

**Universidad de las Ciencias Informáticas**

**Facultad 3**



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

**Sistema informático para la gestión de la información de los procesos  
de las Brigadas Técnicas Juveniles**

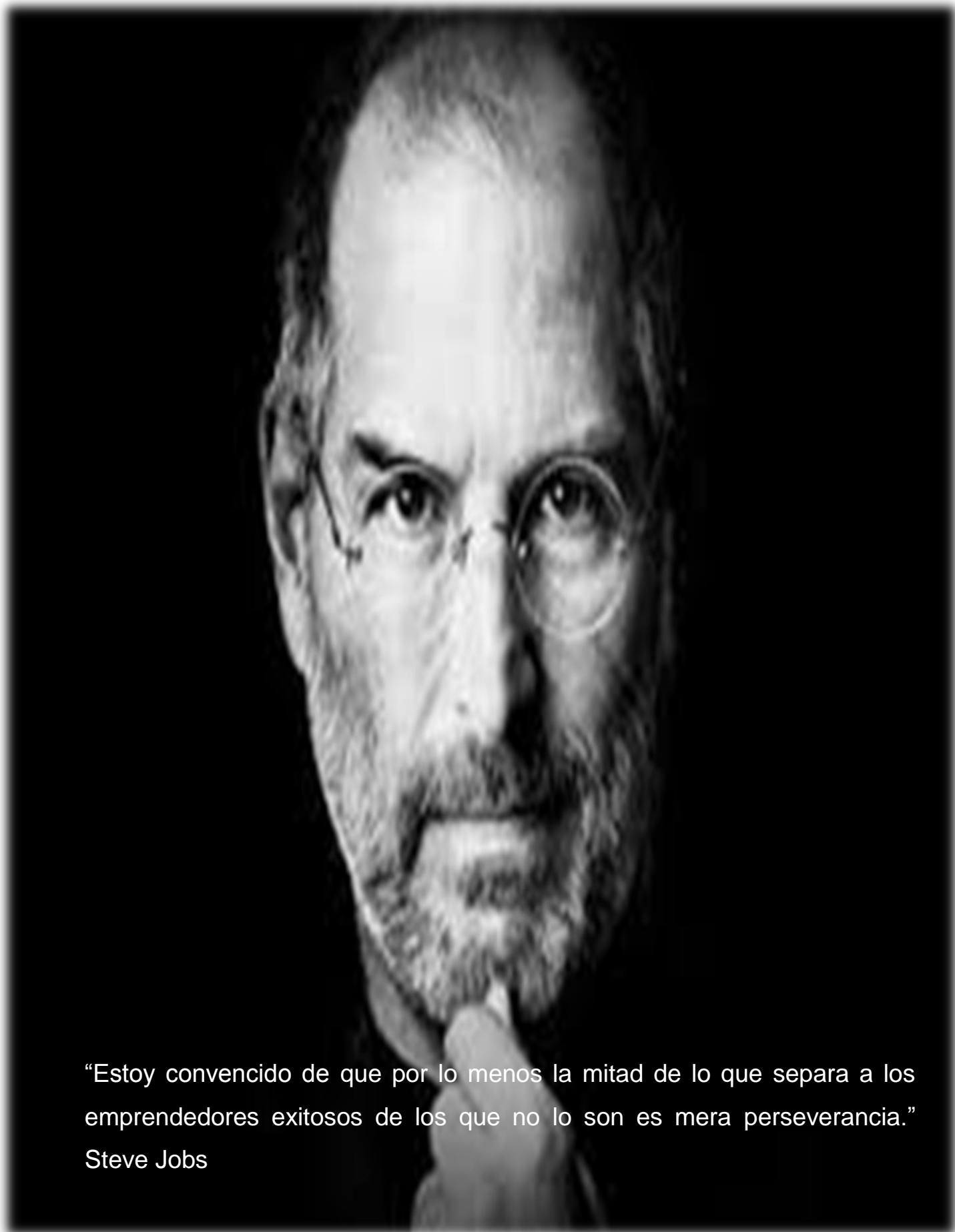
**Autor:** Michel Subert Santa Cruz Pacheco

**Tutor(es):** Ing. Yenier Figueroa Machado

Ing. Carlos Rafael Rodríguez Rodríguez

**Ciudad de la Habana, 2015.**

**Año 57 de la Revolución.**



“Estoy convencido de que por lo menos la mitad de lo que separa a los emprendedores exitosos de los que no lo son es mera perseverancia.”  
Steve Jobs



**DECLARACIÓN DE AUTORÍA:**

Declaro ser autor de la presente tesis y se reconoce a la Universidad de las Ciencias Informática, los derechos patrimoniales de la misma, con carácter exclusivo.

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

\_\_\_\_\_  
Firma del autor

Michel Subert Santa Cruz Pacheco

\_\_\_\_\_  
Firma del Tutor

Ing. Yenier Figueroa Machado

\_\_\_\_\_  
Firma del Tutor

Ing. Carlos R Rodríguez Rodríguez



**DATOS DEL CONTACTO**

**Síntesis de Tutor:** Ing. Yenier Figueroa Machado

**Correo electrónico:** yfigueroa@uci.cu

**Síntesis del Tutor:** Carlos Rafael Rodríguez Rodríguez

**Cargo:** Jefe de departamento del Centro de Gobierno Electrónico

**Correo electrónico:** crodríguezr@uci.cu



*Agradecimientos: A mis familiares por su apoyo incondicional. A mis compañeros de grupo que compartieron conmigo todos estos años. Agradecimiento especial a mis tutores Yenier Figueroa y Carlos Rafael por su apoyo incondicional en todo momento. A Gabriel Coello por el asesoramiento y la ayuda durante la realización de la tesis. Gracias a todos los que de una forma u otra me ayudaron con el desarrollo de este trabajo y durante mi carrera.*

*Michel Subert Santa Cruz Pacheco*



*Dedicatoria:*

*A mis padres, por su apoyo incondicional y ser un ejemplo de esfuerzo y dedicación en mi vida.*

*A mi mamá que siempre ha creído en mí en todos los momentos difíciles de mi carrera.*

*A mi papá que siempre se ha sacrificado por mí para lograr este resultado.*

*A mis abuelos Alberto y mis abuelas Estrella y Dionela que ya no están.*

*A mis hermanos Gabriel y Camilo por su apoyo incondicional.*

*A mi tía Nuris y mis primos Lizandra y Yudiel que siempre me han sabido dar el empujón necesario.*

*A todos muchas gracias.*

*Michel Subert Santa Cruz Pacheco*



## **Resumen**

El presente trabajo está dedicado a la realización de una herramienta informática que permita la disponibilidad en tiempo de los expedientes para su revisión y aprobación y así poder optar por una de las diferentes categorías de premios que otorga las Brigadas Técnicas Juveniles (BTJ). Para ello se realiza un estudio del estado del arte de la temática a tratar, identificando los conceptos principales de la investigación referente a la gestión documental y la gestión de la información. Además, se realiza un análisis de los sistemas informáticos que permiten la gestión de documentos y la gestión de información de las organizaciones de masa a nivel mundial y nacional, demostrándose que estas no resuelven los problemas de las BTJ pero sirven como base para la realización del software que se pretende desarrollar. Para el cumplimiento del propósito de esta investigación se realiza una descripción de la metodología de desarrollo, las herramientas y los lenguajes empleados en el desarrollo del sistema informático. Al finalizar se muestran los resultados obtenidos después de haber aplicado las pruebas pertinentes para detectar y corregir los errores antes de la entrega al cliente. Este sistema proveerá a las BTJ una herramienta para la gestión de los principales procesos, así como la gestión de otros procesos como la gestión de la estructura de la brigada y la conformación de las comisiones del evento Concurso Científico Técnico Juvenil.

**Palabras claves:** Herramienta informática, disponibilidad, principales procesos, gestión



Índice

Introducción.....	1
Capítulo 1: Fundamentación teórica .....	5
1.1. Introducción.....	5
1.2. Principales conceptos .....	5
1.2.1. Definición de proceso .....	5
1.2.2. Gestión de proceso.....	5
1.2.3. Gestión de la información .....	6
1.2.4. Gestión Documental .....	7
1.2.5. Sistemas de Gestión Documental.....	7
1.3. Estado del arte .....	8
1.3.1. Aplicaciones de gestión documental .....	8
1.3.2. Aplicaciones de gestión de información. ....	10
1.3.3. Análisis de soluciones estudiadas.....	11
1.4. Herramientas y tecnologías utilizadas.....	12
1.4.1. Aplicaciones Web vs Aplicaciones de Escritorio .....	12
1.4.2. Marco de trabajo Symfony 2.6 .....	13
1.4.3. Lenguajes de programación.....	14
1.4.4. Servidor Web.....	17
1.4.5. Gestor de base de datos.....	17
1.4.6. Metodología de desarrollo .....	18
1.4.7. Herramienta y lenguaje de modelado .....	20
1.4.8. Entorno de Desarrollo Integrado (IDE).....	22
1.5. Técnicas , métricas y pruebas a emplear para la verificación y validación del sistema.....	22
1.6. Conclusiones .....	25
Capítulo 2: Descripción de la solución propuesta .....	26
2.1. Descripción de la solución .....	26
2.1.1. Descripción de los procesos del negocio .....	26
2.2. Fases y artefactos generados en la metodología XP.....	30
2.2.1. Planificación .....	30
2.2.2. Historias de usuario .....	30
2.2.3. Requisitos funcionales .....	32





2.2.4.	Requisitos no funcionales del sistema .....	34
2.2.5.	Plan de iteraciones .....	35
2.2.6.	Plan de entrega .....	36
2.3.	Diseño.....	38
2.3.1.	Tarjeta CRC.....	38
2.4.	Modelo de datos .....	39
2.4.1.	Patrones de diseño de bases de datos .....	39
2.4.2.	Patrones de diseño.....	40
2.5.	Implementación.....	45
2.5.1.	Estándares de codificación .....	45
2.5.2.	Modelo de despliegue.....	46
2.6.	Conclusiones .....	47
Capítulo 3: Validación de resultados .....		48
3.1.	Aplicación de técnicas de validación de requisitos y utilización de métricas.....	48
3.2.	Pruebas funcionales o de aceptación .....	55
3.3.	Pruebas unitarias .....	57
3.4.	Validación de las variables .....	58
3.5.	Conclusiones parciales.....	60
Conclusiones generales: .....		62
Recomendaciones: .....		63
Bibliografía.....		64
Glosario de términos: .....		68



**Índice de Tablas**

Tabla No. 1 Historia de usuario "Gestionar expediente" .....	31
Tabla No. 2 Historias de usuarios detectadas .....	32
Tabla No. 3 Requisitos funcionales .....	34
Tabla No. 4 "Plan de iteraciones" .....	36
Tabla No. 5 "Plan de entrega" .....	37
Tabla No. 6 Tarjeta CRC # 2 .....	39
Tabla No. 7 Rango de valores para la evaluación de los atributos relacionados con la métrica TOC.....	51
Tabla No. 8 Resultados de la aplicación de la métrica TOC a las clases principales del sistema.....	51
Tabla No. 9 Rango de valores para la evaluación de los atributos relacionados con las métricas.....	53
Tabla No. 10 Resultados de la aplicación de la métrica RC a 38 clases. ....	54
Tabla No. 11 Caso de prueba "Gestionar expediente". ....	57
Tabla No. 12 Resultados de pruebas de aceptación .....	57
Tabla No. 13 Variables de la investigación.....	59
Tabla No. 14 Validación de las variables .....	60



**Índice de Figuras**

Figura No. 1 Fases de la metodología XP.....	20
Figura No. 2 Diagrama de la estructura de las BTJ.....	27
Figura No. 3 Diagrama de proceso del negocio " sello Forjadores del Futuro ".....	29
Figura No. 4 Ejemplo del Patrón llaves subrogadas.....	40
Figura No. 5 Patrón Modelo Vista Controlador.....	41
Figura No. 6 Ejemplo del patrón Unidad de trabajo.....	42
Figura No. 7 Clase experta en la información de los tipos de clasificación de los expediente.....	42
Figura No. 8 Ejemplo del uso del patrón controlador en la clase ExpedienteController.....	43
Figura No. 9 Paquete despachador de eventos de Symfony2.....	44
Figura No. 10 Ejemplo donde se evidencia el patrón composite (árbol simple).....	45
Figura No. 11 Propuesta de escenario del despliegue.....	47
Figura No. 12 Representación de la cantidad de procedimientos por clase.....	52
Figura No. 13 Resultados de la aplicación de la métrica TOC.....	52
Figura No. 14 Representación gráfica de los resultados de la aplicación de las métricas RC.....	55
Figura No. 15 Resultados de las pruebas de caja blanca.....	58



## **Introducción**

En la actualidad, es necesaria la incorporación de un sistema de información eficaz, capaz de alcanzar cambios significativos en la productividad y administración de una organización que agrupa a una gran cantidad de personas. Es imprescindible la utilización de tecnologías de la información y comunicación que brindan la posibilidad de aunar funcionalidades de almacenamiento, procesamiento y transmisión de datos.

Básicamente las TIC, siglas de tecnologías de la información y las comunicaciones, consisten en un grupo de técnicas, desarrollos y dispositivos de última generación. La cual permiten integrar diferentes funcionalidades relacionadas con el almacenamiento, el posterior procesamiento y por último la transmisión de los datos. Por ello, en el momento de diseñar un sistema de gestión de la información eficaz, es necesaria la incorporación de las TIC, que permitirán hacer posible la puesta en marcha del sistema que ha sido desarrollado en base a los requerimientos de los datos en las diversas áreas de la organización.

Los sistemas de gestión de la información han tenido un auge en los últimos años que ha provocado que la mayoría de las organizaciones tengan la necesidad de enfrentar este desafío cambiante y en aras del desarrollo, a tal punto que se les ha hecho imprescindible contar con estos sistemas para un mejor desarrollo.

En los inicios de la Revolución Cubana el Comandante en Jefe Fidel Castro Ruz concibió la creación de una brigada integrada por jóvenes, a los cuales era necesario mantener vinculados sistemáticamente a la superación científico - técnica con vista a que pudieran contribuir, en mejores condiciones a la construcción de la base técnico material del socialismo en el país. Este fruto genuino de la revolución son las Brigadas Técnicas Juveniles (BTJ).

Estas, son el movimiento científico técnico de la Unión de Jóvenes Comunistas, que es la organización juvenil del Partido Comunista de Cuba. Son dirigidas y se rigen a todos los niveles por los estatutos de esta; contribuyendo a crear en todos sus integrantes la modestia, la profundidad de análisis, la objetividad y el trabajo en colectivo, sustentado en los más altos valores de la sociedad, la fidelidad a la Patria y en un profundo espíritu antiimperialista e internacionalista; además de mantenerlos vinculados a la superación científico – técnica.

En la UCI el movimiento de las BTJ ha participado activamente en todos esos procesos obteniendo resultados relevantes. Pero el trabajo de gestión de la información de estos procesos dentro de la Universidad se ha visto afectado por varias limitaciones. Entre estas limitaciones se puede mencionar:

- Todas las tareas de organización, planeación y control están parcialmente informatizadas, debido a que solo se utilizan las herramientas de la suite de productos de



Microsoft Office.

- Para formar parte de estos procesos los aspirantes deben completar una serie de formularios o planillas para optar por los diferentes premios, en estos se observan dificultades tales como la inexactitud y la omisión de datos que deben ser recolectados y la no estandarización de las planillas utilizadas para el proceso, lo cual provoca que los expedientes que conforman estos documentos se invaliden y sean rechazadas.
- Se incumplen los periodos establecidos para la recepción, revisión y aprobación pues el proceso es ineficiente porque todo el traslado de la información se realiza de forma manual y no se puede verificar los datos, además no se cuenta con la disponibilidad de la información a tiempo porque transitan por varios destinos (brigada base, nivel centro, nivel facultad) hasta llegar al nivel de universidad.
- No persiste la información de años anteriores porque no se guardan en ningún formato o son fáciles de extraviar, lo que dificulta la obtención de estadísticas que contribuyan a mejorar el desarrollo de esos procesos.

Por lo planteado anteriormente es necesario para los encargados de las BTJ contar con alguna herramienta que permita tener una mejor organización, control y compilación de los resultados de los procesos que les competen a ellos.

En el análisis anterior se identifica el siguiente **problema a resolver**: ¿Cómo contribuir a una mejor gestión de la información de los procesos que desarrollan las Brigadas Técnicas Juveniles en la UCI?

Como **objeto de estudio** se precisa: desarrollo de software de gestión.

Para dar solución al problema a resolver se define como **objetivo general**: Desarrollar una herramienta informática que contribuya a mejorar la gestión de la información de los procesos de las Brigadas Técnicas Juveniles en la UCI.

Se considera como **campo de acción**: Sistemas informáticos para la gestión de la información de procesos en las BTJ.

A partir del problema general se derivan los siguientes **objetivos específicos**:

- Elaborar el marco teórico para dar a la investigación un sistema coordinado y coherente de conceptos y proposiciones que permitan abordar el problema planteado.
- Desarrollar una herramienta para informatizar la gestión de la información de los principales procesos de la BTJ en la UCI.
- Validar la herramienta propuesta a partir de pruebas funcionales y del criterio de sus usuarios potenciales.



Para dar cumplimiento a estos objetivos se proponen las siguientes **tareas de investigación**:

1. Estudio del estado del arte sobre el desarrollo de sistemas de gestión enfocados en la gestión documental y de información de los procesos de las BTJ.
2. Investigación para determinar:
  - ¿Cuáles son los principales procesos que desarrollan las BTJ en la UCI?
  - ¿Cuáles son los documentos que se gestionan?
  - ¿Quiénes son los actores que participan en los procesos?
3. Confección de un mapa conceptual donde relacione lo aprendido en las tareas anteriores.
4. Estudio y selección de la metodología de desarrollo de software a utilizar.
5. Estudio y selección de las herramientas y tecnologías a utilizar para el desarrollo de la propuesta de solución.
6. Elaboración de los artefactos que la metodología seleccionada propone para el análisis de la propuesta.
7. Elaboración de los artefactos que la metodología seleccionada propone para el diseño de la propuesta.
8. Implementación de las funcionalidades de la capa de presentación, negocio y acceso a datos de la herramienta.
9. Aplicación de métricas de producto para evaluar los requisitos y el diseño de la herramienta.
10. Verificación de la solución propuesta aplicando pruebas de caja blanca.
11. Validación de la solución propuesta aplicando pruebas de caja negra.
12. Confección de una muestra de expedientes para gestionarlos en la herramienta durante las pruebas funcionales.

Para desarrollar las tareas que posibiliten cumplir los objetivos trazados se utilizan métodos teóricos y empíricos.

De los **Métodos Teóricos** se emplean los siguientes:

- **Analítico–Sintético**: Para realizar el estudio del estado del arte de tecnologías, herramientas y metodologías para desarrollar aplicaciones web, y a partir del estudio realizado obtener los elementos significativos y arribar a conclusiones para dar solución al problema.
- **Histórico–Lógico**: Para conocer la evolución y desarrollo del tema tratado en la investigación, y además realizar un estudio crítico de trabajos anteriores como puntos de referencia y de comparación con los resultados alcanzados.



De los **Métodos Empíricos** se emplean los siguientes:

- **Entrevista:** Facilita la obtención de información acerca de las actividades que realizan los encargados de las BTJ.
- **Medición:** Se utilizan en las métricas para la validación de los requisitos, el diseño del sistema y las variables de la investigación así como las pruebas para verificar el correcto funcionamiento de la aplicación.

El presente trabajo está estructurado en tres capítulos, tal y como se describe a continuación:

### **Capítulo 1: Fundamentación teórica**

En este capítulo se realiza un estudio del estado del arte que permita conocer la situación mundial del tema, además se hace referencia a los principales conceptos referentes al mismo. Se describen los sistemas informáticos identificados que permiten la gestión documental y la gestión de información. Por último se caracterizan la metodología de desarrollo, tecnologías y herramientas posibles a utilizar, seleccionando las que más se ajustan a las necesidades del sistema a implementar, con el fin de desarrollar una aplicación web que permita gestionar la información de los procesos de las Brigadas Técnicas Juveniles.

### **Capítulo 2: Descripción de la solución propuesta**

En este capítulo se presentan las funcionalidades que deben cumplir el sistema y la planificación de la entrega. Además se exponen los artefactos propios del diseño, así como el modelo de datos, la arquitectura del sistema y se muestra la configuración de los elementos de hardware (nodos) y la comunicación que existe entre estos.

### **Capítulo 3: Validación de resultados**

Este capítulo tiene como objetivo fundamental aplicar métricas y pruebas de software para evaluar la calidad de los artefactos obtenidos durante el desarrollo del sistema. Se evidencian los resultados obtenidos al evaluar la calidad de los artefactos.



## **Capítulo 1: Fundamentación teórica**

### **1.1. Introducción**

El presente capítulo aborda los elementos que permitieron realizar una correcta caracterización del sistema, sustentados en un estudio de los conceptos fundamentales a los que está relacionado. Se incluyen la tendencia del desarrollo web actual en el país y particularmente en la UCI relacionados con el sistema a desarrollar. Se ofrece un estudio de las principales herramientas, plataforma de desarrollo y metodología de desarrollo de software que son sugeridas. Se estudian los diferentes patrones empleados en el desarrollo de sistemas informáticos y finalmente se hace alusión a las pruebas para evaluar la calidad de las disciplinas descritas en dicho capítulo.

### **1.2. Principales conceptos**

Se plantean los conceptos que pueden ser relacionados al tema planteado en el trabajo de diploma.

#### **1.2.1. Definición de proceso**

Existe una variedad de autores que han definido lo que es un proceso, esto ha contribuido a una evolución de dicho concepto en el tiempo. Se puede observar esta tendencia analizando algunos de estos autores.

Gabriel Pall, define un proceso como “la organización lógica de personas, materiales, energía, equipamiento e información en actividades de trabajo diseñadas para producir un resultado final requerido” (1). También se puede ver que González Riesco se refiere al proceso como “el conjunto de actividades secuenciales que realizan una transformación de una serie de entradas (material, mano de obra, capital, información) en los salidas deseados (bienes y/o servicios) añadiendo valor” (2). Cabrerías también coincide con González Riesco acerca de ver la información como una posible entrada de un proceso (3). Mientras que Cooper puntualiza que las salidas también pueden ser información (4). Otras definiciones interesantes de procesos han sido dadas por Juran que lo describe como “Una serie de acciones sistemáticas dirigidas al logro de un objetivo previamente definido” (5).

Se puede concluir que los autores antes mencionados convergen en un mismo sentido, por lo que se asume por el autor de la investigación, como proceso: entorno donde se modifica una entrada ordenada de recursos sobre la cual actúan una serie de acciones que la transforman en función de un objetivo final.

#### **1.2.2. Gestión de proceso**

Según Susana Pepper la gestión por procesos puede definirse como una forma de enfocar el trabajo, donde se persigue el mejoramiento continuo de las actividades de una organización mediante la





identificación, selección, descripción, documentación y mejora continua de los procesos. Toda actividad o secuencia de actividades que se llevan a cabo en las diferentes unidades constituye un proceso y por consiguiente hay que gestionarlo. Los principios que orientan la gestión de procesos se sustentan en los siguientes conceptos (6):

- La misión de una organización es crear valor para sus clientes; la existencia de cada puesto de trabajo debe ser una consecuencia de ello: existe para ese fin.
- Los procesos siempre han de estar orientados a la satisfacción de los clientes.
- El valor agregado es creado por los empleados a través de su participación en los procesos; los empleados son el mayor activo de una organización.
- La mejora del proceso determinará el mayor valor suministrado o entregado por el mismo.
- La eficiencia de una organización será igual a la eficiencia de sus procesos.

Juan Bravo también define la gestión de procesos como una forma sistemática de identificar, comprender y aumentar el valor agregado de los procesos de la empresa para cumplir con la estrategia del negocio y elevar el nivel de satisfacción de los clientes. La gestión de procesos con base en la visión sistémica apoya el aumento de la productividad y el control de gestión para mejorar en las variables clave, por ejemplo, tiempo, calidad y costo. (7)

Se puede concluir que los autores antes mencionados convergen en un mismo sentido, por lo que se asume por el autor de la investigación, como gestión de proceso: la identificación, selección, descripción y mejora paulatina de los procesos para aumentar el valor agregado de estos.

### **1.2.3. Gestión de la información**

La gestión de información se puede definir como el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, permitiendo mejorar los procesos, productos y servicios de la organización. Realizar procesos de gestión de información posibilita entre otras cosas identificar, organizar, representar y recuperar información dispersa en áreas. (8) La gestión de la información implica:

- Determinar la información que se precisa
- Recoger y analizar la información
- Registrarla y recuperarla cuando sea necesaria
- Utilizarla
- Divulgarla

Existen componentes esenciales de la gestión de la información, ellos son:

- Los recursos informativos: identificación, evaluación y uso de recursos tanto internos como externos.



- La tecnología: medios que recubren los métodos de entrada, almacenamiento, recuperación y difusión de la información.
- La administración y gestión: planificación general, recursos humanos, comunicación interpersonal, contabilidad presupuestos y marketing.

#### **1.2.4. Gestión Documental**

La Gestión Documental (GD) está conformada por un conjunto de operaciones comprometidas en la búsqueda de la eficiencia en la producción, mantenimiento, uso y destino final de los documentos, es decir, desde el momento de su concepción en las oficinas administrativas hasta su ingreso en las instituciones de archivo. Es un área de la administración general que se encarga de garantizar la economía y disposición de los documentos administrativos (9).

Por su parte Ana R Brito Valdés y Lianet B Zaldívar en su tesis “Gestión Documental para los Proyectos de Investigación del MIC” plantea que la gestión documental es el conjunto de técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, permitir la recuperación de información desde ellos, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no se utilizan y asegurar la conservación indefinida de los documentos más valiosos. En resumen la GD es un proceso que se encarga del tratamiento que se le da a todos los documentos de una organización, desde su concepción hasta su eliminación o conservación permanente, teniendo en cuenta todas las etapas que engloba el ciclo de vida de un documento. Concepto con el que se trabajó durante el desarrollo de esta investigación.

Por lo tanto el autor del presente trabajo concluye que la gestión documental no es más que las operaciones comprometidas en administrar el flujo de documentos de una organización, permitir la manipulación de estos y asegurar su conservación.

#### **1.2.5. Sistemas de Gestión Documental**

Los Sistemas de Gestión Documental (DMS, del Inglés Document Management System) son programas de gestión de bases de datos que disponen de una tecnología idónea para el tratamiento de documentos.

Un DMS se conforma como el almacén central de información para toda la organización. Es el punto de referencia y recuperación para la documentación de cualquier índole. Garantiza la integridad de la misma y su acceso consistente desde cualquier punto (10).

Las características generales que identifican los DMS son las siguientes:

- Seguridad: Proporcionando un almacenamiento único, que unifique documentos, la catalogación de los mismos y que simplifique la definición de políticas de mantenimiento.



- Capacidad de recuperación y acceso: En forma sencilla y proporcionando métodos de búsqueda ágiles y completos que garanticen la reutilización adecuada de cualquier material o parte del mismo.
- Centralización: Facilitando el mantenimiento de versiones y garantizando la homogeneidad de formatos y contenidos a lo largo de toda la organización. Además, asegurando la consistencia de los criterios de catalogación.
- Manejo de grandes volúmenes de documentos.
- Garantía de acceso a la información más actual.
- Definición de flujos de trabajo en el sistema para la gestión de procesos operativos entre departamentos y empresas externas.
- Gestión de la información en formato nativo. Control de acceso a la información.
- Seguridad ante la posible pérdida de documentación.

Según el concepto y características mencionadas el autor del presente trabajo concuerda que los DMS son el almacén central de toda organización y constituye punto de referencia para la recuperación y manipulación de documentos de cualquier índole.

### **1.3. Estado del arte**

Se realizó una investigación en busca de una aplicación informática que permita dar solución a los objetivos de este trabajo. Esta tiene que ser multiplataforma, de código abierto, basadas en componentes que sean reutilizables y adaptable a la gestión de la información de los procesos de las BTJ. A continuación se presentan las características más relevantes de algunos sistemas que han sido desarrollados tanto para la gestión documental como para gestionar información de organizaciones internacionalmente, en nuestro país y en la Universidad de las Ciencias Informáticas.

#### **1.3.1. Aplicaciones de gestión documental**

##### **Sistema de gestión documental Alfresco.**

Es una herramienta de código fuente libre desarrollado en Java, basado en estándares abiertos y de escala empresarial para sistemas operativos tipo Windows, Unix Solaris y algunas versiones de Linux. Está diseñado para usuarios que requieren un alto grado de modularidad. Este sistema incluye un repositorio de contenidos, un marco de trabajo de portal web para administrar y usar contenido estándar. (11)

Esta herramienta permite:

- La gestión de documentos, contenido web (incluyendo aplicaciones web y virtualización de sesiones), imágenes y registros.



- Versionado a nivel de repositorio (similar a Subversión)
- Superposición transparente (similar a UnionFS)

### **AvilaDoc Pro 1.0.0**

Aplicación Web desarrollada sobre una plataforma de software libre con base de datos centralizada, destinada a la gestión, tramitación y resguardo de archivos electrónicos. Herramienta de trabajo que gestiona el tramitado interno de los documentos a través de la intranet, sin necesidad de copias o envíos físicos y permite el control de la entrada y salida de documentos, el manejo de anotaciones a los documentos por usuarios, el tratamiento diferencial para los documentos pendientes, así como la recuperación de la información a través del contenido de la ficha del documento (12).

#### **Algunas características:**

- Constituye un archivo digitalizado, que agiliza las tareas de registro, búsqueda, reproducción y distribución de los documentos.
- El tramitado interno de los documentos se realizará sin necesidad de reproducirlos o distribuirlos físicamente a las oficinas involucradas garantizando rapidez en la gestión.
- Ofrece un historial del documento desde que fue creado hasta su eliminación. Conserva la acción que se hizo sobre el mismo, el usuario que la acometió y la fecha en que lo hizo
- Logra integrar la gestión documental y la gestión de archivo.
- Este sistema tiene como inconveniente que no posee un módulo para la digitalización de los documentos que se encuentran en papel.

### **Sistema Gestor de Documentos Administrativos eXcriba.**

Esta herramienta posibilita la gestión de los documentos que se generan o reciben a partir de las funciones y responsabilidades dentro de una entidad mediante una interfaz de usuario sencilla, cómoda y flexible. Permite además gestionar flujos documentales, almacenar documentos en diferentes formatos electrónicos y salvaguardar el patrimonio documental. Está dirigido a todos los usuarios que generan y reciben documentos.

Las principales ventajas del eXcriba están dadas por el ahorro físico en espacio para archivos tradicionales en papel, ahorro en infraestructura y personal dedicado a las tareas de gestión de documental y el consiguiente ahorro de tiempo, así como la mejora de los procesos en las entidades, incidiendo en un conjunto de funcionalidades tales como:

- Garantiza la automatización de los documentos a través de su ciclo de vida.
- Inmediata localización y búsqueda precisa de los documentos.
- Disponibilidad y centralización de los documentos.



- Soporte del almacenamiento de la información en diferentes formatos.
- Evitar el deterioro de la información (física) administrativa y valiosa.

Esta herramienta presenta una interfaz que guía y ayuda al usuario a realizar una gestión de sus documentos de manera más organizada y sencilla, también contiene una nueva funcionalidad que es la de digitalización de documentos que permite integrar al sistema aquellos documentos que anteriormente se encontraban en papel.

Tiene como núcleo o motor para la gestión de documentos de archivo el Administrador de Contenidos Empresariales Alfresco, que de esta forma le proporciona un buen número de sus funcionalidades, las cuales fueron mencionadas antes, en este documento (13).

### **1.3.2. Aplicaciones de gestión de información.**

#### **SIGIFI**

Es una aplicación Web creada por la Facultad de Informática de la Universidad de Camagüey (UC) que cuenta con una interfaz amigable y fácil de utilizar. Este sistema está basado en la tecnología Cliente-Servidor, lo cual permite una mayor centralización del control (acceso, recursos e integridad de los datos) de la aplicación (14).

Este sistema fue desarrollado haciendo uso de la tecnología Apache-PHP-MySQL. Este sistema de gestión de la información posee como funcionalidades:

- Brinda la posibilidad de que la información de los procesos pueda ser observada por toda la intranet de la UC.
- Cada profesor puede entrar sus datos individuales en la parte docente (en la gestión del registro de asistencia, de las asignaturas que imparte, proyectos educativos, trabajo metodológico, entre otros) como científico técnica (Categoría docente y científica, postgrados, diplomados, grupos científicos estudiantiles, maestrías y doctorados cursados, publicaciones que posee, eventos en los que ha participado, premios obtenidos, entre otros.)

#### **Sistema de gestión de los procesos de la FEU en la UCI (DataFeu)**

La herramienta DataFEU es un sistema de gestión para la automatización de los principales procesos llevados a cabo por la Federación Estudiantil Universitaria en la Universidad de las Ciencias Informáticas y apoyar a sus principales dirigentes en el proceso de toma de decisiones. Fue desarrollada sobre herramientas multi-plataformas y de código abierto (marcos de trabajo Symfony y Ext JS empleando MySQL como gestor de bases de datos y apache2 como servidor). El sistema cuenta con módulos para la seguridad y la administración; así como también un módulo para la generación de



reportes en formato PDF con el objetivo de extraer estadísticas y centralizar la información hacia los factores de la universidad. (15)

### **Sistema de gestión de información en las secciones sindicales de la Facultad 3**

Este sistema permite la gestión de toda la información referente a las secciones sindicales de la Facultad 3 de la Universidad de las Ciencias Informáticas. Se desarrolló con la metodología ágil XP, el marco de trabajo Symfony 2.4.1, el entorno de desarrollo NetBeans 7.4, el SGBD PostgreSQL y el servidor web Apache las cuales resultaron ser las más convenientes para el desarrollo del sistema (16).

#### **Características.**

- Permite gestionar una facultad y todas las secciones sindicales de esta.
- Permite dar alta y baja a los trabajadores.
- Gestionar acuerdos tomados en las reuniones de las secciones sindicales.
- Permite gestionar reportes de los procesos de las secciones sindicales.
- Genera modelos con datos de la sección en formato PDF.
- Generar gráficos de pastel teniendo en cuenta los datos de reportes.

#### **1.3.3. Análisis de soluciones estudiadas**

Luego de estudiar estos productos y teniendo en cuenta que las Brigadas Técnicas Juveniles (BTJ) requiere de un sistema que permita la gestión de los expedientes y documentos (tanto generados como adjuntos) que conforman estos y funcionalidades que permitan la gestión de la información interna y estructural de la organización se evalúa que:

- Los sistemas de gestión documental a pesar que poseen funcionalidades para la gestión de expedientes y documentos no permiten la gestión de documentos generados en la misma aplicación solo permiten cargar documentos digitales (imágenes, doc, pdf ect). Además no poseen funcionalidades que posibiliten la gestión de una estructura interna de las BTJ.
- Los sistemas de gestión de información presentan funcionalidades que posibilitan la gestión de una estructura de grupos de colectivos y la gestión de información de un individuo pero no permiten la gestión de expedientes y documentos ni permite establecer el flujo de estos expedientes.

Por lo anteriormente planteado el autor de este trabajo puede concluir que todos los productos son descartados pues individualmente solo enmarca una parte del problema. Esta razón, unido a que ningún sistema estudiado permite le gestión de documentos generados manualmente mediante



formularios en el sistema (solo gestionen documentos digitales), son los motivos por lo cual se decide el desarrollo de una nueva aplicación.

Aunque estos sistemas no permiten dar cumplimiento a la gestión de la información de los procesos de las BTJ sirven como base para la realización del software que se pretende desarrollar brindando algunos aportes que se pueden tomar en cuenta como:

- Brindar información procesada en forma totalmente sistematizada y estandarizada.
- Características de un sistema de gestión aplicado al contexto de una organización en la UCI.
- Poder almacenar y consultar las evidencias de forma segura en cualquier instancia de tiempo.
- Gestión y almacenamiento de documentos (imágenes, .docx, .doc) que se exportan formato PDF
- Principales herramientas y tecnologías definidas para el desarrollo del sistema a implementar.
- Gestión de un expediente.

#### **1.4. Herramientas y tecnologías utilizadas**

##### **1.4.1. Aplicaciones Web vs Aplicaciones de Escritorio**

Se realizó un estudio de factibilidad de implementación de la aplicación con enfoque Web o de escritorio. Las aplicaciones de escritorio se instalan en máquinas individuales las cuales son las responsables de la ejecución y los resultados que se muestran en la pantalla. Sin embargo las aplicaciones Web son emplazadas en una sola máquina que desempeña el papel de servidor, y son ejecutadas a través de la conexión con un navegador Web, quien se encarga de transportar las solicitudes del cliente y las respuestas del servidor (17).

##### **Plataforma seleccionada (Aplicaciones Web)**

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es un software que se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, etc.) en la que se confía la ejecución al navegador. Las aplicaciones web deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación web se escribe una vez y se ejecuta igual en todas partes.

Por lo general una aplicación web está estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa y un motor capaz de usar alguna tecnología web



dinámica (ejemplo: PHP) constituye la capa de en medio. Por último, una base de datos constituye la tercera y última capa (18).

### **Ventajas**

- Acceso ubicuo, sin necesidad de distribución e, idealmente, con pocos requisitos técnicos. Datos centralizados y fácil integración de datos de múltiples fuentes.
- Una empresa puede migrar de sistema operativo o cambiar el Hardware libremente sin afectar el funcionamiento de las aplicaciones de servidor.
- No se requieren complicadas combinaciones de Hardware/Software para utilizar estas aplicaciones. Solo unas estaciones de trabajo llamadas clientes y un navegador compatible a los requisitos no funcionales de software de la aplicación.
- Existe solo una versión en el servidor lo que implica que no hay que distribuirla entre los demás computadores. El proceso es rápido y limpio.
- Se facilita el trabajo a distancia. Se puede trabajar desde cualquier PC o computador portátil con conexión al servidor de aplicaciones.
- Al funcionar en un navegador, se requiere un conocimiento básico de informática para utilizar una aplicación Web.

#### **1.4.2. Marco de trabajo Symfony 2.6**

Symfony 2 es un completo marco de trabajo diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Symfony2 está construido para cumplir los principios bases de crear herramientas que permitan desarrollar y construir rápidamente robustas aplicaciones. Symfony 2 está construido sobre la idea de las mejores tecnologías” (19).

Se escoge además este marco de trabajo por su abundante documentación, el potencial de trabajo que presenta para el desarrollo de aplicaciones web de forma rápida, robusta y segura. Es uno de los marcos de trabajo más maduros en cuanto a desarrollo web, que incluye varios bundles de terceros que constantemente están mejorando temas de seguridad, rendimiento y otras características de todas las versiones.

Entre otras características que se conocen se encuentran:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.





- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Implementa la mayoría de las mejores prácticas y patrones de diseño para la web.
- Implementa aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Contiene código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con bibliotecas desarrolladas por terceros.

Entre otras ventajas se destaca sobre todo la independencia de módulos que han creado. El propio núcleo del marco de trabajo está dividido en módulos que tienen una alta cohesión, lo que permite la reutilización del mismo fuera de un proyecto basado en Symfony 2. (19)

### **ORM Doctrine 2.3.0**

El marco de trabajo Symfony 2 utiliza Doctrine para el mapeo de objetos relacional escrito en lenguaje PHP que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa justo encima de un Sistema Gestor de Base de Datos (SGBD). Puede generar clases a partir de una base de datos existente y después se pueden especificar relaciones y añadir funcionalidades extras a las clases autogeneradas. (20)

Una característica importante de Doctrine es la posibilidad de escribir consultas de base de datos en un lenguaje de consulta estructurado (SQL por Structured Query Language) denominado lenguaje de consulta doctrine (DQL por Doctrine Query Language). (20)

Doctrine es seleccionado por la necesidad de utilizar un marco de trabajo que permita trabajar con los objetos de la base de datos. Doctrine viene integrado por defecto al marco de trabajo Symfony2, además de que permite la generación automática del modelo, logrando agilizar el trabajo. Para crear el modelo, doctrine genera una clase por tabla de base de datos e indica mediante PHP el tipo de datos que se almacenará.

### **1.4.3. Lenguajes de programación**

#### **PHP 5.4**

Teniendo en cuenta que el marco de trabajo seleccionado Symfony 2 es un Framework PHP, entonces PHP versión 5.4 es el lenguaje escogido para la programación de la aplicación en el servidor.

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Puede ser desplegado en la



mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno (21).

Es un lenguaje multiplataforma, completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura (21).

Para el desarrollo de la propuesta solución es seleccionado PHP 5.4 debido a que el marco de trabajo Symfony 2 necesita la versión 5.3 o una superior. Además es muy sencillo para el entendimiento del desarrollador.

### **Marco de trabajo (del lado del cliente)**

#### **Extjs 2.2**

Es un framework para construir Aplicaciones de Internet Enriquecidas (RIA por sus siglas en inglés). Es basado en librerías JavaScript de código abierto, ligera y de alto rendimiento, compatible con la mayoría de los navegadores que permiten crear páginas e interfaces gráficas web dinámicas. El mismo incluye tecnologías como Ajax, DHTML, XML, XSLT, JSON, CSS y DOM. Tiene incluidos la mayoría de los controles de los formularios Web incluyendo tablas para mostrar datos y elementos semejantes a la programación desktop como los formularios, paneles, barras de herramientas, menús y muchos otros.

#### **HTML 5.0**

Es un lenguaje para escritura de hipertexto, es decir, documentos de texto estructurado, incluye enlaces (links) que conducen a otros documentos o a otras fuentes de información y permite la inclusión de información por formularios, entre otros. HTML5 se caracteriza por ofrecer una mejor estructura eliminando el uso excesivo de contenido de las etiquetas <div>, esta se emplea para definir un bloque de contenido a sección de la página; con el objetivo de que la web sea más coherente y comprensible. (22)

#### **CSS 3.0**

CSS o Cascading Style Sheets (Hojas de estilo en cascada) es un lenguaje utilizado para definir cómo se despliega el contenido basado en HTML en la web. CSS se puede utilizar para controlar un diseño, espacio, colores, fuentes y mucho más (22).

CSS3 introduce nuevos estilos para un mejor control sobre la presentación de estas sofisticadas interfaces.

### **Ventajas**



- Se obtiene un mayor control de la presentación del sitio al poder tener todo el código CSS reunido en uno, lo que facilita su modificación.
- Al poder elegir el archivo CSS que se desea mostrar, puede aumentar la usabilidad ya que se puede asignar un código CSS concreto a personas con deficiencias visuales, por ejemplo. Esto lo detecta el navegador web.
- Se consigue hacer mucho más legible el código HTML al tener el código CSS aparte
- Pueden mostrarse distintas hojas de estilo según el dispositivo que se estén utilizando (versión impresa, versión móvil, leída por un sintetizador de voz) o dejar que el usuario elija.
- Las novedades de CSS3 nos permiten ahorrarnos tiempo y trabajo al poder seguir varias técnicas (bordes redondeados, sombra en el texto, sombra en las cajas, etc.) sin necesidad de usar un editor gráfico.

### **Motor de Plantillas Twig 2.1**

Twig es un motor de plantillas para PHP muy rápido y seguro ya que compila las plantillas en código PHP optimizado. Además es flexible debido a que el desarrollador puede definir sus propias etiquetas y filtros personalizados (23). Implementa un novedoso mecanismo de herencia múltiple de plantillas y bloques de código, con esto se tienen todas las opciones posibles para crear plantillas; es fácil de aprender y con una amplia documentación.

### **JavaScript 1.6**

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario, páginas web dinámicas y permite la validación de los datos. JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java (24).

### **Principales características:**

- Maneja objetos dentro de nuestra página Web y sobre ese objeto se puede definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades, modificar archivos etc.
- Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Con esto se puede cambiar totalmente el aspecto de nuestra página al gusto del



usuario, evitándonos tener en el servidor un página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario, etc.

#### **1.4.4. Servidor Web**

##### **Apache.**

Es un servidor web que implementa el protocolo HTTP, es software libre y soporta los sistemas operativos Windows y Linux, es de fácil configuración, pues su estructuración en módulos permite al usuario utilizar los servicios y funcionalidades que ofrece (25).

##### **Principales características**

- Puede ejecutarse en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código abierto.
- Es un servidor altamente configurable de diseño modular. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para instalarlo cuando se necesiten. Otra cosa importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

Se definió Apache como software para el servidor de aplicaciones por su flexibilidad, rapidez, porque es gratuito, multiplataforma, interpreta el lenguaje PHP y por su configurabilidad. Además estas características promueven la eficiencia y rendimiento del sistema que debe ser capaz de dar respuestas a las peticiones con un nivel aceptable de desempeño. Teniendo en cuenta que la concurrencia pueda existir, debe prestar servicios sin que se amplíen los rangos de tiempo de repuesta. El manejo y optimización de la cache del marco de trabajo seleccionado se integra fácilmente con este servidor web, permitiendo mejorar el rendimiento de la aplicación y el acceso a las rutas de la misma.

#### **1.4.5. Gestor de base de datos.**

##### **PostgreSQL 9.4**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, con su código fuente disponible libremente y distribuido bajo licencia BSD. Es el sistema de gestión de bases de datos de



código abierto más potente del mercado. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi-hilos para garantizar la estabilidad del sistema, un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Su estabilidad, potencia, robustez, facilidad de administración han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema (26).

PostgreSQL es seleccionado porque es ideal para tecnologías web y fácil de administrar. Su código fuente es libre y de alta calidad, además de tener un rendimiento excelente y es muy utilizado en la universidad en aplicaciones de gestión.

#### **1.4.6. Metodología de desarrollo**

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto (software). Estas metodologías se agrupan en dos grandes bloques las pesada o tradicionales y las metodologías ágiles.

Las metodologías pesadas se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar y requiere una extensa documentación pues pretende prever todo de antemano. Entre las metodologías pesadas más conocidas se encuentran RUP (Rational Unified Procces) (27), MSF (Microsoft Solution Framework) (28) e Iconix (29) entre otras.

Las metodologías ágiles permiten incorporar cambios con rapidez en el desarrollo de software. Estas metodologías deben responder a ciertos indicadores que valoran:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funciona más que conseguir una buena documentación.
- La colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrictamente un plan.

Las metodologías ágiles más usadas son entre otras Programación Extrema (XP), SCRUM, AUP (Proceso Unificado Ágil) y Crystal Clear (30).

#### **Metodología de desarrollo utilizada: XP (Programación extrema)**

Considerando la situación actual de la investigación y luego del estudio de las metodologías existentes se decide utilizar XP ya que es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo y un pequeño equipo. La metodología consiste en los pasos para realizar una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.



La metodología XP tiene un conjunto importante de reglas y prácticas. En forma genérica, se pueden agrupar en (31):

- Reglas y prácticas para la Planificación
- Reglas y prácticas para el Diseño
- Reglas y prácticas para el Desarrollo
- Reglas y prácticas para las Pruebas

### **Planificación**

Se definen fechas de entrega de las primeras versiones del software, la organización del trabajo y el equipo de desarrollo y por último se detalla dentro de una versión del producto los problemas que deben ser resueltos con carácter urgente (31). En esta fase se definen las historias de usuarios y se identifican el número y tamaño de las iteraciones

### **Historia de usuario**

Las historias de los usuarios son descripciones que realiza el cliente en su propio lenguaje que da a conocer las necesidades del sistema. El nivel de detalle de las historias de usuario debe ser el mínimo posible, permitiendo hacerse una ligera idea de cuánto costará implementar el sistema (31).

### **Plan de iteraciones**

Una vez identificadas las HU y estimado el esfuerzo propuesto para el desarrollo del sistema de cada una de estas historias, se procede a la planificación de la etapa de implementación del sistema. Teniendo en cuenta la prioridad que tiene una determinada HU en el desarrollo del componente se decide en que iteración será implementada. Las HU que cuentan con mayor importancia por ser funcionalidades indispensables para la aplicación deben ser implementadas en las primeras iteraciones del ciclo de desarrollo.

### **Diseño**

El diseño adecuado para el software es aquel que supera con éxito todas las pruebas, refleja claramente la intención de implementación del equipo de desarrollo y tiene el menor número posible de clases y métodos. Una de las partes importantes en XP es la simplicidad en todos los aspectos. Se considera que un diseño sencillo se logra más rápido y se implementa en menos tiempo. La complejidad innecesaria y el código extra debe ser removido inmediatamente (31).

Para el diseño de las aplicaciones, la metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando notación UML, aunque el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante.

### **Tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración).**

Las tarjetas CRC permiten desprenderse del método basado en procedimientos y trabajar con una metodología basada en objetos, además de posibilitar que el equipo completo contribuya en la tarea del diseño. Las tarjetas CRC representan objetos; se utilizan normalmente cuando se determinan las clases necesarias y cómo van a interactuar, luego se implementa la solución.

### Implementación

La implementación o codificación es el flujo de trabajo donde se producen los componentes del sistema: ejecutables, ficheros de código fuente, scripts, entre otros. Tiene como objetivo principal desarrollar la arquitectura y el sistema como un todo, así como definir la organización del código. La Metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores con la opinión de éste.

A continuación se muestra la figura no. 1 donde se muestra cada uno de los artefactos generados en cada una de las fases de la metodología XP.

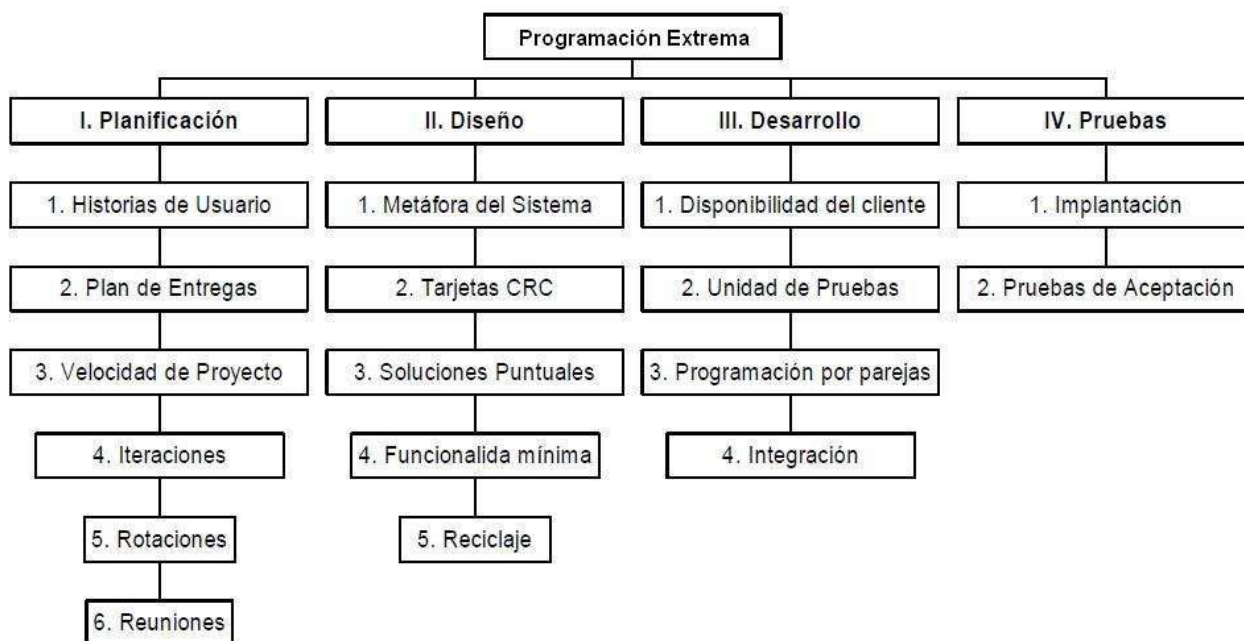


Figura No. 1 Fases de la metodología XP

#### 1.4.7. Herramienta y lenguaje de modelado

##### Visual Paradigm 5.0

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML, ideal para ingenieros de software, analistas y arquitectos que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos.



Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Es muy completa y fácil de usar, con soporte multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Permite además invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación (32).

Se selecciona esta herramienta para agilizar el diseño ya que posee una interfaz fácil de usar. Además de ser un producto de alta calidad y prestaciones, la universidad cuenta con la posibilidad de tener su licencia de uso.

### **UML 8.0**

Es un lenguaje para el desarrollo de software orientado a objetos, su propósito es visualizar, especificar, construir y documentar proyectos de software. Ayuda al usuario a entender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, reduciendo el coste y el tiempo empleado en la construcción de las piezas que constituirán el modelo; posee una gran cantidad de propiedades que han sido las que han contribuido a hacer de UML el estándar de la industria que es en realidad, algunas de estas propiedades son (32):

- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.

Se utiliza en el presente trabajo de diploma para modelar el diagrama de despliegue así como el diagrama de clases.

### **Notación de Modelado de Procesos de Negocios (BPMN)**

Es una notación gráfica que describe la lógica de los pasos de un proceso de Negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades (33).

### **Características de BPMN:**

- Es un estándar internacional de modelado de procesos.





- Es independiente de cualquier metodología de modelado de procesos.
- Permite resolver las dificultades de comunicación que tiene el lenguaje común.
- Proporciona un método normalizado para representar procesos de negocio.
- Facilita su entendimiento debido a la poca complejidad de su notación.
- Facilita la diagramación de los procesos de negocio.

En el presente trabajo de diploma se modela el negocio a través de la realización de diagramas de procesos de negocio.

#### **1.4.8. Entorno de Desarrollo Integrado (IDE)**

Un entorno de desarrollo es una aplicación de software que proporciona servicios integrales para facilitarle al programador el desarrollo de software. Normalmente un IDE consiste en un editor de código fuente, herramientas de construcción automática, un depurador y la mayoría poseen auto-completado de código.

##### **NetBeans 8.0**

Es un entorno de desarrollo gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a las siguientes tecnologías: Java, PHP, C/C++, HTML5. Además puede instalarse en varios sistemas operativos: Windows, Linux, Mac OS (34).

Características principales que influyeron en su selección para emplearlo en la solución:

- Incorpora asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos frameworks.
- Posee un editor de código robusto, multilenguaje, con el habitual coloreado y sugerencias de código, acceso a clases pinchando en el código, control de versiones, localización de ubicación de la clase actual, comprobaciones sintácticas y semánticas, plantillas de código.
- Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistentes de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario.
- Contiene herramientas para depurado de errores: el debugger que incluye el IDE es bastante útil para encontrar dónde existen errores en el código.

#### **1.5. Técnicas , métricas y pruebas a emplear para la verificación y validación del sistema**

Para la verificación y validación del sistema se hizo necesaria la aplicación de técnicas, métricas y pruebas, garantizando así que el software desarrollado cumpla los requisitos del cliente.

##### **Técnicas de validación de los requisitos**



El proceso de validación de requisitos tiene por finalidad comprobar que los requisitos del software son consistentes, precisos, realistas y definen lo que el usuario desea del producto final.

La validación se realiza a través de diversos métodos. Los métodos más utilizados son:

**Revisión de requisitos:** Consisten en una o varias reuniones planificadas, donde se intenta confirmar que los requisitos poseen los atributos de calidad deseados. Los requisitos son analizados sistemáticamente por un equipo de revisores. (35)

**Construcción de prototipos:** esta técnica consiste en la creación de una versión simplificada del sistema de software deseado, a partir de los requisitos recogidos en la especificación. Muestra, además de la interfaz, la secuencia de acciones que debe de realizar el sistema, aunque no en su totalidad. Los prototipos serán evaluados por el cliente y usuarios para comprobar su corrección y completitud. (35)

### **Métricas**

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo del software y los proyectos de mantenimiento (36).

Para llevar a cabo la validación se utilizarán las siguientes métricas:

#### **Métricas para Requisitos**

En esta disciplina se tendrán en cuenta las métricas de estabilidad, especificidad y grado de validez de los requisitos; los cuales brindaran información sobre la comprensibilidad y validez de los mismos. (37)

#### **Métricas para Diseño**

Para medir el diseño se utilizarán las métricas básicas inspiradas en el estudio de la calidad del diseño orientado a objeto que propone Pressman. Las métricas escogidas para la validación del diseño fueron la métrica orientada a clases: Tamaño Operacional de Clase (TOC) y Acoplamiento entre Clases (AC). (38)

### **Pruebas**

Las pruebas son un paso importante en el proceso de culminación de un sistema informático ya que permiten aumentar la calidad del mismo reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección.

La metodología XP divide las pruebas del sistema en dos grupos las pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final (39).

#### **Pruebas unitarias**



Las pruebas unitarias son las formas de probar el correcto funcionamiento en código de un sistema informático, facilitando que cada uno de los módulos que componen dicho sistema funcione correctamente por separado.

En la presente investigación se propone ir añadiendo pruebas unitarias para que se encuentren y solucionen errores en el sistema informático propuesto trayendo consigo que después de distintas iteraciones el código no sólo sea adecuado, sino que cada vez sea mayor el porcentaje del sistema que esté cubierto por pruebas.

### **Herramienta PHPUnit 3.4**

PHPUnit es un framework PHP para realizar pruebas unitarias. Permite realizar las pruebas pertinentes al código, verificando que el funcionamiento de las aplicaciones PHP es el deseado y en su caso encontrando errores que una vez solucionados mejorarán la calidad de nuestros desarrollos web. La actual versión PHPUnit 3.4 requiere PHP 5.1.4 o superior.

Algunos de los beneficios de PHPUnit son (40):

- Reducción del esfuerzo necesario para poner a prueba con frecuencia un mismo código
- Resultados más robustos en sus aplicaciones
- Mayor confianza en el código

### **Pruebas funcionales o de aceptación.**

Las pruebas funcionales o de aceptación son la mejor forma de probar que se tiene un sistema informático con las características que desea el cliente, desde la petición realizada por el navegador hasta la respuesta enviada por el servidor. Las pruebas funcionales prueban todas las capas del sistema informático dígame el sistema de enrutamiento, el modelo, las acciones y las plantillas (39).

Las pruebas de aceptación son una parte integral del desarrollo incremental según lo practicado por la metodología XP. Todas las historias de usuario son apoyadas por las pruebas de aceptación, las cuales son definidas por el cliente en el sistema. Estas pruebas se realizan frente a los temores de que el negocio ha sido mal entendido por parte del equipo de desarrollo. La culminación de las pruebas de aceptación garantiza la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente. Las pruebas de aceptación exigen al cliente a profundizar en su conocimiento del dominio y, precisamente, afirmar lo que el sistema informático debe hacer en circunstancias específicas, por esto, el cliente es la persona adecuada para diseñar las pruebas de aceptación.



## **1.6. Conclusiones**

Como conclusiones finales del presente capítulo se relacionan las siguientes:

- El estudio realizado permitió definir los conceptos asociados a los sistemas de gestión documental y sistemas de gestión de información vinculando ambos términos y determinando que la investigación se va a enfocar en el proceso de gestión de información de documentos generados.
- El estudio de los sistemas similares existentes demostró que estos carecen de varias funcionalidades que se desean implementar para darle solución al problema planteado. Además ninguno cumple con las características necesarias para darle solución al problema planteado en esta investigación, por lo que se concluyó que era necesario desarrollar un nuevo sistema para la gestión de la información de los procesos de las BTJ.
- Se escogieron para este contexto herramientas, lenguajes y tecnologías atendiendo a diseño, entorno de desarrollo y gestión de datos que cumplan las necesidades de la aplicación a desarrollar. Además, se seleccionó la metodología a utilizar como guía en el ciclo de desarrollo de software



## **Capítulo 2: Descripción de la solución propuesta**

### **Introducción**

Durante este capítulo se describirán un conjunto de artefactos de la metodología XP que serán generados en cada fase y ayudarán en el desarrollo del sistema informático. En la fase de Planificación se describen elementos como Historias de Usuarios, Plan de iteraciones y plan de entregas. En la fase de diseño se definen las metáforas del sistema y las tarjetas CRC y en la fase de implementación se expondrá el escenario físico del despliegue y se definen los estándares de codificación.

### **2.1. Descripción de la solución**

Se ha propuesto desarrollar un sistema informático que primeramente reduzca los tiempos de recopilación, revisión y aprobación de expedientes, así como lograr estandarizar las plantillas para la recopilación de los datos y elimine el error y la omisión de los datos en la confección de los expedientes. Su desarrollo será basado en tecnologías libres, potencializando la seguridad de los datos y asegurando la independencia tecnológica, el control de la información, confiabilidad y seguridad del sistema.

El sistema propuesto simula los procedimientos que en la práctica se llevan a cabo para conformar un expediente, una ficha técnica, una comisión o el simple hecho de formar parte de las BTJ. A grandes rasgos esos procedimientos son:

- Recopilación y conformación de la información.
- Conformación de la estructura de las BTJ (Consejo de Universidad, Consejo de Base, Brigadas de Base).
- Avalar los expedientes por los miembros de la UJC o BTJ, PCC y la administración.
- Revisar expediente en el Consejo de Base correspondiente.
- Aprobar expediente en el Consejo de Universidad.
- Conformar comisión y asignar a estas los trabajos que forman parte del concurso Científico Técnico Juvenil.

Estos procedimientos se inician con la publicación de una convocatoria en la cual se recoge las fechas de inicio y fin de los procesos (Sello Forjadores del Futuro, sello Personalidad Forjadora del Futuro, sello Colectivo Forjadores del Futuro, concurso Científico Técnico Juvenil).

#### **2.1.1. Descripción de los procesos del negocio**

En la UCI, las Brigadas Técnicas Juveniles (BTJ) están en constante accionar para cumplir los objetivos del movimiento. En esta se desarrollan diversos procesos que durante todo el año mantiene en

actividad a las brigadas y brigadistas. En estos procesos se logra que participen la mayor cantidad de jóvenes posible para mantener la superación científico-técnica de la organización.

Las Brigadas Técnicas Juveniles (BTJ) en la universidad muestran una estructura compuesta por niveles de trabajo como se muestran en la siguiente figura



**Figura No. 2 Diagrama de la estructura de las BTJ**

Cada uno de estos niveles es conformado de la siguiente manera:

- Primero se encuentran las brigadas de base de las BTJ encargado de agrupar a todo joven entre 17 y 35 años que desee investigar. Se pueden constituir brigadas de base en lugares donde existan más de 5 y hasta 15 jóvenes trabajadores entre obrero calificado, graduado de técnico medio y universitarios, y estén dadas todas las condiciones objetivas para la acción. El proceso de constitución y crecimiento está reglamentado a realizarse en el segundo y cuarto trimestre de año y en cada brigada debe ser designado un presidente y un vice-presidente todo aprobado por la UJC.
- El Consejo de Base (CB) representa un área en la universidad y es integrado por, los presidentes de cada brigada, brigadistas destacados, un representante administrativo y un representante de la Federación Estudiantil Universitaria (FEU) de esta área. En este consejo se designa un Presidente y dos vice-presidentes y cada uno de los miembros deben ser igual aprobados por la dirección de la UJC a ese nivel.
- El Consejo de Universidad (CU), integrado por los presidentes de cada Consejo Base (CB), representantes de la administración y la FEU de la universidad. Cada uno de los miembros del CU es aprobado por la dirección de la Unión de Jóvenes Comunistas (UJC) en la universidad.

Como los principales procesos de las BTJ que aborda esta investigación se encuentra el evento **Concurso Científico - Técnico Juvenil** el cual se realiza a través de jornadas científicas. Se conforman una variedad de comisiones por categoría los cuales se encargan de revisar los trabajos



(con ficha técnica y en formato digital) que les son asignados a estas por el presidente del consejo de universidad y evaluar tanto la exposición como el trabajo de estos. Después de realizar la evaluación de todos los trabajos cada comisión debe emitir un dictamen para cada trabajo asignado los cuales se deben recopilar y publicar los resultados para el conocimiento de la comunidad y de los participantes. Los trabajos que obtienen el premio de relevantes son impresos junto al dictamen, firmados por la comisión y el titular del trabajo y entregados al consejo municipal para su participación en instancias superiores de este proceso fuera de la universidad concluyendo así el proceso en esta. En el (Ver Anexo 1) diagrama de proceso de negocio Concurso Científico Técnico Juvenil se muestran las actividades que se realizan en este proceso.

Otro de los procesos abordados en la investigación otorgamiento del **sello Personalidad Forjadora del Futuro** premio que se le otorga excepcionalmente a personalidades con resultados destacados en las actividades científico-técnicas y a cercanos colaboradores del Movimiento de las BTJ con elevada incidencia en la formación de las nuevas generaciones de hombres de ciencia. El expediente que se confecciona para este proceso se compone por los datos generales del aspirante más un currículum vitae. En el (ver anexo 4) diagrama de procesos de negocio "Sello Personalidad Forjadora del Futuro" se muestran las actividades que se realizan en este proceso.

El **sello Colectivo Forjadores del Futuro** se le otorga a aquellos colectivos temporales o definitivos que hayan obtenido resultados relevantes en el desarrollo científico-técnico, con una elevada consagración al trabajo, y en los que exista un funcionamiento correcto de las BTJ.

Los aspectos a tener en cuenta para argumentar las propuestas son las siguientes:

- Funcionamiento de la UJC
- Funcionamiento de las BTJ
- Cumplimiento de los objetivos
- Logros científicos, productivos y económicos alcanzados
- Principales estímulos obtenidos individual o colectivamente

En el (ver anexo 3) diagrama de procesos de negocio "Sello Colectivo Forjadora del Futuro" se muestran las actividades que se realizan en este proceso.

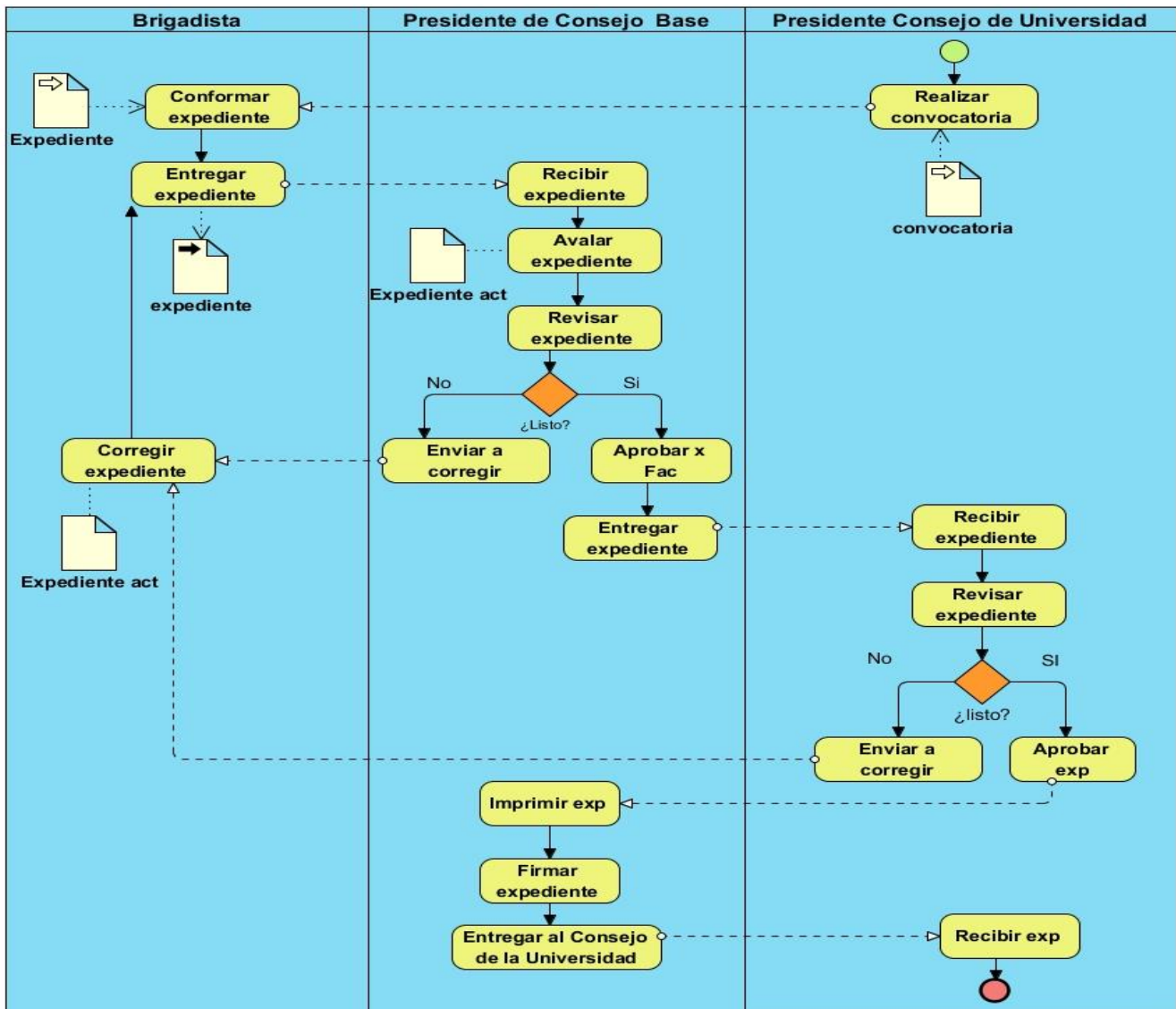
En el proceso de otorgamiento del **sello Forjador del Futuro** se procede a seguir varios pasos para la posterior aprobación por el concejo a nivel UCI. Primeramente se realiza una convocatoria que comprende los meses de enero y marzo para entregar, revisar y aprobar todos los expedientes de los trabajos presentados. Durante esta etapa los que optan al premio, deben presentar los expedientes ante el consejo de las BTJ a nivel de facultad. El consejo a nivel de facultad es el encargado de revisar y hacer cumplir todos los parámetros que se miden en la confección de los expedientes. Al término de

la revisión de cada uno de estos expedientes, el encargado de las BTJ en cada facultad debe enviar al consejo a nivel UCI todos los expedientes que fueron revisados para su posterior aprobación e impresión de cada uno de los expedientes aprobados por el consejo a nivel UCI.

Elementos que conforman un expediente a presentar para optar por distinción sello Forjador del Futuro:

- Datos generales de quien presenta el trabajo
- Ficha técnica (de 3 trabajos en calidad de autor o 5 en calidad de coautor).
- Avalués.
- Evidencias.

En la figura No. 3 se muestra el diagrama de procesos del negocio " sello Forjadores del Futuro ". Donde se evidencia las tareas que desarrollan los actores involucrados en este proceso.



**Figura No. 3 Diagrama de proceso del negocio " sello Forjadores del Futuro ".**





## 2.2. Fases y artefactos generados en la metodología XP

A continuación se describen las fases y artefactos generados en cada una de estas mediante la aplicación de la metodología XP abordada en el capítulo 1.

### 2.2.1. Planificación

En esta fase fue necesaria la constante interacción con el cliente, basado en esto se logró recolectar las historias de usuarios, se determinó el número y tamaño de las iteraciones y se determinó las fechas de las primeras entregas.

### 2.2.2. Historias de usuario

En conjunto con el cliente se realizó una descripción de las historias de usuario (HU) ayudado por el equipo de desarrollo teniendo en cuenta las funciones que debía realizar para lograr identificarlas. También es necesario destacar que las HU han jugado un papel importante en la estimación de los tiempos requeridos para el desarrollo del proyecto.

A continuación se muestran un ejemplo del contenido de la ficha de la HU, para visualizar el resto debe consultar (Anexo 1) Fichas de historias de usuarios al final del documento. Estas fichas quedan estructuradas de la siguiente forma:

**Número:** A cada HU se le asigna un número para facilitar su identificación por parte del equipo de desarrollo.

**Usuario:** Es el usuario del sistema que ejecuta la funcionalidad.

**Nombre:** Nombre descriptivo de la HU.

**Programador:** Nombre del programador encargado de desarrollar la HU.

**Prioridad en Negocio:** Grado de prioridad que le asigna el cliente a la HU en dependencia del valor en el negocio. Los valores que puede tomar son (Alta, Media o Baja).

**Riesgo en Desarrollo:** Grado de complejidad que le asigna el equipo de desarrollo a la HU luego de analizarla. (Alto, Medio o Bajo).

**Puntos Estimados:** Esfuerzo estimado por el equipo de desarrollo para darle cumplimiento a la HU.

**Iteración Asignada:** Número de la iteración en la cual será implementada la HU.

**Descripción:** Descripción simple sobre lo que debe hacer la funcionalidad a la que se hace referencia.

Historia de Usuario	
<b>Número: # 12</b>	<b>Usuario:</b> Usuario
<b>Nombre de Historia de Usuario:</b> Gestionar expediente	
<b>Programador:</b> Michel Subert Santa Cruz Pacheco	
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1



(Alta/Medio/Bajo)	
Riesgo en Desarrollo: Alta (Alta/Medio/Bajo)	Iteración Asignada:3
<b>Descripción:</b> El sistema debe permitirle al usuario registrar, modificar, eliminar y mostrar un expediente para optar por unos de los sellos de las BTJ.	
<b>Observaciones:</b>	

Tabla No. 1 Historia de usuario "Gestionar expediente"

Luego de ser recolectadas las HU el equipo de desarrollo se reúne para estimar cuanto tiempo, idealmente, llevaría implementar cada una. El proceso de estimación se realizó utilizando la técnica Planning Poker (41). El objetivo del Planning Poker es obtener una medida del tamaño relativo de todas las HU respecto a sí mismas mediante la utilización de puntos. Cada punto significa un peso, un esfuerzo o complejidad para completar un objetivo. La numeración está basada en la sucesión de Fibonacci. La distancia entre números crece conforme se hacen mayores. De esta manera, se facilita la decisión sobre qué tamaño (complejidad de implementación de la HU) tiene un objetivo según la experiencia de los desarrolladores.

También fue necesaria la priorización de las HU, esto se hizo teniendo en cuenta el valor en el negocio de cada historia para el cliente. Al finalizar se obtuvo una lista de la HU donde se refleja los siguientes campos:

**Número:** Es el número asignado a la HU para identificarla

**Nombre:** Nombre de la HU que ha sido definido.

**Prioridad:** Valor en el negocio que tiene la HU para el cliente.

**Esfuerzo:** Valor estimado mediante puntos de HU.

No.	Historia de Usuario	Prioridad	Esfuerzo
1.	Publicar convocatoria	Baja	1
2.	Administrar Comisiones del evento	Alta	3
3.	Gestionar trabajos	Alta	1
4.	Publicar dictamen da las comisiones	Alta	1
5.	Generar modelos en formato PDF	Alta	5
6.	Gestionar usuario	Alta	1
7.	Autenticar usuario	Alta	1
8.	Gestionar rol	Alta	1
9.	Administrar Consejo base	Alta	3
10.	Administrar brigada	Alta	3
11.	Gestionar evento	Alta	1
12.	Administrar Consejo de Universidad	Alta	3



13.	Gestionar expedientes.	Alta	3
14.	Avalar expediente	Alta	1
15.	Obtener reporte de brigadistas que han alcanzado el Sello Forjadores de Futuro	Media	1
16.	Obtener reporte de colectivos que han alcanzado el Sello Forjadores de Futuro	Media	1
17.	Obtener reporte de Personalidades que han alcanzado el Sello Forjadores de Futuro	Media	1
18.	Obtener reporte de total de premios obtenidos por consejo base.	Media	1
19.	Obtener reporte de total de premios por tipo por consejo base.	Media	1
20.	Asignar responsabilidades	Baja	1
21.	Asignar trabajos/expedientes para revisión	Baja	3
22.	Gestionar nomencladores	Baja	1

Tabla No. 2 Historias de usuarios detectadas.

Se obtuvo un total de 22 HU con un tiempo estimado de esfuerzo para el desarrollo del sistema de 38 puntos que representan la cantidad de días ideales para el desarrollo del sistema.

### 2.2.3. Requisitos funcionales

Una vez realizadas por el cliente las distintas historias de usuario se procede a documentar las necesidades del mismo capturando los requisitos identificados a partir de las historias de usuario. Los requisitos son la base para un desarrollo exitoso así como para una plena conformidad con el entregable final.

**Entrevista:** Es de gran utilidad para obtener información cualitativa, requiere seleccionar bien a los entrevistados para obtener la mayor cantidad de información en el menor tiempo posible. Es muy aceptada y permite acercarse al problema de una manera natural (42). Esta técnica fue aplicada a los Ing. Carlos Rafael Rodríguez y al Ing. Yasser Concepción Fernández ambos fueron presidentes del consejo de universidad y del consejo base de la organización BTJ.

A continuación se muestra el listado de requisitos definidos para el sistema informático propuesto por la investigación.

Requisitos Funcionales	
<b>RF 1 Crear convocatoria</b>	<b>RF 15 Asignar trabajo/expediente para revisión</b>
<b>RF 2 Administrar Comisión</b> <ul style="list-style-type: none"><li>• Registrar datos comisión</li><li>• Modificar datos comisión</li><li>• Eliminar comisión</li><li>• Asignar miembros a la comisión</li><li>• Asignar trabajo a la comisión</li></ul>	<b>RF 16 Gestionar Nomencladores</b> <ul style="list-style-type: none"><li>• <b>Organizaciones</b><ul style="list-style-type: none"><li>- Insertar organizaciones</li><li>- Modificar organizaciones</li><li>- Activar organizaciones</li><li>- Desactivar organizaciones</li></ul></li><li>• <b>Nivel Técnico</b></li></ul>
<b>RF 3 Gestionar trabajo</b>	



<ul style="list-style-type: none"><li>• Registrar datos del trabajo</li><li>• Modificar datos del trabajo</li><li>• Eliminar trabajo</li><li>• Mostrar detalles del trabajo</li></ul>	<ul style="list-style-type: none"><li>- Insertar nivel técnico</li><li>- Modificar nivel técnico</li><li>- Activar nivel técnico</li><li>- Desactivar nivel técnico</li><li>• <b>Tipo de premio</b><ul style="list-style-type: none"><li>- Insertar tipo de premio</li><li>- Modificar tipo de premio</li><li>- Activar tipo de premio</li><li>- Desactivar tipo de premio</li></ul></li><li>• <b>Clasificación de expediente</b><ul style="list-style-type: none"><li>- Insertar clasificación</li><li>- Modificar clasificación</li><li>- Activar clasificación</li><li>- Desactivar clasificación</li></ul></li><li>• <b>Clasificación de documento</b><ul style="list-style-type: none"><li>- Insertar clasificación</li><li>- Modificar clasificación</li><li>- Activar clasificación</li><li>- Desactivar clasificación</li></ul></li><li>• <b>Tipo aval</b><ul style="list-style-type: none"><li>- Insertar tipo de aval</li><li>- Modificar tipo de aval</li><li>- Activar tipo aval</li><li>- Desactivar tipo de aval</li></ul></li><li>• <b>Responsabilidad</b><ul style="list-style-type: none"><li>- Insertar tipo de responsabilidad</li><li>- Modificar tipo de responsabilidad</li><li>- Activar tipo de responsabilidad</li><li>- Desactivar tipo de responsabilidad</li></ul></li><li>• <b>Clasificación de Consejo</b><ul style="list-style-type: none"><li>- Insertar clasificación</li><li>- Modificar clasificación</li><li>- Activar clasificación</li><li>- Desactivar clasificación</li></ul></li><li>• <b>Como se presenta</b><ul style="list-style-type: none"><li>- Insertar denominación</li><li>- Modificar denominación</li><li>- Activar denominación</li><li>- Desactivar denominación</li></ul></li><li>• <b>Tipo resultados</b><ul style="list-style-type: none"><li>- Insertar tipo de resultados</li><li>- Modificar tipo de resultados</li><li>- Activar tipo de resultado</li><li>- Desactivar tipo de resultados</li></ul></li></ul>
<b>RF 4 Gestionar usuario</b> <ul style="list-style-type: none"><li>• Registrar datos del usuario</li><li>• Modificar datos del usuario</li><li>• Eliminar usuario</li><li>• Mostrar datos del usuario</li><li>• Buscar usuario</li></ul>	
<b>RF 5 Generar documento en formato PDF</b>	
<b>RF 6 Autenticar usuario</b>	
<b>RF 7 Administrar Consejo Base</b> <ul style="list-style-type: none"><li>• Registrar datos del Consejo Base</li><li>• Modificar datos del Consejo Base</li><li>• Eliminar Consejo Base</li><li>• Mostrar datos del Consejo Base</li><li>• Asignar miembros al Consejo Base</li></ul>	
<b>RF 8 Administrar Consejo Universidad</b> <ul style="list-style-type: none"><li>• Registrar datos del Consejo Univ.</li><li>• Modificar datos del Consejo Univ.</li><li>• Eliminar Consejo Base Univ.</li><li>• Mostrar datos del Consejo Univ.</li><li>• Asignar miembros al Consejo Univ.</li></ul>	
<b>RF 9 Administrar Brigada</b> <ul style="list-style-type: none"><li>• Registrar datos la Brigada</li><li>• Modificar datos de la Brigada</li><li>• Eliminar Brigada</li><li>• Mostrar datos de la Brigada</li><li>• Asignar miembros a la brigada</li></ul>	
<b>RF 10 Gestionar expediente</b> <ul style="list-style-type: none"><li>• Registrar datos del expediente</li><li>• Modificar datos del expediente</li><li>• Eliminar expediente</li><li>• Mostrar detalles del expediente</li></ul>	
<b>RF 11 Gestionar evento</b>	
<b>RF 17 Obtener reporte de brigadistas que han alcanzado el Sello Forjadores de Futuro</b>	
<b>RF 18 Obtener reporte de colectivos que han alcanzado el Sello Forjadores de Futuro</b>	



<ul style="list-style-type: none"><li>• Registrar datos del expediente</li><li>• Modificar datos del expediente</li><li>• Eliminar expediente</li><li>• Mostrar detalles del expediente</li></ul>	<b>RF 19 Obtener reporte de Personalidades que han alcanzado el Sello Forjadores de Futuro</b>
<b>RF 12 Avalar expediente</b>	<b>RF 20 Obtener reporte de total de premios obtenidos por consejo base</b>
<b>RF 13 Publicar dictamen de la comisión</b>	
<b>RF 14 Asignar responsabilidades</b>	<b>RF 21 Obtener reporte de total de premios por tipo por consejo base</b>
	<b>RF 22 Gestionar rol</b>

Tabla No. 3 Requisitos funcionales

#### 2.2.4. Requisitos no funcionales del sistema

##### Requisito no Funcional (RnF) 1. Interfaz de usuario.

- El sistema debe ofrecer una interfaz amigable, fácil de operar. Las interfaces deben ostentar un diseño sencillo, con pocas entradas, permitiendo un balance adecuado entre funcionalidad y simplicidad de tal manera que no se haga difícil para los usuarios utilizar el sistema.

##### Requisitos de software:

##### RnF 2. Servidor

- El servidor de base de datos debe ser PostgreSQL 9.2 o superior.
- El servidor de la aplicación debe tener instalado los módulos: php5-xsl, php5-pgsql, php5-memcache, php5-cli, php-apc, php-soap y el php5-intl.
- El servidor de aplicación debe poseer Apache 2.0 o superior.
- El servidor debe tener instalado un intérprete de PHP en su versión 5.3 o superior.

##### RnF 3. PC Cliente

- Las estaciones clientes deberán tener instalado el navegador. Se recomienda como navegador Mozilla Firefox.

##### Requisitos de hardware

##### RnF 4. Servidor

- Los servidores deben poseer memoria RAM de 2 GB o superior. El disco duro debe ser de 60 GB o superior. El procesador debe ser Intel Core i3 a una velocidad de 2.4 GHZ procesamiento o superior.

##### RnF 5. PC Cliente

- Las estaciones clientes deben poseer como mínimo 512 MB de RAM, una capacidad de 60 GB de disco duro y una velocidad de procesamiento de 2.0 GHZ.

##### Seguridad:



#### RnF 6. Restringir el acceso al sistema

- Se establecerán roles para acceder al sistema, garantizando que la información almacenada que solo podrá ser modificada y/o visualizada por los usuarios autorizados.

#### RnF 7. Almacenamiento de contraseñas

- Las contraseñas no se almacenarán en texto plano o en espacios lógicos que permitan el acceso o modificación por personas no autorizadas.

#### Rendimiento:

#### RnF 8. Reducir el tiempo de respuesta de la aplicación.

- Los tiempos de respuesta a las peticiones deben ser inferiores a los 30 segundos.

#### 2.2.5. Plan de iteraciones

Una vez determinada la prioridad de las HU, el grupo de trabajo decide cuales realizar primero y se planifica su desarrollo en la primera iteración, luego de esto se determina la velocidad del proyecto, que no es más que la suma del esfuerzo de las HU, completadas al final de la primera iteración y se utiliza para planificar las iteraciones siguientes.

Se determinó desarrollar las HU Publicar convocatoria y Administrar comisiones del evento porque son las que dan inicio al desarrollo de uno de los procesos principales y le da continuidad al desarrollo de la aplicación. Además también se desarrolla en esta iteración la HU Gestionar trabajo, HU publicar dictamen de las comisiones y la HU generar modelo de trabajos en formato PDF por su importancia en el desarrollo del proceso Evento Científico Técnico Juvenil. Las demás iteraciones se distribuyeron en dependencia del orden en que se implementan. Cuando se toma el total de puntos estimados entre la velocidad del proyecto determinada en la primera iteración, se definen cuantas iteraciones son necesarias para el desarrollo de las HU. La suma de los días que costara desarrollar todas las tareas de la iteración no debería sobrepasar la velocidad del proyecto de la iteración anterior.

Plan de Iteraciones				
Iteración	Historia de usuario	Prioridad	Esfuerzo	Tiempo de duración
1	Publicar convocatoria	Baja	1	11
	Administrar Comisiones	Alta	3	
	Gestionar trabajos	Alta	1	
	Publicar dictamen da las comisiones	Alta	1	
	Generar modelos de documentos en formato PDF	Alta	5	
2	Gestionar usuario	Alta	1	12



	Autenticar usuario	Alta	1	
	Gestionar rol	Alta	1	
	Administrar Consejo base	Alta	3	
	Administrar brigada	Alta	3	
	Administrar Consejo de Universidad	Alta	3	
3	Gestionar datos expedientes.	Alta	3	11
	Avalar expediente	Alta	1	
	Gestionar eventos	Alta	1	
	Obtener reporte de brigadistas que han alcanzado el Sello Forjadores de Futuro	Media	1	
	Obtener reporte de colectivos que han alcanzado el Sello Forjadores de Futuro	Media	1	
4	Obtener reporte de Personalidades que han alcanzado el Sello Forjadores de Futuro	Media	1	8
	Obtener reporte de total de premios obtenidos por consejo base.	Media	1	
	Obtener reporte de total de premios por tipo por consejo base.	Media	1	
	Asignar responsabilidades	Baja	1	
	Asignar trabajos/expedientes para revisión	Baja	3	
	Gestionar nomencladores	Baja	1	

Tabla No. 4 "Plan de iteraciones".

### 2.2.6. Plan de entrega

Ya elaboradas por el equipo de desarrollo las distintas HU y confeccionado por los desarrolladores el plan de iteración se procede a la confección del plan de entrega. El cronograma de entrega se realiza en base a las estimaciones de tiempos de desarrollo realizadas por el equipo de desarrollo, fijándose un periodo de tiempo determinado por puntos sobre el cual se debe implementar cada HU. El equipo de desarrollo determinó que cada punto equivale a un día ideal de trabajo.

Para aproximar el tiempo de la iteración se tomó como medida una semana, cada semana consta de cinco días en los que se trabaja 8 horas sin distracción.

A continuación se verá cómo queda elaborado el plan de entrega que tiene los siguientes campos.

**No.:** Numero de la iteración.

**Historia de usuario:** Nombre de la HU.

**Esfuerzo:** Valor estimado mediante puntos de HU.

**Duración (semanas):** Tiempo aproximado de la duración de la iteración en semanas.

**Fecha de inicio:** Fecha en la que comienza a desarrollarse la iteración.

**Fecha de Fin:** Fecha de entrega de la iteración.



Plan de entrega					
No.	Historia de usuario	Esfuerzo	duración	Fecha inicio	Fecha fin
1	Publicar convocatoria	1	2.2	18/03/2015	04/04/2015
	Administrar Comisiones del evento	3			
	Gestionar trabajos	1			
	Publicar dictamen da las comisiones	1			
	Generar modelos de trabajos en formato PDF	5			
2	Gestionar usuario	1	2.4	05/04/2015	22/04/2015
	Autenticar usuario	1			
	Gestionar rol	1			
	Administrar Consejo base	3			
	Administrar brigada	3			
Administrar Consejo de Universidad	3				
3	Gestionar datos expedientes.	3	2.2	23/04/2015	07/05/2015
	Avalar expediente	1			
	Generar modelos de expediente en formato PDF	5			
	Obtener reporte de brigadistas que han alcanzado el Sello Forjadores de Futuro	1			
	Obtener reporte de colectivos que han alcanzado el Sello Forjadores de Futuro	1			
4	Obtener reporte de Personalidades que han alcanzado el Sello Forjadores de Futuro	1	1.6	08/05/2015	23/05/2015
	Obtener reporte de total de premios obtenidos por consejo base.	1			
	Obtener reporte de total de premios por tipo por consejo base.	1			
	Asignar responsabilidades	1			
	Asignar trabajos/expedientes para revisión	3			

Tabla No. 5 "Plan de entrega".





### 2.3. Diseño

El diseño es realizado durante todo el tiempo de vida del proyecto, siendo revisado y modificado debido a cambios presentados durante el desarrollo de forma casi constante. Este epígrafe muestra los resultados prácticos de la descripción teórica sobre el diseño.

Durante el diseño se invirtió el tiempo exclusivamente necesario en la elaboración de diagramas y diseños de interfaces gráficas. Se invirtió el tiempo necesario en el diseño de interfaces para ubicar los elementos según el cliente los solicitó. En los que refiere a diagramas se crearon tarjetas CRC y el modelo entidad-relación del cual han surgido varias versiones a la medida que se han incorporado funcionalidades a la aplicación.

#### 2.3.1. Tarjeta CRC

Las tarjetas CRC fueron las bases para la obtención del modelo entidad relación. Cada clase reflejada en la tarjeta se convirtió en un objeto sus responsabilidades en métodos públicos y sus colaboradores en llamados a otras clases.

En XP el proceso de diseño es iterativo por lo que la creación de las tarjetas no es en un mismo tiempo, se crean según las iteraciones y se le añaden responsabilidades y colaboradores según se haga necesario. A continuación se muestran las clases del diseño identificadas así como dos ejemplos de las tarjetas CRC, el resto podrá verse en el **Anexo 3** al final del documento.

#### Clases:

- Comisión
- Consejo
- Documento
- Expediente
- Ficha \_ técnica
- Usuario
- Colectivo
- Evento
- Logros
- Convocatoria
- Premio
- Aval
- DocumentoAdjunto
- ComisionExpediente

En las Tarjetas CRC el nombre de la clase se coloca a modo de título, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha.

**Nombre:** Nombre de la clase con que se está modelando.

**Responsabilidades:** Las responsabilidades de una clase son los elementos que conoce y las que realizan, sus atributos y métodos.

**Colaboradores:** los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.



A continuación se muestra el ejemplo:

Tarjeta CRC # 2	
Datos de la Clase	
<b>Nombre: TbExpediente</b>	
Responsabilidades	Colaboradores
RegistrarExpediente()	DatUsuario, DatDocumento, NomEstado, DatAval, NomClasificacionExpediente
ModificarExpediente()	
EliminarExpediente()	
AvalarExpediente()	
MostrarExpediente()	

Tabla No. 6 Tarjeta CRC # 2

## 2.4. Modelo de datos

El modelo de datos representa todas las tablas (entidades) que permiten el funcionamiento del negocio y muestra los datos que serán contenidos en el sistema. El modelo de datos obtenido cuenta con un total de 37 tablas donde 18 son nomencladores y encargados de gestionar conceptos específicos del negocio ya predefinidos como ejemplo se tiene la tabla nomClasificaciónExpediente, que se relaciona con la tabla datExpediente y se definen los tipos de expedientes que existirán en el sistema (sello Forjadores del Futuro, personalidad Forjadora del Futuro, colectivo Forjadores del Futuro y Trabajo Científico - Técnico). Las restantes tablas facilitan el registro de datos que son necesarios para el funcionamiento de la aplicación. En el modelo de datos las tablas están representadas con el identificador **dat** son las tablas generales y con el identificador **nom** las tablas nomencladoras. En el **Anexo 4** se puede visualizar el modelo de datos del sistema.

### 2.4.1. Patrones de diseño de bases de datos

Los patrones de diseño de bases de datos son plantillas que ya han sido evaluadas como las responsables de resolver un problema, son guía para apoyarse en realizar el trabajo. En esencia los patrones de diseño de bases de datos constituyen la base para la búsqueda de soluciones a problemas comunes en el proceso de diseño de las mismas.

#### Patrón Llaves subrogadas

El proceso de creación de los atributos identificadores no se realizó arbitrariamente, sino que se aplicó el patrón "Llaves subrogadas". Este patrón es de gran utilidad pues plantea la creación de una llave primaria única para cada entidad cuyo único requisito es almacenar un valor numérico único para cada

fila de la tabla, actuando como una clave sustituta, de forma totalmente independiente a los datos de negocio. (43) En la siguiente figura se puede visualizar un ejemplo del mismo.

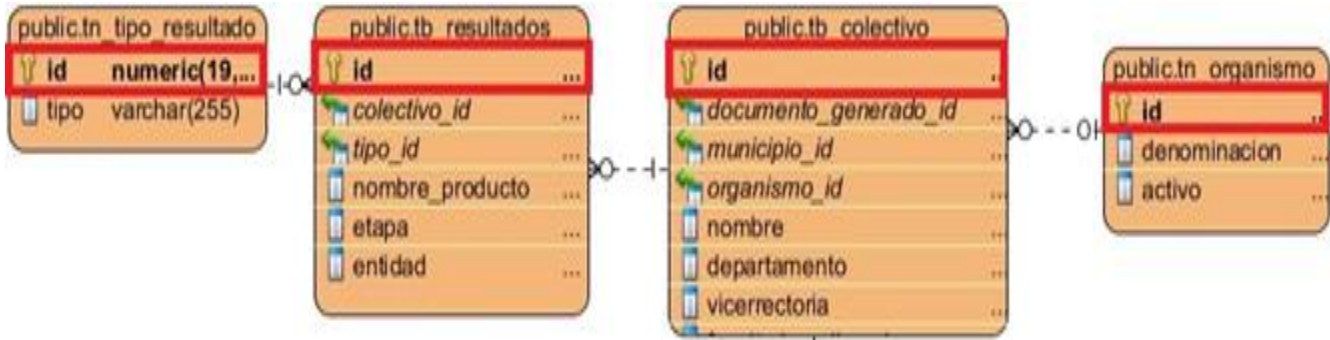


Figura No. 4 Ejemplo del Patrón llaves subrogadas

## 2.4.2. Patrones de diseño

### Patrón de arquitectura

Los patrones arquitectónicos son aquellos que se expresan un esquema organizativo estructural fundamental para sistemas de software. Un patrón de arquitectura de software es un esquema genérico probado para solucionar un problema particular recurrente que surge en un cierto contexto. Este esquema se especifica describiendo los componentes con su responsabilidades, relaciones y las formas en que colaboran (44).

### Patrón Modelo-Vista-Controlador (MVC)

El patrón de arquitectura conocido como Modelo Vista Controlador (MVC), separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario; es decir separa en tres capas diferentes los datos de una aplicación, la interfaz de usuario, y la lógica de control (45):

- **Modelo:** Esta capa administra el comportamiento y los datos del dominio de la aplicación, responde a requisitos de información sobre su estado (usualmente formulados desde la vista) y a instrucciones de cambiar el estado (habitualmente desde el controlador). En esta capa es conformada por las clases entidad y repositorio que encapsula parte del negocio.
- **Vista:** Esta capa maneja la visualización de la información, es decir que presenta el modelo en un formato adecuado para interactuar, que usualmente es la interfaz de usuario. Esta capa está compuesta por las clases javascripts del framework Extjs 2.2.
- **Controlador:** Esta capa controla el flujo de datos entre la vista y el modelo; es decir que responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y

probablemente tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases.

Esta separación permite construir y probar el modelo, independientemente de la representación visual



Figura No. 5 Patrón Modelo Vista Controlador

### Patrones de diseño que implementa Symfony.

El marco de trabajo Symfony utiliza varios patrones de diseño, a continuación se ejemplificará cuáles son y cómo son utilizados.

**Unidad de trabajo:** Ya que Doctrine es consciente de todas las entidades gestionadas, cuando se realiza la llamada al método flush(), este calcula el conjunto de cambios y ejecuta la(s) consulta(s) lo más eficiente(s) posible(s). Por ejemplo, si se persiste un total de 100 objetos y, posteriormente se llama a flush(), Doctrine creará una sola declaración preparada y la volverá a utilizar en cada inserción. Este patrón se conoce como Unidad de trabajo, y se utiliza porque es rápido y eficiente. A continuación ejemplo de la utilización del patrón unidad de trabajo en la clase DatEventoRepository.

```
function eliminar($objetos) {
    $em = $this->getEntityManager();
    foreach ($objetos as $id) {
        $e = $em->getRepository('AppBundle:DatEvento')->find($id);
        $em->remove($e);
    }
    $em->flush();
    $msg = (count($objetos) > 1) ? 'Los eventos han sido eliminados satisfactoriamente.' : 'El evento ha sido eliminado satisfactoriamente.';
    return array('codMsg' => 3, 'msg' => $msg);
}
```

Figura No. 6 Ejemplo del patrón Unidad de trabajo.

Patrones GRASP utilizados:

- **Experto:** este patrón es aplicado a todas las clases debido a que cada una de ellas es experta pues contiene la información necesaria para cumplir con las responsabilidades que les fueron asignadas; facilitando así el entendimiento, extensión y mantenimiento del sistema. Además de conservar el encapsulamiento y contribuir al bajo acoplamiento.

Nom_clasificacion_expediente
-denominacion : String
-descripcion : String
+getDenominacion()
+getDescripcion()
+setDenominacion() : void
+setDescripcion() : void
+__construct() : void

Figura No. 7 Clase experta en la información de los tipos de clasificación de los expediente.

- **Alta Cohesión:** Una de las características principales de Symfony es la organización del trabajo en él mismo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo, las clases con sufijo Controller en su nombre contienen varias funcionalidades estrechamente relacionadas entre ellas, teniendo un sentido común y un propósito único, siendo las encargadas de controlar las acciones de las plantillas y por lo tanto pertenecen a la capa del Controlador dentro de la arquitectura Modelo-Vista-Controlador. Esto hace posible que el software sea flexible a cambios sustanciales con efecto mínimo.
- **Bajo Acoplamiento:** Mediante este patrón se disminuyen las dependencias entre clases. No puede verse separado de otros patrones como son el experto y la alta cohesión. Su utilización se evidencia por ejemplo en la capa de acceso a datos ya que las mismas clases tienen un alto grado de independencia de las clases de abstracción de datos, es decir, las clases entidades no tienen asociación con la vista ni con el controlador. La poca dependencia entre esas clases permite que en la aplicación sean las entidades las de mayor reutilización. Con el uso de este patrón los componentes no se afectan por cambios de otros componentes, son fáciles de entender por separado y fáciles de reutilizar (44).
- **Controlador:** El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

(44) Este patrón se puede observar por lo general en la clase controladora ExpedienteController y en el controlador frontal de Symfony (app.php): siendo este el único punto de entrada a la aplicación en un entorno determinado. A continuación se muestra la figura 6 donde se ejemplifica el uso del patrón controlador.



Figura No. 8 Ejemplo del uso del patrón controlador en la clase ExpedienteController.

#### Patrones GOF utilizados:

Los patrones GoF (Gang of Four o banda de los 4, formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides) se clasifican en 3 categorías basadas en su propósito:

**Patrones de comportamientos:** Más que describir objetos o clases, describen la comunicación entre ellos.

**Patrones estructurales:** Separan la interfaz de la implementación. Se ocupan de como las clases y objetos se agrupan, para formar estructuras más grandes.

**Patrones creacionales:** Se encargan de la inicialización y configuración de objetos.

El marco de trabajo Symfony2 el cual es uno de las herramientas que se utilizara para la implementación del sistema a desarrollar ya trae consigo implementados muchos de estos patrones en su sistema interno, dándole solución a muchos problemas que se presentan continuamente e implementando así numerosas funcionalidades para facilitar el trabajo de los desarrolladores

En la categoría de patrones estructurales:

- **Decorator (Decorador):** Este patrón nos permitirá añadir funcionalidades o responsabilidades a un objeto de forma dinámica. Este patrón nos provee de una alternativa muy flexible a la creación de subclases para extender funcionalidades. Esto nos permitirá, en tiempo de ejecución, añadir esta funcionalidad o responsabilidad extra que necesita nuestro objeto. Esto se podría hacer a través de la herencia pero no sería lo suficientemente flexible ya que de esta forma serían añadidas de forma estática (44). En vez de esto, nuestro objeto decorador rodeará al objeto inicial añadiendo estas responsabilidades o funcionalidades. En la aplicación el archivo

Base.html.twig o plantilla global almacena el código HTML que es común a todas las páginas de la herramienta.

En la categoría de patrones de comportamiento

- **Observador:** Cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes. Un complemento debe ser capaz de agregar métodos, o hacer algo antes o después de ejecutar un método, sin interferir con otros complementos. Este patrón viene implementado por defecto en el marco de trabajo Symfony 2. Ejemplo de esto es el despachador de eventos de Symfony2 (ver Figura 7) en el cual se implementa el patrón observador de manera sencilla y efectiva. Este permite que una vez creado un objeto respuesta, pueda ser útil permitir que otros elementos en el sistema lo modifiquen (por ejemplo, añadan algunas cabeceras de caché) antes de utilizarlo realmente. Para hacer esto posible, el núcleo de Symfony2 lanza un evento “*kernel.response*”.

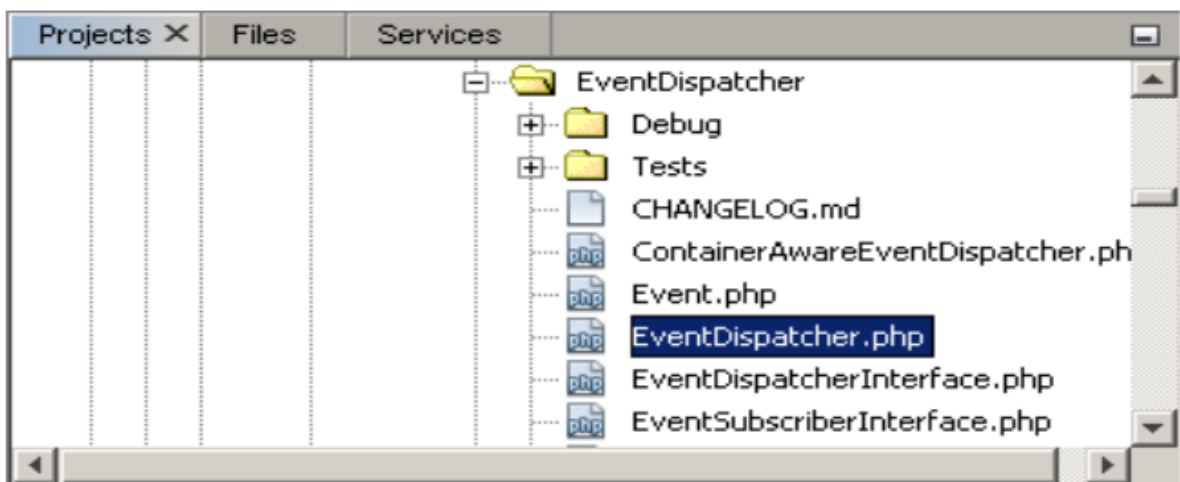


Figura No. 9 Paquete despachador de eventos de Symfony2.

- **Composite:** El objetivo de este patrón es el de montar objetos en una estructura de árbol (en nuestro caso árbol simple) para representarlos de una forma jerarquizada. De este modo, se podrán manejar de forma similar y uniforme tanto los objetos individuales como los objetos compuestos (44). Este patrón es aplicable en situaciones en la que se necesita trabajar con elementos simples y colecciones que contienen elementos simples u otras colecciones, obteniendo de esta forma la estructura de árbol comentada. Para visualizar la utilización de este patrón en la siguiente figura.

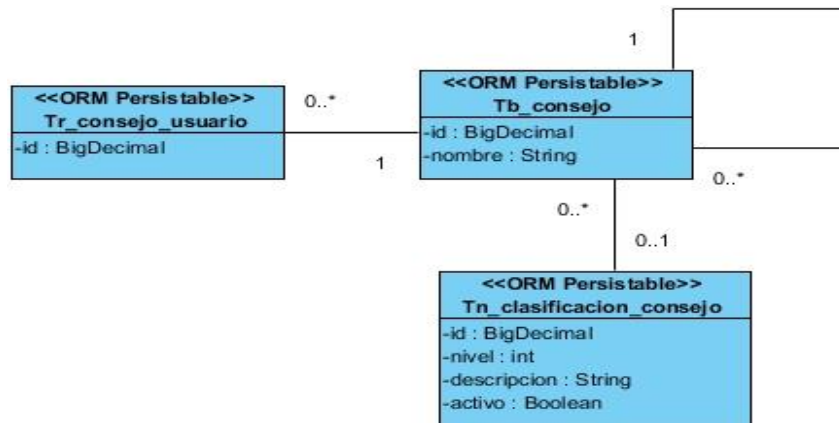


Figura No. 10 Ejemplo donde se evidencia el patrón composite (árbol simple)

## 2.5. Implementación

Algunos de los elementos en cuanto a implementación más importantes son, que el cliente debe estar presente todo el tiempo, se debe trabajar en parejas y debe haber una propiedad colectiva del código para su mejor entendimiento.

En esta fase se genera todo el código fuente necesario para satisfacer las HU definidas para la solución y se describen todas las tareas realizadas en cada iteración. Al inicio de cada iteración, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. Todas las HU son traducidas en tareas de programación, cada una de ellas se asignó a un programador.

### 2.5.1. Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. El equipo de desarrollo decidió definir varios estándares de codificación que son mostrados a continuación.

#### Cabecera del archivo

Siempre es importante que todos los archivos .php inicien con una cabecera específica que indique información de la versión, autor de los últimos cambios, etc. Cada clase contiene un bloque de documentación acorde con el estándar de PHPDocumentor<sup>1</sup>. Ejemplo:

```
/**
 * @Clase persistente área. "tb_expediente.php"
 * @versión:1.0.1
 * @Modificado:14 de Abril del 2015
```

<sup>1</sup> PHPDocumentor: Script desarrollado en php que nos ayuda a extraer la documentación de nuestros proyectos.





\*@Autor: Michel Subert

\*/

### **Clases**

- Las clases formularios comienzan con el nombre del formulario según su función, seguido de la palabra Type (ConsejoType).
- Las clases controladoras de negocio comienzan con el nombre seguido de la palabra Controller.
- Las clases serán colocadas en un archivo .php aparte, donde solo se colocara el código de la misma. El nombre del archivo será el mismo de la clase y siempre comenzara con mayúscula.
- Debe colocarse un comentario antes de la declaración de la clase explicando su utilidad. Los nombres de las clases se escriben con la primera letra de cada palabra que lo compone en mayúscula, haciendo uso de la notación PascalCase o más conocida como CamelCase (versión UpperCamelCase).

### **Nombre de variables**

- El identificador para las variables y los parámetros serán con letras en minúsculas y en caso de ser un nombre compuesto se divide cada palabra por un signo de underscore "\_".
- Los nombres deben ser descriptivos y concisos. No usar grandes frases, pequeñas abreviaciones para las variables.

Siempre es mejor saber qué hace una variable con sólo conocer su nombre. Esto aplica para los nombres de variables, funciones, argumentos de funciones y clases.

### **2.5.2. Modelo de despliegue**

El modelo de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se relacionan en esos nodos.

En la siguiente figura se muestra de la forma que el usuario tendrá acceso a la aplicación, esto será a través de un navegador web en una PC Cliente, con una conexión por HTTPS. La aplicación se conecta al servidor de base de datos a través del protocolo TCP/IP y realiza las consultas a la BD de la aplicación.

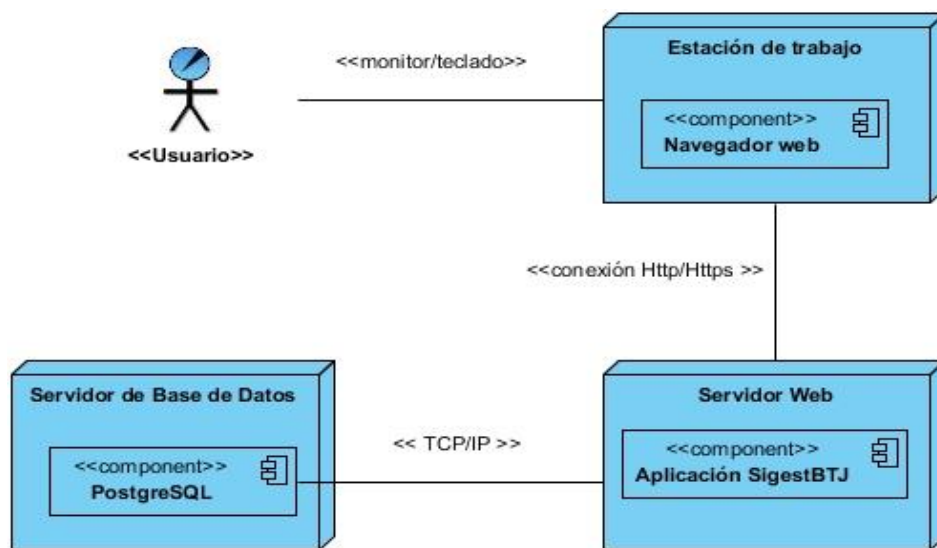


Figura No. 11 Propuesta de escenario del despliegue

La descripción de los nodos de la figura anterior se muestra a continuación:

- **Computadora personal (PC) cliente:** este nodo representa la computadora utilizada por el usuario para conectarse al servidor web mediante el protocolo HTTPS haciendo uso de un navegador web.
- **Servidor Web:** En este nodo se encuentra desplegado o instalado el servidor web apache donde estará desplegada la aplicación.
- **Servidor de base de datos:** en este nodo se encuentra desplegado o instalado el servidor de base de datos con el que interactúa el sistema.

## 2.6. Conclusiones

- La modelación de negocio permitió comprender los procesos o etapas por las que pasa un expediente en las BTJ.
- La descripción de los procesos de las BTJ permitió conocer la necesidad de una herramienta acorde a las necesidades de la organización BTJ.
- Se realizó la planificación de las entregas y la planificación del esfuerzo por cada una de las historias de usuario.
- Se aplicaron buenas prácticas en el diseño de la solución mediante la utilización de los diferentes patrones
- La utilización de los estándares de implementación, facilitaron en entendimiento del código por el desarrollador y el futuro mantenimiento del sistema informático.



## Capítulo 3: Validación de resultados

### Introducción

En este último capítulo se pretende verificar y validar el correcto funcionamiento del sistema, la verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica y en la validación se ejecutan un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente.

Para la comprobación final del sistema SigestBTJ (Sistema informático para la gestión de la información de los procesos de las Brigadas Técnicas Juveniles) se aplicarán técnicas de validación de requisitos y métricas de software, además se realizarán pruebas unitarias y pruebas de aceptación.

### 3.1. Aplicación de técnicas de validación de requisitos y utilización de métricas

#### Técnicas de validación de requisitos

La validación de los requisitos tiene como objetivo comprobar que estos son correctos. Esta fase debe realizarse o de lo contrario se corre el riesgo de implementar una mala especificación, con el costo que eso conlleva. Es muy importante asegurar la validez de los requisitos antes de comenzar el desarrollo del software. Para ello debe hacerse una comprobación de la correspondencia entre las descripciones iniciales y la definición de los requisitos realizada, para verificar que responden a lo que desea el usuario final. Para llevar a cabo este proceso, se aplicaron las siguientes técnicas de validación de requisitos:

- Revisiones de requisitos: los requisitos se analizan sistemáticamente por un revisor técnico en este caso por el desarrollador de la investigación y el cliente.

Al poner en práctica estas revisiones, se detectaron en una primera iteración cinco problemas relacionados con una mala expresión de los requisitos y falta de información a la hora de definirlos. Al realizar la segunda iteración se comprobó que se habían resuelto los errores antes encontrados y que la definición efectuada era correcta.

- Construcción de prototipos: el prototipado de interfaz de usuario es una técnica de representación aproximada de la interfaz de usuario de un software que permite a clientes y usuarios entender fácilmente la propuesta de los ingenieros de requisitos para resolver sus problemas de negocio.

Esta técnica se aplicó diseñando prototipos de interfaz de usuario para que el cliente entendiera de qué forma iba a quedar el sistema que se comenzaba a desarrollar, para ello se realizaron reuniones para mostrar los prototipos, en una primera reunión, el cliente no estuvo de acuerdo con tres de los prototipos realizados, luego se modificaron dichos prototipos y en la segunda reunión efectuada se consiguió la aprobación del cliente.

### Métricas

Un elemento primordial de cualquier proceso de ingeniería es la medición. Se emplean medida para valorar la calidad de los sistemas que se construyen. Para conseguir esta medición se va a hacer uso de métricas para los requisitos y para el diseño.

#### Métricas para los requisitos

Una vez definidos los requisitos del sistema, estos deben ser validados para asegurar que los que los requisitos que se determinaron especifican realmente el sistema que el cliente desea obtener.

#### Aplicación de la métrica Estabilidad de requisitos:

$$ETR = \left[ \frac{RT - RM}{RT} \right] * 100 \quad (38)$$

Donde:

**RT:** Total de requisitos definidos.

**RM:** Cantidad de requisitos modificados.

**Se identificaron un total de 22 requisitos funcionales, de los cuales 4 fueron modificados, por tanto:**

$$ETR = [(22 - 4) / 22] * 100 = 82$$

Como resultado se obtuvo un valor redondeado de 82, este número demuestra que no se han realizado cambios significativos sobre los requisitos y por lo tanto estos son estables.

#### Aplicación de las métricas Especificidad de requisitos

La especificidad de los requisitos se calcula como:

$$Q_1 = \frac{n_{ui}}{n_r} \quad (38)$$

**Q1:** Especificidad de los requisitos.

**nr:** Total de requisitos definidos.

**nui:** Total de requisitos donde se tuvo una interpretación idéntica.

El valor de esta métrica debe estar siempre entre 0 y 1. Mientras más cerca de 1 esté el valor de Q1, mayor será la consistencia de la interpretación de los revisores para cada requisito y menor será la ambigüedad en su especificación.

**Se obtiene como resultado:**

$$Q_1 = 22/22 = 1$$



El valor que se obtuvo de  $Q_1$  es 1, lo cual se puede interpretar que la especificación de los requisitos no posee ambigüedad. Esta característica garantiza mayor calidad en el proceso de especificación.

#### **Aplicación de la métrica Grado de Validación de los requisitos:**

El Grado de Validación de los requisitos mide la corrección en la definición de dichos requisitos. Este valor se calcula con la siguiente fórmula:

$$Q_3 = \frac{n_c}{(n_c + n_{nv})} \quad (38)$$

**Donde:**

**$Q_3$ :** Grado de Validación de los requisitos.

**$n_c$ :** Total de requisitos validados correctamente.

**$n_{nv}$ :** Total de requisitos no validados.

El resultado de la métrica esta siempre entre 0 y 1. El valor óptimo, es el más cercano a 1 e indica un alto nivel de corrección en la definición de los requisitos.

**Se obtiene como resultado:**

$$Q_3 = 22/(22+0) = 22/22 = 1$$

La aplicación de esta métrica dio como resultado **1**, por lo tanto se concluye que la definición de los requisitos es correcta.

#### **Métricas para diseño**

Para medir el diseño se utilizarán la siguientes métricas orientadas a clases: Tamaño Operacional de Clases (TOC) y Relación entre Clases (RC).

#### **Aplicación de las métricas Tamaño Operacional de Clases:**

A continuación se muestran los resultados obtenidos tras la aplicación de las métricas TOC donde:

CA: cantidad de atributos de la clase.

R: responsabilidad de la clase.

Re: reutilización de clase.

CP: cantidad de procedimientos.

CI: complejidad de implementación de la clase.

Teniendo en cuenta los siguientes criterios:



Atributo	Categoría	Criterio
Responsabilidad	Baja	CP <= Promedio
	Media	Promedio <= CP <= 2* Promedio
	Alta	CP > 2* Promedio
Complejidad de Implementación	Baja	CP <= Promedio
	Media	Promedio <= CP <= 2* Promedio
	Alta	CP > 2* Promedio
Reutilización	Baja	CP > 2* Promedio
	Media	Promedio <= CP <= 2* Promedio
	Alta	CP <= Promedio

Tabla No. 7 Rango de valores para la evaluación de los atributos relacionados con la métrica TOC

No	Clases	CA	CP	R	CI	Re
1	TbEvento	6	13	Media	Baja	Alta
2	TbComision	2	5	Baja	Baja	Alta
3	TbConsejo	4	9	Baja	Baja	Alta
4	TbAval	5	11	Media	Media	Media
5	TbConvocatoria	14	9	Baja	Alta	Baja
6	TbDocumento	3	7	Baja	Alta	Baja
7	TbDocumentoAdjunto	3	7	Baja	Baja	Alta
8	TbDocuemntoGenerado	2	5	Baja	Baja	Alta
9	TbExpediente	5	11	Media	Baja	Alta
10	TbFichaTecnica	12	25	Alta	Baja	Alta
11	TbFlujo	5	11	Media	Media	Media
12	TbLogros	6	13	Media	Media	Media
13	TbPremio	4	9	Baja	Baja	Alta
14	TbResultados	6	13	Media	Media	Media
15	TbRol	4	9	Baja	Baja	Alta
16	TbTramite	4	9	Baja	Baja	Alta
17	TbColectivo	18	37	Alta	Alta	Baja
18	TbUsuario	17	35	Alta	Alta	Baja

Tabla No. 8 Resultados de la aplicación de la métrica TOC a las clases principales del sistema.

Se trabajó con un total de 60 clases para un promedio de CP (cantidad de procedimientos) de 9 obteniéndose como resultados los datos que a continuación se muestran.

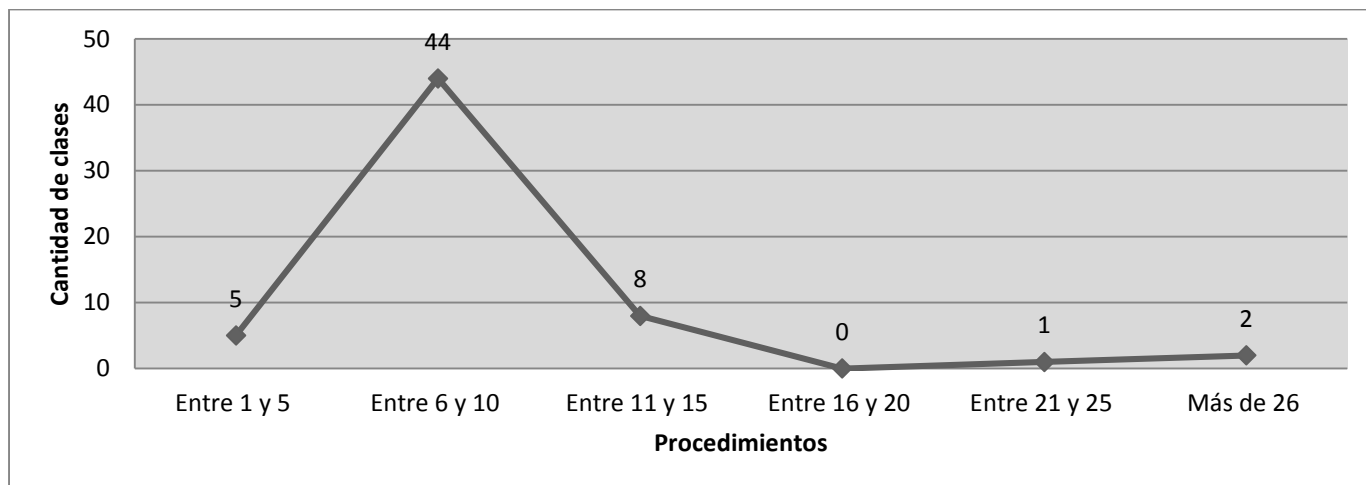


Figura No. 12 Representación de la cantidad de procedimientos por clase

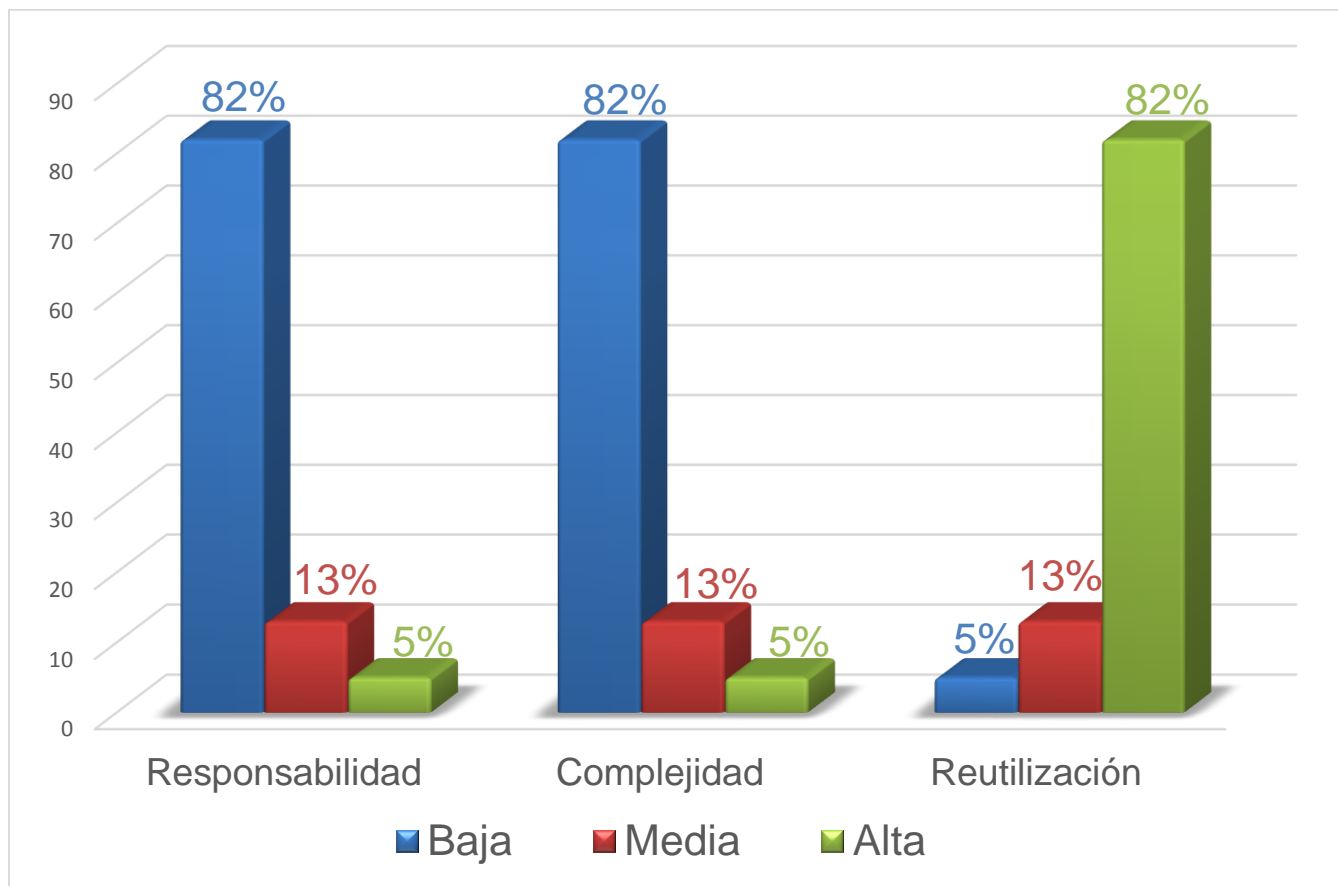


Figura No. 13 Resultados de la aplicación de la métrica TOC.

**Responsabilidad:** luego de aplicar la métrica se obtuvieron resultados satisfactorios que reflejan que un 82%.de las clases tienen responsabilidad baja.

**Complejidad de la implementación:** después de haberse realizado la medición de la métrica, arrojó también resultados positivos ya que la complejidad de las clases es baja en un 82%.

**Reutilización:** se obtuvieron valores que según muestra la gráfica de la figura anterior se comporta en un nivel alto con un 82%.

Haciendo un análisis de los resultados obtenidos para los atributos de la métrica TOC se puede observar que el atributo reutilización cuenta con un porcentaje alto demostrando así que el componente cuenta con una elevada reutilización, baja responsabilidad y complejidad en el diseño propuesto. Por lo que se concluye que los resultados obtenidos en esta métrica son positivos.

### Aplicación de la métrica Relación entre Clases (RC):

A continuación se muestran los resultados obtenidos tras la aplicación de la métrica RC, donde:

CR: cantidad de relaciones de uso.

CM Complejidad de mantenimiento.

A: Acoplamiento.

CP: cantidad de pruebas.

Atributo	Categoría	Criterio
Acoplamiento	Bajo	CR <= 1
	Medio	2
	Alto	CR > 2
Complejidad de mantenimiento	Baja	CR <= Promedio
	Media	Promedio <= CR <= 2 *
	Alta	CR > 2 * promedio
Cantidad de pruebas	Baja	CR <= Promedio
	Media	Promedio <= CR <= 2 *
	Alta	CR > 2 * promedio
Reutilización	Baja	CR > 2 * promedio
	Media	Promedio <= CR <= 2 *
	Alta	CR <= Promedio

**Tabla No. 9** Rango de valores para la evaluación de los atributos relacionados con las métricas.

No	Clase	CR	A	CM	CP	R
1	TbEvento	3	Alto	Alta	Baja	Alta
2	TbComision	2	Medio	Media	Media	Media
3	TbConsejo	2	Medio	Media	Media	Media
4	TbAval	3	Alto	Alta	Baja	Alta
5	TbConvocatoria	1	Bajo	Baja	Alta	Baja





6	TbDocumento	2	Medio	Media	Media	Media
7	TbDocumentoAdjunto	2	Medio	Media	Media	Media
8	TbExpediente	3	Alto	Alta	Baja	Alta
9	TbFichaTecnica	2	Medio	Media	Media	Media
10	TbLogros	1	Bajo	Baja	Alta	Baja
11	TbPremio	3	Alto	Alta	Baja	Alta
12	TbResultados	2	Medio	Media	Media	Media
13	TbRol	0	Bajo	Baja	Alta	Baja
14	TbColectivo	2	Medio	Media	Media	Media
15	TbUsuario	3	Alto	Alta	Baja	Alta
16	TnCategoria	0	Bajo	Baja	Alta	Baja
17	TnClasificacionConsejo	0	Bajo	Baja	Alta	Baja
18	TnClasificacionDocumento	1	Bajo	Baja	Alta	Baja
19	TnClasificacionExpediente	0	Bajo	Baja	Alta	Baja
20	TnEstado	0	Bajo	Baja	Alta	Baja
21	TnMunicipio	1	Bajo	Baja	Alta	Baja
22	TnNivelEvento	0	Bajo	Baja	Alta	Baja
23	TnNivelTecnico	0	Bajo	Baja	Alta	Baja
24	TnOrganizaciones	0	Bajo	Baja	Alta	Baja
25	TnNivel	0	Bajo	Baja	Alta	Baja
26	TnOrganismo	0	Bajo	Baja	Alta	Baja
27	TnCategoriaLogros	1	Bajo	Baja	Alta	Baja
28	TnPremio	0	Bajo	Baja	Alta	Baja
29	TnPresenta	0	Bajo	Baja	Alta	Baja
30	TnProvincia	0	Bajo	Baja	Alta	Baja
31	TnResponsabilidad	0	Bajo	Baja	Alta	Baja
32	TnTipo	0	Bajo	Baja	Alta	Baja
33	TnTipoAval	0	Bajo	Baja	Alta	Baja
34	TrConsejoUsuario	3	Bajo	Alta	Baja	Alta
35	TrComisionExpedienteUsuario	3	Alto	Alta	Baja	Alta
36	TrFichaTecnicaUsuario	3	Alto	Alta	Baja	Alta
37	TrOrganizacionesUsuario	2	Medio	Media	Media	Media
38	TrUsuarioRol	2	Medio	Media	Media	Media

**Tabla No. 10 Resultados de la aplicación de la métrica RC a 38 clases.**

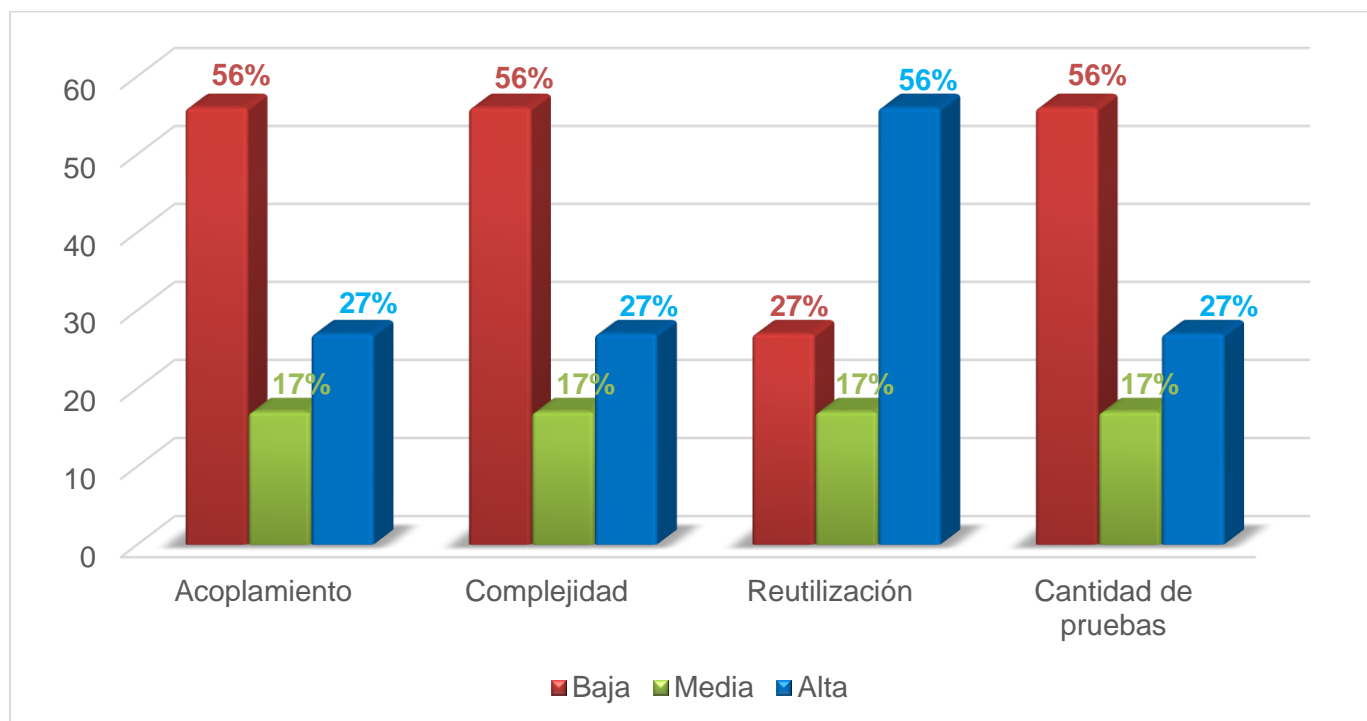


Figura No. 14 Representación gráfica de los resultados de la aplicación de las métricas RC.

**Acoplamiento:** según los resultados que se muestran en la figura anterior un poco más de la mitad (56%) de las clases tienen bajos valores de acoplamiento validando una realización correcta del diseño.

**Complejidad de mantenimiento:** según los resultados que se muestran en la figura se comportan de forma satisfactoria pues cerca del 56% de las clases son de fácil soporte.

**Cantidad de pruebas:** luego de aplicar la métrica se obtuvo que sólo un 27% requiere de un alto esfuerzo a la hora de realizar pruebas de unidad.

**Reutilización:** la métrica generó resultados positivos al informar que el diseño de la solución posee un 56% de clases altamente reutilizables.

Según lo analizado anteriormente los valores de RC se comportan de forma satisfactoria siendo discretos en la mayoría de las clases lo cual implica una disminución del acoplamiento, así como mayor facilidad de mantenimiento de las mismas. La reutilización de las clases es alta y es factible el diseño realizado.

### 3.2. Pruebas funcionales o de aceptación

A continuación se muestran los casos de prueba diseñados a las historias de usuario Gestionar Expediente, Gestionar Trabajo y Generar expediente en formato PDF debido a que las mismas incluyen el conjunto de funcionalidades críticas para el desarrollo exitoso del sistema informático propuesto en la investigación.

Dichos casos de prueba se describirán en las tablas que contendrán los siguientes campos:



**Código:** Identificador de la prueba realizada, a su vez será sugerente al nombre de la prueba a la que hace referencia.

**Historia de usuario:** Nombre de la HU a la que hace referencia la prueba a realizar.

**Nombre:** Nombre de la prueba.

**Descripción de la prueba:** Breve descripción de la prueba.

**Condiciones de ejecución:** Muestra las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.

**La evaluación** de la prueba se hará según el resultado de la misma, la tendrá uno de los tres resultados que a continuación se describen:

**Satisfactoria:** Cuando el resultado de la prueba es exactamente el esperado por el usuario.

**Parcialmente bien:** Cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.

**Insatisfactoria:** Cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la HU.

El siguiente caso de prueba corresponde a la funcionalidad "Gestionar expediente" encargada de recolectar todos los datos y documentos que conforman un expediente:

Caso de prueba de aceptación	
<b>Código:</b> HU10_P1	<b>Historia de usuario :</b> Gestionar expediente
<b>Nombre:</b> Gestionar expediente	
<b>Descripción:</b> Probar que se registren, modifiquen, eliminen y se muestren correctamente los datos de un expediente del sistema	
<b>Conducciones de ejecución:</b> La aplicación debe ser ejecutada con privilegios de usuario, los datos de los expedientes deben guardarse según el proceso inicializado (o sea tipo de expediente) en la base de datos del sistema. Se debe permitir que el expediente sea modificado y eliminado en el sistema utilizando datos válidos. Además deben mostrarse los datos del expediente en formato PDF	
<b>Entrada / Pasos de ejecución:</b> Se intenta adicionar, modificar y eliminar un expediente utilizando datos válidos. Y después se intenta mostrar esos datos.	
<b>Resultado esperado:</b> Los datos del expediente son adicionados correctamente en la base de datos del sistema. El expediente seleccionado es modificado , eliminado y también se pueden mostrar los datos en formato PDF	



Evaluación de pruebas: Prueba satisfactoria

Tabla No. 11 Caso de prueba "Gestionar expediente".

El resto de los casos de pruebas asociados a las restantes historias de usuarios pueden consultarse en el anexo 7.

**Análisis de los resultados:**

Para este tipo de sistemas se realizaron 2 iteraciones para validar que la salida emitida por el sistema informático concordara con el resultado esperado por el cliente. En una primera iteración se diseñaron un total de 22 pruebas funcionales en conjunto cliente-desarrollador para las 22 historias de usuarios que se implementaron en esta investigación de las cuales arrojaron un total de 27 no conformidades en su mayoría errores ortográficos en las interfaces del sistema. Antes de realizar la segunda iteración se realizó la regresión para verificar que los errores de la primera habían sido corregidos. La segunda iteración arrojó 0 no conformidades por lo que la aplicación fue liberada por el grupo de calidad del Centro CEGEL (Centro de Gobierno Electrónico) En la siguiente tabla se muestran los resultados obtenidos durante las pruebas.

Iteraciones	Cantidad de pruebas	Iteración 1	Resultados	Iteracion2	Resultados
Evaluación de las pruebas	22	22 pruebas	27 no conformidades	22	0 no conformidades

Tabla No. 12 Resultados de pruebas de aceptación

**3.3. Pruebas unitarias**

Las pruebas unitarias son aplicadas para verificar que el software cumple los requisitos funcionales y no funcionales de su especificación y también son empleadas para asegurar la calidad del código entregado. Además, son la mejor forma de detectar fallas tempranamente en el desarrollo y está demostrado que mientras más pronto se encuentren los errores, menos costara corregirlos.

Para llevar a cabo las pruebas unitarias al código de la aplicación se hizo uso del framework de prueba PHPUnit el cual permite crear y ejecutar juegos de test unitarios de manera sencilla. PHPUnit utiliza assertions para verificar que el comportamiento de una unidad de código sea el esperado. Seguidamente los resultados de la aplicación de dichas pruebas al bundle de la aplicación.

## APPBUNDLE

Current directory: /var/www/apps/subert/src/AppBundle (dashboard)

Legend: **Low: 0% to 35%** **Medium: 35% to 70%** **High: 70% to 100%**

	Coverage								
	Lines		Functions / Methods		Classes				
Total		88.54%	1368/1545		87.05%	363/417		96.42%	54/56
Controller		90.00%	568/632		81.82%	36/44		100.00%	3/3
DependencyInjection		87.50%	7/8		100.00%	2/2		100.00%	2/2
Entity		88.68%	500/568		86.56%	250/290		96.70%	30/31
Form		87.20%	156/180		90.63%	46/51		94.17%	16/17
Negocio		86.24%	137/159		96.62%	29/30		100.00%	2/2
AppBundle.php		100.00%	1 / 1		100.00%			100.00%	1 / 1

Figura No. 15 Resultados de las pruebas de caja blanca

Después de aplicadas las pruebas a cada una de las funcionalidades y responsables se obtuvo como resultado un 88.54% de las líneas de código ejecutadas, un 87% de las funcionalidades ejecutadas y un 96.42% de las clases ejecutadas. Las pruebas unitarias demostraron que el estado real del software coincide con el esperado, comprobándose los flujos de datos a través de las pruebas realizadas mediante el framework PHPunit, donde se corroboró que la ejecución de las condiciones funcionó satisfactoriamente.

### 3.4. Validación de las variables

La investigación realizada plantea como problema a resolver: "¿Cómo contribuir a una mejor gestión de la información de los procesos que desarrollan las Brigadas Técnicas Juveniles en la UCI?", teniendo como objetivos esperados una herramienta que permita la gestión de la información de los procesos de las Brigadas Técnicas Juveniles (BTJ), así como plantillas y flujos de procesos estandarizados.

Variable Conceptual	Dimensión	Indicadores
Mejorar las deficiencias existentes en la gestión de la información de los procesos, gestión de información de la estructura, gestión de	Entorno del Sistema de Gestión de la información de las BTJ	<ul style="list-style-type: none"> <li>• Disponibilidad de los expedientes para su revisión.</li> <li>• Disponibilidad de los expedientes para su aprobación</li> <li>• Gestión de la estructura de las BTJ</li> <li>• Estandarización de las planillas</li> </ul>



estadísticas y obtención de reportes de las BTJ.		<ul style="list-style-type: none"> <li>• Obtención de reportes</li> <li>• Almacenamiento de la información</li> </ul>
--	--	---

**Tabla No. 13 Variables de la investigación**

A continuación se evalúan cada uno de los indicadores definidos para las distintas dimensiones de la variable de la investigación, antes y después del desarrollo del sistema de gestión de la información de los procesos de las BTJ, para verificar cuanto mejoró la gestión de la información de los procesos, gestión de información de la estructura, gestión de estadísticas y obtención de reportes de las BTJ.

Indicadores	Antes	Después
Disponibilidad de los expedientes para su revisión.	El aspirante al terminar de confeccionar el expediente entregaba este, por vía email o almacenado en un dispositivo extraíble al presidente del Consejo Base en un plazo que no excedía los 60 días desde el inicio de la convocatoria.	El sistema permite una vez conformado el expediente enviarlo automáticamente al presidente del Consejo de Base para su revisión. Si es rechazado automáticamente es enviado al aspirante
Disponibilidad de los expedientes para su aprobación	Se debía esperar porque los consejos de base entregaran los expedientes y así revisar si estos cumplen los parámetros de calidad y confección de las BTJ.	El sistema permite que los expedientes se agrupen automáticamente después de ser revisados en cada consejo base son revisados solo para ver si tienen la calidad requerida para ser aprobado ya que el parámetro de confección el sistema lo valida. Si el expediente no posee la calidad requerida es rechazado y es enviado de vuelta el aspirante.
<b>Gestión de la estructura de las BTJ</b>	Limitada. Toda la estructura de las BTJ era conformada de forma manual, se registraba la información suministrada por las propuestas a miembros de las BTJ y una vez aprobados a cada instancia por la UJC se disponía a conformar el Consejo de Universidad, los Consejos de Base y las Brigadas de Base	El sistema permite la gestión de la estructura de forma centralizada. Cuando los miembros propuestos para integrar las BTJ son aprobados por cada una de las instancias de la UJC el responsable del Consejo de Universidad y de cada uno del Consejo de Base puede asignar los miembros de su consejo. Los presidentes del Consejo de Base pueden conformar además sus Brigadas de Base.

<b>Plantillas estándar</b>	Cada uno de los consejos había conformado una planilla para cada uno de los expediente. Esas planillas eran diferentes en cada uno de los consejos.	El sistema permite la estandarización ya que se utiliza la misma planilla para la confección de cada uno de los expedientes aprobada por el Presidente del Consejo de la Universidad.
<b>Obtención de reportes</b>	Limitada. Se conformaban reportes siempre que se tuviera la información (los documentos no eran heredados de presidente a presidente tanto de los consejos de base como el de universidad)	El sistema permite la consulta de la documentación almacenada y elaborar reportes.
<b>Almacenamiento de información</b>	La información en el mejor de los casos era almacenada en formato duro (papel) o documentos digitales lo que dificulta la consulta de los datos y la obtención de estos.	El sistema presenta una base de datos completa, sencilla y de cómoda gestión que facilita las búsquedas y garantiza la consistencia e integridad de los datos.

**Tabla No. 14 Validación de las variables**

Luego de realizada la evaluación de la variable, analizados los resultados y las características de la solución de la presente investigación se puede demostrar que el empleo del Sistema de gestión de las BTJ (SigestBTJ). Mejora la gestión de la información de los procesos de las BTJ y ofrece nuevas funcionalidades para mejorar el funcionamiento de la organización. Además apoya el correcto accionar de la organización y contribuye a su desarrollo.

### **3.5. Conclusiones parciales**

A lo largo del capítulo se elaboraron y analizaron los elementos fundamentales que forman parte de la verificación y validación del sistema, así como la validación de la investigación, obteniendo los siguientes resultados:

- Se emplearon Técnicas de Validación de requisitos lo que permitió asegurar la validez de los requisitos antes de comenzar el desarrollo del software.
- Se aplicaron métricas para requisitos obteniendo como resultado que los requisitos definidos son estables y por tanto, es confiable el diseño efectuado sobre ellos, además se concluyó que la definición realizada es correcta ya que el 100% de dichos requisitos no poseen ambigüedad.
- Se aplicaron métricas para diseño, en el caso de la métrica RC, los resultados demuestran que las clases del diseño poseen un bajo acoplamiento y tras la aplicación de la métrica TOC se llegó a la conclusión de que existe una baja responsabilidad y complejidad de implementación y que existe



una alta reutilización en el diseño propuesto con un valor del 82% del total. Por lo tanto, se concluye que los resultados obtenidos según estas métricas son positivos.

- Se aplicaron pruebas unitarias utilizando el marco de trabajo para pruebas PHPunit lo que permitió verificar que el software cumple los requisitos funcionales y no funcionales de su especificación.
- La aplicación de las pruebas de aceptación garantizó que el sistema desarrollado funciona correctamente y permitió verificar la aceptación del sistema por parte del cliente.
- Por último, se validaron las variables que forman parte del problema de la investigación, demostrando que con el sistema que se desarrolló, se mejora la gestión de la información que generan los principales procesos de las Brigadas Técnicas Juveniles.





**Conclusiones generales:**

Con la culminación del presente trabajo de diploma se puede concluir que se desarrollaron todas las tareas para cumplir con el objetivo propuesto, destacando que:

- Se realizó el marco teórico conceptual de la investigación, que permitió tener el conocimiento de los problemas y dificultades detectadas en la gestión de la información de las BTJ y mejorar la gestión de toda la información relacionada con la estructura y conformación de la BTJ.
- Las pruebas realizadas arrojaron resultados satisfactorios, demostrando la disminución del tiempo de gestión, revisión y aprobación de los expedientes y se constató la calidad del sistema de gestión (SigestBTJ) utilizando para ello técnicas y métricas aplicables al desarrollo de la solución.
- Se implementó un sistema informático que gestiona la información los procesos de la BTJ, tanto los procesos de participación como los procesos estructurales. Este contribuyo a disminuir las afectaciones existentes en el trabajo de gestión de la información de los procesos así como la organización, la planeación y el control de estos.



**Recomendaciones:**

- Desplegar la herramienta implementada para mejorar, de manera general, el trabajo de gestión de las Brigadas Técnicas Juveniles en todas las universidades del país.
- Implementar un componente de notificaciones (vía Email) que permita comunicar a los aspirantes el estado de su expediente.



## Bibliografía

1. **Pall, Gabriel.** *Quality Process Management.* Mexico : Prentice Hall, 1987.
2. **Riesco, Monserrat González.** *Gestión de la producción. Como planificar y controlar la producción industrial.* Vigo : Ideaspropias, 2006. ISBN/978-84-9839-014-8.
3. **Torres, Luis Cabrera.** *Procedimiento de análisis y mejoramiento de procesos.* s.l. : Aplicación a una corporación comercial, 2003.
4. **Cooper, Robin.** *When Lean Enterprises Coolide: Compeing Through Confrontation.* United State of America : Harvard Business Press, 1995. ISBN/0-87587-571-1.
5. **Uran, J.M.** *Juran y la planificación para la calidad.* Madrid : Diaz de Santos Casa S.A, 1990. ISBN/84-87189-37-7.
6. **Bergholz, Sussana Pepper.** [www.madwave.cl](http://www.madwave.cl). *Gestión por procesos.* [Online] mayo 11, 2011. [Cited: febrero 6, 2015.] <http://www.medwave.cl/link.cgi/Medwave/Series/GES03-A/5032>.
7. **Carrasco, Juan Bravo.** *Gestión de procesos.* Santiago de Chile : Evoluciones S.A, 2008. 956-7604-08-8.
8. **autores, Compilacion de.** Certificado por Instituto de Información Científica y Tecnológica (IDICT). [www.ecured.cu](http://www.ecured.cu). [Online] [Cited: febrero 02, 2015.] [http://www.ecured.cu/index.php/Gestion\\_de\\_la\\_Informacion](http://www.ecured.cu/index.php/Gestion_de_la_Informacion).
9. **Mena, Mayra Mujica.** *La Gestión Documental y Organización de Archivos.* La Habana : Felix Varela, 2005.
10. **S.A, Grupo Archicentro.** <http://www.archicentro.com>. [Online] Grupo Archicentro S.A, 03 12, 2011. [Cited: 5 1, 2015.] <http://www.archicentro.com/El-Mundo-de-la-Oficina-sin-Papel>.
11. **Alfresco.** [www.alfresco.com](http://www.alfresco.com). [Online] Alfresco Software inc, 2015. [Cited: abril 28, 2015.] <http://www.alfresco.com/es/products>.
12. **S.A, desoft.** [www.nubelo.com](http://www.nubelo.com). [Online] 1 4, 2012. [Cited: abril 28, 2015.] [www.nubelo.com/files/.../thumb\\_Ficha\\_Producto\\_AvilaDOC\\_Pro.doc](http://www.nubelo.com/files/.../thumb_Ficha_Producto_AvilaDOC_Pro.doc).
13. **Universidad de las Ciencias Informaticas.** [eXcriba.uci.cu](http://www.eXcriba.uci.cu). [Online] 2013. [Cited: abril 24, 2015.] <http://www.eXcriba.uci.cu>.
14. *Sistema de Gestión de Información para la Educación.* **Anisley Cano Inclán, Irima Campillo Torres, Floriselda Cuesta Rodríguez.** 02, La Habana, Cuba : Instituto de Información Científica y Tecnológica (IDICT), 04 2, 2014, Cinfo, Vol. 45.
15. **Karel Riverón Escobar\*, Yasiel Pompa Vázquez, Victor Gabriel González Cardoso.** <http://biblioteca.uci.cu>. [http://repositorio\\_institucional.uci.cu](http://repositorio_institucional.uci.cu). [Online] 03 22, 2012. [Cited: febrero 9, 2015.] [http://repositorio\\_institucional.uci.cu/jspui/handle/ident/4489](http://repositorio_institucional.uci.cu/jspui/handle/ident/4489). ISBN/ 978-959-286-019-3.



16. **Javier Saname Mena, Javier Alejandro Dominguez Prado.** *Sistema de gestión de información en las secciones sindicales de la Facultad 3.* La Habana Cuba : Universidad de las Ciencias Informáticas, 2014.
17. **Development, OSD-Open System.** <http://www.osdglobal.com>. [Online] OSD, 2015. [Cited: 4 14, 2015.] <http://www.osdglobal.com/faq/desarrollo-software/comparativo-web-vs-escritorio>.
18. **Mora, Sergio Luján.** <http://rua.ua.es>. [En línea] 26 de Febrero de 2006. [Citado el: 5 de 5 de 2015.] <http://rua.ua.es/dspace/bitstream/10045/4412/5/03c-AplicacionesWeb.pdf>.
19. **SensioLabs.** The Book. For Symfony 2.6. <ftp://ftp.prod.uci.cu>. [Online] 01 14, 2015. [Cited: 02 1, 2015.] [ftp://ftp.prod.uci.cu/PHP/Documentacion/Symfony2/2.6/Symfony\\_book\\_2.6.pdf](ftp://ftp.prod.uci.cu/PHP/Documentacion/Symfony2/2.6/Symfony_book_2.6.pdf).
20. **Holscher, Eric, Leifer, Charles y Grace, Bobby.** Doctrine Documentation. *Doctrine-project.org*. [Online] [Cited: Enero 20, 2015.] [http://www.doctrine-project.org/documentation/manual/1\\_0/en/introductiono-models..](http://www.doctrine-project.org/documentation/manual/1_0/en/introductiono-models..)
21. **Stig Sæther Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Zeev Suraski,.** *Manual PHP*. [ed.] Rafael Martínez. 5ta. s.l. : Free Software Foundation, 2001.
22. **Hogan, Brain P.** *HTML5 and CSS5. Develop with Tomorrow's Standards Today*. Raleigh, North Carolina Dallas, Texas : Pragmatic Programmers , LLC, 2011. ISBN 978-1-934356-68-5.
23. **SensioLabs.** *The Twig Book*. s.l. : SensioLab.org, 2015.
24. **Eguiluz, Javier.** *Introduccion a JavaScript*. Barcelona, España : LIBROSWEB, 2011.
25. **Bowen, Rich.** *¿What's New in Apache Web Server 2.2?* EE.UU : O'Reilly Media, Inc, 2007.
26. **Group, The PostgreSQL Global Development.** *PostgreSQL 9.3.8 Documentación*. California : The PostgreSQL Global Development Group, 2015.
27. **Zaragoza, María de Lourdes Santiago.** Desarrollando aplicaciones informáticas. <http://www.utvm.edu.mx>. [Online] UTVM, 2015. [Cited: 5 15, 2015.] <http://www.utvm.edu.mx/OrganoInformativo/orgJul07/RUP.htm>.
28. **Rojas, Juan Carlos Olivares.** [Online] 2011. [Cited: 2 1, 2015.] <http://antares.itmorelia.edu.mx/~jcolivares/courses/am071q/unidad7.pdf>.
29. **Scot, Doug Rosenberg Kendall.** *ICONIX Desarrollo de Soluciones en Software Libre*. 1993.
30. **Suarez, Diego A Rojas.** <http://www.academia.edu>. [Online] agosto 20, 2013. [Cited: 2 10, 2015.] <http://www.academia.edu/4324067/Metodolog%C3%ADas>.
31. **Perez, Maria J Perez.** *Guía comparativa de Metodologías Ágiles*. Valladolid (España) : Universidad de Valladolid.



32. **VP-UML\_Users\_Guide.** [Online] Visual Paradigm International, 2014. [Cited: febrero 10, 2015.] [ftp://10.0.0.22/documentacion/Ingenieria%20Software/Visual%20Paradigm/VP-UML\\_Users\\_Guide.pdf](ftp://10.0.0.22/documentacion/Ingenieria%20Software/Visual%20Paradigm/VP-UML_Users_Guide.pdf).
33. **Stephen A. White, Derek Meirs.** *Guía de referencia y modelado BPMN.* Lighthouse Point, Florida, USA : Future Strategies Inc., 2009. ISBN13: 978-0-9819870-3-3.
34. **Gembeta: dev Desarrollo y software.** [Online] [Cited: enero 25, 2015.] <http://www.genbetadev.com/herramientas/netbeans-1>.
35. **Sommerville, Ian.** *Ingeniería de Software. Séptima edición.* 2005.
36. **Muñoz, Coral Calero.** *Métricas del software: conceptos básicos, definición y formalización.* 2006.
37. **Fornaris, Maité Sánchez. Rabí, Dayanis Alcántara.** *Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica.* 2010.
38. **Pressman, Roger S.** *Ingeniería del Software : un enfoque práctico 7ma Edicion.* La Habana : Felix Varela, 2008. Sin ISBN .
39. **Noble Martin, Angela y Biddle James, Robert.** *The XP Customer Role in Practice: Three Studies Agile.* 2004.
40. **Fabian Potencier, Ryan Weaver, Javier Eguiluz.** *Buenas practicas para aplicaciones Symfony . s.l. : LibrosWeb, 2014.*
41. **Estimación Empírica de Proyectos con Planning Poker. Hector Cuesta Arvizu, Sergio Ruiz Castilla.** DF, Mexico : s.n., 2013, SG, Vol. 39.
42. **Beyris Souлары, Liliam Celia, et al.** *Proceso de medición y análisis para el polo de hardware y automática.* 2010.
43. **GuilleSQL. GuilleSQL .** *Un portal sobre Microsoft SQL Server en castellano.* [Online] 11 18, 2008. [Cited: 5 1, 2015.] [http://www.guillesql.es/Articulos/Claves\\_Subrogadas\\_Slowly\\_Changing\\_Dimension\\_SCD\\_Tipo\\_2.aspx](http://www.guillesql.es/Articulos/Claves_Subrogadas_Slowly_Changing_Dimension_SCD_Tipo_2.aspx).
44. **Larman, Craig.** *UML y Patrones. Introduccion al analisis y diseño orientado a objetos.* Mexico : Prentice Hall, 1999. ISBN 970-17-0261-1.
45. **Schmidt, Douglas.** *PATTERN ORIENTED SOFTWARE ARCHITECTURE VOLUME 2: Patterns for Concurrent and Networked Objects.* Chichester , England : Wiley and Sons, 2000. ISBN-13: 978-0471606956.
46. (UNICEN), Facultad de Ciencias Exactas. Sitio web de la Facultad de Ciencias Exactas (UNICEN). [Online] 2003. [Cited: febrero 1, 2015.] [http://www.exa.unicen.edu.a/catedras/modemp/03\\_Introduccion\\_Integracion\\_Modelado.pdf](http://www.exa.unicen.edu.a/catedras/modemp/03_Introduccion_Integracion_Modelado.pdf).



47. Adam, Kelli y Stanley, Lisa. *Internet Information Services Administration*. s.l. : Sams Publishing, 2000.



**Glosario de términos:**

**Microsoft Office:** es una suite ofimática que abarca el mercado completo en Internet e interrelaciona aplicaciones de escritorio, servidores y servicios para los sistemas operativos Microsoft Windows y Mac OS X.

**Subversion** (abreviado frecuentemente como **SVN**, por el comando *svn*): es una herramienta de control de versiones de código abierto basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros.

**UnionFS:** es un servicio para sistemas de archivos de Linux que permite montar un sistema de archivos formado por la unión de otros sistemas de archivos de Linux.

**Potencial de uso:** Valor dado por el cálculo de la mediana. Representa el nivel obtenido según la cantidad de indicadores que posee.

**Licencia BSD:** Berkeley Software Distribution (en español, distribución de software berkeley) es un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

**Bundles:** Conjunto de archivos que implementan una única funcionalidad

**Sucesión de Fibonacci:** En matemáticas, la sucesión de Fibonacci es la sucesión infinita de números naturales: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144... La sucesión comienza con los números 0 y 1, y a partir de estos, cada elemento es la suma de los dos anteriores.

**Smalltalk-80:** Lenguaje de programación reflexivo orientado a objeto, de tipo dinámico.

**DHTML:** Se denomina HTML dinámico (DHTML) a la combinación de Javascript y definición de formato de página en CSS (Hojas de Estilo en cascada), para conseguir efectos dinámicos en las páginas web

**XML:** (Extensible Markup Language) Lenguaje de Marcado Extensible.

**XSLT:** (Extensible Stylesheet Language Transformations) Lenguaje extensible de transformación de hojas de estilos.

**JSON:** (JavaScript Object Notation) Notación de Objetos JavaScript.

**DOM:** (Document Object Model) Modelo de Objetos del Documento.

**ECMAScript:** Define un lenguaje de tipos dinámicos ligeramente inspirado en Java y otros lenguajes del estilo de C. Soporta algunas características de la programación orientada a objetos mediante objetos basados en prototipos y pseudoclases.