sus módulos, tanto para los usuarios como para los administradores.
- ✓ Módulos: las funcionalidades que brinda el CMS se pueden ampliar instalando nuevos módulos contribuidos que implementen dichas funcionalidades.
- ✓ Personalización: tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario, producto al entorno de personalización implementado en su núcleo.
- ✓ Autenticación de usuarios: se puede registrar e iniciar sesión de forma local, utilizando un sistema de autenticación externo, otro sitio Drupal o integrando con un servidor LDAP (*Lithweight Access Protocol*).
- ✓ Permisos basados en roles: los administradores de Drupal no tienen que establecer permisos para cada usuario, en lugar de eso, pueden asignar permisos a un rol determinado y agrupar los usuarios por roles.
- ✓ Enlaces permanentes: todo el contenido creado tiene un enlace permanente asociado a él para que pueda ser enlazado externamente sin temor que el enlace falle en el futuro.
- ✓ Objetos de contenido: el contenido creado es funcionalmente un objeto, permitiendo el tratamiento uniforme de la información, con una misma cola de moderación

Capítulo 1: Fundamentación Teórica

para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios sobre cada objeto.

✓ Plantillas: el sistema de temas separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto de la aplicación web. Se pueden crear plantillas con HTML y/o con PHP.

✓ Independencia de la base de datos: aunque la mayor de las instalaciones de Drupal utilizan MySQL, existen otras opciones, debido a que incorpora una capa de abstracción de base de datos mantenida para MySQL, SQLite, PostgreSQL y Oracle.

1.5.3 Selección de CMS a emplear

Se selecciona el CMS Drupal en su versión 7.23, pues su diseño modular permite extender sus funcionalidades con la inclusión de nuevos módulos, temas y perfiles de instalación. A su vez, es un CMS orientado a comunidades y el sistema que se propone está enfocado en esa línea (usuarios, permisos, registros, etc.), su código simplificado es limpio, permite debido a la aplicación de patrones en la implementación de su núcleo hacer cambios fácilmente en el sistema gestor de base de datos. La flexibilidad de Drupal, proporcionada por su arquitectura en capas mantiene los componentes organizados. Lo anterior, sumado al empleo del patrón arquitectónico Modelo-Vista-Controlador (MVC¹⁵) posibilita desarrollar por separado (y acoplar después) diseño y funcionalidades. También más que un CMS, Drupal es un CMF (*Content Management Framework* por sus siglas en inglés), según el autor Noble *“el tablero de desarrollo ha evolucionado hacia algo tan extenso y completo que ya se ha considerado un entorno de trabajo más que un simple panel” (Noble, 2010)*.

1.6. Lenguaje de programación

Eguíluz define un lenguaje de programación como *“aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes” (Eguíluz, 2008)*. En el desarrollo de aplicaciones web, los lenguajes de programación se suelen clasificar como: lenguajes del lado del cliente y del lado del servidor.

¹⁵ El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica del negocio.

Capítulo 1: Fundamentación Teórica

1.6.1 Lenguajes y tecnologías del lado del cliente

Padilla define las tecnologías del lado del cliente como *“aquellas que se ejecutan en el navegador del usuario, son las páginas dinámicas que se procesan en el cliente. En estas páginas, toda la carga del procesamiento de los efectos y funcionalidades la soporta el navegador”* (Padilla, 2008).

Drupal emplea el lenguaje XHTML como base para toda la estructura del sistema del lado del cliente dando formato a la información y CSS para cambiar la apariencia visual de los elementos en el diseño web de manera que se presente de forma estructurada y agradable e incluye el *framework*¹⁶ JQuery para alcanzar mayor dinamismo mediante el lenguaje JavaScript.

JavaScript 1.5

El lenguaje de programación JavaScript es utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página web. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas, así como las interacciones con el usuario. Su compatibilidad con la mayoría de los navegadores modernos, lo posiciona como uno de los lenguajes de programación del lado del cliente más utilizado. El autor Eguíluz, asegura que, *“los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios”* (Eguíluz, 2009).

Actualmente es utilizado en las páginas web para validar formularios, crear *cookies*, detectar navegadores y mejorar el diseño, su fácil aprendizaje lo hace un lenguaje muy demandado. JavaScript, como una parte integrada de Drupal, proporciona características dinámicas con la inclusión de una librería potente para los desarrolladores de módulos. Este *framework*, llamado JQuery es un tipo de librería de código para facilitar la manipulación del código. Es un estándar más rápido y conciso que simplifica el trabajo con los documentos HTML y el manejo de eventos, animaciones e interacciones de Ajax para el rápido desarrollo web. JQuery se diseña para cambiar la manera de escribir en JavaScript.

JQuery es una biblioteca o marco de trabajo para el lenguaje JavaScript, al decir de Álvarez, *“ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente (...) con JQuery obtendremos*

¹⁶ Son unas librerías de código que contienen procesos o rutinas ya listos para usar.

Capítulo 1: Fundamentación Teórica

ayuda en la creación de interfaces de usuarios, efectos dinámicos, aplicaciones que hacen uso de AJAX” (Álvarez, 2009).

Una de las características y ventajas que ofrece Drupal 7 es que esta versión se distribuye con las versiones jQuery 1.4.4 y jQuery 1.8.7, las cuales vienen integradas en su núcleo, convirtiéndose en la librería oficial para desarrollos de aplicaciones del lado del cliente con el CMS. Esta ventaja de integración y compatibilidad permite que JQuery sea la librería a usar para el trabajo en el lado del cliente. En caso que un módulo o componente nuevo no emplee JavaScript 1.5, se recomienda emplear el módulo *Jquery update* que permite la actualización del mismo en administración y usuario.

XHTML 1.1

El lenguaje XHTML (Lenguaje de Marcado de Hipertexto Extensible), es la versión XML de HTML, por lo que tiene las mismas funcionalidades y características de HTML, además de ser una importante mejora. Al mismo tiempo, utiliza la sintaxis de XML, gana toda la potencia y flexibilidad del mismo y es una base perfecta para CSS.

En el desarrollo de la solución, el uso de este lenguaje propone diversas ventajas. La extensibilidad de XHTML mejora las posibilidades de estructuración, usabilidad, accesibilidad, mejores formularios, etc. Por todo lo antes expuesto y la flexibilidad que brinda este lenguaje, el autor de este trabajo coincide en utilizar esta potente combinación de estándares para el desarrollo del sistema a desarrollar.

Para el desarrollo de cualquier sistema web es inherente el uso de HTML (*HyperText Markup Language*), ya que es uno de los lenguajes interpretados por el navegador para presentar el contenido al usuario final. Para los autores Musciano y otros, “HTML es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque sí le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados” (Musciano, 2014).

En la actualidad HTML es el lenguaje predominante para el desarrollo de páginas web y su más reciente versión (**HTML5**) se encuentra en constante mejora para aprovechar todos sus beneficios y conseguir una superior experiencia del usuario.

El autor Van Lancker, expone algunas ventajas que propone la utilización de HTML5 en las aplicaciones web, estas son:

- ✓ Mejora de la estructura de las páginas web: utilizando HTML5 es posible mejorar

Capítulo 1: Fundamentación Teórica

considerablemente la estructura de las páginas de una aplicación web con determinadas etiquetas.

- ✓ Trabajar *offline*: brinda la posibilidad de trabajar con utilidades web sin necesidad de estar *online*.
- ✓ Limpieza de etiquetas: HTML5 se encarga de limpiar aquellas etiquetas obsoletas (estilos en las tablas, `<frames>`, ``, `<center>` y ``, entre otras).
- ✓ Sitios web más rápidos: HTML5 trabaja descargando elementos livianos primero e inmediatamente los muestra en pantalla mientras se cargan los de mayor peso. Las páginas cargan mucho más rápido que en un sitio desarrollado en versiones anteriores de HTML.

Por todas las posibilidades anteriormente expuestas que brinda HTML5 se decide emplear esta última versión de HTML para el desarrollo y diseño de la propuesta de la presente investigación.

CCS3

Las hojas de estilo en cascada, conocidas como (*Cascading Style Sheets* en inglés, abreviado como CSS) se definen según los autores Rivera, Zamora y Soria, como “*un lenguaje de hojas de estilo creado para controlar la presentación de los documentos*” (Rivera et al. 2012). También, De Luca refiere que “*se utilizan para darle forma y aplicar diseño o presentación al contenido o estructura HTML y XML que forma una página o aplicación web*” (De Luca et al. 2014). CSS es la mejor forma de separar contenidos de presentación y es imprescindible para la creación de páginas web complejas ya que permite realizar cambios a múltiples elementos dentro del código, agilizando considerablemente así el proceso de cambio.

En los últimos años el código CSS ha evolucionado hasta su tercera revisión (CCS 3), la cual incluye propiedades que permiten alcanzar efectos complejos, facilitando la aplicación, edición y actualización de formato en todo un sitio web. En la propuesta de solución se decide optar por CSS como lenguaje para mejorar el diseño visual de la solución.

1.6.2 Lenguaje del lado del servidor

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato

Capítulo 1: Fundamentación Teórica

comprensible para él. En la presente investigación se analiza el lenguaje PHP pues es el utilizado en Drupal y sus módulos.

PHP 5.3.5

Según el autor Travis, *“PHP (Hipertext Preprocessor) es un lenguaje del lado del servidor para la generación de HTML. Actualmente, PHP se puede utilizar solo (stand alone scripting language), pero para Drupal, PHP es utilizado como un poderoso lenguaje para extraer información almacenada en base de datos, aplicar reglas del negocio e instrucciones para temas, y eventualmente crear HTML que será enviado al servidor”* (Travis, 2011).

Para los autores Converse, Park y Morgan, *“el lenguaje PHP es gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. El tiempo de aprendizaje de PHP es muy corto gracias a su potencial y simplicidad”* (Converse et al. 2004). Un artículo citado por la Universidad Tecnológica de Mixteca en México, asegura que *“las principales características de PHP son: rapidez, puede ser utilizado en diversos sistemas operativos, servidores web y la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos”* (KadaSoftware, 2013). Para Converse y otros investigadores *“PHP ha evolucionado como lenguaje orientado a objetos permitiendo incluir nuevas funcionalidades, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor”* (Bakken et al. 2015).

Otro elemento lo aporta el autor Travis cuando asegura que, *“Drupal está escrito puramente en PHP, lenguaje que utiliza arreglos multidimensionales para mucho de su procesamiento, y Drupal continúa con esta metodología (principalmente en sus módulos)”* (Travis, 2011). Por esta razón es necesario el empleo de PHP en su versión 5.3.5 en el desarrollo de la presente solución.

1.7 Sistemas Gestores de Bases de Datos (SGBD)

El estudio de la bibliografía consultada arrojó que un SGBD (*DataBase Management System* en inglés, abreviado como DBMS), según el autor Asenjo, *“es un sistema de software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos”* (Asenjo, 2009). En estos sistemas se proporciona un conjunto de funcionalidades, procedimientos y lenguajes que permiten a los distintos usuarios realizar tareas habituales con los datos, garantizando además la seguridad de

Capítulo 1: Fundamentación Teórica

los mismos.

El autor Gil afirma que “*Drupal soporta la integración con diferentes SGBD como MySQL, PostgreSQL, SQLite, entre otros. Esto se logra mediante una capa de abstracción de la base de datos que convierte las instrucciones genéricas proporcionadas por Drupal en instrucciones particulares de cada base de datos*” (Gil, 2011). El mismo autor indica que, “*de esta forma es posible cambiar el gestor de base de datos sin necesidad de cambiar el código de programación de Drupal o de los módulos desarrollados*” (Gil, 2011). A continuación se exponen las características fundamentales de los motores de bases de datos que el CMS Drupal es capaz de soportar.

1.7.1 MySQL

Para los investigadores Casillas, Gibert y Pérez MySQL es un SGBD muy conocido y ampliamente usado por su simplicidad y notable rendimiento, algunas de sus características son (Casillas, 2011):

- ✓ Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- ✓ Está optimizado para equipos de múltiples procesadores.
- ✓ Es muy destacable su velocidad de respuesta.
- ✓ Cuenta con un rico conjunto de tipos de datos.
- ✓ Su administración se basa en usuarios y privilegios.
- ✓ Es altamente confiable en cuanto a estabilidad se refiere.

Sin embargo Oracle desde hace algunos años decidió que MySQL tuviese versiones privativas y dificulta el acceso a las versiones libres, obligando a los desarrolladores al consumo de estas versiones privativas. Esto hace que no sea una opción viable para el desarrollo de la solución que se propone.

1.7.2 PostgreSQL

PostgreSQL es un gestor de base de datos orientado a objetos, muy conocido y usado en entornos de software libre que posee un conjunto de funcionalidades avanzadas. Para el autor Ginesta, “*PostgreSQL se distribuye bajo licencia BSD¹⁷, lo que permite su uso, redistribución y modificación con la única restricción de mantener el copyright del software*

¹⁷Es una licencia de software libre permisiva que permite el uso del código fuente en software no libre.

Capítulo 1: Fundamentación Teórica

a sus autores” (Ginesta et al. 2011).

De la fuente bibliográfica antes consultada se extraen algunas de las características de PostgreSQL (Ginesta et al. 2011) como son:

- ✓ Cuenta con un rico conjunto de tipos de datos, permitiendo su extensión mediante tipos y operadores definidos y programados por el usuario.
- ✓ Su administración se basa en usuarios y privilegios.
- ✓ Sus opciones de conectividad abarcan el protocolo TCP/IP.
- ✓ Es altamente confiable en cuanto a estabilidad se refiere.
- ✓ Puede extenderse con librerías externas para soportar encriptación o búsquedas por similitud fonética.
- ✓ Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas y casi todos los tipos y operadores soportados en SQL92 y SQL99.
- ✓ Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.

1.7.3 SGBD seleccionado

El SGBD que se selecciona, para desarrollar las funcionalidades de la propuesta de solución, es PostgreSQL en su versión 9.2. Presenta un conjunto de características y funcionalidades que lo hacen más profundo en comparación con MySQL que pueden ser consultadas al acceder a **(PostgreSQL: Dossier de Prensa de PostgreSQL 9.3, 2014)**.

Algunos de los elementos que avalan su selección como SGBD son: código abierto, presencia de funciones de indexación, incluyendo exploraciones funcionales en índices parcial, funcional y de múltiple indexado, establece confirmaciones asincrónicas para todo el sistema, por usuario o transacción haciendo su funcionamiento más eficaz y con arquitectura cliente-servidor, los conectores de datos foráneos están habilitados para escritura, lo que permite el intercambio bidireccional de datos entre sistemas, posee una gran estabilidad¹⁸, soporta el lenguaje de programación PHP. A su vez, está establecido en las políticas de desarrollo de software de la UCI.

¹⁸ Indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida.

Capítulo 1: Fundamentación Teórica

1.8 Entorno de Desarrollo Integrado (IDE)

De la bibliografía consultada son varios autores los que coinciden en definir un Entorno de Desarrollo Integrado o IDE (*Integrated Development Enviroment*) como un “programa informático que agrupa diversas herramientas de programación para facilitar la tarea al programador y obtener rapidez en el desarrollo” (González, 2003). Luego, concluye “es un sistema que facilita el trabajo del desarrollador de software, integrando sólidamente la edición orientada al lenguaje, la compilación o interpretación, la depuración, las medidas de rendimiento, la incorporación de las fuentes al sistema de control de fuentes, etc., normalmente de una forma modular” (González, 2003). Algunas de las características que incluyen los IDE son:

- ✓ Multiplataforma.
- ✓ Soporte para diversos lenguajes de programación.
- ✓ Integración con sistemas de control de versiones.
- ✓ Reconocimiento de sintaxis.
- ✓ Extensiones y componentes para el IDE.
- ✓ Depurador.
- ✓ Importar y exportar proyectos.
- ✓ Integración con *framework* populares.
- ✓ Múltiples idiomas.
- ✓ Manual de usuarios y ayuda.

Entre los ejemplos de IDE más utilizados en entornos de Linux se encuentran: NetBeans y Zend Studio, los cuales se analizan a continuación:

1.8.1 Zend Studio

Zend es uno de los IDE que agrupa todos los componentes de desarrollo necesarios para el ciclo de desarrollo de aplicaciones PHP, disponible para desarrolladores profesionales.

Incluye un conjunto de herramienta de edición, depurado, análisis y optimización, lo que permite simplificar los proyectos complejos. Proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. Consta de dos partes: las funcionalidades de parte del cliente (contiene interfaz de edición y ayuda) y las del servidor (instala Apache y el módulo PHP o en caso que estén instalados los configura

Capítulo 1: Fundamentación Teórica

para trabajar juntos en depuración). A pesar de todas estas ventajas es distribuido a través de una licencia comercial.

1.8.2 NetBeans 7.14

En el portal web del IDE se afirma que, “*NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, aunque permite programar en diferentes lenguajes*” (NetBeans, 2011).

NetBeans es una base sólida para desarrollar aplicaciones complejas con un enfoque modular y posee características como la extensibilidad y la escalabilidad.

Se selecciona NetBeans en su versión 7.14 pues ofrece un excelente entorno y proporciona un rendimiento superior para los desarrolladores de PHP, proporcionándole editores. Este IDE tiene editores y herramientas para los lenguajes HTML, PHP, JavaScript y CSS que son los empleados para la solución de software. A su vez, permite el marcado de error de PHP, plantillas, macros, tiene un excelente balance entre una interfaz con múltiples opciones, completamiento de código y posee una amplia comunidad de usuarios y desarrolladores mundialmente.

Según el autor Rea otras características de esta herramienta son (Rea, 2013):

- ✓ Interfaz de usuario para la personalización de la aplicación.
- ✓ *Framework* para la creación de interfaces de usuario.
- ✓ Ayudas del sistema.
- ✓ Rendimiento óptimo en tiempo de ejecución y optimización de recursos.
- ✓ Poder usar las licencias *open source* para realizar mejoras futuras.
- ✓ Tener un respaldo *online* por parte de otros programadores.
- ✓ Tener un IDE que soporta el desarrollo de todos los tipos de aplicación Java.
- ✓ Internacionalización.

1.8.3 IDE seleccionado

El IDE Zend Studio al tener carácter privativo no cumple con la filosofía de software libre. Para el desarrollo de la propuesta de solución se selecciona NetBeans 7.14 que contiene una plataforma donde desarrollar las aplicaciones a partir de un conjunto de componentes de software llamados módulos. Los módulos dan la posibilidad de ser extendidos agregándole nuevas funcionalidades, facilitando y reduciendo el tiempo de desarrollo de

Capítulo 1: Fundamentación Teórica

la aplicación. Además, ofrece formato de código acorde a los estándares de Drupal y reconocimiento a los principales tipos de archivo específicos de Drupal (.info, .module, .install etc.). NetBeans facilita el auto-completado de código para lenguajes HTML, JavaScript y CSS, que serán empleados para el desarrollo de la propuesta de solución, marca los errores de PHP, plantillas, macros, control de versiones y ofrece soporte para *frameworks* como JQuery que es uno de los utilizados agilizando el desarrollo de la solución.

1.9 Herramientas para el diseño de prototipos de interfaz de usuario

Hassan y sus colaboradores definen que un “*prototipo es una representación limitada del diseño de un producto, probarlo en situaciones reales y explorar su uso*” (Hassan et al. 2004). En las interfaces de usuarios se emplean los prototipos puesto que son útiles para comunicar, discutir y definir ideas, a su vez, ayudar a idear, identificar y probar un producto antes de crearlo.

La consulta bibliográfica realizada arrojó la presencia de herramientas para diseñar prototipos de interfaz de usuario como *Axure RP* y *Justinmind*, son herramientas de prototipado profesional que permiten crear interfaces de usuario para aplicaciones web, pero disponibles para plataforma Windows y Mac; *MockFlow*, es una herramienta web sencilla para diseño de aplicaciones web y de software, aunque su versión gratuita limita el número de prototipos que se pueden hacer y se requiere de Internet para su trabajo; *Pencil Project*, es un *plugin*¹⁹ para Firefox que permite la elaboración de bocetos de forma sencilla, multiplataforma y se emplea directamente en el navegador, sin embargo no permite la colaboración, esto refiere que la herramienta pueda manejar íconos de otras herramientas y *Balsamiq Mockups*. Para este estudio que se presenta se selecciona *Balsamiq Mockups* en su versión 2.2.10 por ser una herramienta que permite desplegar la creatividad de diseñadores sin ser limitados, permitiendo que el equipo de desarrollo respete y comparta una filosofía común de usabilidad. Es multiplataforma, permite la creación y modificación de diseños en tiempo real y la exportación de estos diseños en imágenes con formato PNG. También, el uso de esta herramienta por parte del cliente sin ningún conocimiento técnico, permite que este comunique sus ideas y necesidades al equipo de desarrollo de manera clara. A su vez, que el equipo de desarrollo pueda

¹⁹ Es una aplicación que se relaciona con otras para aportarle nuevas funcionalidades y suelen ser muy específicas.

Capítulo 1: Fundamentación Teórica

comunicar la propuesta de solución, sin invertir tiempo en la implementación del software sin la aceptación del cliente.

1.10 Servidor web

Para el autor Duque, un servidor web “es un programa que se ejecuta continuamente, manteniéndose a la espera de peticiones por parte de un cliente (un navegador de Internet) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error” (Duque, 2012).

Para su correcta ejecución, una aplicación en Drupal necesita alojarse en un servidor web. Para el autor Travis, “los servidores web recomendadas para Drupal son IIS (Internet Information Server) y Apache” (Travis, 2011). A su vez, el lenguaje de programación PHP (elegido para el desarrollo de la propuesta de solución) para su funcionamiento necesita tener instalado uno de estos dos servidores.

1.10.1 Internet Information Server

Según los investigadores Hu, Irfan y otros, “Internet Information Server o ISS es un servidor de Microsoft destinado a la publicación, mantenimiento y gestión de páginas y portales web” (Hu et al. 1997). La razón de peso por la que no se selecciona ISS para el sistema a implementar, es su carácter privativo, razón suficiente por la cual no se procede al posterior análisis del mismo, teniendo en cuenta que la UCI aboga por el uso de tecnologías gratuitas y de código libre.

1.10.2 Apache 2.2.22

El servidor web Apache es un software de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras. Apache es un potente servidor web de software libre cuyo objetivo es servir o administrar páginas web a los clientes o navegadores que las solicitan.

Entre las principales características de este servidor web, se destacan las siguientes:

- ✓ Apache se ha concentrado en la escalabilidad, en la seguridad y en el rendimiento.
- ✓ Es un servidor web altamente configurable.
- ✓ Plataforma de servidores web de código fuente abierto.
- ✓ Trabaja con diversas tecnologías como PHP, *mod_perl*, *servlets* de Java, etc.

Capítulo 1: Fundamentación Teórica

1.10.3 Servidor web seleccionado

Se selecciona como servidor web Apache en su versión 2.2.22 debido a que es una tecnología gratuita, de código abierto, soporta la mayoría de los lenguajes de desarrollo del lado del servidor, cuenta con bases de datos de autenticación, es compatible con el SGBD y el CMS y encabeza la lista, según el autor Kabir “*de las plataformas de servidores web de código fuente más poderosa del mundo*” (Kabir, 1999). Dos estudios, uno publicado por *Netcraft* en enero del 2015 y disponible en (Netcraft, 2015) y otro del autor (Greenstein et al. 2014), afirman que algo más del 60% de los servidores de Internet confían en Apache como servidor web, por su robustez y estabilidad. También, alrededor del 90% de los servidores con más disponibilidad funcionan con Apache.

1.11 Conclusiones parciales del capítulo

Luego del estudio y análisis realizado del objeto de investigación del presente trabajo de diploma, apoyado en los métodos de la investigación científicos identificados, se concluye lo siguiente:

- ✓ Se construyó el marco conceptual que soporta la investigación y se adquirió el conocimiento necesario sobre las metodologías, herramientas y lenguajes, seleccionándose las más adecuadas para el cumplimiento del objetivo general propuesto.
- ✓ Se identificó la metodología XP como guía del proceso de desarrollo, pues provee un marco metodológico integrado de principios, prácticas y técnicas concretas que se ajustan al entorno y/o condiciones del proyecto y tienen el potencial de acelerar el tiempo de desarrollo.
- ✓ Se definió Drupal 7.23 como CMS ya que es una plataforma de desarrollo web de última generación, PostgreSQL 9.1.8 como SGBD, NetBeans 7.14 como IDE, PHP 5.3.5 como lenguaje del lado del servidor, JavaScript, CCS3 y HTML5 como lenguajes del lado del cliente y como servidor web Apache 2.22, guiados por XP como metodología de desarrollo puesto que presenta prácticas muy afines para darle solución al problema planteado.

Capítulo 2: Propuesta de solución

Capítulo 2: Propuesta de solución

2.1. Introducción

En XP no se enfatiza la definición temprana de una arquitectura bien especificada para el sistema. Dicha arquitectura evolutiva y los posibles inconvenientes que se generaría por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. En el libro *Extreme Programming Explained*, Kent Beck plantea: *“la metáfora del sistema es algo que todos los clientes y programadores entienden sobre cómo funciona el sistema. Las metáforas ayudan con la abstracción y modelado del sistema. Encontramos más útil el uso de una metáfora que un modelo, como por ejemplo el modelo de dominio”* (Beck, 2000).

Este concepto es una manera bastante sencilla de explicar el propósito del proyecto y guiar la estructura y arquitectura del mismo. Por otra parte Martin Fowler explica que la práctica de la metáfora consiste en *“formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Las palabras usadas para identificar entidades deben estar consistentemente tomadas de la metáfora principal. Mientras avanza el desarrollo del sistema y la metáfora madura, el equipo de desarrollo podrá hallar nueva inspiración a partir de la examinación de la metáfora”* (Fowler, 2003). De esta forma, este conjunto de palabras puede ayudar además a la nomenclatura de clases y métodos del sistema.

En la presente investigación, la metáfora del sistema que se propone es: un sistema que logre gestionar las actividades de la Agenda de trabajo del Consejo de dirección de la Facultad 4 y la información generada. La metodología XP sugiere utilizar *“conjuntamente con la metáfora, otras técnicas importantes de diseño como son: “historias de usuario y tarjetas CRC (Clase-Responsabilidad-Colaborador)”* (Bajo de Luque et al. 2011). Además, en este capítulo se determinan los servicios que debe brindar la solución y se describen las posibles funcionalidades a través de la exposición de los principales artefactos generados por la metodología en las fases de Exploración, Planificación y Diseño.

2.2. Descripción de los procesos vinculados al campo de acción

Actualmente la gestión de la Agenda de trabajo del Consejo de dirección (C/D) de la

Capítulo 2: Propuesta de solución

Facultad 4 se realiza manualmente y de forma descentralizada. El flujo actual para gestionar la Agenda de trabajo del Consejo de dirección se comporta de la siguiente manera: a los miembros del Consejo de dirección les llega (vía correo electrónico) la citación para la reunión, donde se especifica: día, hora, lugar y se les recuerda el orden del día a tratar. Cuando se reúne el Consejo de dirección, la secretaria del Consejo hace un recuento de los temas a tratar y de otros que se deban incluir y que no estaban planificados, debido a la dinámica propia con que se trabaja en la universidad y la facultad. A su vez, los miembros que deben rendir cuentas ante el Consejo, exponen sus informes, los que previamente se debieron haber circulado a todos los miembros. Las opiniones y criterios que se emiten en el Consejo quedan reflejados en un acta que se confecciona manualmente por parte de la secretaria del Consejo de dirección. Esta acta se redacta manualmente para luego transcribirla a formato digital por lo que se puede incurrir en la omisión de ideas que pueden ser de peso para la toma de decisiones. También, los informes que se exponen en el Consejo, no están disponibles en un espacio donde los usuarios puedan consultarlo cuando así lo requieran.

Hasta el momento no se cuenta con una herramienta informática que permita informatizar las tareas y actividades previas y posteriores a la realización del Consejo de dirección, que facilite a los miembros de este órgano de dirección, disponer de toda la información necesaria y de manera organizada. Por lo que si se contara con una herramienta informática se lograría que: el sistema fuese capaz de enviar notificaciones a los usuarios de cambios en la reunión del Consejo de dirección, así como el envío de las citaciones; que los informes que se presentan en el C/D estén disponibles para cuando se decida acceder a ellos y se evitaría el trabajo de redactar manual el acta para luego llevarlo a formato digital.

2.3. Objetos de informatización

Para llevar a cabo la gestión de la Agenda de trabajo del Consejo de dirección de la Facultad 4, existen cinco procesos que deben ser informatizados:

1. Enviar citación para Consejo de dirección.
2. Notificar confirmación de asistencia al Consejo de dirección.
3. Confeccionar el acta del Consejo de dirección.
4. Archivar actas e informes del Consejo de dirección.
5. Imprimir y exportar las actas e informes del Consejo de dirección.

Capítulo 2: Propuesta de solución

2.4. Propuesta de solución

Como propuesta que fundamenta esta investigación se plantea, la creación de un sistema de gestión desarrollado mediante el CMS Drupal, que permita elevar el control de las actividades de la Agenda de trabajo del Consejo de dirección de la Facultad 4. Para informatizar los cinco procesos descritos anteriormente, el autor propone la implementación de tres módulos que a continuación se describen con sus respectivas funciones:

1. Módulo citación (*m_planifica*): el usuario autorizado accede al sistema y llena un formulario que contiene los datos: fecha, hora, lugar, orden del día y participantes, para enviar una notificación a todos los participantes en el Consejo de dirección. Los usuarios que participarán deben estar previamente autenticados en el sistema.
2. Módulo de informe (*m_informe*): solo cuando un usuario accede al sistema puede crear un informe, modificar uno existente y del que sea creador y eliminar sus propios informes. El sistema guardará los informes en la base de datos y mostrará un mensaje de que la operación sobre el informe se ha ejecutado.
3. Módulo actas (*m_actas*): la secretaria/o del Consejo accede a la opción de crear una nueva acta cada vez que se realice un Consejo de dirección y permite seleccionar la citación a la que corresponde dicha acta y enviar al grupo de audiencia (Consejo de dirección) una notificación para que la tengan en su poder.

2.5. Personal relacionado con el sistema

El sistema GESTA de la Facultad 4 permitirá a los usuarios ejecutar un grupo de funcionalidades y servicios para cumplir con los objetivos planteados. Su implementación será a través de módulos donde se establecerán un grupo de roles para asignar los diferentes permisos para su acceso. Los roles identificados son aquellas personas que van a interactuar con el sistema, en dependencia del espacio de trabajo en que se encuentre.

Tabla 1: Usuarios relacionados con el sistema y su descripción.

Usuarios	Justificación
Anónimo	Es el usuario que navega por la aplicación sin haberse autenticado.

Capítulo 2: Propuesta de solución

	Tiene la posibilidad de interactuar con contenidos que no requieran autenticación.
Usuario autenticado	Usuario que se encuentra registrado en el sistema y se autenticó correctamente. Puede acceder solo a las opciones definidas para su rol.
Administrador	Usuario encargado de realizar la gestión de los contenidos en el sistema. Tiene libertades ilimitadas para ejecutar todas las funciones administrativas del sistema: administrar todas cuentas de los usuarios, darle soporte y mantenimiento a la aplicación.
Decano de la Facultad	Es un usuario autenticado que gestiona todos los procesos que ejecuta el sistema, tiene acceso a la edición de todos los elementos de una actividad, como es el caso de la planificación, modificación, acceso a la información que se genere en cada reunión, etc.
Secretaria del Consejo de dirección	Es un usuario autenticado que tiene privilegios para editar algunos contenidos en el sistema, como es el caso de las actas y revisar los informes que se presentan y editarlos en caso de errores y notificárselos a sus autores.

2.6. Definición de contenidos y servicios

Mediante el desarrollo de las funcionalidades definidas, el sistema podrá ofrecerles a los

Capítulo 2: Propuesta de solución

usuarios un conjunto de contenidos y servicios con el fin de satisfacer sus necesidades y expectativas, alcanzando los objetivos trazados para la aplicación. Dentro de los servicios que brindará la propuesta de solución se encuentran:

- ✓ Información relacionada con la estructura, misión y visión del Consejo de dirección de la Facultad 4.
- ✓ Servicio para la creación y gestión de actividades de la Agenda de trabajo del Consejo de dirección.
- ✓ Servicio de consulta de calendarios.
- ✓ Servicios de descarga de documentación.
- ✓ Noticias informativas sobre el acontecer de la universidad.
- ✓ Imprimir y exportar la documentación necesaria.
- ✓ Enlaces a otros sitios.

2.7. Modelo de dominio

El modelo de dominio es un artefacto de la disciplina de negocio, construido con las reglas de UML durante la fase de concepción. De la consulta bibliográfica realizada el autor de la investigación que se presenta constató que, el modelo de dominio presenta uno o más diagramas de clases que no contienen conceptos propios de un sistema de software sino de la realidad. A su vez, es un subconjunto del modelo de negocio y se realiza cuando no están claros los procesos o cuando no se identifican claramente los actores y trabajadores del negocio.

Los modelos de dominio suelen emplearse, para capturar y expresar el entendimiento ganado en un área como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de negocio es utilizado por los analistas de los equipos de desarrollo como un medio para comprender el sector del negocio al cual el sistema va a servir.

2.7.1. Diagramas de modelado de negocio

En la Ilustración 1 se muestra el diagrama del flujo de trabajo “Citación al Consejo de dirección”, modelado en la herramienta Bigazi, lo que permite tener una visión más clara de cómo funciona el negocio y del flujo de trabajo del proceso de “gestionar agenda de trabajo de la Facultad 4”. Para visualizar los demás flujos de trabajo de los subprocesos

Capítulo 2: Propuesta de solución

implementados puede consultarse el [Anexo # 1: Flujo de trabajo del proceso agenda de trabajo de la Facultad 4.](#)

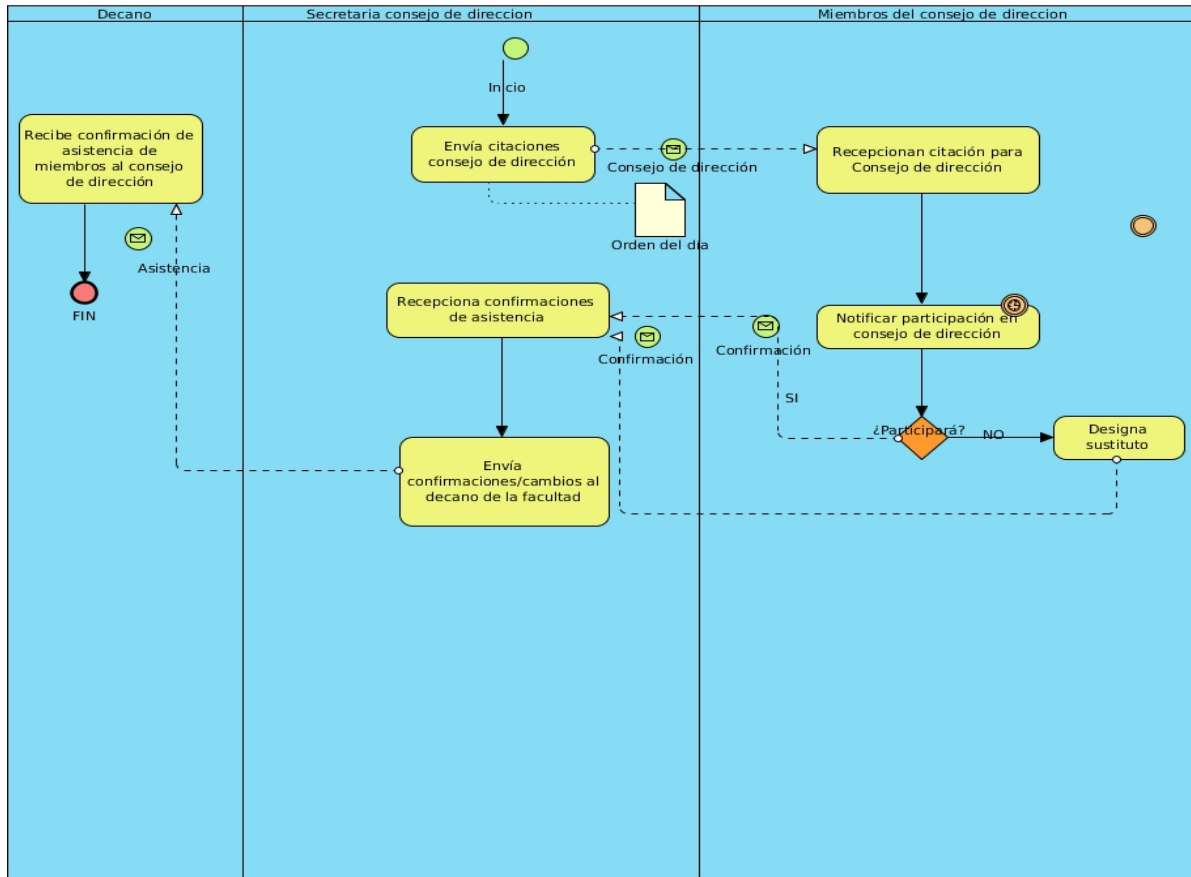


Ilustración 1: Diagrama del flujo de trabajo Citación al Consejo de dirección.

2.8. Exploración

Es la primera fase de la metodología XP, en la que se define el alcance general del proyecto, diversos autores coinciden en que se exploran las diferentes formas de resolver problemas concretos de implementación que se pueden presentar y sobre todo se definen las Historias de usuario, que son la forma de definir los requisitos del sistema a implementar. El autor Jaskowicz refiere que en esta fase, “*los programadores estiman los tiempos de desarrollo...y se hacen las estimaciones de cada iteración, quedando claro que las estimaciones son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalles en cada iteración*” (Jaskowicz, 2008).

Capítulo 2: Propuesta de solución

El propósito de la fase de Exploración es, para el autor Urbino Hidalgo, “llegar a un entendimiento global entre desarrolladores y el cliente acerca de qué es lo que el futuro sistema debe hacer” (Urbino, 2010). Para ello se toman los requerimientos funcionales que se han identificado y comienza el proceso de descripción de “las historias de usuario, los análisis en grupos y las tormentas de ideas. Se esclarecen las dudas sobre procesos que deben ser automatizados y que no han sido entendidos” (Urbino, 2010).

2.8.1 Historias de usuario

El autor Penadés refiere que las Historias de usuario (HU) son la técnica utilizada en XP “para especificar los requisitos del software tanto funcionales como no funcionales” (Beck et al. 2005). El tratamiento de las HU es muy flexible haciendo que cada una de ellas sea lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

La información de una HU suele variar y ajustarse a las características específicas del proyecto. Existen varias planillas sugeridas para representar cada HU, pero no existe un consenso al respecto. A continuación se incluyen los aspectos esenciales de cualquier proyecto de desarrollo, que ha sido resultado de consultar las HU presentadas como ejemplo por los autores (Shore et al. 2008) y (Wake, 2000).

- ✓ **Número:** Posee el número asignado a la HU.
- ✓ **Nombre de la HU:** Atributo que contiene el nombre de la HU.
- ✓ **Usuario:** El usuario del sistema que utiliza o protagoniza la HU.
- ✓ **Prioridad en el negocio:** Evidencia el nivel de prioridad de la HU en el negocio.
- ✓ **Riesgo de desarrollo:** Evidencia el nivel de riesgo en caso de no realizarse la HU.
- ✓ **Puntos estimados:** Atributo que contiene la estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como cinco días hábiles trabajando 40 horas, es decir, ocho horas diarias.
- ✓ **Iteración asignada:** Precisa la iteración en la que será desarrollada la HU.
- ✓ **Descripción:** Posee una breve descripción de lo que realizará la HU.
- ✓ **Observaciones:** Brinda información extra que se estime agregar para hacer más comprensible la HU.

Capítulo 2: Propuesta de solución

Para las HU que describen las características y cualidades con que debe cumplir la solución, el autor de la presente investigación considera excluir los siguientes aspectos: riesgo en desarrollo, puntos estimados y puntos reales. A continuación se muestra la HU Registrar usuario en el sistema, que relacionada a esta se encuentran los requisitos funcionales: Registrar usuarios en el sistema, Mostrar los datos del perfil del usuario y Editar perfil de usuario. El resto de las HU generadas que representan funcionalidades que serán implementadas se pueden consultar en el [Anexo # 2: Historias de usuario](#).

Tabla 2. HU Registrar cuenta en el sistema.

Historia de Usuario	
Número: 1	Nombre: Registrar usuarios del sistema
Usuario: Todos	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Muestra un formulario de creación de la cuenta, donde se introducen los datos que se especifican y luego puede visualizar los datos personales de su perfil, además de poder editarlos.	
Observaciones: Se mostrará un mensaje al usuario si existe alguna dificultad durante la introducción y validación de sus datos.	

2.9. Planificación

Para el autor Joskowicz la planificación es una “fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las HU, y asociadas a estas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas o Release Plan” (Joskowicz, 2008). Varios investigadores exponen que esta fase es una de las más importantes dentro de la metodología XP, ya que se estiman los esfuerzos por parte de los desarrolladores para implementar las funcionalidades de la aplicación. Se necesita un constante intercambio con el cliente para suplir cualquier duda que surja durante el desarrollo del sistema y de existir algún cambio, que todas las partes involucradas sean informadas del mismo y las consecuencias que arraigará tal decisión. A

Capítulo 2: Propuesta de solución

su vez, se planifica la utilización de recursos humanos y materiales para su utilización en cualquier situación del proceso.

2.9.1 Estimación de esfuerzo por HU

Refiere el autor Letelier que *“la estimación de esfuerzo asociado a la implementación de las HU la establecen los desarrolladores utilizando como medida el punto. Un punto equivale a una semana ideal de programación. Las historias generalmente equivalen de uno a tres puntos. Por otra parte, los clientes deciden sobre el ámbito, tiempo de las entregas y de cada iteración” (Newkirk et al. 2001).*

En el [Anexo # 3: Estimación de esfuerzo por HU](#), se puede consultar este artefacto generado por la metodología XP.

2.9.2 Plan de iteraciones

En la metodología XP, la creación del sistema se divide en iteraciones. La duración ideal de una iteración está entre una y tres semanas. Un grupo de investigadores encabezado por el autor Bajo de Luque afirma que *“para cada una de las iteraciones se establecen un conjunto de HU definidas por el cliente que serán implementadas en cada iteración del sistema, superando las pruebas de aceptación las cuales son realizadas al final de cada ciclo” (Bajo de Luque et al. 2013).* Al terminar cada iteración la aplicación tendrá implementadas funcionalidades para dar cumplimiento a los objetivos propuestos, por las características del proyecto el mismo se divide en tres iteraciones que se detallan a continuación:

Primera iteración: se implementan las funcionalidades referentes a las HU 1 Registrar usuarios del sistema, HU 2 Autenticar usuario en el sistema, HU 3 Gestionar citación y HU 4 Visualizar citación. Con estas se conforma la base de la estructura del negocio, obteniendo una versión de prueba y aportándole al sistema las funcionalidades básicas.

Segunda iteración: se realiza la implementación de las HU 5 Publicar informe del Consejo de dirección, HU 6 Mostrar informe y acta del Consejo de dirección, HU 7 Mostrar listado de citaciones al Consejo de dirección, HU 8: Gestionar acta del Consejo de dirección, HU 9: Eliminar citaciones y la HU 10: Filtrar búsqueda de usuarios y documento por atributo. Además, se corrigen errores o inconformidades del usuario con las HU implementadas en la iteración anterior. De esta forma, se obtiene la segunda versión de pruebas del software. Esta segunda versión es mostrada a los clientes con el único

Capítulo 2: Propuesta de solución

objetivo de realizar cambios de acuerdo con la aceptación del mismo.

Tercera iteración: se implementan las HU 11: Mostrar listado de actas del Consejo de dirección, HU 12: Eliminar informe del Consejo de dirección, HU 13: Ver listado de miembros del Consejo de dirección, HU: 14 Administración de usuarios, HU 15 Exportar a PDF actas e informe del Consejo de dirección y HU 16 Imprimir acta e informe del Consejo de dirección. Además, se corrigen errores o inconformidades del usuario con las HU implementadas en las iteraciones anteriores.

2.9.3. Plan de duración de las iteraciones

El plan de duración de las iteraciones se realiza luego de tener estimado en días cuánto demora implementar cada HU. Se tendrá en cuenta además la prioridad que el cliente le asigna a cada Historia de usuario y el nivel de complejidad que estas poseen. El plan de duración de las iteraciones se puede consultar en el [Anexo # 4: Plan de duración de las iteraciones](#).

2.9.4. Plan de entregas

Para un grupo de investigadores encabezado por Luque en el plan de entregas se realiza un cronograma de entregas, *“donde el cliente establece la prioridad de cada HU, cuáles serán agrupadas para conformar una entrega, el orden de las mismas. En correspondencia, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.)”* (Luque et al. 2011). Esta fase dura unos pocos días. En el [Anexo #5: Plan de entregas](#), se puede consultar todo lo referente a los entregables por cada iteración de la implementación.

2.10. Diseño del sistema

Un prototipo es una visión preliminar del sistema a implementar, es un modelo operable, fácilmente ampliable y modificable, que tiene todas las características que hasta el momento debe tener el sistema. El autor Oktaba refiere que, *“un prototipo de interfaz de usuario no funcional es una representación parcial de la interfaz de usuario que tendrá el sistema y se utiliza para que el cliente pueda refinar sus necesidades y comunicarlas al desarrollador”* (Oktaba, 2010). A su vez, la metodología XP sugiere que hay que conseguir *“diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible que a la larga costará*

Capítulo 2: Propuesta de solución

menos tiempo y esfuerzo desarrollar” (Castillo, 2011). En la Ilustración 2 se propone un prototipo de interfaz de usuario, que constituye el punto de partida para la interfaz inicial que presentará el sistema para la gestión de la Agenda de trabajo del Consejo de dirección de la Facultad 4.

Sistema de gestión de Agenda de trabajo de la Facultad 4

Usuario Contraseña Recordarme

Logotipo Título del sistema

Inicio Misión Vision Quienes somos Superación Cursos, Diplomados... Otros temas

Inicio > Contenido > Editar

Editar

Asunto:

Asistentes:

Lugar:

Hora:

Adjunto

Word

No se han agregado adjuntos

Enlaces de interés

- [Correo](#)
- [Intranet](#)
- [Biblioteca](#)
- [EVA-Pregrado](#)
- [EVA-Posgrado](#)
- [Octavitos](#)

Usuarios conectados

- Usuario 1
- Usuario 2
- Usuario n

© 2015 | Facultad 4 | UCI [Términos y condiciones](#) | [Quejas y sugerencias](#)

Ilustración 2: Prototipo de interfaz de usuario de Agenda de trabajo.

2.10.1. Arquitectura de información

Refiere el autor Reyes que la Arquitectura de Información (AI), “es el conjunto de métodos y herramientas que permiten organizar los contenidos, para ser encontrados y utilizados por los usuarios, de manera simple y directa” (Reyes, 2012). La AI estará cumpliendo sus objetivos cuando un usuario entre por primera vez al sitio y pueda reconocer a quién pertenece el sitio web; lo que pueda entender en forma rápida, sin esfuerzo y encontrar la información ofrecida fácilmente. A su vez, eso entregará como beneficio que, quienes producen el sitio, podrán ubicar la información sin tener que crear nuevas estructuras.

Capítulo 2: Propuesta de solución

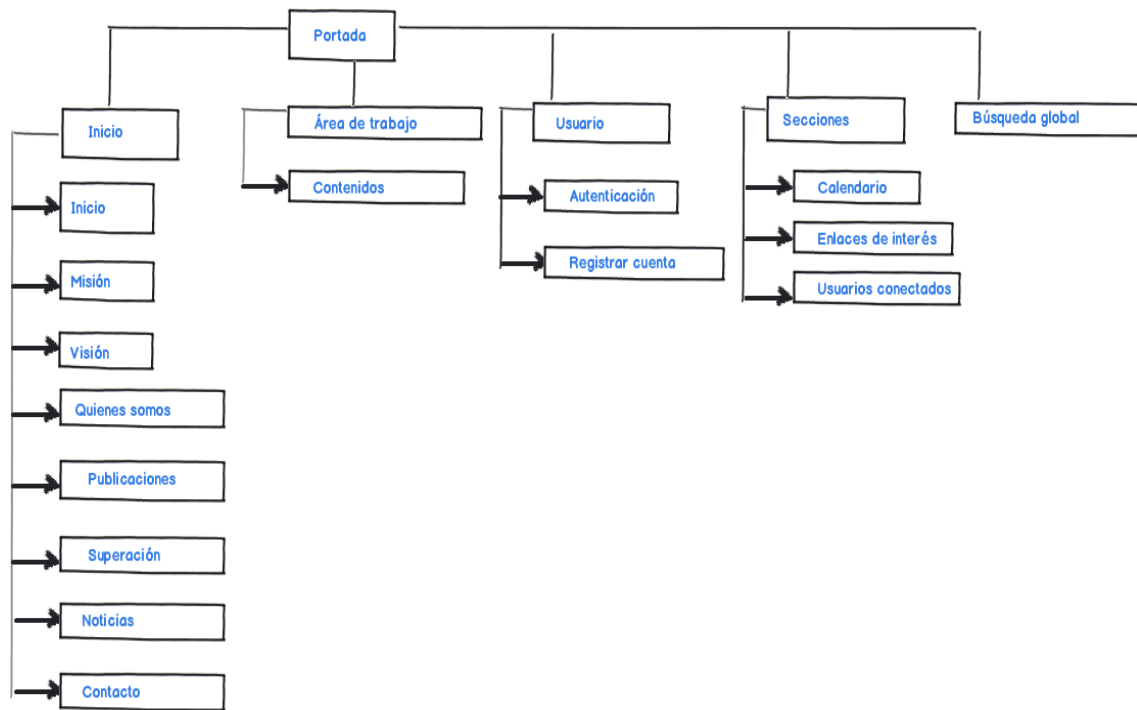


Ilustración 3: Representación de la Arquitectura de la Información del sistema

2.11. Tarjetas CRC

De la bibliografía consultada, se puede afirmar que la metodología XP no requiere la presentación del sistema mediante diagramas de clases empleando notación UML, aunque en su lugar el autor Bajo de Luque refiere que, “se usan técnicas como las tarjetas CRC (Clase, Responsabilidad y Colaboración)” (Bajo de Luque et al. 2011). Más adelante, el investigador refiere que cada tarjeta CRC representa una clase, objeto, módulo, paquete, etc., con su nombre en la parte superior. En la parte inferior izquierda se describen las responsabilidades y a la derecha las clases que le sirven de soporte” (Bajo de Luque et al. 2011).

La filosofía del CMS Drupal, no es basada en POO, pero aun así, en sus módulos utiliza varios conceptos o técnicas del paradigma orientado a objeto (“objetos”, “abstracción”, “polimorfismo”, “encapsulación” y “herencia”). Además, a pesar de la falta explícita de clases, en ellos se destaca el uso de los siguientes patrones de diseño: *Singleton*, *Decorator*, *Bridge*, *Chain of Responsibility* y *Command*. Al igual que en la POO, la elaboración de tarjetas CRC en Drupal es útil para dar una idea de la arquitectura del sistema, conocer la ubicación de las diferentes responsabilidades y conocer la distribución

Capítulo 2: Propuesta de solución

de los módulos a desarrollar. Para aplicar esta idea se adopta lo siguiente: cada tarjeta CRC es un módulo, sus responsabilidades son las funciones que realiza y sus colaboradores los propios métodos del módulo.

Tabla 3. Tarjeta CRC Módulo Citación.

Módulo: m_citación	
Responsabilidades	Colaboraciones (módulos)
Gestionar citación de Consejo de dirección	<i>Node</i> <i>Views</i> <i>Block</i> <i>Menu</i> <i>Upload</i> <i>Path</i> <i>Calendar</i> <i>Multiselect</i> <i>Date</i> <i>Rules</i>
Crear aviso de Consejo de dirección	<i>Calendar</i> <i>Views</i> <i>SMTP</i> <i>Tableform</i> citación

2.12. Conclusiones parciales del capítulo

Luego de realizar un análisis de los principales conceptos que intervienen en el negocio, se logró obtener una propuesta de solución que dará respuesta a la problemática de la investigación. Por ello se concluye que:

1. Atendiendo a los requerimientos que debe presentar el sistema, definidos por parte del cliente, se identificaron un total de 16 HU y se implementarán tres módulos, que serán entregados en 12 semanas.

Capítulo 2: Propuesta de solución

2. Prestando atención a lo planteado por la metodología seleccionada en relación a la distribución y planificación del trabajo de producción, se decidió establecer un total de tres iteraciones con una duración de 3 semanas, cada semana con 40 horas laborales.
3. Se modelaron los principales procesos que tienen lugar en el funcionamiento del sistema, se especificaron las características del modelo de negocio, con el objetivo de comprender la solución, a su vez permitió determinar cómo se desarrolló el sistema a partir de las funcionalidades identificadas.

Capítulo 3: Implementación y Pruebas

Capítulo 3. Implementación y Pruebas

3.1. Introducción

La implementación es un pilar fundamental de la metodología XP que se realiza durante la fase de producción, a partir del análisis y diseño consecuentes en la fase de iteraciones hasta la entrega final, y donde el equipo de desarrollo mantiene un constante intercambio con el cliente, no solo para aclarar las dudas que puedan surgir sino para formar parte de este.

Aunque en las fases anteriores la metodología XP propone claramente artefactos basados en técnicas como son las HU y las tarjetas CRC, para la fase de producción se plantean prácticas concretas. Los investigadores Letelier, Beck, Butcher y Wake definen estas prácticas como “*integración continua y utilización de estándares de código*” (**Letelier et al. 2008**), (**Beck, 1999**), (**Butcher et al. 2013**) y (**Wake, 2000**), pero no existe una opinión única entre los autores sobre cuáles son los artefactos que genera la metodología XP, aunque implícitamente se sugiere que el código desarrollado forme parte de estos. Debido a que el sistema esperado es la implementación de la presente investigación, el código se encuentra incluido en módulos, que aportan un conjunto de funcionalidades al sistema GESTA. Para obtener mayor entendimiento sobre lo antes expuesto, fue necesario definir conceptos elementales en Drupal como son: nodos, bloques, ganchos, temas y menús, lo cual puede consultarse en bibliografía sobre el tema citado por los autores (**Butcher et al. 2013**), (**Reyes et al. 2007**) y (**Travis, 2011**).

Este capítulo está enmarcado en las fases de iteraciones hasta la entrega final y producción que propone la metodología XP, por lo tanto, se explica someramente cómo se desarrollaron las funcionalidades del sistema GESTA de la Facultad 4, comenzando desde la creación de las tablas añadidas a la base de datos, hasta cómo se construyó el diseño visual y la programación de los módulos que se incorporaron al CMS. Para concluir, se analizan los resultados de las pruebas diseñadas para cada una de las HU que se proponen en la fase de Exploración y Planificación.

3.1. Patrón arquitectónico de Drupal

El investigador Gómez refiere que el patrón MVC, “*es una arquitectura de diseño de software, para separar los componentes de aplicación en tres niveles, interfaz de usuario,*

Capítulo 3: Implementación y Pruebas

lógica de control y lógica de negocio. Es una especialización de un modelo de capas, con la diferencia que se usa para entornos web como patrón por excelencia” (Gómez, 2011).

Drupal hace uso del patrón arquitectónico MVC debido a que es el patrón por excelencia para entornos web y cuando se implementan sistemas donde se requiere el uso de interfaces de usuario. El Modelo, contiene la lógica del negocio de la aplicación, la Vista muestra al usuario la información que este necesita y el Controlador recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.

En Drupal este patrón arquitectónico está implementado como se muestra en la Ilustración 4.

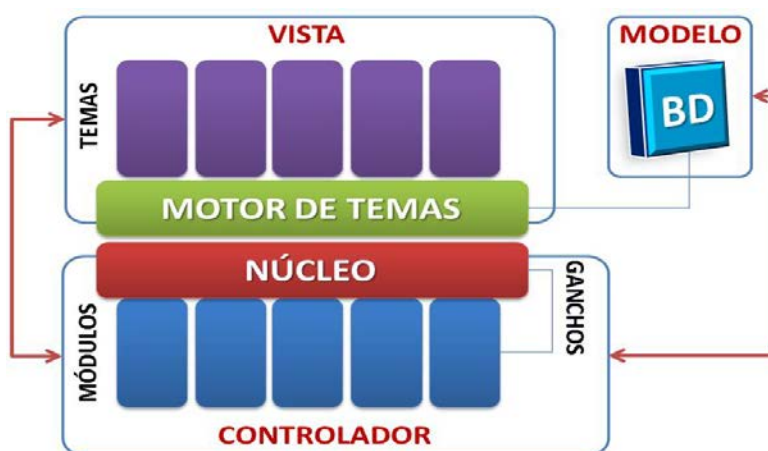


Ilustración 4: MVC en Drupal (Butcher et. al 2013).

En la utilización que hace Drupal del patrón arquitectónico MVC, no existe interacción directa entre la Vista y el Modelo, esto se debe a que dichas interacciones siempre se realizan a través de la lógica del negocio o el Controlador.

3.2. Estructura de Drupal

En lugar de considerar una aplicación web como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en un conjunto de elementos básicos. Estos son plantillas, usuarios y permisos, bloques y menús, módulos y base de datos.

Capítulo 3: Implementación y Pruebas

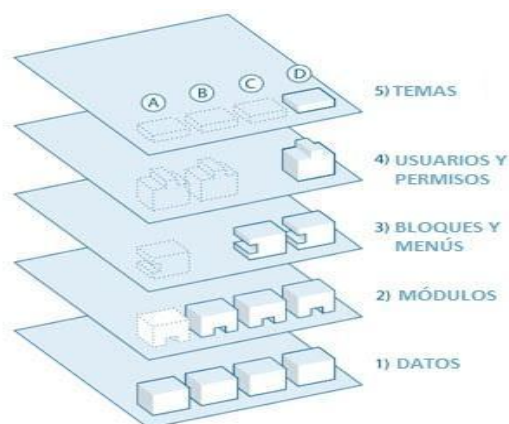


Ilustración 5: Estructura de Drupal (Butcher et. al 2013).

3.2.1. Temas en Drupal

El investigador Rodríguez refiere que un tema “establece la apariencia gráfica o estilo en que la información se mostrará al usuario” (Rodríguez, 2011). Por otro lado Butcher lo define como “conjunto de archivos inter-relacionados responsables del aspecto visual. El empleo del patrón MVC y el uso de temas en Drupal hacen que este sea excepcionalmente flexible a la hora de trabajar con la interfaz, debido a que las funcionalidades del sitio están en gran medida desvinculadas de la presentación y es muy fácil cambiar el aspecto visual” (Butcher et al. 2013). Es posible descargar temas desarrollados, personalizar un tema existente o desarrollar uno propio. Los temas incluyen algunos aspectos del desarrollo web, como el trabajo con hojas de estilo, código JavaScript y/o código PHP.

Los temas en Drupal están compuestos por diferentes tipos de ficheros, incluyendo aquellos que tienen formato *.tpl.php* y se conocen como plantillas, estos controlan la estructura de los elementos que serán mostrados en pantalla y aplicando CSS garantizan la disposición y visualización específica de nodos, bloques u otros contenidos. Las plantillas de un tema incluyen largas secciones de HTML y pequeños fragmentos de código en lenguaje PHP que serán reemplazados por contenido dinámico cuando se construya la página. Cambiar el HTML u otra etiqueta de estas secciones requiere conocimiento del sistema de capas de Drupal, lo que marca una diferencia significativa entre construir un tema de Drupal y un sitio estático en HTML.

Para el desarrollo del tema de la propuesta de solución, primeramente se partió del

Capítulo 3: Implementación y Pruebas

prototipo de interfaz de usuario definido en el capítulo anterior, para tener una idea clara de la ubicación de los bloques y contenidos en el sistema. El autor modificó un tema que se ajustaba en gran manera a las necesidades del sistema. El tema modificado posee un diseño adaptable al entorno del usuario, es un tema ligero, para su confección se hace uso de CSS y JavaScript. A este tema se le puso por nombre “agenda” y se aloja en el directorio *sites/all/themes/agenda*.

Para facilitar el trabajo con las hojas de estilo y elementos HTML se utilizó la extensión DrupalForFirebug²⁰ en su versión 2.2.1 del navegador web Firefox 36. Drupal facilita la instalación directa de temas desde Internet, sin embargo en este caso se instala el tema “agenda” de forma manual, puesto que fue el que se modificó para la presente tesis. La instalación manual, requiere privilegios adecuados para mover ficheros en la estación de trabajo que se esté utilizando o en el servidor. Con estos permisos la carpeta del tema se debe ubicar en *sites/all/themes* y con privilegios administrativos aparecerá una vez instalado el sistema en la dirección *admin/appearance*, o navegando hacia el vínculo “Apariencia” en el menú que provee el módulo *Dashboard* en la parte superior del sistema. En esa página se muestra una lista de todos los temas disponibles a instalar; se selecciona la opción habilitar y cambiar por defecto, entonces el tema inmediatamente se actualiza. En la Ilustración 6 queda reflejado el diseño de la interfaz del tema en la página principal.



Ilustración 6: Tema del sistema.

²⁰ Extiende la extensión Firebug añadiendo a Drupal depuración específica y mensajes de estado. Herramienta ideal para desarrolladores de temas en Drupal.

Capítulo 3: Implementación y Pruebas

3.3. Creación de las tablas adicionales en la base de datos

Por defecto, la instalación estándar de Drupal crea 74 tablas que son empleadas por el CMS para almacenar y manejar datos de nodos, usuarios, bloques, menús, contenidos, etc. Los módulos implementados siguen completamente la estructura de Drupal y utilizan el *core* del CMS, permitiendo la gestión de tablas de la base de datos a nivel de *core*. La gestión de base de datos es posible mediante el trabajo con las API del módulo *node* y *field*. También, se hace uso de las tablas *node*, *field*, *rules* y las variables de almacenamiento implementadas por el módulo *Views*.

3.4. Fase de implementación

En esta fase se implementan las HU definidas anteriormente, como parte del cumplimiento de los requerimientos del sistema. Al principio de esta actividad se realiza la revisión del plan de iteraciones y se modifica en caso de ser necesario. Mediante este plan se descomponen las HU en tareas de desarrollo y se asignan responsables de implementación a cada una de las tareas. Estas tareas son para el uso exclusivo de los programadores, definiéndose en lenguaje técnico y no necesariamente entendible por el cliente. La descripción de las tareas de ingeniería correspondientes a cada iteración, pueden ser consultadas en el [Anexo #6: Tareas de implementación](#).

3.5. Módulos en Drupal

Una de las capas que conforman Drupal son los módulos (*modules*) que operan sobre los nodos y que le otorgan funcionalidades permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada aplicación web. Son extensiones que se instalan en las aplicaciones otorgándole nuevas funcionalidades.

Un colectivo de autores encabezado por Butcher, define un módulo en Drupal como *“la unión de varias funciones escritas en PHP que permiten dotar al sitio de un mayor número de funcionalidades. Las funciones actúan como enganche a la hora de construir una página web y gestionar el contenido. Los módulos poseen una arquitectura basada en ganchos (hooks), los cuales son la vía más común de adicionar una nueva funcionalidad al núcleo de Drupal, llamados por este en el momento apropiado”* (Butcher et al. 2013).

Los módulos suelen clasificarse en tres tipos: los módulos del núcleo que son los que vienen en la instalación por defecto de Drupal, los módulos contribuidos que son los desarrollados por la comunidad que da soporte a Drupal y los personalizados que son los creados por los desarrolladores de un sistema específico.

Capítulo 3: Implementación y Pruebas

Para crear un módulo en Drupal son necesarios los ficheros de tipo PHP, uno de ellos de extensión *.info*; para portar la información que es pertinente al módulo como tal, la versión y el autor del módulo. Otro fichero es la extensión *.install*, para crear entre otros las tablas; y por último un fichero de extensión *.module*; que será donde se escriban las funciones que actúen como enganche al ser llamadas por Drupal a la hora de construir una página web y gestionar el contenido. Ejemplo de estas funciones descritas son: *hook_help*, *hook_perm*, *hook_menu*. Los módulos se pueden desactivar temporalmente para reducir la carga del servidor.

3.5.1. Módulos empleados y desarrollados en la solución

A continuación se ofrece una breve explicación de aquellos módulos que implementan funcionalidades necesarias para el sistema GESTA, aunque se estudiaron muchos otros para determinar cuáles podrían ser utilizados.

De las clasificaciones conocidas de los módulos en la presente solución se incluyen 11 de los incluidos en el núcleo de Drupal, estos son: *Filter*, *Watchdog*, *Node*, *Block*, *User*, *System*, *Path*, *Menu*, *Upload*, *Locale*, *Taxonomy* y *Search*. Por otra parte dentro de los módulos contribuyentes incluidos se encuentran: *LDAP Integration*, *beautytips*, *calendar*, *ckeditor*, *jquery_plugin*, *multiselect*, *ctools*, *date* y *views*. Las nuevas funcionalidades contenidas en el sistema GESTA están incluidas en 3 módulos desarrollados que se explicaron anteriormente.

3.5.2. Patrones de diseño

Para el funcionamiento de Drupal se emplean un grupo de patrones de diseño, entendido este concepto como, una solución a un problema común en el desarrollo de software aplicable a otros contextos referentes al diseño. Los patrones de diseño están embebidos dentro de la arquitectura de Drupal. Estos son propios de sistemas orientados a objetos y se les suele llamar patrones GOF (*Gang of Four*).

Algunos estudiosos del tema indican que los patrones GOF “*brindan soluciones concretas ya que proponen soluciones a problemas concretos, no son teorías genéricas y son soluciones técnicas pues indican resoluciones basadas en la Programación Orientada a Objetos (POO)*” (Gamma et al. 1994). A su vez, estos patrones suelen clasificarse en tres categorías basadas en su propósito: creacionales, estructurales y de comportamiento. En la implementación de los módulos empleados en el sistema GESTA están reflejados los patrones de diseño: *Bridge* (puente), *Chain of Responsibility* (cadena de

Capítulo 3: Implementación y Pruebas

responsabilidades), *Command* (comando), *Decorator* (decorador) y *Singleton* (instancia única), lo cual no indica que hayan sido empleados conscientemente, sino por la manera particular que establece Drupal para la escritura de sus módulos, lo que además hace que el código sea más elegible, fortaleciendo la práctica de XP que propone la utilización de estándares de codificación. A su vez, se favorece en la obtención de un diseño elegante y robusto de un producto web desarrollado con el CMS Drupal.

3.6. Pruebas de software

Para el autor Pressman, las pruebas *“son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática, para lo que se debe definir una plantilla para las pruebas de software (conjunto de pasos en que se pueden incluir técnicas y métodos específicos del diseño de casos de prueba)”* (Pressman, 2005). Para lograr que un software cumpla con todos los requerimiento propuestos por el cliente, debe someterse a una serie de pruebas o exámenes, con el fin de descubrir errores en el funcionamiento externo o interno del sistema. XP enfatiza mucho de los aspectos relacionados con las pruebas, clasificándolas en: unitarias y de aceptación. El investigador Echeverry refiere que *“en las pruebas de aceptación el usuario juega un papel fundamental seleccionando los casos de prueba para cada HU e identificando los resultados esperados, en las unitarias no tiene ninguna intervención, son creadas por los programadores y verifican el código de manera automática”*(Echeverry, 2007). A continuación se especifican los dos tipos de pruebas que se aplicaron.

3.6.1. Pruebas unitarias

Las pruebas unitarias o pruebas de unidad según el autor Joskowicz son la piedra angular de la metodología XP, debido a que, *“todos los módulos deben pasar estas pruebas antes de ser liberados o publicados. Estas pruebas deben ser definidas antes de realizar el código, y se guardan junto a este en caso de futuras correcciones y actualizaciones. Cuando se encuentren errores, estos deben ser corregidos inmediatamente, y se deben definir nuevas pruebas para verificar que el error haya sido resuelto”* (Joskowicz, 2008).

Las pruebas unitarias se aplicaron sobre las funcionalidades críticas que más se utilizan y otras que se implementaron en los módulos generados para ciertas especificaciones, entre ellas verificar los permisos de usuario sobre un nodo, la cantidad de menús generados por un módulo así como manejo de los bloques en las regiones del tema y su

Capítulo 3: Implementación y Pruebas

configuración. De acuerdo a los módulos implementados y sus funciones varias, la interacción con el núcleo y la forma de obtener respuesta que es similar; se decide ejecutar las pruebas unitarias sobre cada uno de los módulos desarrollados

Para realizar las pruebas unitarias se utilizó *SimpleTest* que es un marco de pruebas unitarias de código abierto. A su vez, proporciona algunas pruebas básicas, otras se pueden escribir pruebas para módulos personalizados, formularios, etc. Las pruebas emplean archivos de prueba y se deben colocar en el directorio del módulo que se está probando. En Drupal 7, *SimpleTest* está incorporado en el núcleo como el módulo de prueba, y todos los módulos del núcleo presentan un conjunto de pruebas. Para el desarrollo de estas pruebas automatizadas empleando *SimpleTest* se tuvo en cuenta el capítulo 2, del libro *Drupal 7 Module Development*, el capítulo 8 del libro *Pro Drupal por 7 Windows Developers*, y el capítulo 25 del libro *Pro Drupal 7 Development*.

Como resultado de las pruebas en *SimpleTest* el sistema arrojó los siguientes resultados: 143 pases, 0 fallo, o excepciones y 18 mensajes de depuración. Una muestra de cómo se implementaron las pruebas unitarias y los resultados que se obtuvieron luego de aplicarlas, pueden ser visualizados en el [Anexo #7: Pruebas unitarias](#).

3.6.2 Pruebas de aceptación

El autor (**Joskowicz, 2008**) plantea que las pruebas de aceptación son creadas en base a las HU definidas, y son el equivalente a las pruebas de caja negra definidas bajo la metodología RUP. El cliente debe definir diversos escenarios para comprobar que una HU ha sido correctamente implementada. Son los clientes los responsables de verificar que los resultados de estas pruebas sean correctos. Se hace necesario publicar los resultados de las pruebas de aceptación, para que todo el equipo de desarrollo esté al tanto de las mismas. Para la ejecución de estas pruebas se especificó el diseño de casos de pruebas del sistema para cada iteración, correspondientes a cada HU, los cuales pueden ser consultados en el [Anexo # 8: Diseño de Casos de Prueba](#).

3.7. Resultados obtenidos

Las pruebas de aceptación se realizaron en tres iteraciones, con el fin de verificar que las funcionalidades para cada entrega del software, arrojaran el resultado esperado. En las iteraciones se obtuvieron los siguientes resultados:

- **Primera iteración:** 10 No Conformidades (NC), de ellas 5 significativas, 3

Capítulo 3: Implementación y Pruebas

no significativas y 2 recomendaciones.

- **Segunda iteración:** 8 NC, de ellas 5 significativas, 2 no significativas y 1 recomendación.
- **Tercera iteración:** 3 NC, de ellas 2 significativas, 1 no significativa y sin recomendaciones.

De manera general, las no conformidades, significativas se refieren a errores de validación y cambios en el diseño y las no significativas se refieren a errores ortográficos: como omisiones de tildes y cambio de mayúscula por minúsculas.

3.8. Conclusiones parciales del capítulo

Al finalizar el presente capítulo se obtuvo un sistema totalmente funcional, cumpliendo con los principales procesos del negocio que se plantearon para su desarrollo. Para lograr este objetivo se tuvo en cuenta los siguientes aspectos:

1. Se definió la arquitectura por la que está compuesta el sistema, así como los patrones de diseño que se utilizaron para su implementación. El empleo de estos patrones logró establecer un uso adecuado de buenas prácticas en la implementación del sistema.
2. Se definieron las principales tareas de programación que dieron paso a la implementación del sistema, creando la base central de la aplicación y describiendo cómo los elementos del diseño se implementan y organizan.
3. Una vez desarrolladas las funcionalidades previstas, se ejecutó un proceso de pruebas de software, donde se definieron los tipos de pruebas a aplicar en el sistema, con el objetivo de detectar y corregir fallas en el funcionamiento del sistema, aspecto determinante para obtener un producto de calidad.

Conclusiones Generales

Conclusiones generales

Con la culminación de la presente investigación se arribaron a las siguientes conclusiones:

1. Se realizó un estudio de las necesidades de los Sistemas de gestión documental a nivel nacional e internacional, evidenciando la necesidad de crear un sistema informático para mejorar la gestión de la Agenda de trabajo del Consejo de dirección de la Facultad 4, mediante la utilización de herramientas libres y multiplataforma.
2. Se construyó un sistema que brinda la posibilidad de gestionar la agenda de trabajo de las reuniones de Consejo de dirección, citación de los involucrados y la visualización de los informes presentados, satisfaciéndose las necesidades del cliente.
3. Se realizaron las pruebas al sistema mediante pruebas unitarias con 21 no conformidades en tres iteraciones y pruebas de aceptación con 24 casos de prueba diseñados, lo que permitió mejorar la calidad del sistema al solucionar los errores detectados.
4. Después de la entrega de la propuesta de solución al cliente, se pudo constatar que este está satisfecho y el sistema puede implantarse en el Consejo de dirección de la Facultad 4 para gestionar su agenda de trabajo.

Recomendaciones

Una vez concluido el desarrollo del presente trabajo de diploma se enuncian las recomendaciones a tener en cuenta para futuros trabajos, tomando como referencia el actual:

1. Perfeccionar el maquetado del acta cuando se genera y está en proceso de impresión.
2. Lograr un mejor seguimiento de las actividades que se planifican en el plan de temas anuales del Consejo de dirección.

Glosario de términos

API de Drupal: *Application programming interface*, conocidas como Interfaz de programación de aplicaciones no es más que una librería de funciones que se pone a disposición de los desarrolladores para acceder a determinados datos del sistema, sin necesidad de conocer cómo se organiza el sistema internamente.

Bloques: Los bloques son elementos del CMS Drupal que se emplean para incluir información, ya sea menú o cualquier otra función que se desee, casi siempre aparece en las columnas izquierda y derecha de la página.

CMS: Content Management System, sistema que facilita la gestión de contenidos en todos sus aspectos: creación, publicación y presentación.

Código Abierto: Es una tendencia internacional del desarrollo de software que profesa la distribución del código junto a las aplicaciones, se rigen por licencias GNU/GPL.

CSS: Las hojas de estilo en cascada (en inglés Cascading Style Sheets) contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una web. Se llama “cascada” porque una hoja puede heredar los formatos definidos en otra hoja de tal forma que no hace falta que vuelva a definirlos. Estas hojas permiten la separación entre el contenido y la presentación de un sitio web.

Frameworks: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y metodología de trabajo la cual extiende o utiliza aplicaciones del dominio.

Hook: Del inglés gancho, palabra usada en la comunidad de Drupal para nombrar los métodos de la interfaz que deben implementar los módulos desarrollados para este CMS.

HTML: *HyperText Markup Language* (Lenguaje de marcas de hipertexto). Es el método más común de intercambio de información en la *World Wide Web*, mediante el cual se transfieren las páginas web a un ordenador.

HTTP: *HyperText Transfer Protocol* (Protocolo de transferencia de hipertexto). Es el protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, vídeo y otros archivos multimedia) en la world wide web.

JavaScript: Lenguaje de programación interpretado, utilizado principalmente en páginas web, con una sintaxis semejante al lenguaje Java y el lenguaje C. Actualmente todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web.

LDAP: *Lightweigh Directory Access Protocol*, Protocolo Ligero de Acceso a Directorios es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido.

Licencia GNU/GPL: En español Licencia Pública General, es una licencia creada por la *Free Software Foundation* y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre.

Metodología: Son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software.

Multiplataforma: Programa o aplicación que puede utilizarse sin inconvenientes en distintas plataformas de hardware y sistemas operativos.

PHP: *Hypertext Preprocessor* es un lenguaje multiplataforma, *script* (no se compila para conseguir códigos máquina sino que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene este código) para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto y a que es de código.

Proceso de Pruebas: Es donde se desarrollan todas las actividades de la fase de pruebas dentro del software como son: planificación de las pruebas, diseño de las

Glosario de términos

pruebas, implementación de las pruebas, ejecución de las pruebas y evaluación de los resultados.

Proceso: Secuencia de actividades invocadas para producir un producto de software.

Producto de Software: Conjunto de programas de computadora, procedimientos y posiblemente documentación y datos asociados.

Producto: Cualquier software que será construido a petición de otros.

SCRUM: Es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Software: Componentes intangibles de una computadora, es decir, conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.

TIC: Tecnologías de la Información y la Comunicación, se encargan del estudio, desarrollo, implementación y distribución de la información mediante la utilización de hardware y software como medio de sistema informático.

Unix: Sistema operativo atribuido a Ken Thompson y Dennis Ritchie y comercializado en la década de los 70 que alcanzó éxito, sobre todo en las universidades y posteriormente en las empresas. Entre sus principales características se encuentran: su código corto, simple, claro, modular y extensible, que puede ser mantenido y reutilizado con otros propósitos por otros desarrolladores que no sean los creadores originales; es portable, multitarea, y multiusuario. Su éxito se debe en gran parte a que no fue diseñado para cumplir ningún objetivo predefinido.

Web: Es un sistema para presentar información en Internet basado en hipertexto.

XHTML: *Extensible HyperText Markup Language* (lenguaje extensible de marcado de hipertexto), es el lenguaje para sustituir a HTML como estándar para las páginas web.

Referencias bibliográficas

DANTE, Gloria Ponjuan. Gestión documental, gestión de información y gestión del conocimiento: evolución y sinergias. Comunicación preliminar. *Ciencias de la Información*, 2005, vol. 36, no 3.

LATINO, Robert J.; LATINO, Kenneth C.; LATINO, Mark A. *Root cause analysis: improving performance for bottom-line results*. CRC Press, 2011. Disponible en [\[http://www.leanexpertise.com/TPMONLINE/articles_on_total_productive_maintenance/rootcause/rootcauseqltyofprocess02.htm\]](http://www.leanexpertise.com/TPMONLINE/articles_on_total_productive_maintenance/rootcause/rootcauseqltyofprocess02.htm)

YIGOTSKY, L. Pensamiento y Lenguaje Buenos Aires, Pléyade, 1985.

DÍAZ-BALART, F. Ciencia, innovación y futuro. Barcelona. España, Grupo Editorial Random House Mondadori. ISBN: 84-253-3650-3, 2002.

ALAVI, et al. Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly* 25 (1): 107–136. doi: 10.2307/3250961. JSTOR 3250961, 2001.

SÁNCHEZ, M et al. La gestión del conocimiento y su relación con otras gestiones. Vol. 37, 2006.

NONAKA, I et al. Teoría de la creación del conocimiento organizacional. La dimensión ontológica y la dimensión epistemológica, 1999.

RUESTA, Carlota Bustelo. Inforarea. [En línea]. 2008. [Disponible en: <http://www.inforarea.es/Documentos/GC.pdf>.]

GONZÁLEZ, F et al. Propuesta de Modelo para la Gestión del Conocimiento y la Información en el Centro de Tecnologías de Almacenamiento y Análisis de Datos (CENTALAD). 135. p. La Habana, UCI, 2009.

SÁNCHEZ, B. Gestión y uso integral de la información en la administración pública municipal cubana. La Habana: Facultad de Comunicación, 2002.

CITMA. Bases para la introducción de la gestión del conocimiento en Cuba, 2002.

GOÑI CAMEJO, Ivis. Algunas reflexiones sobre el concepto de información y sus implicaciones para el desarrollo de las ciencias de la información. *Acimed*, 2000, vol. 8, p. 201-207.

LANZILLOTTA, Analía. Mastermagazine. Definición de Información. [En línea] 2004. [Citado el: 10 de febrero de 2009.] <http://www.mastermagazine.info/termino/5366.ph66.php>.

Referencias y Bibliografía

FERNÁNDEZ, Norma M. Barrio; BALBÓN, María Aurora Soto. Gestión del conocimiento. Parte I. Revisión crítica del estado del arte. *Acimed: revista cubana de los profesionales de la información y la comunicación en salud*, 2006, vol. 14, no 2, p. 3.

CAPOTE MARRERO, Belina; GONZÁLEZ MACHÍN, Diego; RODRÍGUEZ Durán, Emma. La gestión de información como herramienta fundamental en el desarrollo de los centros toxicológicos. *Acimed*, 2003, vol. 11, no 2, p. 5-6.

HEREDIA HERRERA, Antonia. Gestión de documentos y administración de archivos. *Códices*, 2008, vol. 4, no 2, p. 43-50.

GALENDE DÍAZ, Juan Carlos, GARCÍA Ruipérez, Mariano. El concepto de documento desde una perspectiva interdisciplinaria: de la diplomática a la archivística. [En línea] 2003. [Disponible en: <http://dialnet.unirioja.es/servlet/articulo?codigo=838229>].

BJÖRK, Bo-Christer. The impact of electronic document management on construction information management. En *Proc of the International Council for Research and Innovation in Building and Construction, CIB W78 Conference*. 2002. p. 12-14.

FREIRE, Luis. Estudio de factibilidad para la implementación del Sistema de Gestión Documental "Quipux" en la Universidad Regional Autónoma de los Andes. *UNIANDÉS EPISTEME*, 2014, vol. 1, no 1.

FORCADA Matheu, Núria, et al. Life cycle document management system for construction. 2005.

RENDÓN VELOSA, Ana María, et al. Diseño de un modelo de gestión documental en la Universidad Libre basado en el sistema de gestión de calidad ISO 9001: 2008 e ISO 15489: 2001. 2015.

SHARIFF, Munwar. *Alfresco 4 Enterprise Content Management Implementation*. Packt Publishing Ltd, 2013.

BÁEZ ÁLVAREZ, E. J. Babel, sistema automatizado para la gestión de información para los servicios de traducción e interpretación. *Tono, revista técnica de la Empresa de Telecomunicaciones de Cuba SA*, 2005, vol. 3, no 5, p. 45-50.

BADILLO Juan Carlos, MARTÍNEZ Ramón. 2012. AvilaDOC. Control de documentos. [En línea]. [Disponible en: <http://www.expomatanzas.cu/index.php/ofertas/avila-doc>]

VIERA Lorenzo, Yanay, MUSTELIER Sanchidrian Daimara. 2007. Gestión documental del Sistema de Gestión Penitenciaria. [En línea]. [Disponible en: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_0444_07/1/TD_0444_07.pdf]

Referencias y Bibliografía

NAVARRO PÉREZ, Zuleyda , REDONDO González Anett. 2011. Sistema para la gestión de documentos electrónicos correspondientes al Área de Calidad de la Dirección de Producción Audiovisual (DPA), perteneciente al Centro de Tecnologías para la Formación (FORTES) de la Facultad 4. [En línea]. [Disponible en: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_04592_11/1/TD_04592_11.pdf]

GÓMEZ MÁRQUEZ, José Jorge. Experiencias de un ingeniero. [En línea] 2011. [Disponible en: <http://jorge.queideas.com/wp-content/uploads/2011/11/Arquitectura-MVC.pdf>.]

MORALES, Cristóbal Romero, SOTO, Sebastián Ventura, MARTÍNEZ, Cesar Hervás. *Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web.* Córdoba: Departamento de Informática y Análisis Numérico, 2005.

AVISON, D.E. y FITZGERALD, G. *Information Systems Development: Methodologies, Techniques, and Tools.* s.l.: McGraw-Hill, 1995.

SUÁREZ Jorge, Alinoet . *Estudio Comparativo sobre 11 Metodologías de Desarrollos Web.* La Habana: s.n., 2009.

Newkirk, J., Martin R.C. "Extreme Programming in Practice". Addison-Wesley. 2001

LETELIER, P. y Penadés, M^a C. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Valencia: Universidad de Valencia, 2008. Vol. 05, 26. ISSN 1666-1680.

CANÓS, JOSÉ H., LETELIER, PATRICIO y PENADÉ, MARÍA DEL CARMEN. *Metodologías Ágiles en el Desarrollo de Software.* s.l.: Universidad Politécnica de Valencia, 2005.

HIGHSMITH, Jim. *Agile software development ecosystems.* Addison-Wesley Longman Publishing Co., Inc., 2002.

KHRAMTCHENKO, Serguei, 2004, Comparing eXtreme Programming and Feature Driven Development in academic and regulated environments. Final paper for CSCIE-275: Software Architecture and Engineering. Harvard University.

BECK, Kent. *Embracing Change with Extreme Programming.* s.l.: Addison Wesley Longman, Inc, 1999. ISBN-10: 0201616416 | ISBN-13: 978-0201616415.

Referencias y Bibliografía

JOSKOWICZ, José. Reglas y prácticas en eXtreme Programming. Universidad de Vigo, 2008. [Disponible en: <http://ie.fing.edu.uy>]

HIDALGO URBINO, Rafael Jacobo, 2010, Análisis, diseño e implementación de una herramienta que permita la centralización de la información gestionada por el módulo Resultados de la Colección Multisaber. [En línea]. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana, Cuba: Universidad de las Ciencias Informáticas. [Disponible en: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_02857_10/1/TD_02857_10.pdf.]

SHORE, James, WARDEN, Shane. *The Art of Agile Development.* Estados Unidos: O'Reilly Media, Inc., 2008. ISBN-13: 978-0-596-52767-9.

RIOLA, Jose Carlos Carvajal. *Metodologías Agiles:Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial.* Barcelona: UPC - Barcelona, 2008.

GRENNING, J. Launching XP at a Process-Intensive Company. s.l.: IEEE Software, 2001. Vol. 18.

GJONI, Oskeol. Comparison of Two Model Driven Architecture Approaches for Automating Business Processes, Moskitt Framework and Bizagi Process Management Suite. *Mediterranean Journal of Social Sciences*, 2015, vol. 6, no 2, p. 615.

SERRANO-COBOS, Jorge. Evolución de los sistemas de gestión de contenidos (CMS). Del mainframe al open source. 2007. Vol. 16, 3. p. 213-215.

GIL, F. *Experto en Drupal 7.* s.l.: Forcontu S.L., 2011.

CUERDA GARCÍA, Xavier. *Mosaic. Introducción a los sistemas de gestión de contenidos (CMS) de código abierto.* [En línea] 2004. [Disponible en: <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>. ISSN: 1696-3296].

RODRÍGUEZ, Fran Gil. Experto en Drupal 7. Nivel Avanzado. s.l. : Forcontu S.L., 2011. 978-84-939410-2-4. —. Experto en Drupal 7. Nivel Inicial. s.l. : Forcontu S.L., 2011. 978-84-939410-0-0.

BUTCHER, M., y otros, y otros. *Drupal 7 Module Development.* Birmingham,Mumbai: s.n., 2013. ISBN 978-1-849511-16-2.

NOBLE, Mark. *Drupal 7 First Look.* Packt Publishing Ltd, 2010.

DRUPAL.ORG. Open Source CMS Portal del CMS Drupal. [En línea] 2015. [Disponible

Referencias y Bibliografía

en: <http://www.drupal.org>]

EGUÍLUZ PÉREZ, Javier. Introducción a JavaScript. Madrid: Autoedición, 2009.

PADILLA Matos, Reynier. Programación Web y Tecnologías Informáticas. [En línea] 2008.

ÁLVAREZ, Miguel Angel. Manual de jQuery. [En línea] 2009. [Disponible en: <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>]

MUSCIANO, C. y KEMEDY, B. HTML. La Guía Completa. 2014. Segunda ed. ISBN 1-56592-235.

VAN LANCKER, Luc. *HTML5 y CSS3: Domine los estándares de las aplicaciones Web [2ª edición]*. Ediciones ENI, [En línea]. 2013. Disponible en: [https://books.google.com/cu/books?id=hpW82IMDmxkC&printsec=frontcover&dq=isbn:2746084066&hl=es&sa=X&ei=QeNIVc_5D7GOsQSF5IDYCA&ved=0CB0Q6AEwAA#v=onepage&q&f=false]

RIVERA, E.A., ZAMORA, R.G. y SORIA, M.G. *Sistema de Educación a Distancia*. s.l.: IV Congreso de Tecnología en Educación, 2012.

DE LUCA, Damián y De Luca, A.A. *CSS3 & HTML5. Los nuevos estándares para el diseño y desarrollo web*. [En línea] 2012. [Disponible en: <http://css3html5.com.ar/>].

TRAVIS, B. *Drupal 7 for Windows Developers*. s.l.: Apress, 2011.

CONVERSE, Tim, Park, Joyce y Morgan, Clark. *PHP5 and MySQL Bible*. Indiana: Wiley Publishing, Inc, 2004. ISBN: 0-7645-5746-7.

BAKKEN, S. S. y AULBACH, A. Manual de PHP. 2015. Grupo de documentación de PHP, [Disponible en: www.php.net]

KADASOFTWARE. KadaSoftware. [En línea] Universidad Tecnológica de la Mixteca, 2013. [Disponible en: <http://www.kadasoftware.com/index.php/tecnologias-de-desarrollo.html>].

ASENJO Sánchez, Jorge. *Sistemas Gestores de Bases de Datos*. España: Creative Commons, 2009.

POSTGRESQL: Dossier de Prensa de PostgreSQL 9.3, 2014. [En línea]. [Accedido 4 de octubre del 2014]. [Disponible en : <http://www.postgresql.org/about/press/presskit93/es/>]

CASILLAS Santillán, Luis Alberto , GINESTA Marc Gibert, PÉREZ Mora Óscar. Universitat Oberta de Catalunya. Universitat Oberta de Catalunya. [En línea] 2011. [Disponible en: http://ocw.uoc.edu/computer-science-technology-and-multimedias-bases-de-datos/bases-de-datosp06_m2109_02151].

Referencias y Bibliografía

GINESTA Marc Gibert, Oscar Pérez Mora. Scribd. Scribd. [En línea] 2011. [Disponible en: <http://es.scribd.com/doc/65741986/postgreSQL>].

GONZÁLEZ Barahona, J.M., PASCUAL Seoane, Joaquín, ROBLES Gregorio. Introducción al software libre. [En línea] 2003. [Disponible en: <http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/sec-ide.html>].

NETBEANS.ORG. Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto. [En línea] 2011. [Disponible en: http://netbeans.org/index_es.html].

REA Inca, FERNANDO Paúl. *Diseño y Desarrollo de un Prototipo de Control mediante SMS para Casas Inteligentes*. Sangolquí: s.n., 2012.

DUQUE González, Raúl. Mundo geek. [En línea] 2011. [Disponible en: <http://mundogeek.net/etiqueta/servidor-web>].

HU, James C.; PYRALI, Irfan; SCHMIDT, Douglas C. Measuring the impact of event dispatching and concurrency models on web server performance over high-speed networks. En *Global Telecommunications Conference, 1997. GLOBECOM'97., IEEE*. IEEE, 1997. p. 1924-1931.

KABIR, Mohammed J. *La Biblia Servidor Apache 2*. s.l.: Anaya Multimedia, 1999. ISBN: 8441514682.

NETCRAFT: Internet Research. [En línea]. 2015. [Disponible en: <http://news.netcraft.com/>]

GREENSTEIN, Shane; NAGLE, Frank. Digital dark matter and the economic contribution of Apache. *Research Policy*, 2014, vol. 43, no 4, p. 623-631.

ALFRESCO, Alfresco Software © 2014. Acerca de Alfresco. Todos los derechos reservados. [Disponible: <http://www.alfresco.com/es/acerca-de-alfresco>].

SHARIFF, Munwar, Vinita CHOUDHARY, Amita BHANDARI a Pallika MAJMUDAR, 2009. *Alfresco 4 Enterprise Content Management Implementation* [En línea]. Copyright © 2009 Packt Publishing. B.m.: Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK. 2009. Production Reference: 1020609. ISBN ISBN 978-1-847197-36-8. [Disponible:http://sunshine.prod.uci.cu/gridfs/sunshine/books/Alfresco_3_Enterprise_Content_Management_Implementation]

FONSECA Mata, Misael, SUÁREZ Michel David, SÁNCHEZ Góngora Marcel Rodolfo . 2012. EXCRIBA, GESTOR DE DOCUMENTOS ADMINISTRATIVOS [En línea]. ISSN ISBN: 978-959-286-019-3. [Disponible:

Referencias y Bibliografía

http://repositorio_institucional.uci.cu/jspui/bitstream/ident/4116/1/uciencia-2012-t46-p945-ponencia-1392.pdf

GAMMA, Erich, et al. *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1994.

PRESSMAN, ROGER S., 2005, **Ingeniería del Software. Un enfoque practico**. Sexta Edición. McGraw-Hill Interamericana. ISBN 970-10-5473-3. Disponible en: http://eva.uci.cu/mod/resource/view.php?id=9302subdir=/Pressman_6ta_Edicion

ECHEVERRY Tobón, Luis Miguel y Delgado Carmona, Luz Elena. *Caso práctico de la metodología XP al desarrollo de software*. Pereira: Universidad Tecnológica de Pereira, 2007.

Anexo 1: Flujo de trabajo "Plan de temas" del proceso Agenda de trabajo del Consejo de dirección

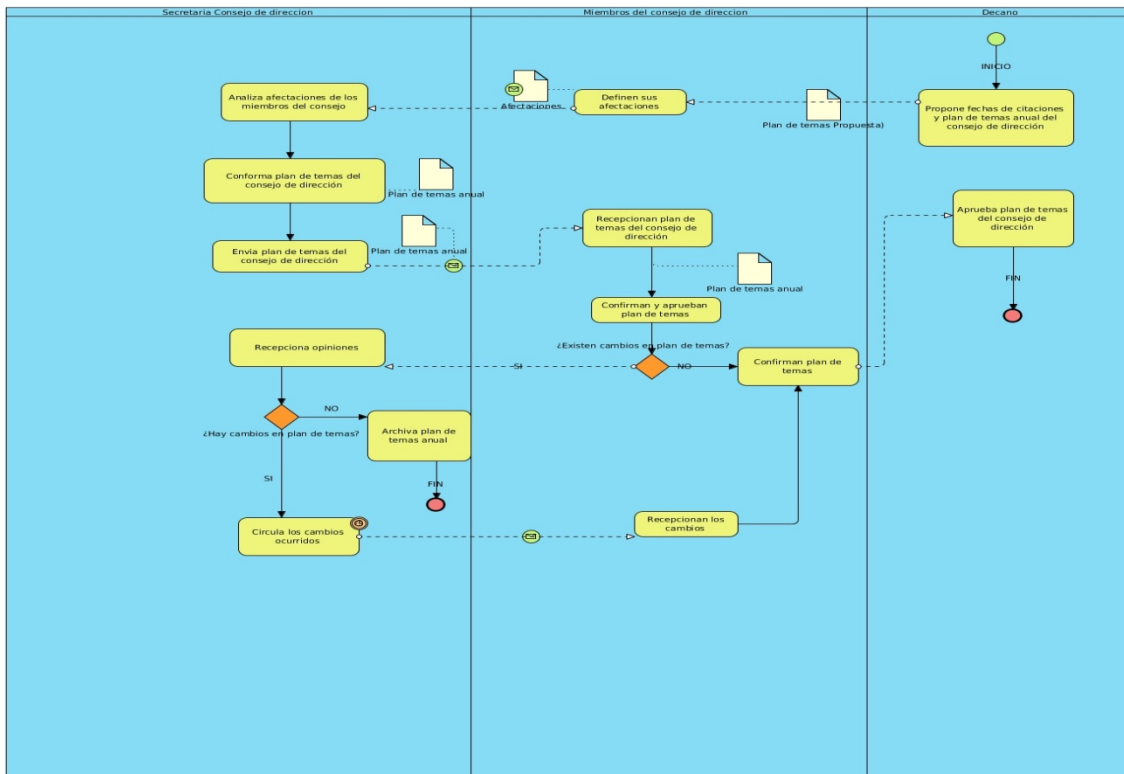


Ilustración 7: Flujo de trabajo "Plan de temas" del proceso Agenda de trabajo.

Anexo 2: Historias de usuario del sistema GESTA

Tabla 4. HU Autenticar usuario en el sistema.

Historia de Usuario	
Número: 2	Nombre: Autenticar usuario en el sistema
Usuario: Administrador, Personal	
Prioridad en el negocio: Media	Puntos Estimados: 2d
Riesgo de desarrollo: Alta	Iteración asignada: 1

Descripción: Permitir a los usuarios anónimos registrarse en el sistema, se verificará el rol por el cual se le asignarán los permisos.
Observaciones:

Tabla 5. HU Gestionar citación.

Historia de Usuario	
Número: 3	Nombre: Gestionar citación
Usuario: Decana	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Permite crear y editar una o varias citaciones para el Consejo de dirección. Una citación debe contener los siguientes campos: lugar, hora, orden del día y participantes.	
Observaciones:	

Tabla 6. HU Visualizar citación.

Historia de Usuario	
Número: 4	Nombre: Visualizar citación
Usuario: Decana	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Permite que la decana visualice la citación que se le va a enviar a los miembros del Consejo de dirección indicándoles el día, hora, lugar y orden del día de la reunión.	
Observaciones:	

Tabla 7. HU Publicar informe y acta del Consejo de dirección.

Historia de Usuario	
Número: 5	Nombre: Publicar informe y acta del Consejo de dirección
Usuario: Decana, Jefes de Departamento Docente, Director del Centro FORTES, Secretaria Docente y demás miembros del Consejo de dirección.	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Permite que la Secretaria Docente publique las acta de los Consejos de dirección y que el resto de los miembros la consulten. También que se publiquen los informes que se han ido analizando en cada una de las reuniones.	
Observaciones:	

Tabla 8. HU Mostrar informe y acta del Consejo de dirección.

Historia de Usuario	
Número: 6	Nombre: Mostrar informe y acta del Consejo de dirección
Usuario: Decana, Jefes de Departamento Docente, Director del Centro FORTES, Secretaria Docente y demás miembros del Consejo de dirección.	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Muestra las actas e informes que se han presentado en el Consejo de dirección y se brinda la posibilidad de que se puedan descargar.	
Observaciones:	

Tabla 9. HU Mostrar listado de citaciones al Consejo de dirección.

Historia de Usuario

Anexos

Número: 7	Nombre: Mostrar listado de citasiones al Consejo de dirección
Usuario: Decana, Jefes de Departamento Docente, Director del Centro FORTES, Secretaria Docente y demás miembros del Consejo de dirección.	
Prioridad en el negocio: Media	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Permite que los usuarios vean las citasiones que se han ido generando en los Consejos de dirección realizados, dichas citasiones estarán almacenadas en el sistema y por cada página se mostrarán al menos 5 citasiones.	
Observaciones:	

Tabla 10. HU Gestionar acta del Consejo de dirección.

Historia de Usuario	
Número: 8	Nombre: Gestionar acta del Consejo de dirección
Usuario: Secretaria del Consejo de dirección	
Prioridad en el negocio: Alta	Puntos Estimados: 1
Riesgo de desarrollo: Media	Iteración asignada: 1
Descripción: Permite crear, ver, listar y editar una o varias actas del Consejo de dirección. Para la confección de cada acta existe una planilla que va a estar disponible en el sistema.	
Observaciones:	

Tabla 11. HU Eliminar citasiones.

Historia de Usuario	
Número: 9	Nombre: Eliminar citasiones
Usuario: Administrador	

Anexos

Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Mostrar las diferentes citas que se han enviado y permitir que el usuario seleccione la/las citas que desee eliminar.	
Observaciones: Se mostrará un mensaje al usuario cuando se haya ejecutado la acción o si existe alguna dificultad en la ejecución de esta tarea.	

Tabla 12. HU Filtrar búsqueda de usuarios y documentos por atributo.

Historia de Usuario	
Número: 10	Nombre: Filtrar búsqueda de usuarios y documentos por atributo.
Usuario: Jefes de Departamento Docente, Director del Centro FORTES, Secretaria Docente y demás miembros del Consejo de dirección.	
Prioridad en el negocio: Media	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Permite hacer una búsqueda de un usuario, acta o informe del Consejo de dirección, para ello el usuario debe introducir patrones de búsqueda como: fecha en que fue generado el acta o informe, el nombre de usuario, etc.	
Observaciones:	

Tabla 13. HU Mostrar listado de actas del Consejo de dirección.

Historia de Usuario	
Número: 11	Nombre: Mostrar listado de actas del Consejo de dirección
Usuario: Decana, Jefes de Departamento Docente, Director del Centro FORTES, Secretaria Docente y demás miembros del Consejo de dirección.	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1

Descripción: Permite que los usuarios vean las actas que se han ido generando en los Consejos de dirección realizados, las que estarán almacenadas en el sistema.
Observaciones:

Tabla 14. HU Eliminar informe y acta del Consejo de dirección.

Historia de Usuario	
Número: 12	Nombre: Eliminar informe y acta del Consejo de dirección
Usuario: Decana, Jefes de Departamento Docente, Director del Centro FORTES, Secretaria Docente y demás miembros del Consejo de dirección.	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Muestra las actas e informes que se han presentado en el Consejo de dirección y se brinda la posibilidad de que el usuario busque por el campo de la fecha el elemento que desea eliminar.	
Observaciones: Se mostrará un mensaje al usuario si existe alguna dificultad para ejecutar esta acción y se confirmará su ejecución.	

Tabla 15. HU Ver listado de miembros del Consejo de dirección.

Historia de Usuario	
Número: 13	Nombre: Ver listado de miembros del Consejo de dirección.
Usuario: Decana, Jefes de Departamento Docente, Director del Centro FORTES, Secretaria Docente y demás miembros del Consejo de dirección.	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 3
Descripción: Muestra al usuario el listado de todos los miembros del Consejo de dirección, teniendo en cuenta todos los usuarios que están registrados y forman parte del	

Consejo.
Observaciones:

Tabla 16. HU Administración de usuarios.

Historia de Usuario	
Número: 14	Nombre: Administración de usuario del sistema
Usuario: Administrador	
Prioridad en el negocio: Alta	Puntos Estimados: 2d
Riesgo de desarrollo: Alta	Iteración asignada: 3
Descripción: El sistema debe permitirle al administrador visualizar y eliminar la información personal de cada usuario del sistema.	
Observaciones:	

Tabla 17. HU Exportar a PDF acta e informe del Consejo de dirección.

Historia de Usuario	
Número: 15	Nombre: Exportar a PDF acta e informe del Consejo de dirección.
Usuario: Decana, Jefes de Departamento Docente, Director del Centro FORTES, Secretaria Docente y demás miembros del Consejo de dirección.	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 3
Descripción: Muestra al usuario la posibilidad de exportar a formato .PDF un acta o informe del Consejo de dirección que el usuario seleccione.	
Observaciones:	

Tabla 18. HU Imprimir acta e informe del Consejo de dirección.

Historia de Usuario	
Número: 16	Nombre: Imprimir acta e informe del Consejo de dirección.
Usuario: Decana, Jefes de Departamento Docente, Director del Centro FORTES, Secretaria Docente y demás miembros del Consejo de dirección.	
Prioridad en el negocio: Alta	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 3
Descripción: Brinda la posibilidad al usuario de imprimir el acta e informe del Consejo de dirección que este seleccione.	
Observaciones:	

Tabla 19.HU Apariencia.

Historia de Usuario	
Número: 17	Nombre: Apariencia
Descripción: El sistema contará con una interfaz amigable, manteniendo el formato en los navegadores.	
Observaciones: El sistema contará con enlaces bien identificados, permitiendo acceder a cualquiera de las secciones con un número mínimo de clicks y estará optimizado para una resolución de 1024x768. El diseño debe permitir el uso de colores que identifican a la facultad y no debe contar con animaciones ni imágenes pesadas que obstaculicen la rapidez de las transacciones.	

Tabla 20. HU Seguridad.

Historia de Usuario	
Número: 18	Nombre: Seguridad

Anexos

Descripción: La autenticación está garantizada a través de los roles y permisos definidos a cada usuario en el sistema.

Observaciones: Los usuarios no necesariamente deberán estar autenticados para visualizar algunos de los contenidos que brinda la aplicación. Sin embargo los administradores y miembros del Consejo de dirección se comportarán como usuarios, para realizar cambios en el sistema, tendrán que obligatoriamente estar autenticados. Cada usuario autenticado podrá realizar las operaciones correspondientes con su rol definido en el sistema.

Tabla 21. HU Condiciones tecnológicas.

Historia de Usuario	
Número: 19	Nombre: Condiciones tecnológicas
Descripción: Garantizar que el acceso al sistema se realice desde las máquinas con características óptimas.	
Observaciones: El sistema debe tener al menos 512 MB de capacidad de disco duro.	

Tabla 22. HU Condiciones legales.

Historia de Usuario	
Número: 20	Nombre: Condiciones legales
Descripción: Usar herramientas de software libre bajo las licencias GNU/GPL.	

Anexo 3: Estimación de esfuerzo por HU

Tabla 23. Estimación de esfuerzo por HU.

No.	Historia de Usuario	Estimación (semanas)
1	Gestionar usuarios del sistema	1

Anexos

2	Autenticar usuario en el sistema	1
3	Gestionar citación	3
4	Visualizar citación	2
5	Publicar informe y acta del Consejo de dirección	1
6	Mostrar informe y acta del Consejo de dirección	1
7	Mostrar listado de actas del Consejo de dirección	1
8	Gestionar acta del Consejo de dirección	2
9	Eliminar citación	1
10	Filtrar búsqueda de usuarios y documentos por atributo	2
11	Mostrar listado de actas del Consejo de dirección	2
12	Eliminar informe del Consejo de dirección	2
13	Ver listado de miembros del Consejo de dirección	1
14	Administración de usuarios del sistema	1
15	Exportar a PDF acta e informe del Consejo de dirección	1
16	Imprimir acta e informe del Consejo de dirección	1

Anexo 4: Plan de duración de las iteraciones.

Tabla 24. Plan de duración de las iteraciones.

Iteraciones	Orden de las HU a implementar	Duración Total
-------------	-------------------------------	----------------

Anexos

1	Registrar usuarios del sistema. Autenticar usuario en el sistema. Gestionar citación. Visualizar citación.	7 semanas
2	Publicar informe y acta del Consejo de dirección. Mostrar informe y acta del Consejo de dirección. Mostrar listado de citaciones al Consejo de dirección. Gestionar acta del Consejo de dirección. Eliminar citación. Filtrar búsqueda de usuario y documento por atributo.	7 semanas
3	Mostrar listado de actas del Consejo de dirección. Eliminar informe del Consejo de dirección. Ver listado de miembros del Consejo de dirección. Administración de usuarios del sistema. Exportar a PDF acta e informe del Consejo de dirección. Imprimir acta e informe del Consejo de dirección.	8 semanas