



**Facultad 3**

*dataFEM*

Sistema para la gestión de los procesos de la Federación Estudiantil  
Universitaria en la Universidad de las Ciencias Informáticas.

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

**Autores:**

Victor Gabriel González Cardoso

Eiler Figueredo Bustamante

**Tutores:**

Ing. Alelí Sánchez Méndez

Ing. Yidian Yosbel Castellanos Sabarí

Ing. Yoan Carlos Machado Espinosa

*Junio 2014*

*“Año 56 de la Revolución”*

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 5 días del mes de junio del año 2014.

---

Victor Gabriel González Cardoso

---

Eiler Figueredo Bustamante

---

Ing. Alelí Sánchez Méndez

---

Ing. Yidian Yosbel Castellanos Sabarit

---

Ing. Yoan Carlos Machado Espinosa.

## AGRADECIMIENTOS

Son muchas a las personas que debo agradecerle haber podido llegar a este punto, lo cual constituye un gran hito en mi vida. A todos, mi agradecimiento infinito.

A la Revolución por todas las oportunidades dadas y por permitirme formarme como ingeniero sin pagar un centavo. A la FEU, que para mí es la segunda escuela, por darme permanentemente la posibilidad de ser una persona de bien y de ayudar a los demás. Por enseñarme a levantarme cuando todo estaba oscuro y no había más salida, por darme las fuerzas para continuar y por darme algo en qué creer.

A mis padres (Idolidia y Victor) y hermana (Nelly), por darme todas las fuerzas en todo momento, por el apoyo incondicional, por las noches de preocupación y desvelo que causé. Gracias por estar ahí cuando más se necesitaba.

A mi hermano Yunior Feria, el chino, el hermano de sangre; por aguantarme con mucha paciencia todas mis pesadeces. Por estar siempre en el momento justo y preciso, y por decirme siempre la verdad con una crudeza incalculable. Gracias por todo.

A Jorge Hidalgo por sacarme de muchos aprietos y por todo lo que me enseñó en cuestiones de “dar cuero a la gente”. A los integrantes de la “Familia real”. A Joyce, Yusdel, Luisma (otro de mis grandes amigos) y Yoandry; a ellos gracias por haberse convertido en una familia para mí. A Yidian, gracias por todas las enseñanzas, el que siempre me guio en los temas FEU. Mi agradecimiento por todo tu aporte a mi persona.

A todos los piquetes de la FEU que tuve oportunidad de dirigir y formar parte, a esos que me acompañaron en todo momento y se pusieron duro cuando los momentos lo exigían: a Nolvis, Karel, Polanco, Carmen, Anel, Addiel, Niurka (gracias por tu ayuda en esta última etapa, te creciste de verdad, rompiste las expectativas), José Augusto, Livan, Lester, José Joel, todo el Consejo de la FEU, Nani, Jose Daniel, Pedro, Carlo Carlo y otros muchos cuya lista sería interminable.

Al Consejo de Dirección de la universidad que siempre me trató como un hijo, por la confianza y el cariño que depositaron en mí, por todas las enseñanzas que se convirtieron en consejo para la vida. Gracias por formarme.

A esos profesores que me marcaron de forma especial, por sus enseñanzas y por su cariño: a Dariela, Sariannys y Rainer.

A todas esas personas que conocí algunas por azares de la vida y otras por las responsabilidades que ocupaba. A Melo, Adelina, Maritza, Carmen, Pastora, Dalila, Rosita y otros muchos que no recuerdo.

A mi grupo, por ser tan diverso y especial. A Yaimari y a Leanet por su preocupación permanente y por defenderme siempre a capa y espada contra quien fuera.

A mi compañero de tesis Eiler por los días de intenso trabajo y por el trabajo en la construcción de la aplicación. Eres un genio del Ext, aprovéchalo bien. A nuestros tutores Alelí y Yoan Carlos.

**Victor.**

En primer lugar a mi tía Beley, que no pudo ver este sueño hecho realidad.

Agradezco a mi familia por ser constante veladora de mis largos años de estudios y estar ahí para mí siempre.

En especial a mi madre, que ha sabido jugar los roles de padre, hermana y amiga incondicional en todos estos años, por todo su amor y entrega, gracias mima.

A mi “hermano primo” Iván por todo ese amor que me da.

A mi “chukulita” y futura esposa, Anabí, por ese amor que me brinda y por ser la inspiración de ser una mejor persona cada día.

A mis queridos abuelos Rosa, Nelson y Walquiria, por saber ser mis padres de la forma en que lo han hecho en todos estos años.

A mi tía Betty, gracias por toda la ayuda que me ha brindado.

A todos mis amigos y amigas, en especial a Eddy, Damarys, Dian, Indra, Carlos Llama, Yunior Feria por permitirme tomar ese preciado tesoro que poseemos los mortales, el tiempo.

A mis hermanos postizos Jethell y Victor.

A mi compañero de tesis por compartir esta aventura y estar ahí en las buenas y malas.

A mi gente del grupo por haberme soportado durante estos cinco años.

A todos los que de una forma u otra tuvieron un papel decisivo en mi educación y formación.

A mis tutores.

A todos aquellos que me ayudaron a realizar este sueño.

**Eiler.**

## DEDICATORIA

A la FEU, mi segunda universidad. La organización que me formó y me dio la posibilidad de conocer a mucha gente especial. A todos los estudiantes de la UCI por darme la posibilidad de representarlos y hacerme sentir útil en cada situación que compartí con ellos.

**Victor**

A mi familia, especialmente a mi madre Yaumara, mi novia Anabí, mi hermano Iván, mis abuelos Rosa, Nelson y Walquiria.

A mis amigos de aquí y de allá, especialmente a Eddy, a Dian, a Damarys y a Indra.

**Eiler.**

## RESUMEN

La Federación Estudiantil Universitaria en la Universidad de las Ciencias Informáticas lleva a cabo un importante número de procesos que requieren reducir el tiempo de procesamiento de la información que se genera, estandarizar la documentación que aportan las diferentes áreas en relación con la organización y elevar la persistencia de los datos.

En el presente informe se muestran los resultados de la implementación de la herramienta dataFEU. Esta es una aplicación informática que asegura la calidad del proceso de evaluación estudiantil y apoya a sus principales dirigentes en el proceso de toma de decisiones. Se centra principalmente en el proceso de integralidad y caracterización de los estudiantes registrando mediante evidencias la trayectoria de cada uno de ellos en un expediente digital.

DataFEU fue desarrollado sobre los marcos de trabajo Symfony y Ext JS empleando MySQL como servidor de base de datos, ajustándose a la soberanía tecnológica por la cual aboga Cuba. Cuenta con cinco subsistemas que se integran entre sí y permiten el correcto funcionamiento, integridad y persistencia de los datos que se introducen y procesan. Brinda información a sistemas externos mediante servicios web que garantizan la interoperabilidad del mismo.

Como parte de la investigación se valida la herramienta a partir de diferentes pruebas al sistema y se muestra el resultado de una encuesta para medir la satisfacción de los usuarios que interactúan con el software.

## ÍNDICE:

<b>INTRODUCCIÓN</b> .....	10
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	15
1.1 INTRODUCCIÓN.....	15
1.2 PROCESOS QUE SE LLEVAN A CABO DENTRO DE LA FEU.....	15
1.3 PRINCIPALES CONCEPTOS .....	16
1.3.1 GESTIÓN DE PROCESOS.....	16
1.3.2 SOFTWARE DE GESTIÓN .....	17
1.3.3 APLICACIÓN WEB .....	17
1.3.4 SOFTWARE LIBRE.....	18
1.4 PROCESO DE DESARROLLO DE SOFTWARE .....	19
1.5 ARQUITECTURA BASADA EN COMPONENTES .....	20
1.6 TECNOLOGÍAS .....	20
1.7 HERRAMIENTAS .....	22
1.8 ANÁLISIS DE OTRAS SOLUCIONES .....	24
1.8.1 KAINOS.....	24
1.8.2 SISTEMAS PARA LA GESTIÓN DE EVENTOS.....	24
1.8.3 SISTEMA PARA LA GESTIÓN DE INQUIETUDES .....	25
1.8.4 SISTEMAS PARA LA GESTIÓN DEL FUNCIONAMIENTO .....	25
1.8.5 GESTIÓN UNIVERSITARIA .....	26
1.8.6 CONCLUSIONES DE LAS SOLUCIONES ESTUDIADAS.....	27
1.9 CONCLUSIONES.....	27
<b>CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA</b> .....	28
2.1 INTRODUCCIÓN.....	28
2.2 DESCRIPCIÓN DE LOS PROCESOS DE NEGOCIOS. ....	28
2.2.1 PROCESO DE INTEGRALIDAD.....	28
2.2.2 PROCESO DE CARACTERIZACIÓN .....	30
2.2.3 GESTIÓN DE EVENTOS Y ACTIVIDADES.....	31
2.2.4 GESTIÓN DE LA MEMBRESÍA .....	31
2.2.5 FUNCIONAMIENTO INTERNO .....	32
2.3 DEFINICIÓN DE LOS REQUISITOS FUNCIONALES Y NO FUNCIONALES	33
2.3.1 TÉCNICAS DE OBTENCIÓN DE REQUISITOS.....	33
2.3.2 REQUISITOS FUNCIONALES.....	34
2.3.3 REQUISITOS NO FUNCIONALES.....	42
2.4 VALIDACIÓN DE REQUISITOS .....	44

2.5	DESCRIPCIÓN DE REQUISITOS.....	45
2.6	DEFINICIÓN DE LA ARQUITECTURA.....	47
2.6.1	SUBSISTEMAS QUE SE MODELAN .....	47
2.6.2	ESTILO ARQUITECTÓNICO .....	49
2.6.3	PATRONES DE DISEÑO .....	49
2.6.4	MODELO DE DATOS .....	51
2.7	OBTENCIÓN DEL RANKING DE INTEGRALIDAD.....	52
2.8	CONCLUSIONES.....	53
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA.....</b>		<b>54</b>
3.1	INTRODUCCIÓN .....	54
3.2	ESTÁNDARES DE CÓDIGO .....	54
3.2.1	IDENTACIÓN, LLAVES DE APERTURA Y CIERRE, Y TAMAÑO DE LAS LÍNEAS.....	54
3.2.2	CONVENCIÓN DE NOMENCLATURA.....	54
3.2.3	ESTRUCTURAS DE CONTROL.....	55
3.2.4	DOCUMENTACIÓN .....	55
3.3	INTEGRACIÓN CON SISTEMAS EXTERNOS .....	55
3.4	SEGURIDAD Y TRAZAS.....	56
3.5	SERVICIOS WEB .....	56
3.6	PRUEBAS .....	57
3.6.1	NIVELES DE PRUEBAS .....	57
3.6.2	TIPOS DE PRUEBAS .....	58
3.6.3	DISEÑO DE LOS CASOS DE PRUEBAS .....	61
3.6.4	IMPLEMENTACIÓN DE LAS PRUEBAS .....	61
3.7	MODELO DE DESPLIEGUE.....	63
3.8	IMPLANTACIÓN DEL SISTEMA.....	63
3.9	VALIDACIÓN DE LAS VARIABLES DEL PROBLEMA DE LA INVESTIGACIÓN .....	66
3.10	ENCUESTA PARA MEDIR NIVEL DE SATISFACCIÓN DE LOS USUARIOS.....	66
3.11	CONCLUSIONES.....	67
<b>CONCLUSIONES GENERALES.....</b>		<b>68</b>
<b>RECOMENDACIONES .....</b>		<b>69</b>
<b>BIBLIOGRAFÍA REFERENCIADA .....</b>		<b>70</b>
<b>ANEXOS .....</b>		<b>73</b>
	ANEXO 1: Encuesta realizada en mayo-junio del 2013 sobre el sistema dataFEU..	73
	ANEXO 2: Resultados de la encuesta (Anexo 1).....	75



ANEXO 3: Código de Nueva instancia actividad.....	82
ANEXO 4: Modelo de datos.....	83
ANEXO 5: Parámetros para el cálculo del ranking de integralidad para el proceso de evaluación estudiantil de primero a cuarto año.....	84
ANEXO 6. Parámetros para el cálculo del ranking de integralidad para el proceso de caracterización.....	86
ANEXO 7: Criterios de complejidad .....	88

## INTRODUCCIÓN

En la actualidad es muy extendido el uso de las Tecnologías de la Información y las Comunicaciones (TICs) por empresas y organizaciones que desean tener éxito en entornos cada vez más globalizados y competitivos. Esto se ha convertido en un factor a tener en cuenta para la gestión eficiente y eficaz de los recursos materiales y humanos. Cuba no está ajena a este fenómeno, el cual se definió como prioridad en los lineamientos de la política económica y social del 6to Congreso del Partido Comunista (1).

Entre las organizaciones que han utilizado las TICs se encuentra la Federación Estudiantil Universitaria (FEU), organización de masas que agrupa a todos los estudiantes universitarios. Está compuesta por una estructura en diferentes niveles: desde la brigada (célula fundamental) hasta el Consejo Nacional (órgano rector que sesiona entre cada congreso), pasando por las estructuras de facultad y universidad. Cada estructura presenta un equipo de dirección que integra los grupos de trabajo y esferas que atienden el funcionamiento de la organización, todos encabezados por el presidente.

Las exigencias actuales obligan a asegurar la calidad de sus principales procesos, por lo que se hace necesario introducir dentro de su sistema de trabajo herramientas informáticas que faciliten la gestión y el procesamiento de los volúmenes de información que se generan.

Dentro de los procesos llevados a cabo por la FEU son válidos resaltar:

- Gestión y planificación de eventos y actividades.
- El funcionamiento interno incluyendo la gestión de la emulación.
- Gestión de los movimientos estudiantiles, haciéndose énfasis en la atención de los alumnos ayudantes desde su ingreso al movimiento hasta su evaluación por el departamento.
- Gestión de las sanciones aplicadas.
- Control de la trayectoria de los dirigentes estudiantiles.
- Gestión de reconocimientos, distinciones, premios y bonificaciones.
- Atención de los estudiantes en los proyectos de desarrollo.

Estas tareas implican evaluar y atender a los estudiantes en las diferentes esferas de la vida universitaria (docencia, investigación, producción y extensión), lo que se torna complejo a raíz del dinamismo implícito en la universidad. Debido al volumen de

información, las estadísticas que se obtienen son susceptibles a errores, pues la documentación que entrega cada área en relación con la organización no sigue ningún tipo de estandarización.

En la Universidad de las Ciencias Informáticas (UCI) todos estos procesos consumen tiempo y recursos, ya que existen 500 estudiantes como promedio por facultad. Finalizado un año académico se lleva a cabo el proceso de evaluación estudiantil, denominado Proceso de integralidad para los estudiantes de primero a cuarto año y Proceso de caracterización para los estudiantes de quinto. Con el mismo se persigue fomentar el desarrollo de las capacidades, actitudes, valores y potencialidades del futuro profesional. Esto se logra durante todo el período de formación en el pregrado a partir del control y evaluación de los estudiantes en el ejercicio de sus actividades.

En cada curso escolar el estudiante debe entregar una autoevaluación del período señalado que es tomada como base por la brigada para otorgarle una evaluación. La forma de realizar este proceso trae consigo que la veracidad de esa información se vea comprometida, reflejando el poco control de la trayectoria estudiantil ya que las actividades no son registradas en el momento en que ocurren. Según los registros de estos procesos en años anteriores, obtenidos de las actas de los Consejos de la FEU de la UCI, una cifra cercana al 40% de los estudiantes no entrega su caracterización o lo hace con mala calidad. Lo anterior obliga a los presidentes de brigadas a recopilar todas las evidencias necesarias para llevar a cabo esta evaluación, ralentizando el envío de la información a los organismos superiores y el cierre del proceso.

Al no existir una estandarización en la documentación a almacenar, cada estudiante decide en qué formato enviar su datos: PDF, Microsoft Office u Open Office/Libre Office, lo que torna engorroso los pasos de unificar y procesar la información. De manera similar sucede con las evaluaciones de los dirigentes, los alumnos ayudantes, los avales de méritos y distinciones otorgadas; aspectos que son necesarios guardar debidamente para lograr un mejor funcionamiento del sistema de estimulación que implementa la universidad. Tampoco se cuenta con un archivo histórico para la consulta de la información de años anteriores.

Lo anterior evidencia la poca fiabilidad en la información que se recolecta, comprometiéndose la integridad, la calidad y el grado de certeza de la misma.

Tomando en cuenta el contexto anterior cabría plantearse como **problema de la investigación**: ¿cómo asegurar la calidad del proceso de evaluación estudiantil de la Federación Estudiantil Universitaria en la Universidad de las Ciencias Informáticas de forma tal que permita reducir el tiempo de procesamiento de la información,

estandarizar la documentación y elevar la persistencia de los datos?, con el siguiente **objetivo general**: desarrollar una aplicación informática que reduzca el tiempo de procesamiento de la información, estandarice la documentación y eleve la persistencia de los datos del proceso de evaluación estudiantil de la Federación Estudiantil Universitaria en la Universidad de las Ciencias Informáticas.

Se define como **objeto de estudio**: los sistemas informáticos para la gestión de los procesos de la Federación Estudiantil Universitaria. El **campo de acción** son los sistemas informáticos para la gestión de los procesos de la Federación Estudiantil Universitaria en la Universidad de las Ciencias Informáticas. La validación de la investigación está sustentada en la siguiente **idea a defender**: si se desarrolla una aplicación que reduzca el tiempo de procesamiento de la información, estandarice la documentación y eleve la persistencia de los datos, entonces se asegura la calidad del proceso de evaluación estudiantil de la Federación Estudiantil Universitaria de la Universidad de las Ciencias Informáticas.

La calidad es un objetivo decisivo en cualquier actividad dentro de las organizaciones. Para conseguirla es necesario conocer lo que la palabra significa en su sentido más amplio y no solamente referido a un producto o servicio al que se aplica.

Existen múltiples definiciones de la calidad, las cuales provienen de campos diferentes pero según el Diccionario de la Real Academia de la Lengua Española (RAE) define el concepto como: *“Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor”*.

Haciendo énfasis en lo anterior pueden definirse varias variables en el negocio para determinar la calidad en el proceso de evaluación estudiantil:

- **Tiempo**: se reduce el procesamiento de la información que conduce proporcionalmente a acortar el tiempo de desarrollo del proceso de evaluación. Se mide en semanas.
- **Estandarización de la documentación que se aporta**: como todas las áreas que tienen relación con la organización utilizan una única plataforma, la información resultante sigue un único patrón o estándar, lo que permite el fácil acceso a los documentos generados. Como unidad de medida se tendrá el porcentaje de documentos que se eliminan en el intercambio de información.
- **Persistencia de la información**: se centralizan las evidencias en un archivo histórico, elemento que permite al concluir la carrera poder acceder a las evaluaciones que se guardaron durante la vida estudiantil. La información puede ser consultada a través de los cursos y solo los usuarios con los

permisos adecuados pueden cambiar o certificar la misma. Se mide en el porcentaje de expedientes con la evaluación estudiantil que se logran archivar.

Para dar cumplimiento al objetivo anteriormente mencionado, se concibieron las siguientes tareas de investigación:

- Realización de un estudio del estado del arte de sistemas informáticos para la gestión de procesos de la Federación Estudiantil Universitaria.
- Realización de un estudio de las herramientas y lenguajes de programación que se utilizarán para el desarrollo de software.
- Descripción de los procesos de negocios.
- Descripción de los requisitos funcionales necesarios para dar solución al problema planteado.
- Implementación de una aplicación informática que sea capaz de gestionar el proceso de evaluación estudiantil de la Federación Estudiantil Universitaria.
- Realización de pruebas para corregir los errores de la etapa de implementación.

Con la realización de este trabajo se espera la creación de una solución informática que asegure la calidad del proceso de evaluación estudiantil de la organización en la universidad a partir de la reducción del tiempo de procesamiento de la información, la estandarización de la documentación y el aumento de la persistencia de los datos con la creación de un archivo histórico.

La estructura del trabajo de diploma, siguiendo un orden lógico ha sido dividida en tres capítulos:

El **primer capítulo: “Fundamentación teórica”**, consiste en una fundamentación teórica del resto de la investigación y un estudio del estado del arte de los sistemas informáticos para la gestión de los procesos de la Federación Estudiantil Universitaria. Se identifican los procesos de llevados a cabo por la organización. Se selecciona la metodología, las herramientas y las tecnologías a utilizar a partir de una revisión de las existentes en el mundo.

El **segundo capítulo: “Descripción de la solución propuesta”**, abarca la descripción de los procesos de negocios, la definición de los requisitos funcionales y no funcionales, el diseño y arquitectura de la aplicación a desarrollar. Se describen los patrones de diseño y el estilo arquitectónico empleado.

En el **tercer capítulo: “Implementación y validación del sistema”**, se analizan los resultados de la implementación del sistema propuesto, además de verificarse los

requisitos funcionales a través de pruebas y donde se evalúa la calidad del producto desarrollado. Se efectúa además un estudio de satisfacción a los clientes finales que consumirán la aplicación web.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

## 1.1 INTRODUCCIÓN

En el presente capítulo se identifican los procesos que lleva la FEU, además de una visión de los principales conceptos y elementos teóricos para el desarrollo de la aplicación. Se analizan las tecnologías, métodos, técnicas y herramientas necesarias para la implementación del software. Se realiza un estudio del estado del arte de los sistemas informáticos para la gestión de los procesos de la Federación Estudiantil Universitaria, identificando tendencias y requisitos funcionales.

## 1.2 PROCESOS QUE SE LLEVAN A CABO DENTRO DE LA FEU.

El secretariado de la FEU a cada nivel es el encargado de ejecutar y llevar a cabo todos los procesos de la organización para mantener un correcto funcionamiento. Dentro de los cuales se encuentran (3):

1. Proceso de integralidad: se realiza al concluir cada curso académico entre el primer y cuarto año de la carrera. Pretende evaluar a todos los estudiantes según su desempeño durante el período señalado. El resultado final es una evaluación cualitativa de bien, regular o mal, a partir del cumplimiento de un grupo de indicadores que se discuten y validan en cada brigada a inicios del semestre.
2. Proceso de caracterización: se realiza al concluir el último curso de la carrera y constituye una evaluación final del desempeño del estudiante durante los años de estudio, a partir de la recopilación de evidencias y las evaluaciones del proceso de integralidad. Las evaluaciones generadas tributan directamente al proceso de ubicación laboral.
3. Gestión de eventos y actividades: se encarga de controlar, atender y avisar sobre las actividades a desarrollarse en cualquier esfera de la vida universitaria. Cada actividad o evento se relaciona con una esfera de trabajo. Los eventos se clasifican de acuerdo a su magnitud.
4. Gestión de la membresía: controla, analiza y convoca a qué grupos académicos, científicos o extensionistas pertenece cada estudiante. Vela por el estado de sus miembros. Existen grupos cuya entrada es voluntaria y otros donde debe aprobarse como miembro a través de un proceso de selección.
5. Funcionamiento interno: evalúa las tareas que desempeña la organización a través de la gestión de la emulación y el control de las actas y documentos

necesarios para su correcto funcionamiento. En este proceso se gestionan los premios que son otorgados por la organización a los trabajadores y profesores.

### 1.3 PRINCIPALES CONCEPTOS

#### 1.3.1 GESTIÓN DE PROCESOS

Los tiempos modernos han hecho que las organizaciones se encuentren inmersas en entornos y mercados competitivos y globalizados; en los que todas las que deseen tener éxito o, al menos, subsistir tiene la necesidad de alcanzar buenos resultados empresariales.

Para alcanzar estos buenos resultados, las organizaciones necesitan gestionar sus actividades y recursos con la finalidad de orientarlos hacia el éxito de los mismos. Esto a su vez se ha derivado en la necesidad de adoptar herramientas y metodologías que permitan cumplir las metas y objetivos; lo que exige prestar atención a los procesos transversales de la organización tales como: la planeación, el desarrollo del producto, las compras y otros.

A causa de una prolongada falta de atención por parte de la alta dirección, muchos de estos procesos se hacen obsoletos, complicados, costosos, redundantes, mal definidos y no se adaptan a las demandas de un entorno en continuo cambio. En tales procesos señala Juran en (3) que: *“la calidad de sus resultados quedan muy lejos de la requerida para ser competitivos”*.

Una de las enunciaciones de procesos, es la dada en la norma ISO 9000:2000 que lo define como el conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados.

Las organizaciones que cuentan con estructuras organizativas rígidas ejecutan sus actividades de manera fraccionada, dificultando el flujo de la comunicación y haciendo que primen los intereses de las áreas o funciones de una forma aislada sobre los de la propia organización. Para dar solución a esta dificultad surge la adopción de un enfoque de procesos o gestión por procesos. Esta busca que las empresas tengan estructuras con una mayor capacidad de adaptabilidad al entorno cambiante, mayor flexibilidad, más capacidad para aprender, de crear valor y con una mayor orientación hacia el logro de los objetivos. La gestión por procesos es una forma de conducir o administrar una organización, concentrándose en el valor agregado para el cliente y las partes interesadas.



La gestión de procesos busca reducir la variabilidad innecesaria que aparece habitualmente cuando se producen o prestan determinados servicios y trata de eliminar las ineficiencias asociadas a la repetitividad de las acciones o actividades, al consumo inapropiado de recursos, entre otros (4).

### 1.3.2 SOFTWARE DE GESTIÓN

Algunos investigadores creen que el término software fue utilizado por primera vez por John W. Tukey en 1958 cuando publicó un artículo en la revista *“American Mathematical Monthly”* haciendo referencias a los software de computación (2). Una de las definiciones más formales es la dada por los Estándares del Instituto de Ingenieros Eléctricos y Electrónicos IEEE2 (por sus siglas en inglés) en el estándar 729-1983: *“Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación”* (1).

La gestión como proceso, en el amplio mundo de la informática se entiende por el subsistema encargado del control y seguimiento de los repositorios de información, de los grupos de usuarios, y de los procesos de soporte para otros subsistemas. El software de gestión se encarga de definir y controlar los flujos de trabajo que son utilizados por los otros subsistemas, y de la definición de parámetros para el funcionamiento del sistema (7). El procesamiento de información de gestión constituye, casi desde los inicios de la informática la mayor de las áreas de aplicación de los ordenadores. Estos programas utilizan grandes cantidades de información almacenadas en bases de datos con el objetivo de facilitar las transacciones comerciales o la toma de decisiones.

El software de gestión es utilizado por las organizaciones que requieren aumentar la competitividad a partir de una mejor organización de sus recursos tanto humanos como materiales. Las grandes empresas sobre todo aquellas que fomentan aumentar su mercado rápidamente, tienen que optar por implantar un software de gestión debido a que deben enfrentar cada vez con más rapidez y eficiencia los cambios en el entorno externo e interno, lo que demanda una gestión eficiente y a la medida de sus principales procesos.

### 1.3.3 APLICACIÓN WEB

A decir de Conallen (1999) (3) *“Las aplicaciones web son sistemas informáticos que los usuarios utilizan accediendo a un servidor web a través de Internet o de una Intranet”*. Existe gran diferencia entre este tipo de aplicaciones y los sitios web, aun

cuando estos sean dinámicos. También Conallen (2002) (4), en otra de sus publicaciones, apunta que cuando la navegación de un usuario a través de los recursos colocados en la web tienen implicaciones para los negocios respectivos, se dice que estamos en presencia de una aplicación web. O sea que básicamente las diferencias entre una aplicación web y un sitio web, radican en su utilización, ya que las primeras implementan “lógica de negocios” y su uso cambia el estado del mismo. Esto es importante porque define el enfoque del esfuerzo a la hora de modelarlas.

El desarrollo de aplicaciones web tiene inherente una complejidad mayor, ya que requieren implementar una arquitectura que se adapte a los cambios constantes, que facilite su ágil integración con otros sistemas y que resuelva variables picos de interacción con un buen rendimiento.

En la actualidad debido al auge de este tipo de aplicaciones, han surgido diferentes categorías en dependencia del tipo de negocio que modelan como son las informacionales, interactivas, transaccionales, comunidades, portales entre otros. A pesar de estas clasificaciones, existen aplicaciones web que constituyen un híbrido de ellas, pues implementan en su lógica elementos de más de una.

#### 1.3.4 SOFTWARE LIBRE

El software libre, es un movimiento tecnológico que ha revolucionado la sociedad. Presenta características especiales que han permitido la experimentación de nuevas formas de desarrollo y mantenimiento de programas, nuevos modelos económicos, y nuevas normas legales. Es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en “libre” como en “libertad de expresión”.

Según la Free Software Foundation (FSF) (9), Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios: usar el programa, con cualquier propósito (libertad 0); estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1); distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2); mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie (libertad 3). Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, se debe tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el software libre.

Lo anterior da la medida de la viabilidad económica de este sistema, que por ser libre no es necesariamente gratuito, sino que nos da la posibilidad de comercializarlo,

prestarlo con total libertad y protegerlo legalmente. A pesar de las lagunas técnicas que pueda tener aún el software libre, es la alternativa para los países subdesarrollados.

#### 1.4 PROCESO DE DESARROLLO DE SOFTWARE

Pressman (11) apunta que: “un proceso en el contexto de la ingeniería de software no es una descripción rígida de cómo desarrollar software, sino que es un enfoque adaptable que permite al equipo del proyecto escoger las acciones y tareas apropiadas para realizar el trabajo”. A partir de lo anterior se utiliza la metodología: “Desarrollo Ágil con Calidad” (DAC), la cual es definida por la Dirección de Informatización de la UCI (DIN) para el desarrollo de aplicaciones para la universidad. La imagen 1 muestra cómo es la relación entre sus áreas de procesos.

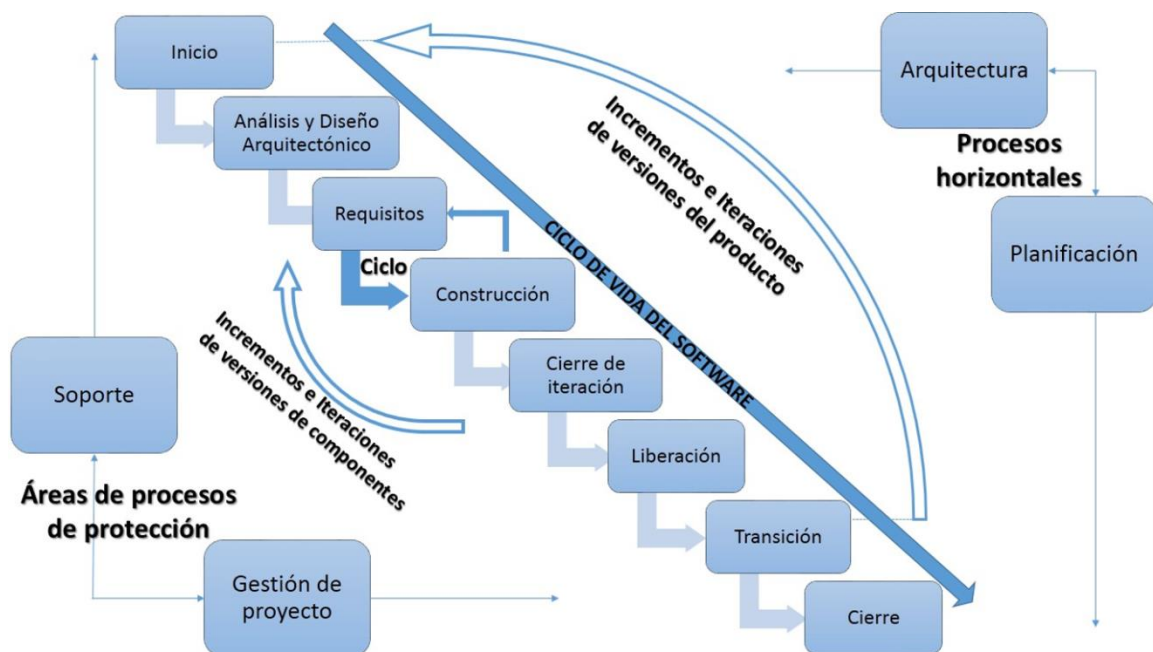


Imagen 1: Metodología Desarrollo Ágil con Calidad (DAC)

Sánchez Méndez (2013) (5), creadora de la metodología hace alusión a que: “El enfoque tradicional para el desarrollo de software ha probado su efectividad en grandes proyectos donde, por lo general, se requiere mayor control, formalismo y documentación. En muchos de los proyectos actuales el contexto es muy cambiante y se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Con el objetivo de ajustarse a estas restricciones, muchos equipos de desarrollo decidieron olvidar las buenas prácticas de la ingeniería de software. Ante esta situación, las metodologías ágiles aparecen como una posible respuesta para llenar este vacío metodológico. Estas metodologías están orientadas especialmente a

*proyectos pequeños y a pesar de su elevada simplificación no renuncian a las prácticas esenciales para asegurar la calidad del producto.”*

DAC, en su adaptación para la DIN, es una metodología de desarrollo de software que combina las metas y prácticas de las áreas de procesos del nivel 2 de Integración de Modelos de Madurez de Capacidades (CMMI) con las buenas prácticas de la dirección y desarrollo ágil de proyectos de software. Es colaborativo, recursivo-iterativo, incremental y guiado por procesos y requisitos. Está enfocado a proyectos pequeños o proyectos grandes divididos en sub-proyectos que desarrollan software de gestión basado en componentes.

### 1.5 ARQUITECTURA BASADA EN COMPONENTES

Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor que los principios de orientación por objetos y no se enfoca en asuntos específicos de los objetos como los protocolos de comunicación y la forma como se comparte el estado (7).

Esta arquitectura enfatiza en la separación de “partes”, entiéndase un servicio web, o un módulo que encapsula un conjunto de funciones relacionadas o de datos. A esto le llamamos componente de software, lo cual se integra a un grupo de funciones para formar un paquete independiente que puede comunicarse con otros paquetes lógicos o funcionales que tienen interfaces bien definidas. Brinda numerosas facilidades, entre ellas la de permitir un mejor desarrollo de los productos, ya que pueden ser diseñados e implementados sin impactar en otras partes del sistema. Su instalación es fácil, ya que se podrán reemplazar los componentes que se afectaron sin interferir en la integridad del sistema como un todo. Al separar el proceso en pequeños sub-procesos, se mitiga la complejidad técnica facilitándose su implementación y diseño, lo que se traduce en mejoras de calidad y la reducción del ciclo de desarrollo.

### 1.6 TECNOLOGÍAS

Para la selección del marco de desarrollo se tuvo en cuenta la correspondencia con las directrices que tiene el departamento de desarrollo de las aplicaciones de informatización de la universidad. La utilización de software libre fue otra de las premisas en apego a las políticas de soberanía tecnológica.

El uso de tecnologías web facilita compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web. En este caso se utiliza el lenguaje **PHP 5.3** para la programación del lado del servidor. Dentro de las ventajas que posee el mismo se encuentran que es de fácil aprendizaje, multiplataforma, multiparadigma, puede ser conectado con la mayoría de los gestores de base de datos e incluye una gran cantidad de funciones. Su uso está muy difundido en el mundo de los programadores debido a su flexibilidad.

### **Symfony**

Dentro de los marcos de trabajo que brindan funcionalidades para agilizar el proceso de construcción de software se encuentra el Symfony 1.4.8. Su sitio web oficial (9), apunta: *“Es un framework PHP de tipo full-stack construido con varios componentes independientes creados por el proyecto Symfony”*. El mismo ha ganado popularidad por su facilidad de trabajo, potencia, solidez y rendimiento, el cual es superior notablemente a su anterior versión. Aunque utiliza la arquitectura Modelo Vista Controlador (MVC), tiene su propia forma de trabajo, con algunas variantes como: la capa de abstracción de base de datos, el controlador frontal y las acciones. Sigue las buenas prácticas y patrones de diseño para la web.

Este marco de trabajo se acopla con amplios beneficios a **Doctrine 1.2**, un Mapeador de Objetos Relacionales (ORM, por sus siglas en inglés) que crea una capa de acceso a datos separando la entrada directa a la base de datos de la lógica del sistema. Es una librería muy completa y configurable. Permite la generación automática del modelo creando las clases que representan al modelo de negocio de la aplicación. Posibilita trabajar con YAML, acrónimo recursivo que significa "YAML Ain't Another Markup Language (en castellano: YAML no es otro lenguaje de marcado), que es un formato de serialización de datos legible muy usado. Facilita la búsqueda de registros basadas en cualquier campo de una tabla, además de lo fácil que resulta una vez definido el modelo, acceder y moverse por las entidades relacionadas. Posee un lenguaje propio: Doctrine Query Language (DQL), para manejar las interacciones con la base de datos (10).

### **JavaScript**

Como lenguaje del lado del cliente se selecciona JavaScript 1.4. El mismo se implementa como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web

se provee a este lenguaje de una implementación del Modelo de Objetos del Documento (DOM, por sus siglas en inglés).

Se utiliza además la librería **Ext JS 3.3.0** (11) ya que logra un equilibrio entre el cliente y el servidor al distribuir la carga de procesamiento, lo que permite que el servidor pueda atender más peticiones al mismo tiempo al tener menor carga. La gran ventaja de ExtJS es que posee un conjunto de componentes ya creados que permite su utilización de forma rápida en los sitios web.

A esto se le suman algunas tecnologías importantes como es el uso de **AJAX** para mitigar la lentitud de renderizado de los componentes de Ext JS, así como de los datos adicionales que se solicitan al servidor cargándose en segundo plano sin interferir con la visualización ni el comportamiento de la página. Se utiliza también la biblioteca **HTML2PDF** para la generación de reportes en formato PDF de forma rápida y dinámica (12).

### **Lenguaje de Marcas de Hipertexto**

La versión 5 de HTML (por sus siglas en inglés), es la quinta revisión importante de este lenguaje base de la World Wide Web, siendo esta la primera vez que HTML y XHTML se han desarrollado en paralelo. Establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Proporciona nuevas funcionalidades a través de una interfaz estandarizada y permite renderizar elementos en tercera dimensión (3D) (16).

### **Las hojas de estilo en cascada.**

CSS, por sus siglas en inglés, hace referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas web escritas en lenguaje HTML. La versión 3 provee de mejores diseños y estilos que crean un ambiente más elegante en los proyectos. La principal característica es la modularización de sus principales características (17).

## **1.7 HERRAMIENTAS**

### **NetBeans**

Es un Entorno de Desarrollo Integrado (IDE) de código abierto escrito completamente en Java, multiplataforma y está registrado bajo Licencia Común de Desarrollo y Distribución (CDDL) y Licencia Pública General de GNU (GPL). La versión 7.4 de esta herramienta brinda servicio de control de versiones. Permite crear aplicaciones web

escritas en PHP 5, HTML 5, CSS 3, Javascript, AJAX, Symfony y otros frameworks PHP. Posee un potente editor y debugger integrados (18).

### **Visual Paradigm**

Pressman (2002) (13) apunta que: *“es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.”*

A pesar de que no es software libre, posee una versión libre llamada Visual Paradigm for UML Community Edition, herramienta multiplataforma que soporta el ciclo de vida completo del desarrollo de una aplicación informática. Teniendo en cuenta lo anterior se selecciona la versión 8.0.

### **Subversion**

Como sistema de control de versiones se selecciona Subversion en su versión 1.6.16. Es un software libre bajo una licencia de tipo Apache/ Distribución de Software Berkeley (BSD). Entre sus ventajas se encuentran que sigue el historial de los archivos y directorios a través de copias y renombrados, modificaciones atómicas, la creación de ramas y etiquetas en operaciones más eficientes enviando sólo las diferencias en ambas direcciones (20).

### **MySQL**

MySQL 5.1.37 es un Sistema Gestor de Base de Datos (SGBD) cliente/servidor, multihilo y multiusuario. Se compone de un servidor SQL, varios programas clientes y bibliotecas, herramientas administrativas y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se enlaza dentro de otras aplicaciones para obtener un producto más pequeño, rápido y fácil de manejar.

Dentro de las ventajas se encuentra que dispone de borrados multi-tablas, mejores utilidades de administración y cuenta con un sistema de replicación multihilo en los servidores esclavos. Soporta cinco tipos de tablas y posee recuperación automática ante fallas e integridad referencial.

Sus principales desventajas son: no es viable para su uso con grandes bases de datos, a las que se acceda continuamente ya que no implementa una buena escalabilidad. Carece de soporte para transacciones, rollback's y subconsultas (20).

## 1.8 ANÁLISIS DE OTRAS SOLUCIONES

Durante la investigación se revisaron diferentes sistemas relacionados con el objeto de estudio, algunos de los cuales se les hace referencia en los epígrafes siguientes. El estudio de los sistemas se realiza para tener una referencia acerca de la forma de informatizar los procesos de la organización. Además para buscar requisitos candidatos para la solución a desarrollar, y descubrir si existen aplicaciones informáticas que solventen los problemas actuales, al menos parcialmente con posibilidades de integración.

### 1.8.1 KAINOS

García Rodríguez y González González (2007), referencian a Kainos en el cual, uno de sus módulos *“...está dirigido a la gestión de todos los datos referentes a caracterizaciones estadísticas de la organización en el país y visualización de reportes informativos. Constituye una herramienta de trabajo para los miembros de la FEU Nacional de gran utilidad”* (14). Este sistema concibe el proceso de evaluación estudiantil de forma general a partir del nivel de facultad, pasando por las estructuras provinciales y nacionales. Otorga significación al trabajo con datos numéricos y la obtención de estadísticas para definir políticas a nivel de país.

Otra de los procesos que son gestionados por Kainos es mencionado por Couso Linares (2007) (15) quien expresa haciendo referencia al módulo de eventos: *“...fue diseñado con el objetivo de crear un espacio digital para la FEU, donde se registran y promueven sus actividades..., a través de la red interna de la UCI (...) incorporando funcionalidades a su espacio digital, el cual no es más que un sitio web con novedosas características.”* Debe hacerse énfasis en que el mismo se concentra en la esfera de cultura, trabajando las actividades de contenido extensionista realizadas dentro de la universidad. Permite la inserción de eventos y manifestaciones dentro del Festival de Artistas Aficionados, gestionando su emulación. La investigación presta atención a la divulgación de este tipo de actividades mediante esta plataforma. Los datos que se recogen no persisten en el tiempo como históricos una vez concluido cada curso escolar.

### 1.8.2 SISTEMAS PARA LA GESTIÓN DE EVENTOS

Suárez Blanco y Hernández Calvo (2008) (16) elaboran el análisis y diseño de un sistema para la gestión de eventos en la UCI. Ellos definen clasificaciones atendiendo a su magnitud: grandes eventos; copas y eventos; talleres, congresos y activos así como otras actividades. La propuesta del sistema se centra en la primera clasificación



teniendo en consideración los eventos de la Jornada Científica y el Seminario Juvenil Martiano. Los autores hacen una valoración exhaustiva acerca de los problemas y la necesidad del software, adaptándolo a las características del momento.

GRECIA (17): *“Es un sistema para la gestión de eventos científicos para la Universidad de las Ciencias Informáticas”*. El mismo se desarrolló en Drupal, *“...diseñado específicamente para mejorar el tiempo y el esfuerzo en el proceso de gestión de un evento científico”*. La tesis de grado de Castro Carrillo (2012), fue realizar esta aplicación hecha a la medida para integrarla con el Sistema de Gestión Universitaria (SGU) y el Repositorio Institucional (RI). Su objetivo era el de mitigar las dificultades existentes en el acceso y almacenamiento de la información. Los autores de esta solución también realizan un estudio sobre cómo relacionar mediante un nuevo protocolo de comunicación dichas aplicaciones en ambientes de trabajo totalmente diferentes.

### 1.8.3 SISTEMA PARA LA GESTIÓN DE INQUIETUDES

Otras de las tesis de grado que realizan una solución es la implementada por de la Torre Quintana y Cruz González (2008) (18) quienes hacen hincapié en que *“El diseño del Sistema de Gestión de Inquietudes de la FEU en la UCI, se basa en una propuesta de aplicación web interactiva, que presenta funcionalidades que mejoran el flujo de la gestión de inquietudes y viabiliza la retroalimentación entre la dirección de la FEU de la universidad y los estudiantes.”* Es válido destacar que dicho sistema abarca todo lo referente al seguimiento de las inquietudes, pero no como el proceso que se da desde la reunión de brigada en el momento en que se genera el planteamiento, hasta su retorno a la misma con la respuesta.

El aporte de este sistema se encuentra en la forma y el procedimiento que utiliza para la obtención de estadísticas que tributan a la toma de decisiones por parte de los dirigentes estudiantiles en la organización.

### 1.8.4 SISTEMAS PARA LA GESTIÓN DEL FUNCIONAMIENTO

García Rodríguez y González González (2007), realizan *“...el modelado, hasta la fase de diseño, de una aplicación web que automatiza los procesos del funcionamiento FEU de la UCI.”* (14) Se basa en la esfera de trabajo de funcionamiento, gestionando los cierres de esta área de trabajo, desde la brigada y hasta la universidad. Está enfocado principalmente a la gestión estadística de las reuniones de brigada, compilando criterios numéricos como asistencia, cantidad de planteamientos y

opiniones. Abarcan también las inquietudes que se generan y los principales debates políticos.

Basso Mesa y Gómez Almaguer (2009) (19) también implementaron un sistema para la gestión de la caracterización de los estudiantes de 5to año, no como un proceso continuo de atención y control desde que el estudiante ingresa en la universidad, sino como una forma de gestionar la documentación recopilada al final de cada curso. Presta especial connotación a la gestión de la documentación y la persistencia de los datos históricos, pero no permite que el proceso se lleve en tiempo real, ni que el estudiante, los dirigentes o los profesores puedan ir evaluando el desempeño del mismo y tomar acciones en función del análisis de esa información. Esta tesis representa una solución muy completa del proceso de caracterización, aunque no permite adaptarse a las nuevas estructuras que han ido cambiando con el tiempo.

Otras soluciones existentes se centran en la gestión de los alumnos ayudantes tal es el caso de los trabajos de González Fernández y Benítez Campo (2010) (20) e Hinojosa Domínguez y Hurtados Díaz (2010) (21), quienes logran implementar la aplicación web en el mismo tiempo por direcciones diferentes. Desarrollaron software hechos a la medida y con aplicaciones solo para sus respectivas facultades, lo que imposibilita gestionar los datos a un nivel superior. Ambas fueron elaboradas con tecnologías y herramientas diferentes, que no permiten que sus datos perduren en el tiempo tras pasar un curso académico.

#### 1.8.5 GESTIÓN UNIVERSITARIA

La DIN, encargada de digitalizar sus principales procesos, lleva a cabo la realización de un sistema integral que abarca todas las áreas de la universidad. El Sistema de Gestión Universitaria (22) fue desarrollado para: *“... mejorar la planeación, organización y dirección de los procesos universitarios facilitando su ejecución y control.”* Su aplicación comenzó en el año 2012, sustituyendo al anterior Sistema Automatizado para la Gestión Académica (Akademos), pues este no incluía todos los roles universitarios ni áreas necesarias en la gestión académica.

A pesar de que es un proyecto en continuo desarrollo, hasta ahora los módulos desplegados no logran abarcar todas las áreas comunes necesarias para la gestión de la información que necesita la FEU para uno de sus procesos fundamentales: el de evaluación estudiantil; solamente cubre las áreas de pregrado. Prevé dentro de sus 12 áreas de gestión universitaria la planificación de eventos, pero la misma no se encuentra implementada y está distante en la línea de tiempo de desarrollo del sistema.

### 1.8.6 CONCLUSIONES DE LAS SOLUCIONES ESTUDIADAS

Se puede concluir que las propuestas anteriores no solventan las nuevas situaciones, a partir de la poca flexibilidad para adaptarse a los escenarios actuales. No todas avanzan hasta la implementación de la solución, solamente describen el diseño, análisis o propuesta de la solución, haciendo una descripción de los casos de uso, las tecnologías empleadas y argumentando el diseño de la base de datos empleada. Ninguna está orientada a procesos, concibiendo el problema como la inserción de datos agrupados lógicamente para la obtención de estadísticas generales.

Los autores afirman que no se conoce ningún sistema que gestione los procesos de las organizaciones políticas y de masas a nivel nacional, lo más aproximado es el gran número de portales informativos y sitios web cuya función es mantener una línea de comunicación entre los miembros con noticias e informaciones. También existen los llamados Sistemas de Gestión del Personal, que son aquellos que utilizan algunas empresas para el control y el seguimiento de sus trabajadores, incluyendo tablas y planillas laborales.

### 1.9 CONCLUSIONES

Se identifican los procesos que lleva a cabo la FEU, los cuales generan mucha dinámica a partir de los diferentes escenarios en el funcionamiento de la organización.

Luego de realizada una revisión de los principales elementos teóricos para el desarrollo del trabajo, se decide desarrollar una aplicación web de tipo interactiva. Se define como marco de trabajo: el lenguaje PHP, el ORM Doctrine integrado al framework Symfony, como lenguaje del lado del cliente JavaScript con su librería ExtJS y como sistema gestor de base de datos fue elegido MySQL.

A partir del estudio del estado del arte se concluye que las soluciones estudiadas no solventan las nuevas situaciones y que no se conoce a nivel nacional ningún sistema que gestione los procesos de la FEU.

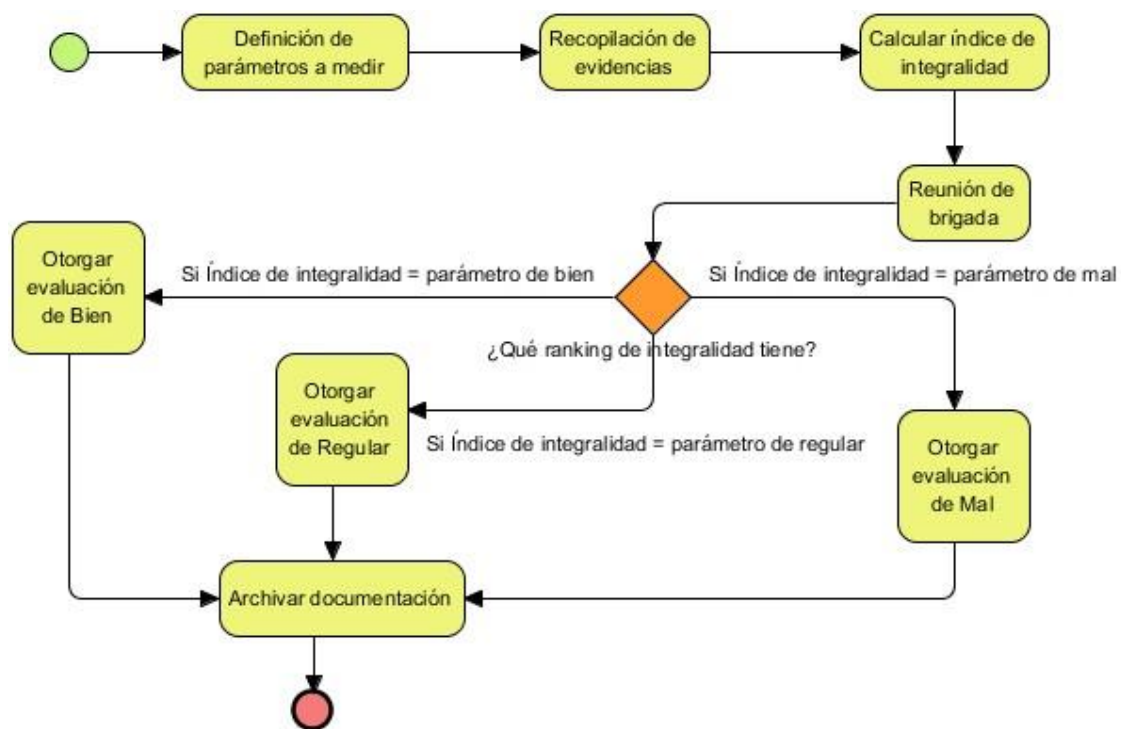
# CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

## 2.1 INTRODUCCIÓN

En este capítulo se describen los procesos de negocios, los requisitos funcionales, no funcionales, así como la arquitectura de la aplicación pasando por los patrones de diseño, los estilos arquitectónicos y los principales elementos del modelo de datos.

## 2.2 DESCRIPCIÓN DE LOS PROCESOS DE NEGOCIOS.

### 2.2.1 PROCESO DE INTEGRALIDAD



*Imagen 2 Modelado del proceso de integralidad*

Este proceso se inicia una vez que comienza el curso académico, donde el estudiante va recopilando evidencias por cada acción realizada en cualquiera de las esferas de la vida universitaria. Culminado el mismo se elabora un resumen donde queda reflejada toda la trayectoria del período en evaluación, haciendo énfasis en los resultados obtenidos y las actividades donde participó o colaboró como organizador. El resultado de este resumen es un documento que es entregado por el evaluado al presidente de brigada.

Una vez que se tengan todos los documentos de la brigada, en apego también a las fechas establecidas por las estructuras superiores; se realiza una reunión donde en

colectivo son discutidas las evaluaciones. Este es el espacio donde se le notifica a cada estudiante la calificación de su desempeño durante el curso.

Son consideradas evidencias los siguientes elementos, que constituyen subprocesos:

Los méritos son definidos por los secretariados a cada nivel donde se generen, siendo estos los encargados de otorgarlos o darlos a conocer a instancias superiores. Se clasifican en tres tipos: reconocimientos (que abarca todos los que se entreguen por las diferentes instancias o factores), distinciones y especiales (aquellos que no entren dentro de las dos categorías anteriores). Para optar por un mérito se deben cumplir ciertos parámetros, los cuales pueden ser modificados por el Consejo de la FEU a la instancia correspondiente.

Las sanciones que pueden ser aplicadas, están definidas en el ABC de la FEU, reglamento aprobado en el VII Congreso de la organización. Una vez demostrado y analizado que el estudiante cometió una indisciplina, se procede a la imposición de la medida. Pueden aplicar sanciones todos los niveles de dirección y en el caso de la brigada lo realiza su presidente, previo consentimiento de la mayoría dentro del grupo. Las sanciones son diferentes para los estudiantes y dirigentes estudiantiles, quienes se rigen por cláusulas especiales de tipificación de la falta.

Un estudiante puede transitar por las diferentes estructuras de dirección de la FEU. Los cargos estudiantiles son atendidos por el vicepresidente y el presidente de cada estructura u otra persona que ellos designen. Estos se guardan en un archivo histórico recogiendo la trayectoria del dirigente. Una vez concluida las elecciones es que se consideran oficiales, en su defecto deben ser aprobados por el Consejo de la FEU correspondiente. Los dirigentes pueden causar bajas por diferentes causas y motivos, esta acción solo la podrá hacer oficial el vicepresidente, previa consulta con el presidente y analizando los factores con todos los implicados. Mensualmente el dirigente recibe una evaluación de su trayectoria durante el período señalado, la cual debe ser discutida en la estructura a la que pertenece. El Consejo de la FEU de la Universidad define cada curso cuáles serán los cargos que serán ocupados en las estructuras inferiores. Para que un dirigente pase a ocupar un cargo a nivel de universidad debe transitar por las estructuras inferiores.

Los resultados obtenidos por publicaciones científicas, exámenes de premio o de suficiencia, así como otros que tributen al estudiante como fruto de su esfuerzo, son considerados bonificaciones. Las mismas se encuentran clasificadas de acuerdo a su tipo y son gestionadas por los secretariados al nivel correspondiente. Una vez que el

estudiante haya obtenido alguna bonificación debe dirigirse al miembro del secretariado que atiende este tema, haciéndole constar con los documentos oficiales de que obtuvo fidedignamente la misma.

Los alumnos ayudantes (AA) son un caso especial de membresía, que se atiende directamente por los responsables de la esfera Vida académica. Para ingresar al Movimiento de Alumnos Ayudantes se deben cumplir un número mínimo de requisitos, además de estar validado por la brigada. Terminado el proceso de ingreso el responsable de la esfera por la cual son atendidos procede a su registro, teniendo en cuenta las funciones que realizará, el tutor y la elaboración de su plan de trabajo mensual. En caso de que los requisitos que le dieron entrada al movimiento son perdidos, se procede a darle baja, especificando las causas que la motivaron y en previa consulta con el presidente de la estructura que lo atiende. Cada semestre los AA son ratificados, verificándose si posee los requisitos para continuar en el movimiento. Mensualmente son evaluados teniendo en cuenta la trayectoria en el período, la opinión de la brigada y del jefe de departamento por la parte administrativa.

Existen varias tareas que durante un curso en específico debe cumplir cada estudiante, ejemplo de ello está la Reunión de Estudios Militares (REM) y el Trabajo Socialmente Útil (TSU). Una vez concluido los procesos de este tipo y enviado por la administración la certificación de las evaluaciones se procede a su registro. Pueden existir tantas tareas evaluadas como se definan, es el Consejo de la FEU a cada nivel quien puede crearlas para su posterior evaluación.

Hay otros elementos que se consideran evidencias y tributan al proceso de integralidad que no son responsabilidad directa de la FEU. Estos tienen un peso importante en la valoración cualitativa del estudiante, tal es el caso de las evaluaciones de la guardia, la cuartelería (ambas de carácter mensual), si está en proyecto y cuál es rol dentro de él, el índice académico, las evaluaciones de los cortes docentes, si lleva arrastre, mundiales o si es repitente (todo esto se engloba en su situación académica).

### 2.2.2 PROCESO DE CARACTERIZACIÓN

Este proceso se realiza al concluir el último curso escolar como resumen de la trayectoria de los cinco años de la carrera. Se procede de igual manera que el proceso de integralidad (descrito en el epígrafe anterior), la diferencia radica en que este agrupa toda la vida universitaria y se retroalimenta de las evaluaciones obtenidas en los procesos de integralidad que se realizaron de primero a cuarto año. En el mismo se define una ecuación que cuantifica cada una de las evidencias que se obtienen a fin

de obtener un ranking de integralidad, lo cual tributa directamente al proceso de ubicación laboral en la confección del escalafón para el otorgamiento de plazas.

### 2.2.3 GESTIÓN DE EVENTOS Y ACTIVIDADES



*Imagen 3 Modelado del proceso de gestión de eventos y actividades*

La gestión de las actividades comienza con la planificación por parte del nivel de dirección que las organice. Se define el lugar, fecha, hora, los tipos de participación, las necesidades logísticas y la esfera a la cual está asociada. El miembro del secretariado de la FEU que atiende dicha esfera es el encargado de darle seguimiento y coordinar los aseguramientos materiales necesarios para su realización. Las que se ejecutan a nivel de brigada, son atendidas por el presidente quien se encargaría de la planificación y aseguramiento.

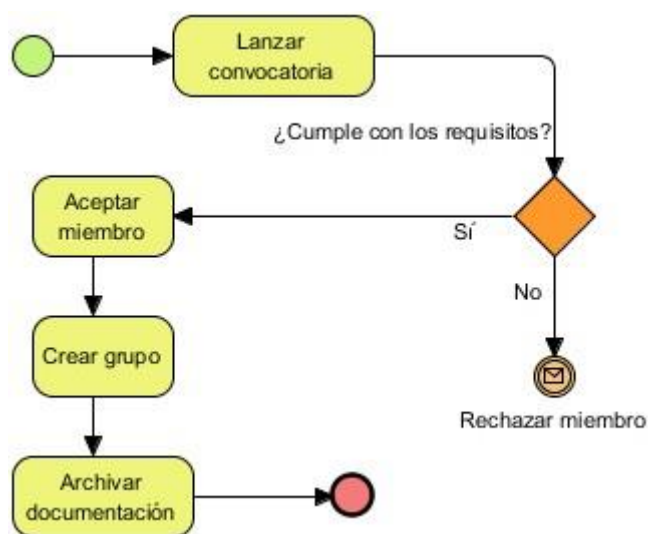
Una vez que la actividad o el evento ha sido realizado, el presidente de brigada archivará los participantes de su grupo teniendo en cuenta los resultados obtenidos y el tipo de participación. En dependencia del nivel donde se realicen dichas actividades, se emitirá por parte del secretariado un listado donde se certificará la asistencia de los estudiantes. La participación puede darse de tres tipos diferentes: como ponente (con la presentación de un trabajo), colaterales (en los sub-eventos que se realizan) y como organizador.

### 2.2.4 GESTIÓN DE LA MEMBRESÍA

Los estudiantes dentro de la organización se asocian en grupos o movimientos, en dependencia de sus gustos, preferencias o deseos de aprender alguna habilidad. Los grupos pueden ser creados por cualquier estudiante, quien será su coordinador y moderador. De los grupos existentes los estudiantes pueden convertirse en invitados permanentes, esto hace que puedan acceder a compartir información con el resto de los integrantes. Los miembros tienen privilegios especiales, pues su entrada no es automática, ya que deben ser aprobados por el responsable del grupo.

Es responsabilidad de las estructuras de la FEU darle atención a los movimientos que agrupan a gran cantidad de estudiantes, garantizando su correcto funcionamiento y

dirección. A decisión del Consejo FEU de la universidad puede seleccionarse un coordinador de cada movimiento, el cual es tratado como miembro del secretariado al nivel correspondiente.



*Imagen 4 Modelado del proceso: Gestión de la membresía*

## 2.2.5 FUNCIONAMIENTO INTERNO

Las tareas que garantizan el funcionamiento interno de la organización dependen de todas las estructuras, pues las inferiores tributan información a las superiores. El flujo también se realiza en la otra dirección cuando se orienta la realización de diferentes procesos o se comparte la documentación.

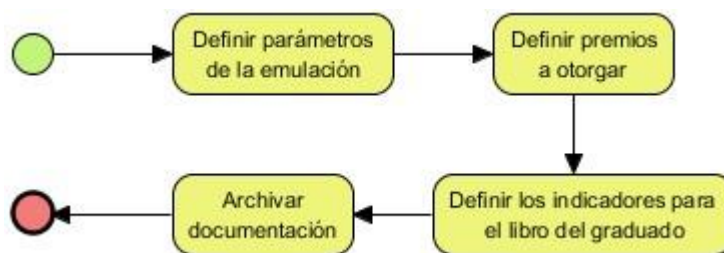
La emulación es uno de los mecanismos base para la FEU, pues a partir de ahí define políticas de estimulación y de evaluación de cada una de las estructuras inmediatamente inferiores. Cada una de ellas puede definir una emulación, estableciendo sus parámetros, la ponderación de cada uno de ellos, los actores y los que fungirán de moderadores, quienes único tienen acceso a modificar los parámetros ya establecidos. Las emulaciones pueden ser públicas o privadas a un reducido grupo de estudiantes o profesores.

El libro del graduado es una forma de recopilar los datos de los estudiantes que se encuentren en quinto año. El estudiante debe entregar una página, actualizando su información personal, los cuales no pasarán por ningún nivel de certificación, el objetivo es dejar sus impresiones de la universidad y que se registre en un archivo histórico sus credenciales.

La entrega de los premios a profesores y trabajadores, es uno de los procesos que lleva el vicepresidente en cada nivel correspondiente, atendiendo a cada uno de los



trabajadores docentes o no docentes que obtienen algún tipo de reconocimiento que otorga la FEU. Una vez otorgado es que se debe proceder a su registro para que sea recogido en el archivo histórico.



*Imagen 5 Modelado del proceso: Funcionamiento interno*

## 2.3 DEFINICIÓN DE LOS REQUISITOS FUNCIONALES Y NO FUNCIONALES

### 2.3.1 TÉCNICAS DE OBTENCIÓN DE REQUISITOS

Existe un gran número de técnicas para obtener requisitos, cada una adaptada a las condiciones del desarrollo de la aplicación. Se debe hacer énfasis en que ninguna de estas técnicas fue suficiente por sí sola sino que hubo que combinarlas para obtener requisitos completos, además de emplearse la ayuda y orientación de los clientes, de los cuales uno, se encuentra dentro del equipo de desarrollo.

#### Entrevistas:

Es una de las técnicas que más se utilizó, siendo de gran utilidad para la obtención de información cualitativa como opiniones o descripciones subjetivas de actividades de la FEU. Abarcó cuatro pasos principales: identificar los entrevistados, preparar y realizar la entrevista y documentación de los resultados. En el caso del proyecto se definió la minuta de reunión, para reflejar los asuntos tratados, los acuerdos tomados y las partes involucradas en la reunión.

#### Observación:

Con la aplicación de esta técnica se pudo obtener de primera mano la forma de realización de las actividades y los principales procesos de la organización, además de verificarse que estos seguían los pasos establecidos para su desarrollo. En muchos de los casos el papel no muestra lo que sucede realmente en la práctica, de ahí la importancia de su empleo.

#### Estudio de la documentación:

Tomando como base el estudio de varios tipos de documentación como manuales, reglamentos, minutas de reuniones y reportes, se obtuvo información con respecto a la

forma de organización y el flujo de los procesos de la FEU. La documentación en ocasiones no reflejaba la forma en que realmente se desarrollan las actividades, o donde se encontraba el poder de la toma de decisiones. Sin embargo, fue de gran importancia para introducir a los desarrolladores el dominio de la operación y el vocabulario que se utiliza.

A partir de la utilización de las técnicas anteriores se obtuvo un total de 117 requisitos funcionales y 16 no funcionales.

### 2.3.2 REQUISITOS FUNCIONALES

Para la especificación de los mismos se utilizaron los artefactos propuestos por la metodología de desarrollo ágil DAC. A continuación se muestra una tabla con el listado de ellos, donde la descripción en algunos casos no fue necesaria introducirla debido a que con el propio nombre del requisito queda explícito la función que realiza. Tanto la prioridad para el cliente como la complejidad pueden ser definidas en Alta (A), Media (M) o Baja (B) de acuerdo a los criterios definidos en el Anexo 7.

#### Paquete Común

N°	Nombre	Descripción	Prioridad para el cliente	Complejidad
RF 1	<i>Insertar nuevo usuario.</i>		A	B
RF 2	<i>Modificar datos de usuario.</i>		A	B
RF 3	<i>Eliminar usuario del sistema.</i>		A	M
RF 4	<i>Otorgar permisos a usuario.</i>		A	B
RF 5	<i>Revocar permisos a usuario.</i>		A	B
RF 6	<i>Insertar nuevo rol.</i>	<i>Define un conjunto de entradas a las cuales podrán tener acceso los usuarios que posean el rol.</i>	A	M
RF 7	<i>Modificar rol.</i>		A	M
RF 8	<i>Eliminar rol.</i>		A	M
RF 9	<i>Autenticar usuario.</i>	<i>Verifica las credenciales del usuario con el sistema para su utilización.</i>	A	M
RF	<i>Bloquear usuario.</i>	<i>Bloquea el acceso de un</i>	A	B

10		<i>usuario específico al sistema.</i>		
RF 11	<i>Modificar variables de configuración.</i>	<i>Modifica las variables globales de configuración en el sistema.</i>	A	B
RF 12	<i>Seguimiento de trazas por grupos y por usuarios.</i>	<i>Lista las trazas de los usuarios que interactúan en el sistema de forma global o individual.</i>	A	A
RF 13	<i>Búsqueda avanzada.</i>	<i>Realiza una búsqueda por diferentes criterios de los usuarios que están en el sistema.</i>	A	M
RF 14	<i>Importar estudiantes.</i>	<i>Inserta en el sistema los estudiantes de un curso académico a través del consumo de servicios web universitarios.</i>	A	A
RF 15	<i>Insertar nuevo módulo.</i>		A	M
RF 16	<i>Modificar módulo.</i>		A	M
RF 17	<i>Eliminar módulo.</i>		A	M
RF 18	<i>Insertar nueva funcionalidad.</i>		A	M
RF 19	<i>Modificar funcionalidad.</i>		A	M
RF 20	<i>Eliminar funcionalidad.</i>		A	M
RF 21	<i>Asignar funcionalidad a rol.</i>		A	M
RF 22	<i>Insertar nomencladores de administración.</i>	<i>Inserta un nuevo nomenclador al sistema, solo aquellos que están relacionados con la administración.</i>	M	M
R 23	<i>Modificar nomencladores de administración.</i>		M	M
RF 24	<i>Eliminar nomencladores de administración.</i>		M	M

## Paquete Integralidad

<b>Nº</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Prioridad para el cliente</b>	<b>Complejidad</b>
RF 25	<i>Insertar mérito a estudiante.</i>	<i>Registrarle a un estudiante la obtención de algún mérito que haya obtenido.</i>	A	B
RF 26	<i>Modificar mérito a estudiante.</i>		A	B
RF 27	<i>Eliminar mérito a estudiante.</i>		A	B
RF 28	<i>Mostrar listado de méritos estudiantiles.</i>	<i>Lista los méritos que fueron registrados a los estudiantes de acuerdo a varios criterios fecha, tipo y nivel.</i>	B	B
RF 29	<i>Insertar cargo a estudiante.</i>	<i>Registrarle el cargo al estudiante para que sea reconocido como dirigente estudiantil.</i>	A	B
RF 30	<i>Modificar cargo a estudiante.</i>		A	B
RF 31	<i>Eliminar cargo a estudiante.</i>		A	B
RF 32	<i>Mostrar listado de dirigentes estudiantiles.</i>	<i>Lista los dirigentes registrados en el sistema de acuerdo a los criterios de nivel, facultad, cargo y factor.</i>	B	B
RF 33	<i>Mostrar evaluaciones de dirigentes estudiantiles.</i>		B	B
RF 34	<i>Insertar baja a dirigente.</i>	<i>Una vez que el estudiante tiene un cargo, puede resultar baja por diferentes motivos, las cuales deben ser registradas como un archivo histórico.</i>	A	B
RF 35	<i>Modificar baja a dirigente.</i>		A	B
RF 36	<i>Eliminar baja a dirigente.</i>		A	M
RF 37	<i>Evaluar dirigente.</i>	<i>Inserta una evaluación a los que son reconocidos como dirigentes estudiantiles.</i>	A	M

RF 38	Mostrar listado de bajas de dirigentes estudiantiles.		B	B
RF 39	Mostrar estadísticas de las evaluaciones de dirigentes estudiantiles.		B	B
RF 40	Mostrar estadísticas por provincias.	Lista los dirigentes estudiantiles por provincias de acuerdo a los criterios nivel, factor y/o facultad.	B	B
RF 41	Aplicar sanción a estudiante.	Inserta una nueva sanción a un estudiante en dependencia de los tipos definidos en nomencladores.	A	B
RF 42	Modificar sanción a estudiante.		A	B
RF 43	Eliminar sanción a estudiante.		A	B
RF 44	Insertar tarea a evaluar.	Corresponde al área de las tareas que se convocan y que luego deben ser evaluadas tal es el caso del Trabajo Socialmente Útil (TSU) y la Reunión de Estudios Militares (REM).	A	B
RF 45	Modificar tarea a evaluar.		M	B
RF 46	Eliminar tarea a evaluar.		M	B
RF 47	Insertar evaluación a estudiante en tarea a evaluar.	Inserta una evaluación de los estudiantes en correspondencia a su participación en algunas de las tareas definidas.	A	B
RF 48	Modificar evaluación a estudiante en tarea a evaluar.		A	B
RF 49	Eliminar evaluación a estudiante en tarea a evaluar.		A	B
RF 50	Mostrar mejores estudiantes.	Lista los estudiantes descendientemente de acuerdo al ranking de integralidad.	B	B
RF 51	Mostrar estudiantes con problemas.	Lista los estudiantes ascendientemente de acuerdo al	B	B

		<i>ranking de integralidad.</i>		
RF 52	Mostrar estudiantes sancionados.		B	B
RF 53	Mostrar estudiantes reconocidos.		B	B
RF 54	Mostrar estadísticas de estudiantes sancionados.		B	B
RF 55	Mostrar estadísticas de estudiantes reconocidos.		B	B
RF 56	Actualizar información personal.	Actualiza los datos personales que puede subir el usuario como: dirección particular, teléfono, móvil, si es militante o no, etc.	A	M
RF 57	Actualizar fondo de perfil.		A	A
RF 58	Cambiar de brigada.	Actualiza la ubicación de la brigada en la que se encuentra el estudiante.	M	B
RF 59	Cambiar de apartamento.	Actualiza la ubicación del estudiante en la residencia.	M	B
RF 60	Ingresar miembro al Movimiento de Alumnos Ayudantes.	Insertar un estudiante al Movimiento de Alumnos Ayudantes para que sea reconocido como tal.	A	M
RF 61	Ratificar alumno ayudante.		A	M
RF 62	Eliminar miembro del Movimiento de Alumnos Ayudantes.	Elimina un estudiante que haya sido introducido como miembro del movimiento.	A	M
RF 63	Modificar membresía del Movimiento de Alumnos Ayudantes.		A	M
RF 64	Dar baja a miembro del movimiento.	Inserta una baja a un estudiante que por diferentes causas salga del movimiento.	A	B
RF 65	Eliminar baja del movimiento.		A	B
RF 66	Evaluar alumno ayudante.		A	B

RF 67	Mostrar alumnos ayudantes.	Lista los alumnos ayudantes registrados en el sistema.	B	B
RF 68	Mostrar evaluaciones de alumnos ayudantes.		B	B
RF 69	Mostrar alumnos ayudantes de nuevo ingreso.		B	B
RF 70	Mostrar alumnos ayudantes por grupos.		B	B
RF 71	Mostrar bajas del Movimiento.	Lista los estudiantes que resultaron bajas del Movimiento de Alumnos Ayudantes.	B	B
RF 72	Mostrar estadísticas del movimiento.		B	B
RF 73	Mostrar estadísticas de las evaluaciones de los alumnos ayudantes.		B	B
RF 74	Insertar nueva bonificación a estudiante.	Inserta las bonificaciones que están tipificadas en los nomencladores, esto lo realiza el estudiante y requiere una certificación de un miembro del secretariado a cualquier nivel.	M	B
RF 75	Modificar bonificación a estudiante.		M	B
RF 76	Eliminar bonificación a estudiante.		M	B
RF 77	Mostrar bonificaciones de estudiantes.		B	B
RF 78	Mostrar estadísticas de las bonificaciones de estudiantes.		B	B

Paquete Eventos y actividades

Nº	Nombre	Descripción	Prioridad para el cliente	Complejidad
RF 79	Insertar nuevo nomenclador de actividad.	Inserta una nueva actividad a la lista de nomencladores genéricos.	A	B

RF 80	Modificar nomenclador de actividades.		A	B
RF 81	Eliminar actividad del nomenclador de actividades.		A	B
RF 82	Registrar nueva instancia de actividad.	A partir de las actividades genéricas que se definen en el nomenclador, se crean instancias de las actividades las cuales contemplan una serie de datos para ubicarlas en tiempo y espacio, además de definirse cómo se realizará la participación.	A	M
RF 83	Modificar nueva instancia de actividad.		A	M
RF 84	Eliminar nueva instancia de actividad.		A	M
RF 85	Insertar participación de estudiante en actividad.	Añade los estudiantes que participaron en alguna de las instancias declaradas de una actividad en específico.	A	M
RF 86	Modificar participación de estudiante en actividad.		A	M
RF 87	Eliminar participación de estudiante en actividad.		A	M
RF 88	Mostrar listado de participantes por actividad.	Lista los estudiantes que participaron en una actividad dada de acuerdo a los criterios fecha y tipo de participación.	B	B
RF 89	Mostrar estadísticas de actividades.	Genera una tabla comparativa con las actividades desarrolladas en un período dado teniendo en cuenta los parámetros de tipo de participación.	B	B
RF 90	Mostrar estadísticas de participantes por actividad.		B	B
RF 91	Mostrar actividades.	Lista las actividades de acuerdo a varios criterios.	B	B



Paquete Funcionamiento interno

<b>Nº</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Prioridad para el cliente</b>	<b>Complejidad</b>
RF 92	<i>Insertar nueva membresía.</i>	<i>Añade un nuevo tipo de membresía a un estudiante.</i>	M	B
RF 93	<i>Modificar membresía.</i>		M	B
RF 94	<i>Eliminar membresía.</i>		M	B
RF 95	<i>Mostrar miembros.</i>	<i>Lista los estudiantes de acuerdo a la membresía que posean.</i>	B	B
RF 96	<i>Mostrar estadística de miembros.</i>	<i>Genera una tabla comparativa con los tipos de membresía de los estudiantes.</i>	B	B
RF 97	<i>Insertar página en el libro del graduado.</i>	<i>Añade una página al libro del graduado y guarda en un archivo histórico la información generada.</i>	M	M
RF 98	<i>Modificar página en el libro del graduado.</i>		M	M
RF 99	<i>Eliminar página en el libro del graduado.</i>		M	M
RF 100	<i>Graduar estudiante.</i>	<i>Elimina a los estudiantes de la base de datos del curso actual, para que no sean consultados o activados en próximos cursos.</i>	A	M
RF 101	<i>Insertar nueva emulación.</i>	<i>Añadir una emulación a la lista de nomencladores del curso vigente.</i>	B	B
RF 102	<i>Modificar una emulación.</i>		B	B
RF 103	<i>Eliminar una emulación.</i>		B	B
RF 104	<i>Insertar parámetros a una emulación.</i>	<i>Define cada uno de los parámetros que se tendrán en cuenta en la emulación.</i>	B	B
RF 105	<i>Modificar parámetro de emulación.</i>		B	B
RF 106	<i>Eliminar parámetro de emulación.</i>		B	B

RF 107	Insertar actores de emulación.		B	M
RF 108	Modificar actores de emulación.		B	B
RF 109	Eliminar actores de emulación.		B	B
RF 110	Insertar moderador de emulación.		B	B
RF 111	Modificar moderadores de emulación.		B	B
RF 112	Eliminar moderadores de emulación.		B	B
RF 113	Visualizar ganadores de emulación.	Mostrar las posiciones de cada uno de los actores en una emulación dada.	B	M

#### Paquete Alertas y notificaciones

Nº	Nombre	Descripción	Prioridad para el cliente	Complejidad
RF 114	Visualizar cambios en el perfil.	Mostrar las modificaciones realizadas al perfil personal de los estudiantes desde la última vez que se entró al sistema.	M	A
RF 115	Enviar correos.	Enviar correos a destinatarios seleccionados con mensajes previamente definidos por el usuario o por los administradores del sistema.	M	A
RF 116	Notificar cambios en el perfil vía correo.		M	M
RF 117	Notificar próximas actividades vía correo.		M	M

### 2.3.3 REQUISITOS NO FUNCIONALES

#### Usabilidad

- Desarrollar una solución web para los procesos de gestión.

- La aplicación debe estar dirigida a registrar la información referente al proceso de evaluación estudiantil de la FEU.
- Solo se mostrarán a los usuarios aquellas acciones o informaciones a las que por su responsabilidad o rol dentro del negocio necesitan acceder mostrando en la vista los íconos y menús correspondientes.
- Las vistas del sistema deben indicar en cada momento la acción que se está realizando así como lo íconos deben estar representados por una imagen acorde a la acción que se realiza mediante el mismo.
- Para el despliegue del software estará instalado en el servidor el sistema operativo CentOS 6.2 donde se encuentre PHP v5.3 con las librerías php5-ldap, php5-gd, php5-mcrypt, php5-xsl, php5-openssl, Apache 2.2 con el módulo rewrite activado.
- Para el uso del sistema se requiere una PC cliente con el navegador web Mozilla Firefox 3.6 o superior.

### **Confiabilidad**

- El sistema puede permanecer inactivo durante 10 minutos. Al cumplirse este término se cerrará la sesión teniendo que autenticarse el usuario nuevamente.
- El sistema de autenticación recogerá datos necesarios de los usuarios y tendrá control de identidades de los mismos.
- Debe proveer algún mecanismo seguro de encriptación y transferencia de datos.
- Se validan los datos recibidos de otras aplicaciones externas.
- Se debe mantener una seguridad a nivel de usuarios y contraseñas codificadas para el acceso a la base de datos.
- El servidor de aplicaciones y de base de datos deberá mantener una seguridad mediante firewall para proteger el código y la información.

### **Rendimiento**

- El tiempo de respuesta de cada una de las páginas debe ser menor que medio minuto, excepto en aquellas que por las actividades que realizan requieran más tiempo, tal es el caso de la generación de los reportes.

### **Soporte**

- La base de datos que utilizará el sistema como medio de almacenamiento de la información estará soportada sobre un gestor de bases de datos MySQL, permitiéndole interactuar con otros sistemas estableciendo vías de compatibilidad.

- Debe ser capaz de obtener de sistemas externos (SGU) información como los datos relacionados con la situación académica, así como del LDAP UCI la autenticación para acceder a la aplicación.
- Debe permitir brindar información a través de servicios web públicos a sistemas externos.

## 2.4 VALIDACIÓN DE REQUISITOS

La validación de requisitos trata de mostrar que los levantados en la fase inicial del proceso de desarrollo del sistema realmente definen el software que el cliente desea. Es importante debido a que los errores en el documento de requisitos pueden conducir a importantes costes al repetir el trabajo cuando son descubiertos durante el desarrollo o después de que el sistema esté en uso.

A decir de Sommerville (2005) (8) durante el proceso de validación de requisitos, se deben llevar a cabo verificaciones que comprenden:

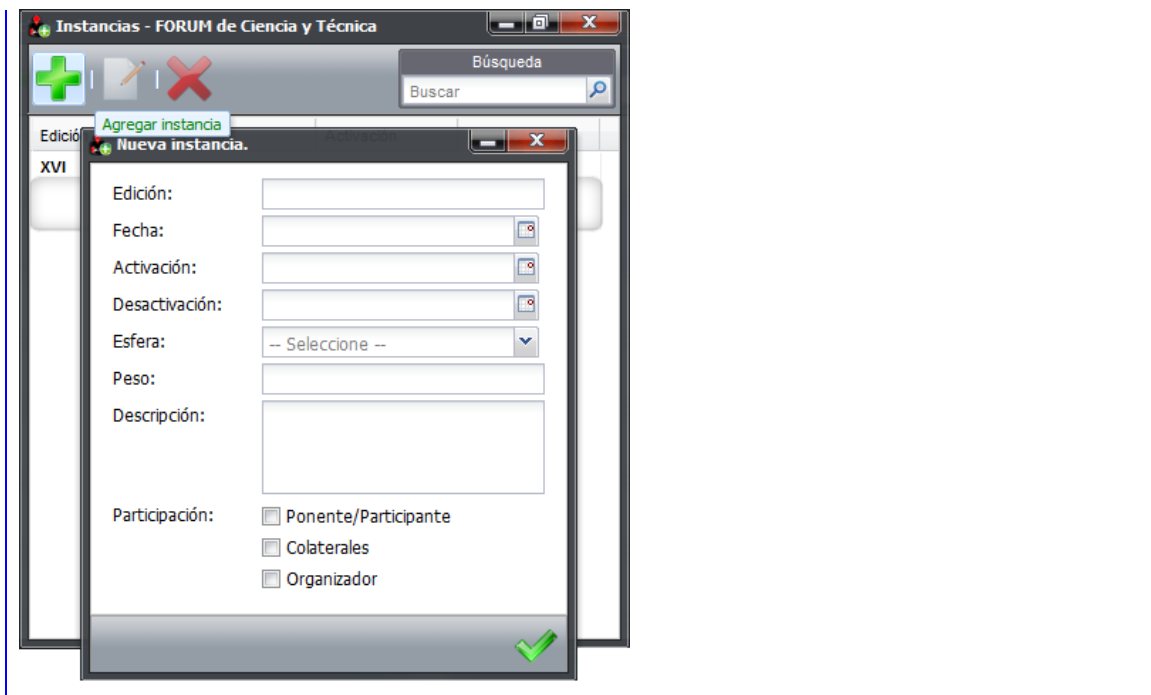
- Validez: un usuario puede pensar que se necesita un sistema para llevar a cabo ciertas funciones. Sin embargo, el razonamiento y el análisis pueden identificar que se requieren funciones adicionales o diferentes. Los sistemas tienen diversos puntos de vistas con diferentes necesidades y para cualquier conjunto de requisitos es inevitable un compromiso en el entorno del negocio.
- Consistencia: los requisitos en el documento no deben contradecirse. Esto es, no debe haber restricciones o descripciones contradictorias en la misma función del sistema.
- Completitud: el documento de requisitos debe incluir requisitos que definan todas las funciones y restricciones propuestas por el usuario del sistema.
- Realismo: utilizando el conocimiento de la tecnología existente, los requisitos deben verificarse para asegurar que se pueden implementar. Estas verificaciones también deben tener en cuenta el presupuesto y la confección de agendas para el desarrollo del sistema.
- Verificabilidad: para reducir la posibilidad de discusiones entre el cliente y el contratista, los requisitos del sistema siempre deben redactarse de tal forma que sean verificables. Esto significa que se debe poder escribir un conjunto de pruebas que demuestren que el sistema a entregar cumple cada uno de los requisitos especificados.

Todos los indicadores mencionados al inicio de este epígrafe fueron seguidos, apeándose al estándar y las buenas prácticas de ingeniería de software definido por Sommerville (2005) (8). Se concluye que todos los requisitos son válidos.

## 2.5 DESCRIPCIÓN DE REQUISITOS

A continuación se muestra la descripción de dos requisitos funcionales como ejemplo utilizando la metodología DAC, el resto se encuentra en el expediente de proyecto.

<b>Historia de usuario</b>	
<b>Código:</b> RF 82	<b>Nombre historia de usuario:</b> Registrar nueva instancia de actividad.
<b>Modificación de historia de usuario número:</b> 2	
<b>Referencia:</b> RF 82 del Expediente de proyecto.	
<b>Programador:</b> Victor González Cardoso	<b>Iteración asignada:</b> 2
<b>Prioridad :</b> Alta	<b>Puntos estimados:</b> 4 días.
<b>Riesgo en desarrollo:</b> Cuidar los métodos empleados en el desarrollo del algoritmo para insertar en la Base de Datos.	<b>Puntos reales:</b> 4 días.
<p><b>Descripción:</b></p> <p>El proceso comienza cuando se realiza la actividad o el evento. El vicepresidente a cualquier nivel procede a insertar una nueva instancia de la actividad en cuestión. Se accede a la funcionalidad a través del menú inicio, en Secretariado. En la ventana activa se da clic en Gestionar participación. Luego se selecciona de la lista de nomencladores la actividad a la cual se desea realizarle una instancia. En la ventana activa se listarán todas las instancias creadas del nomenclador. En la barra de herramienta se selecciona adicionar, representado por un signo de “más” (+), abierto el formulario se introducen todos los campos necesarios y luego se da clic en aceptar. El proceso habrá concluido cuando en la lista de instancias aparezca la nueva introducida. El sistema además mostrará un mensaje de notificación indicando que el proceso concluyó satisfactoriamente.</p>	
<p><b>Observaciones:</b></p> <p>Todos los campos son obligatorios para la inserción de una nueva instancia.</p>	
<b>Prototipo de interfaz:</b>	



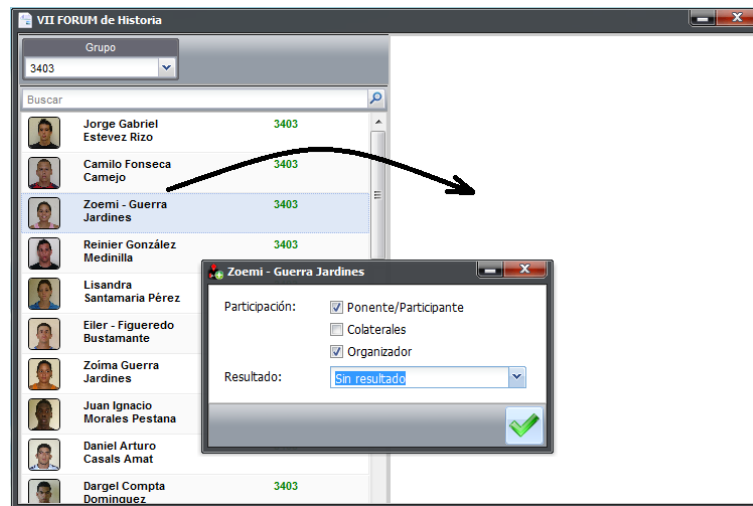
<b>Historia de usuario</b>	
<b>Código:</b> RF 85	<b>Nombre historia de usuario:</b> Insertar participación de estudiante en actividad.
<b>Modificación de historia de usuario número:</b> 3	
<b>Referencia:</b> RF 85 del Expediente de proyecto.	
<b>Programador:</b> Eiler Figueredo Bustamante	<b>Iteración asignada:</b> 2
<b>Prioridad :</b> Alta	<b>Puntos estimados:</b> 5 días.
<b>Riesgo en desarrollo:</b> La inserción de la actividad debe ser registrada en las trazas del sistema de la óptima forma posible.	<b>Puntos reales:</b> 4 días.
<p><b>Descripción:</b></p> <p>Una vez que los miembros del secretariado a los diferentes niveles insertan un nuevo nomenclador e instancia de la actividad los presidentes de brigadas proceden a insertar la participación a un estudiante. Seleccionan la actividad en cuestión de la lista que se les muestra en la interfaz. En la ventana activa se mostrarán dos columnas: en la de la izquierda aparecerán los estudiantes asociados a una brigada. Bastará con que se arrastren los estudiantes que participaron en la misma desde el panel izquierdo hacia el derecho. Se marcan los tipos de participación, se introduce el resultado obtenido y luego se acepta para culminar el proceso. En caso de que se</p>	

produzca un error, el sistema mostrará un mensaje indicando el mismo.

### Observaciones:

Se pueden hacer selecciones múltiples de estudiantes para registrarle una misma actividad. Un estudiante no puede ser insertado dos veces en la misma instancia. Una participación ha sido insertada satisfactoriamente cuando el estudiante aparezca en la columna izquierda del panel.

### Prototipo de interfaz:



## 2.6 DEFINICIÓN DE LA ARQUITECTURA

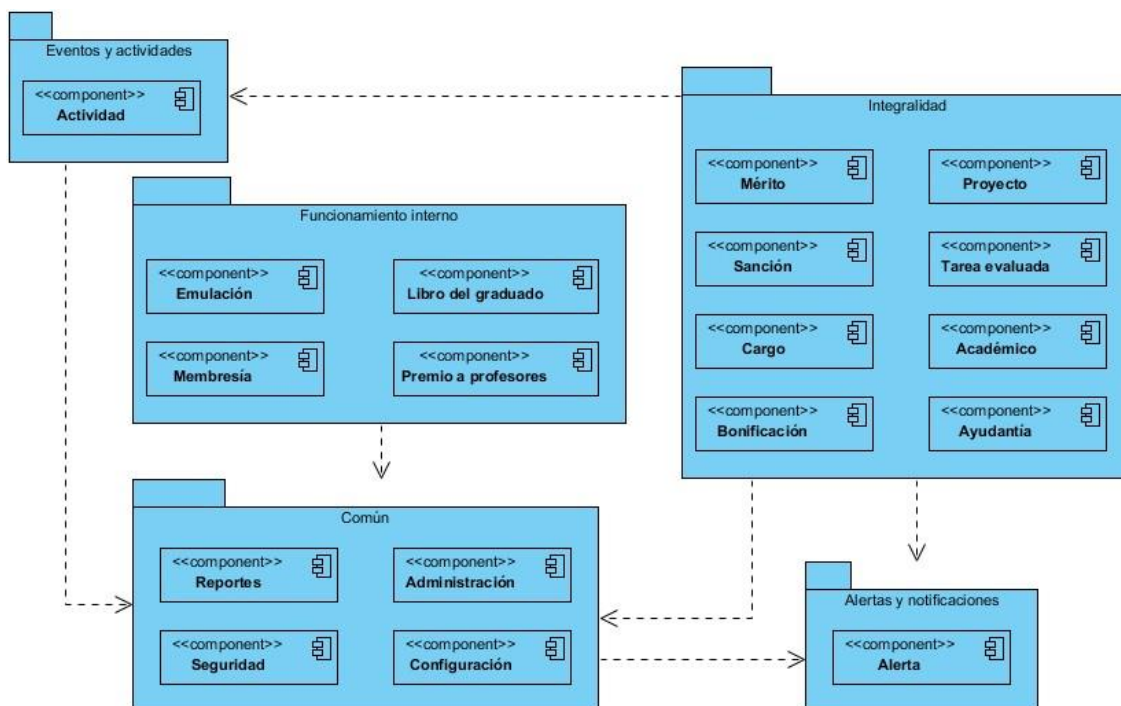
Sommerville (2005) (8) hace énfasis en que “... los grandes sistemas siempre se descomponen en subsistemas que proporcionan algún conjunto de servicios relacionados. El proceso de diseño inicial que identifica estos subsistemas y establece un marco para el control y comunicación de los subsistemas se llama diseño arquitectónico.”

### 2.6.1 SUBSISTEMAS QUE SE MODELAN

Cualquier aplicación que se desee desarrollar por muy pequeña que esta sea, siempre lleva algún grado de complejidad en su estructura. Por eso lo mejor es dividir el sistema en piezas lógicas que puedan conectarse a un todo. Esas piezas que a su vez son componentes, se describirán para que puedan abstraerse las características esenciales.

Teniendo en cuenta los requisitos definidos en el epígrafe 2.2, el software se divide en cinco subsistemas fundamentales, los cuales tienen como núcleo el subsistema

común. El mismo es base dentro del desarrollo del software, conteniendo aquellos módulos que deben ser utilizados por toda la aplicación.



*Imagen 6 División por paquetes*

A continuación se da una breve descripción de los subsistemas que componen el sistema desarrollado:

- Subsistema común: constituye el núcleo del software e integra lo indispensable para su funcionamiento. Está compuesto por los módulos de Seguridad, Administración, Reportes y Configuración.
- Subsistema integralidad: agrupa la mayoría de los módulos que tributan al proceso: módulo Méritos, Sanciones, Cargos, Bonificaciones, Alumnos Ayudantes, Académico, Proyecto y Tareas Evaluadas.
- Subsistema eventos y actividades: permite gestionar todo lo relacionado con los eventos, su planificación, participación de los estudiantes y evaluación. Lo integra un solo módulo de Actividades.
- Subsistema de funcionamiento interno: agrupa los componentes lógicos relacionados con esta área de la organización, divididos en los módulos de Emulación, Membresía, Libro del graduado y Premios a profesores.
- Subsistema alertas y notificaciones: es el encargado de gestionar todas las notificaciones y avisos tanto vía correo como dentro del propio sistema.

Se definen 18 módulos agrupados en los subsistemas por funcionalidades comunes. El de mayor peso y responsabilidad es el de integralidad. Cada uno de ellos es



independiente y puede ser actualizado sin comprometer el resto del sistema como un todo.

## 2.6.2 ESTILO ARQUITECTÓNICO

Somerville (8) describe tres estilos organizacionales ampliamente usados: modelo de repositorio, cliente – servidor y por capas.

Para la realización de la aplicación web se seleccionó la arquitectura cliente-servidor, consistente básicamente en un cliente que realiza peticiones un servidor, quien las responde. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes.

Dentro de las ventajas que ofrece este estilo están:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.

## 2.6.3 PATRONES DE DISEÑO

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón describe una estructura de diseño que resuelve un problema de diseño en particular dentro de un contexto específico y en medio de fuerzas que pueden tener un impacto en la manera en que se aplica y utiliza (19).

Symfony implementa dentro de su marco de trabajo el patrón MVC (24), en el cual todo el proceso está dividido en 3 capas. Típicamente estas capas son el modelo, la

vista y el controlador. MVC separa los conceptos de diseño, por lo que hay un decremento en la duplicación de código.

El Modelo incorpora la capa del dominio y persistencia. Es el encargado de guardar los datos en un medio persistente. Es el proveedor de los recursos, independiente al sistema de almacenamiento de datos.

La Vista se encarga de presentar la interfaz al usuario en sistemas web. En esta capa solo se deben hacer operaciones simples, condicionales, ciclos, formateo, etc. Recibe los datos del modelo y los muestra al usuario. Normalmente, tienen un registro del controlador asociado. En la propuesta de solución, estas vistas son instanciadas por el controlador para la presentación de los datos al usuario.

El Controlador es el que recoge los cambios en la vista y se los envía al modelo, el cual le regresa los resultados del envío.

Otro de los patrones utilizados son los GRASP, de los cuales se pueden destacar 5 principales (Larman, 1999):

- Experto: utilizado con el objetivo de darle a las clases las responsabilidades necesarias siempre que cuentan con la información para cumplirlas. Con esto se logra un mejor comportamiento haciendo que estas fueran más cohesivas y fáciles de comprender y mantener.

En la arquitectura de Symfony, en el modelo, existen 2 tipos de clases, las encargadas de la abstracción de la base de datos y las de acceso a datos. Este framework genera 3 clases por cada tabla de la base de datos, por ejemplo si se tiene una tabla llamada Estudiante, se obtendrían las clases: Estudiante, BaseEstudiante y EstudianteTable. Las clases de abstracción de los datos serían las de tipo "Base", que son las que trabajan directamente con la base de datos. Son las que tienen los atributos necesarios para obtener los registros de las tablas de la base de datos, en esa interacción directa es donde se pone de manifiesto el patrón experto.

- Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. En la solución propuesta fue utilizado el patrón en las clases del paquete de dominio, quienes son las encargadas de crear los objetos de tipo query, que permiten el acceso a la información almacenada a nivel de datos.

- Alta cohesión: es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.
- Bajo acoplamiento: es una medida de la fuerza con que una clase está conectada a otras, las conoce y recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras clases. Este patrón tiene como idea, tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto, potenciando la reutilización, y disminuyendo la dependencia entre las mismas.

Con el propósito de que las clases estuvieran relacionadas solo lo necesario unas con otras, se utilizó este patrón. Definiendo para cada una sus respectivos métodos y atributos con el fin de que el acoplamiento entre ellas fuera débil, para lograr la reutilización y el soporte. La funcionalidad de este patrón se aplica en todas las clases desarrolladas.

- Controlador: consiste en asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Este patrón se pone de manifiesto en las clases `sfFrontController`, `sfWebFrontController`, `sfContext`, los "actions" y el `index.php` del ambiente. La arquitectura del marco de trabajo (MVC) brinda una capa específicamente para los controladores, que son el núcleo del mismo y especifica la presencia de este patrón.

#### 2.6.4 MODELO DE DATOS

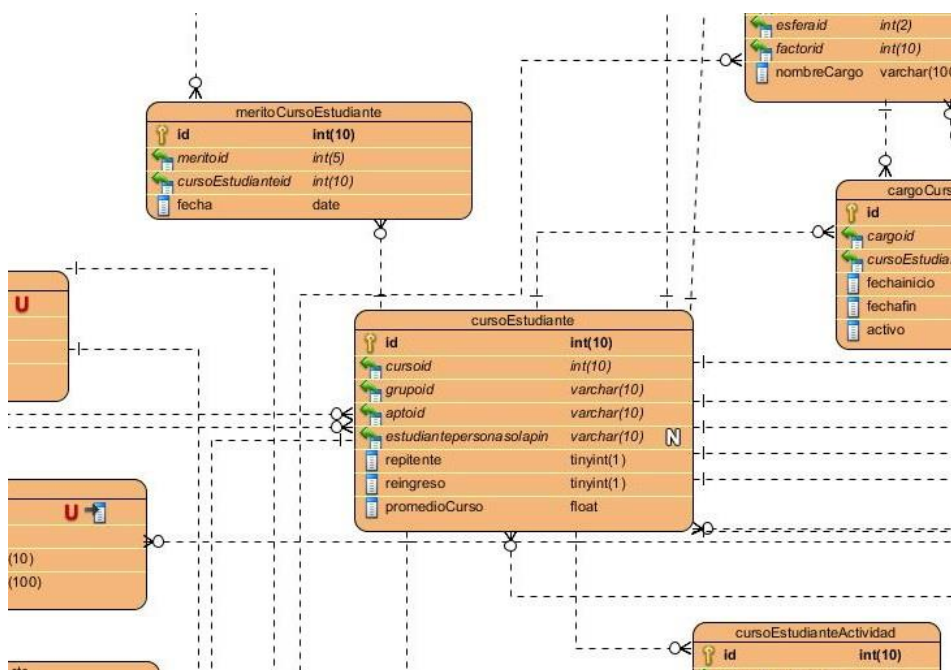
El modelo de datos describe de forma abstracta las entidades y sus características, además de las relaciones entre estas dentro de la base de datos. Las entidades son objetos que guardan información necesaria para el sistema. Su símbolo es un rectángulo. Los atributos son características de una entidad, se representan colocando su nombre dentro del rectángulo de la entidad. Los atributos se clasifican en: obligatorios, opcionales, claves foráneas y claves primarias (estas se dividen en simples y compuestas). Gráficamente la clave primaria se representa con una llave y

la foránea con una flecha en dirección izquierda. Las relaciones muestran la asociación entre dos entidades, representadas por una línea discontinua que une a las entidades involucradas.

El sistema cuenta con 118 tablas en un único esquema. Se tuvo en cuenta el patrón de diseño de base de datos: llave subrogada, a partir de la generación de una llave primara única para cada entidad en vez de usar un atributo identificador en el contexto dado. Se utilizó el tipo de dato entero en las llaves primarias, lo que permite que las tablas sean más fáciles de consultar por el identificador, dado que se conoce el mismo en cada una de ellas.

La notación utilizada para nombrar las tablas fue CamelCase, en esta ocasión el tipo que se utilizó es empleando la letra inicial de cada palabra es mayúscula excepto la primera.

La imagen 7, muestra un fragmento del modelo Entidad-Relación que muestra las relaciones entre las tablas. En el Anexo 4 se encuentra este íntegramente.



**Imagen 7** Tabla cursoestudiante del Modelo de Datos.

## 2.7 OBTENCIÓN DEL RANKING DE INTEGRALIDAD

El ranking de integralidad es una cuantificación de cada una de las evidencias que el estudiante va acumulando a lo largo del curso escolar en cuestión. Cada indicador tiene un peso en dependencia de las prioridades de la organización, los cuales pueden

ser modificados. Esto permite dar seguimiento detallado a los más rezagados y a aquellos que sobresalen.

Los anexos 5 y 6 muestran un ejemplo de los parámetros que se tienen en cuenta para el cálculo de dicho ranking.

## 2.8 CONCLUSIONES

A partir de los elementos estudiados en el capítulo se concluye que la Federación Estudiantil Universitaria en la Universidad de las Ciencias Informáticas lleva a cabo un gran número de procesos relacionados con su funcionamiento, los cuales se pueden agrupar en cinco subsistemas para ser informatizados en una aplicación web.

Se define utilizar para ello una arquitectura basada en componentes lo que permite mayor modularidad e independencia entre los subsistemas. Las técnicas de obtención de requisitos seleccionadas arrojaron 117 funcionales y 16 no funcionales, siendo los mismos descritos en el capítulo utilizando la metodología DAC para el desarrollo del software.

# CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

## 3.1 INTRODUCCIÓN

Durante el presente capítulo se realiza el diseño de la implementación del sistema propuesto. Además del estudio de los algoritmos necesarios para el cálculo de la evaluación estudiantil. Se verifica el cumplimiento de los requisitos a través de pruebas y se evalúa la calidad del producto desarrollado.

## 3.2 ESTÁNDARES DE CÓDIGO

A continuación se describe el estándar de código utilizado para el desarrollo del sistema dataFEU.

### 3.2.1 IDENTACIÓN, LLAVES DE APERTURA Y CIERRE, Y TAMAÑO DE LAS LÍNEAS

Usar una indentación de un tabulador con un equivalente a 4 espacios, para mantener integridad en las revisiones de Subversion. El uso de las llaves “{}” será en la misma línea y la longitud de las líneas de código es aproximadamente de 75-80 caracteres, para mantener la legibilidad del mismo.

### 3.2.2 CONVENCIÓN DE NOMENCLATURA

Variables: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra será con mayúscula.

Clases: se rigen por la nomenclatura Dromedarian y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Funciones: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Controladoras: mismo nombre de la clase que representa seguido de la palabra “Actions”.

### 3.2.3 ESTRUCTURAS DE CONTROL

Las estructuras de control incluyen if, for, foreach, while, switch. Entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Con esto se aumenta la legibilidad del código y se disminuye la probabilidad de errores lógicos.

Si las condiciones son muy largas que sobrepasan el tamaño de la línea, estas se dividen en varias líneas. En el mejor de los casos cuando la condición es muy extensa, se puede dividir en variables y compararlas dentro de la estructura de control.

### 3.2.4 DOCUMENTACIÓN

Todos los archivos deben de tener la documentación asociada al mismo. Para esto debe de cumplirse con la documentación de tipo comentario en aras de facilitar el trabajo a la hora del completamiento de código que brinda los entornos de desarrollo actuales.

### 3.3 INTEGRACIÓN CON SISTEMAS EXTERNOS

Para la carga inicial se hace necesario un conjunto de datos de los estudiantes en la Universidad los cuales son gestionados por el Sistema de Gestión Universitaria (SGU), implantado oficialmente en la institución.

Mediante servicios web esta aplicación brinda las siguientes informaciones, necesarias a la hora de insertar nuevos estudiantes a la Base de Datos:

- Usuario.
- Nombre y apellidos del estudiante.
- Situación académica.
- Grupo.
- Índice académico.
- Carné de identidad.
- Municipio y provincia.

La autenticación se realiza mediante un Protocolo Ligero de Acceso a Directorios (LDAP, por sus siglas en inglés) el cual almacena la información de usuario y contraseña de los actores del sistema.

### 3.4 SEGURIDAD Y TRAZAS

Las trazas constituyen el sistema nervioso de la administración para seguir la actividad de la aplicación, registrándose todo lo que se realiza por parte de los usuarios. Son almacenadas en la base de datos utilizando las facilidades que ofrece el marco de trabajo y el ORM Doctrine.

Los elementos que se auditan son los siguientes:

- Méritos otorgados.
- Bonificaciones recibidas.
- Cargos asignados a estudiantes.
- Bajas realizadas a dirigentes estudiantiles.
- Sanciones aplicadas.
- Asignaturas de arrastre y mundiales asignadas a estudiantes.
- Bajas realizadas a los alumnos ayudantes.
- Bajas académicas.
- Las personas que son insertadas en la aplicación como profesor, instructora o profesor tribunal.
- Los roles asignados a los usuarios dentro del sistema.
- Las actividades insertadas.
- Los participantes a cada una de las actividades.
- Las evaluaciones tanto mensual, por corte como por fechas de todas las áreas dentro de la aplicación.

La seguridad del sistema está estructurada por roles, los cuales tienen una relación directa con los cargos ocupados en los diferentes niveles de la organización.

### 3.5 SERVICIOS WEB

La implementación de los servicios web necesarios para la comunicación con aplicaciones externas se realizó utilizando el plugin de Symfony ckWebServicePlugin el cual se encuentra de forma gratuita en el sitio oficial de Symfony.

Los servicios públicos disponibles son:

- `ObtenerRankingdeIntegriadadporSolapinyCurso($solapin, $curso)`.
- `ObtenerEvaluacionIntegralidadporSolapinyCursi($solapin, $curso)`.
- `ObtenerCargosdeEstudiante($solapin, $curso)`



### 3.6 PRUEBAS

Como parte del proceso de desarrollo, la fase de pruebas añade valor al producto que se maneja, ya que el desarrollo de sistemas de software lleva implícito una serie de actividades en las cuáles, los fallos y los errores son frecuentes desde el momento inicial, de ahí que esta tiene como objetivo específico encontrarlos.

*“Las pruebas de software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y la codificación. Los procesos que verifican y revelan la calidad de un producto de software, son utilizados para identificar los posibles fallos de la implementación, calidad o la usabilidad de una aplicación informática” (Pressman, 2002) (13).*

Por otro lado Sommerville (2005) (8) sobre esta parte del proceso de desarrollo de software apunta: *“Dentro de los objetivos del proceso de pruebas están: demostrar al desarrollador y al cliente que el software satisface sus requisitos y para descubrir defectos en el software, que el comportamiento de éste es incorrecto, no deseable o no cumple su especificación. El primer objetivo conduce a las pruebas de validación, en las que se espera que el sistema funcione correctamente usando un conjunto determinado de casos de prueba que reflejan el uso esperado de aquel. El segundo objetivo conduce a la prueba de defectos, en los que los casos de prueba se diseñan para exponer los defectos. Los casos de prueba pueden ser deliberadamente oscuros y no necesitan reflejar cómo se utiliza normalmente el sistema.”*

A partir de los criterios anteriores se puede concluir que el objetivo de las pruebas es detectar un error no descubierto hasta entonces, y estas tienen éxito si lo descubren.

El software debe probarse desde dos perspectivas diferentes:

- Lógica interna del programa: se comprueba utilizando técnicas de diseño de casos de pruebas de “caja blanca”.
- Los requisitos del software: se comprueban utilizando técnicas de diseño de casos de prueba de “caja negra”.

En ambos casos se intenta encontrar el mayor número de errores con la menor cantidad de esfuerzo y tiempo.

#### 3.6.1 NIVELES DE PRUEBAS

A decir de Rumbaugh (24) el proceso de realización de las pruebas está compuesto por una serie de niveles entre los que se encuentran: pruebas unitarias, pruebas de integración, pruebas del sistema y pruebas de aceptación.

Una vez implementado el sistema fue sometido a los niveles de prueba que a continuación se detallan, los cuales ayudaron a la detección de los errores:

- Pruebas unitarias: se concentran en la lógica del procesamiento interno y en las estructuras de datos de los límites de un componente. Se le aplican a cada módulo de un software de manera independiente. El objetivo de estas es aislar cada segmento del programa y mostrar que las partes individuales son correctas.
- Pruebas del sistema: tienen como propósito fundamental ejercitar profundamente el sistema desarrollado, con el objetivo de verificar que se hayan integrado todos los elementos del mismo y que realizan correctamente las funciones descritas. Este tipo de pruebas estudia el producto completo para analizar defectos globales o para estudiar aspectos específicos de su comportamiento, tales como seguridad y rendimiento.

### 3.6.2 TIPOS DE PRUEBAS

#### Caja negra:

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. La misma permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores. La prueba de caja negra intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación (11).

Como resultado de las pruebas de caja negra se realizó un caso de prueba por cada requisito (tomando los diseños basados en requisitos), en los cuales se detectaron varias no conformidades en revisiones internas realizadas por el equipo de desarrollo, las cuales fueron resueltas.

#### Caja blanca:

Las pruebas de caja blanca, denominadas a veces prueba de caja de cristal es el método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante este tipo, el ingeniero del software puede obtener los casos de prueba que garanticen que se ejercita por lo

menos una vez todos los caminos independientes de cada módulo; se ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; se ejecuten todos los ciclos en sus límites y con sus límites operacionales, y se ejerciten las estructuras internas de datos para asegurar su validez.

La prueba de caja blanca empleada en la solución fue la prueba del camino básico, a partir del cálculo de la complejidad ciclomática del algoritmo a ser analizado. Para realizarla se deben enumerar las sentencias de código y a partir de ahí elaborar el grafo de flujo de esta funcionalidad.

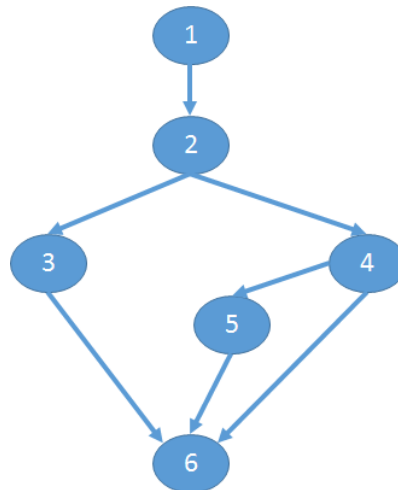
Se definieron una serie de pasos a seguir:

1. Notación del grafo de flujo: usando el código como base se realiza la representación del grafo de flujo, mediante una sencilla notación. Cada construcción estructurada tiene su correspondiente símbolo.
  - Nodo: cada círculo denominado nodo, representa una o más sentencias procedimentales.
  - Arista: las flechas del grafo de flujo, denominadas aristas, representan el flujo de control y son análogas a las flechas del diagrama de flujo.
  - Región: las áreas delimitadas por aristas y nodos se denominan regiones.
2. Complejidad ciclomática: es una métrica que proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado define el número de caminos independientes del conjunto básico de un programa. Esto indica el límite superior para el número de pruebas que se deben realizar, para asegurar que se ejecuta cada sentencia al menos una vez. Se utilizó la siguiente forma:  $V(G)$ , de un grafo de flujo  $G$  se define como:  $V(G) = A - N + 2$ , donde  $A$  es el número de aristas del grafo de flujo y  $N$  es el número de nodos.
3. Determinar un conjunto básico de caminos linealmente independientes: el valor de  $V(G)$  es el número de caminos linealmente independientes de la estructura de control del programa.
4. Obtención de casos de prueba: se realizan los casos de pruebas que forzarán la ejecución de cada camino del conjunto básico.

La figura del Anexo 3 muestra el código que se tomó como base y ejemplo para realizar el procedimiento anteriormente descrito para la técnica de camino básico,

correspondiente al método `executeNuevaInstanciaActividad(sfWebRequest $request)` del controlador.

En la imagen 8 se muestra el grafo de flujo asociado al requisito funcional seleccionado:



*Imagen 8 Grafo de flujo del método nueva instancia de actividad.*

Luego se calcula la complejidad ciclomática:

$$1. V(G) = (A - N) + 2, V(G) = (6 - 5) + 2, V(G) = 3$$

Siendo A la cantidad total de aristas del grafo y N la cantidad de nodos.

A partir de los resultados obtenidos en cada uno, se determina que la complejidad ciclomática del código analizado es 3, que a su vez es el número de caminos posibles a circular el flujo y el límite superior de casos de prueba que se le pueden aplicar a dicho código.

A continuación se muestran los caminos básicos por donde puede circular el flujo:

<b>NÚMERO</b>	<b>CAMINOS BÁSICOS</b>
<b>1</b>	1, 2, 3, 6
<b>2</b>	1, 2, 4, 5, 6
<b>3</b>	1, 2, 4, 6

Cada camino independiente es un caso de prueba a realizar, de forma que los datos introducidos provoquen que se visiten las sentencias vinculadas a cada nodo del camino.

### 3.6.3 DISEÑO DE LOS CASOS DE PRUEBAS

A continuación se presenta en una tabla el diseño de caso de prueba para los requisitos que fueron descritos en el epígrafe 2.7. El resto de los casos de pruebas se encuentran en el expediente de proyecto.

Escenario	Descripción	Nombre actividad.	Respuesta del sistema	Flujo central
<i>RF 82 Registrar nueva instancia de actividad.</i>	<i>A partir de las actividades genéricas que se definen en el nomenclador, se crean instancias de las actividades las cuales contemplan una serie de datos para ubicarlas en tiempo y espacio, además de definirse cómo se realizará la participación.</i>	<i>“Marcha por el 1ero de Mayo” N/A</i>	<i>Se actualiza el listado de los nomencladores de actividades.</i>	<i>Una vez abierta la interfaz para registrar nomencladores se da clic en el botón de agregar (+). En el cuadro que aparece introducir el texto deseado (Variable) y por último clic en aceptar.</i>
<i>RF 85 Insertar participación de estudiante en actividad.</i>	<i>Añade los estudiantes que participaron en alguna de las instancias declaradas de una actividad en específico.</i>	<i>“Estudiante 1” Selección múltiple.</i>	<i>Mueve el estudiante seleccionado hacia la columna de la derecha y lo elimina de la columna izquierda. En la columna derecha se actualiza el tipo de participación.</i>	<i>En dependencia de los niveles de acceso se selecciona la facultad y el grupo. En la columna de la izquierda se listarán todos los estudiantes asociados al grupo docente en cuestión. Se marca a los que se desea actualizarle la participación y se arrastran hacia la columna de la derecha. Cuando se sueltan los elementos seleccionados en la columna izquierda, aparece una ventana emergente para introducir los términos de participación, los cuales una vez añadidos se da clic en aceptar.</i>

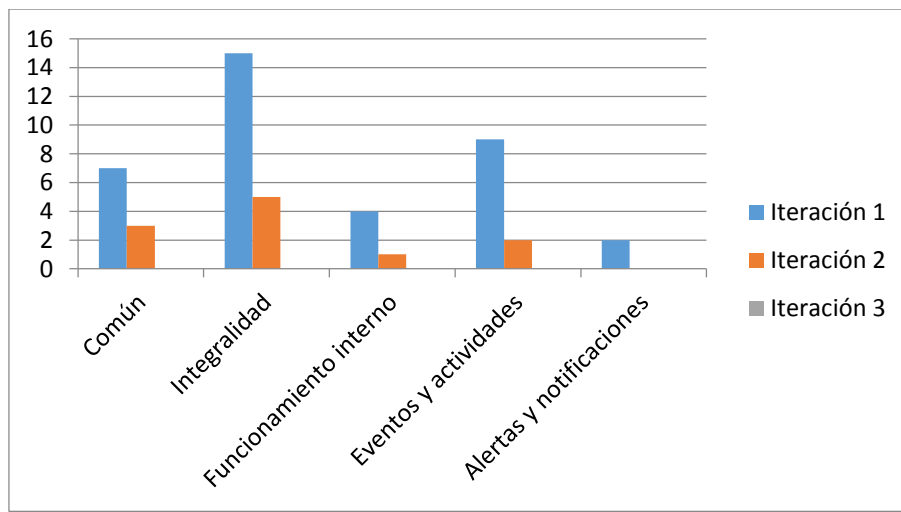
### 3.6.4 IMPLEMENTACIÓN DE LAS PRUEBAS

Durante el desarrollo de las funcionalidades se realizaron pruebas unitarias al código para ir comprobando el funcionamiento del software. Estas pruebas fueron realizadas por el propio desarrollador, apoyándose en la compilación paso a paso que brinda el IDE empleado. Estas no se planificaron ni se registraron sus resultados, fueron haciéndose a medida que la solución se desarrollaba.

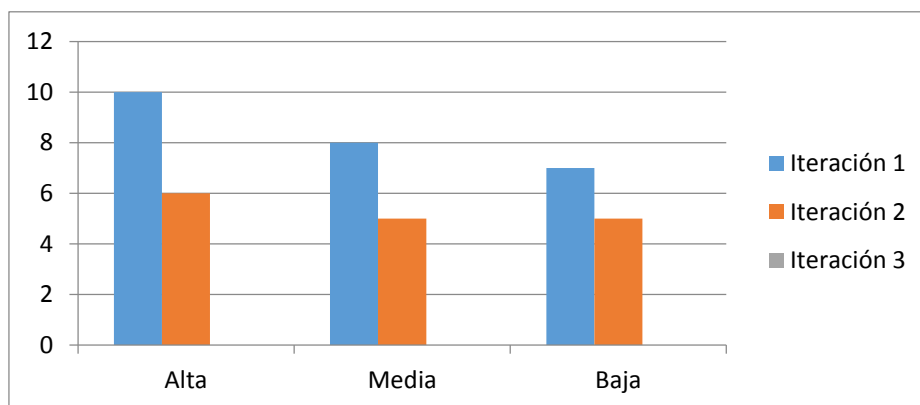
A través del método de caja negra, y apoyados en el diseño de los Casos de Pruebas explicados en el epígrafe anterior, se realizaron 2 iteraciones de pruebas internas pertenecientes al nivel del sistema. Dichas pruebas fueron realizadas con el objetivo de detectar y corregir errores que impidieran el correcto funcionamiento de la solución. A continuación se presentan los resultados arrojados durante las diferentes pruebas aplicadas:

- Pruebas internas: Realizadas por el equipo de desarrollo con el fin de entregar un producto con la menor cantidad de errores posibles. Se centraron en el cumplimiento de las funcionalidades descritas en el listado de requisitos y de casos de uso del sistema, además, se revisó toda la documentación.
- Pruebas cruzadas: Realizadas también por el equipo de desarrollo al sistema con el fin de encontrar la mayor cantidad de errores posibles en términos de validaciones, pautas definidas por arquitectura de información, formato de los campos, entre otras.

Pruebas de caja negra a nivel de unidad:



Pruebas de caja negra a nivel de sistema:



Las no conformidades más comunes encontradas estuvieron relacionadas con el mal funcionamiento de las interfaces, comportamiento de objetos de forma no deseada y mensajes de información no personalizados.

### 3.7 MODELO DE DESPLIEGUE

“En todo sistema de software es necesario describir la distribución física de los elementos que lo componen, mostrando la configuración del hardware en la que se desplegará el sistema, identificando los nodos, que pueden ser procesadores o dispositivos de hardware” (32).

Las relaciones físicas finales entre los componentes de hardware y software son representados a través de nodos, los cuales serían una computadora (PC), los servidores donde está alojada la aplicación y aquellos que brindan los servicios necesarios para el funcionamiento del sistema. La conexión de la PC cliente puede ser a través de la red cableada o inalámbrica. En este caso la aplicación se encuentra hospedada en un servidor web mientras que la base de datos (MySQL) se encuentra en otro servidor, la comunicación entre ambos se realiza mediante servicios.

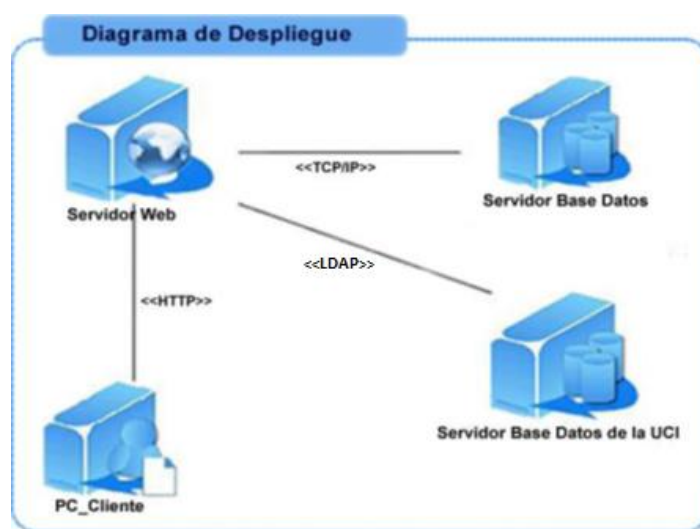


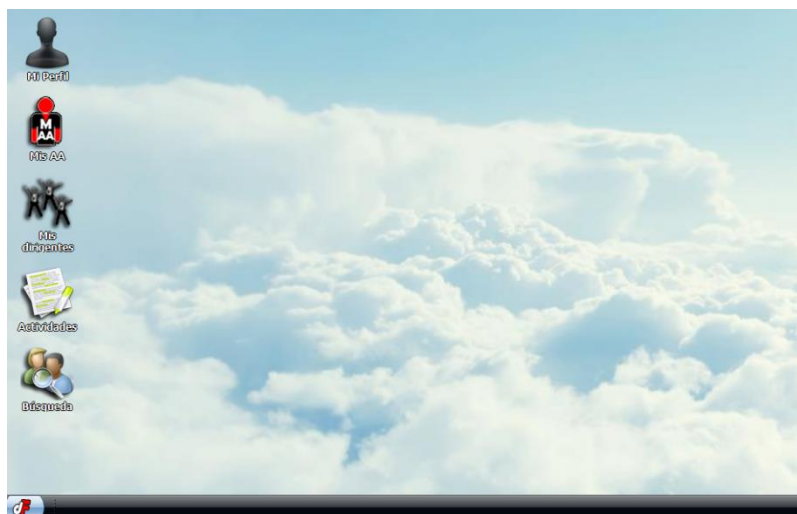
Imagen 9 Diagrama de Despliegue

### 3.8 IMPLANTACIÓN DEL SISTEMA

Concluido el desarrollo del software, fue implantado en todas las facultades de la UCI en el curso 2012-2013.

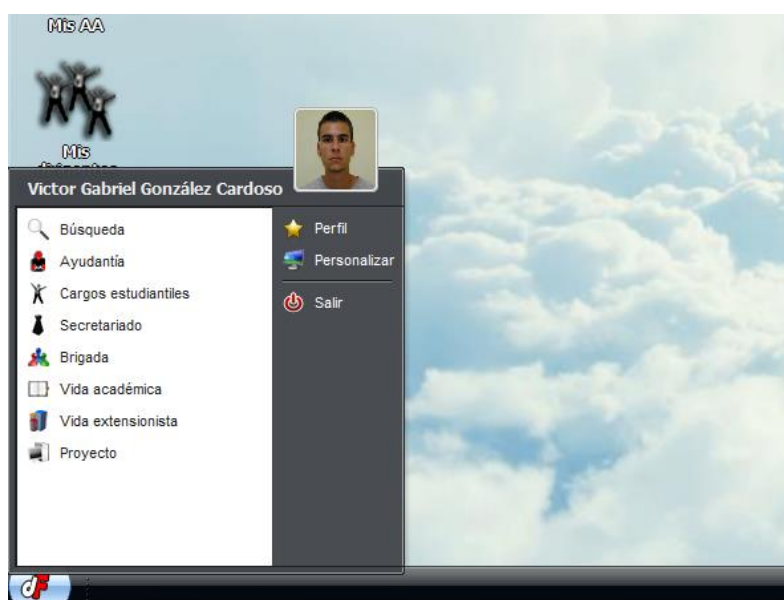
El ambiente de la aplicación semeja al escritorio de un Sistema Operativo, con el fin de hacerlo más familiar e intuitivo para los usuarios que interactúan con la misma y

teniendo en cuenta los requisitos no funcionales, ya que un porcentaje importante de los usuarios finales no son de la rama de la informática.



**Imagen 10 Ambiente de escritorio**

Sus elementos principales pueden ser configurados por el usuario, en la sección Personalizar. Los íconos en el escritorio dan la posibilidad de brindar informaciones importantes de conocimiento de todos los estudiantes como: las próximas actividades, los dirigentes estudiantiles a los diferentes niveles y los alumnos ayudantes asignados a cada una de las brigadas.



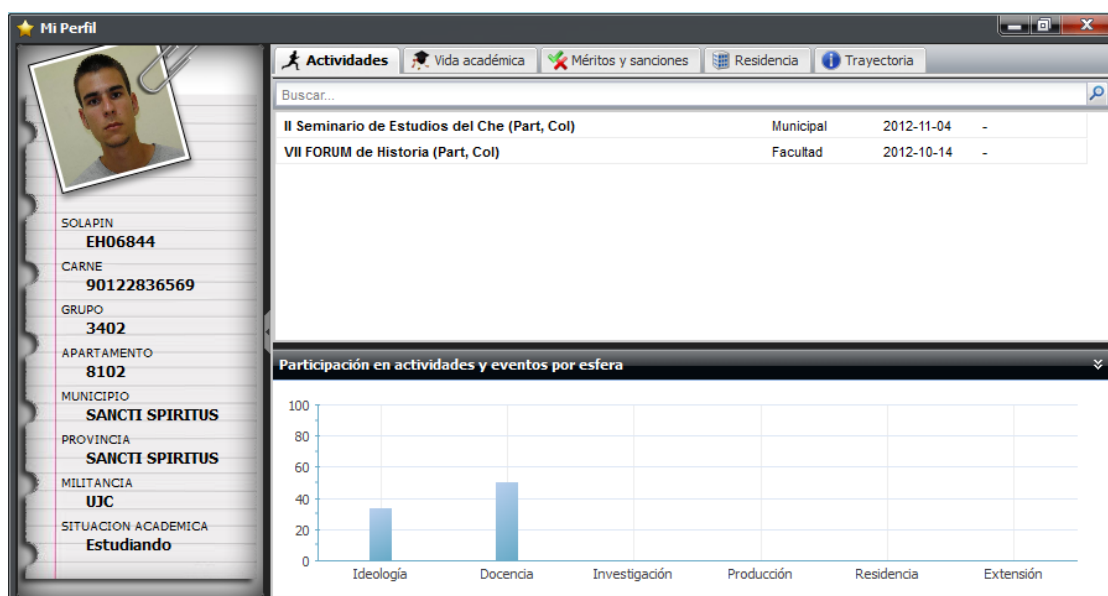
**Imagen 11 Menú inicio**

A través del menú inicio (*Imagen 11*) se puede acceder a los módulos de gestión, definidos en la arquitectura del sistema. La aplicación tiene una vista de administración la cual está reservada para las configuraciones del sistema.



Toda la información generada por las diferentes áreas de la vida universitaria es concentrada en un expediente digital (imagen 12), donde se muestra el accionar de cada estudiante, recopilando evidencias que le servirán para el proceso final de integralidad, caracterización y de ubicación laboral en 5to año de la carrera. La información que se muestra se encuentra en un único formato, lo que permite un procesamiento eficiente.

El sistema permite interactuar con el escritorio de la aplicación, brindándoles la posibilidad a los usuarios de cambiar su fondo de pantalla o íconos del escritorio en dependencia del gusto y confort.



*Imagen 12 Sección donde se registra la participación y los resultados en las actividades convocadas.*

El acceso a las funcionalidades se realiza en bloques ordenados lógicamente por módulos como muestra la imagen 13, donde en cada una de ellas se muestra en una columna una breve descripción de su función. Esto ayuda a la organización del sistema y permite el acceso por roles de usuarios.



*Imagen 13 Módulo Ayudantía.*

### 3.9 VALIDACIÓN DE LAS VARIABLES DEL PROBLEMA DE LA INVESTIGACIÓN

Antes de la implantación del sistema, el proceso de evaluación estudiantil (integralidad y caracterización) se realizaba durante todo un mes teniendo en cuenta los cronogramas que se registran en las actas de los Consejos de la FEU de la Universidad y los acuerdos de los Consejos de Dirección de la Universidad. A partir de la introducción de la herramienta informática dataFEU, este proceso se logró completar en dos semanas, evidenciándose una reducción del tiempo en el procesamiento de la información generada.

Al existir una única plataforma para introducir los datos aportados por las áreas de la vida universitaria se eliminó en un 100% el intercambio de información en archivos de diferentes formatos de las áreas que aportaban información a la organización. También es válido señalar que como los principales indicadores se encuentran listados en el sistema informático, se redujo la entrada de datos con variables diferentes unificándose la forma de gestionar la información.

La creación de un archivo histórico permitió que el grado de persistencia de la información se elevara en un 100%, ya que se almacenaron la totalidad de los perfiles de los estudiantes, situación que no se daba antes de la creación del sistema debido a pérdida de los datos físicos que se generaban como resultado del proceso.

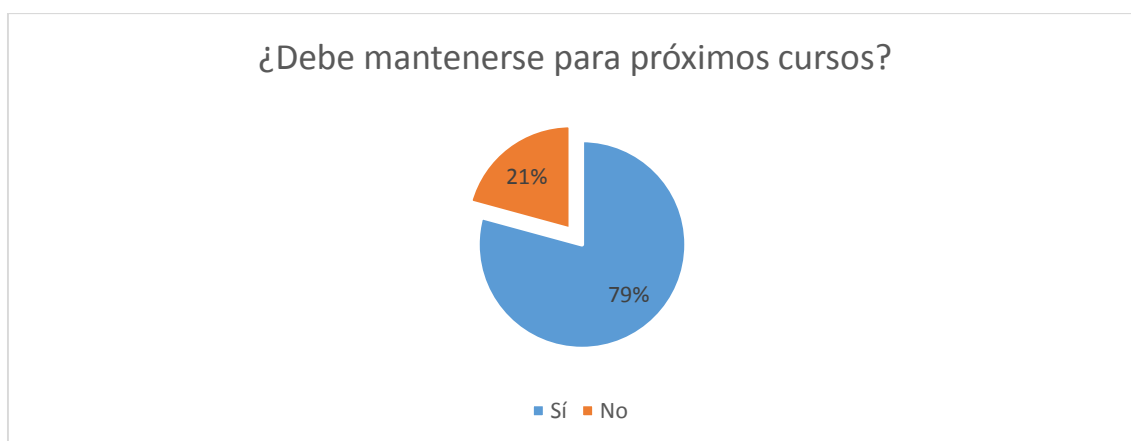
### 3.10 ENCUESTA PARA MEDIR NIVEL DE SATISFACCIÓN DE LOS USUARIOS

A finales de junio del año 2013 se aplicó a través del sitio web <http://encuestas.uci.cu> una encuesta anónima (Anexo 1), donde se realizaron una serie de preguntas a los usuarios que interactúan con el sistema dataFEU. Los resultados (Anexo 2) fueron favorables a la utilización del sistema, haciendo énfasis en que logra una mejor gestión y organización de la información. Los usuarios recomendaron orientarla a una red

social donde se pudieran compartir criterios y estados de opinión respecto a los procesos fundamentales que lleva la organización.

A la pregunta: “¿Debe mantenerse [dataFEU] para próximos cursos?”, el 79% de los encuestados opinó que sí. Del 21% que opina que no: el 69,7% declara que entra casi nunca o nunca al sistema.

Esto evidencia una confianza en el desarrollo del sistema como herramienta de apoyo a la toma de decisiones para los dirigentes estudiantiles.



*Imagen 14 Pregunta de la sobre el sistema dataFEU*

### 3.11 CONCLUSIONES

Durante el desarrollo de este capítulo se mostraron los resultados de la implementación. Los diferentes tipos de pruebas realizadas arrojaron que el sistema funciona correctamente y de acuerdo a los requisitos funcionales definidos en la etapa inicial del desarrollo de software. Se realizaron dos iteraciones, con las cuales se detectaron varias no conformidades, siendo resueltas por el equipo de desarrollo.

Se describe la implantación del sistema obteniéndose como resultado: la reducción del tiempo en el procesamiento de la información, la estandarización de la información que aporta cada área en relación con la FEU y la persistencia de los datos en un archivo histórico.

Se realizó una encuesta a los usuarios de la aplicación, demostrándose que el nivel de satisfacción es alto a favor de la utilización de la aplicación.

## CONCLUSIONES GENERALES

A través del estudio de esta investigación se ha podido arribar a las siguientes conclusiones:

1. El estudio del estado del arte realizado demostró que no se conoce a nivel nacional un sistema para la gestión de los procesos de la Federación Estudiantil Universitaria que solventen las nuevas situaciones.
2. El análisis de un conjunto de herramientas y tecnologías permitió definir el marco tecnológico en ajuste a las políticas de migración a software libre por la cual aboga el país.
3. Se demuestra que el desarrollo de la aplicación permitió reducir el tiempo de procesamiento de la información, estandarizar la documentación y elevar la persistencia de los datos del proceso de evaluación estudiantil.

## RECOMENDACIONES

Luego de haber arribado a las conclusiones generales de esta investigación se recomienda generalizar el sistema en todas las universidades o sedes del país.

Atendiendo a los resultados de la encuesta del Anexo 2, se sugiere integrar futuras versiones a una red social, creando una plataforma más interactiva y dinámica para el usuario.

## BIBLIOGRAFÍA REFERENCIADA

1. *Lineamientos de la política económica y social del Partido y la Revolución*. La Habana : Editora política, 2011, Vol. II.
2. Colectivo de autores. *ABC de la FEU*. La Habana, Cuba : Pueblo y educación, 2006.
3. Juran, J.M. *Manual de calidad*. Madrid : A Blanton, Mc Graw Hill, 2001.
4. Oteo, Luis Ángel. *Gestión clínica: gobierno clínico*. México : Díaz de Santos, 2006.
5. *Biografía de John Wilder Tukey*. O'Connor, John J. y Robertson, Edmund F. Scotland : JOC/EFR, Junio, 2004.
6. *IEEE Software Engineering Standard: Glossary of Software Engineering Terminology*. Std, IEEE. s.l. : IEEE Computer Society Press, 1993.
7. *Herramientas de software libre para la gestión de contenidos*. Tramullas, Jesús. 3, s.l. : Universidad de Zaragoza, 2005.
8. Conallen Wae, Jim. *Modeling Web Application Architectures with UML*. 1999.
9. Maxim Richar J, *Modeling Web Application Architectures with UML*. 2002.
10. Mullenweg, Matt. Free Software Foundation. [En línea] 2004. [Citado el: 2014 de 05 de 27.] [www.fsf.org](http://www.fsf.org).
11. Pressman, Roger S. *Ingeniería de Software, un enfoque práctico*. Quinta edición. s.l. : McGraw-Hill Companies, 2002.
12. Sánchez Méndez, Alelí. *Proceso de desarrollo de software DAC*. Departamento de Desarrollo de la Dirección de Informatización de la Universidad de las Ciencias Informáticas. La Habana : s.n., 2013. pág. 88.
13. Peláez, Juan. Blog de Juan Peláez. [En línea] [Citado el: 29 de 1 de 2014.] <http://www.juanpelaez.com/geek-stuff/arquitectura/arquitectura-basada-en-componentes/>.
14. Symfony. [En línea] [Citado el: 29 de 1 de 2014.] <http://symfony.es/que-es-symfony>.
15. Comunidad de Doctrine. Doctrine Project. [En línea] [Citado el: 29 de 1 de 2014.] <http://www.doctrine-project.org/>.
16. Sencha Corp. JavaScript Framework. [En línea] [Citado el: 29 de 1 de 2014.] <https://www.sencha.com/products/extjs/>.

17. MINGUET, Laurent. HTML2PDF. [En línea] [Citado el: 29 de 1 de 2014.] <http://html2pdf.fr/es/default>.
18. Colectivo de autores. HTML 5. [En línea] [Citado el: 15 de 5 de 2014.] <http://www.w3.org/TR/html5/>.
19. W3Schools. [En línea] [Citado el: 15 de 5 de 2014.] <http://www.w3schools.com/about/default.asp>.
20. Netbeans.org. [En línea] [Citado el: 15 de 05 de 2014.] <https://netbeans.org/>.
21. Subversion. [En línea] [Citado el: 15 de 05 de 2014.] <http://subversion.apache.org/>.
22. MySQL. [En línea] [Citado el: 15 de 4 de 2014.] <http://www.mysql.com/>.
23. García Rodríguez, Yanet y González González, Yoandrys. *Modelado de una aplicación web para la gestión de las tareas de funcionamiento de la FEU en la UCI*. Cuba : s.n., 2007.
24. Couso Linares, Arian Abel. *Análisis y diseño de la gestión de eventos culturales de la FEU en la UCI*. La Habana, Cuba : s.n., 2007.
25. Suárez Blanco, Yoan y Hernández Calvo, Osniel. *Análisis y diseño del sistema de gestión de eventos de la FEU en la UCI*. Cuba : s.n., Junio, 2008.
26. Castro Carrillo, Yaremi. *Integración de la Plataforma para la Gestión de Eventos Científicos con el Sistema de Gestión Universitaria y Repositorio Institucional de la Universidad de las Ciencias Informáticas*. La Habana, Cuba : s.n., 2012.
27. de la Torre Quintana, Iana y Cruz González, Michel. *Análisis y diseño del sistema de gestión de inquietudes de la FEU UCI*. Cuba : s.n., 2008.
28. Basso Mesa, Anileidy y Gómez Almaguer, Dayaima. *Sistema automatizado para el proceso de caracterización de los estudiantes*. Ciudad de la Habana : s.n., 2009.
29. González Fernández, Yisel y Benítez Campo, Yanitza. *Sistema de apoyo al ingreso, seguimiento y control del movimiento de alumnos ayudantes en la facultad 1*. La Habana : s.n., 2010.
30. Hinojosa Domínguez, Yusdani y Hurtados Díaz, Rangel. *Sistema de Gestión del Movimiento de Alumnos Ayudantes*. La Habana : s.n., 2010.
31. *Sistema integral de Gestión Universitaria*. Colectivo de autores. La Habana : s.n., 2012.

32. Sommerville, Ian. *Ingeniería de Software Séptima edición*. Madrid : Pearson Education SA, 2005. 84-7829-0745.
33. Potencier, Fabien y Zaninotto, François . *Symfony 1.4, la guía definitiva*. 2009.
34. Rumbaugh, James, Jacobson, Ivar y Booch, Grady. *El Proceso Unificado de Desarrollo de Software. Manual de referencia*. s.l. : S.I: Addison Wesley.
35. Borges Ramírez, Ygraine y Saura Guerra, José Enrique. *Kainos: Sistema de gestión nacional de la FEU*. Cuba : s.n., Junio, 2007.
36. Beck, Kent, y otros. Scrum Manager, Body of Knowledge. [En línea] 13 de agosto de 2013. [Citado el: 28 de enero de 2014.] [http://www.scrummanager.net/bok/index.php?title=Manifiesto\\_%C3%A1gil](http://www.scrummanager.net/bok/index.php?title=Manifiesto_%C3%A1gil).
37. ArPUG. ArPUG Grupo de usuarios PostgreSQL de Argentina. [En línea] 6 de enero de 2013. [Citado el: 28 de enero de 2014.] <http://www.arpug.com.ar/trac/wiki/PgAdmin>.
38. Pentaho. Pentaho. [En línea] 2013. [Citado el: 29 de enero de 2014.] <http://www.pentaho.com/product/data-integration>.
39. Hernández, L. y otros. *El paradigma cuantitativo de la investigación científica*. Ciudad de la Habana, Cuba : Editorial Universitaria (Eduniv), 2005.
40. Hernández Sampieri, R y otros. *Metodología de la investigación. Segunda edición*. 1998.



## ANEXOS

### ANEXO 1: Encuesta realizada en mayo-junio del 2013 sobre el sistema dataFEU

Estimado estudiante:

A fin de mejorar el servicio que brinda la herramienta dataFEU ponemos a tu disposición una encuesta anónima para conocer tus opiniones y criterios. Nos será de gran ayuda y utilidad contar con tu transparencia y claridad en las respuestas.

1. ¿De qué Facultad eres?

- a. Facultad 1.
- b. Facultad 2.
- c. Facultad 3.
- d. Facultad 4.
- e. Facultad 5.
- f. Facultad 6.
- g. Facultad 7.

Única selección.

2. ¿Qué año cursas?

- a. Primero.
- b. Segundo.
- c. Tercero.
- d. Cuarto.
- e. Quinto.

Única selección.

3. ¿Eres dirigente estudiantil?

- a. Sí.
- b. No.

4. ¿Con qué frecuencia entras al sistema?

- a. Todos los días.
- b. Una vez por semana.
- c. Regularmente.
- d. Casi nunca.
- e. Nunca.

Única selección.

5. ¿Cómo ves a dataFEU?

	Sí.	No.	Tengo mis dudas.
Ayuda al control y seguimiento de la trayectoria estudiantil.			
Es un mecanismo para obligarte a ir a las actividades.			
Permite evaluar cómo marchó durante el curso, además de fijarme metas altas.			
El perfil de los estudiantes debe estar público para que todos lo vean.			
Los profesores deben tener acceso para poder seguir la evolución integral del estudiante.			
El sistema debe darme la evaluación a partir			

de las evidencias que tiene registradas.			
Debe usarse en el proceso de caracterización de 5to año para la ubicación laboral.			
Todas las actividades por mínimas que sean deben registrarse sin excepción.			
Es un sistema confiable y veraz en la información.			

6. ¿Quién crees deba actualizar tu perfil?
- Yo mismo.
  - Presidente de brigada.
  - Profesor guía.
  - Miembro del Secretariado de facultad o universidad.

Única selección.

7. ¿Qué funcionalidades crees que debería brindar dataFEU?

Respuesta abierta.

8. De los criterios de medida que se muestran debajo, seleccione un valor de 1 (Baja/Mala) a 5 (Alta/Excelente)

	1	2	3	4	5
Interfaz gráfica.					
Estructuración de las áreas de contenido. (¿Pasas trabajo para encontrar la información?)					
Velocidad en la navegación.					
Aviso y notificaciones vía correo.					
Usabilidad					
Complejidad (¿Es fácil de utilizar?).					

9. ¿Crees que el sistema debe ser más interactivo, enfocado a una red social donde puedas emitir criterios sobre tus compañeros, actividades y demás procesos de la organización y su funcionamiento?
- Sí.
  - No.
  - Perdería la esencia.
  - No sé, tengo mis dudas.

Única selección.

10. ¿Debe mantenerse para próximos cursos?
- Sí.
  - No. (En caso de seleccionar no, poner la pregunta: ¿Por qué?).
11. alguna sugerencia o criterio que quieras transmitir.

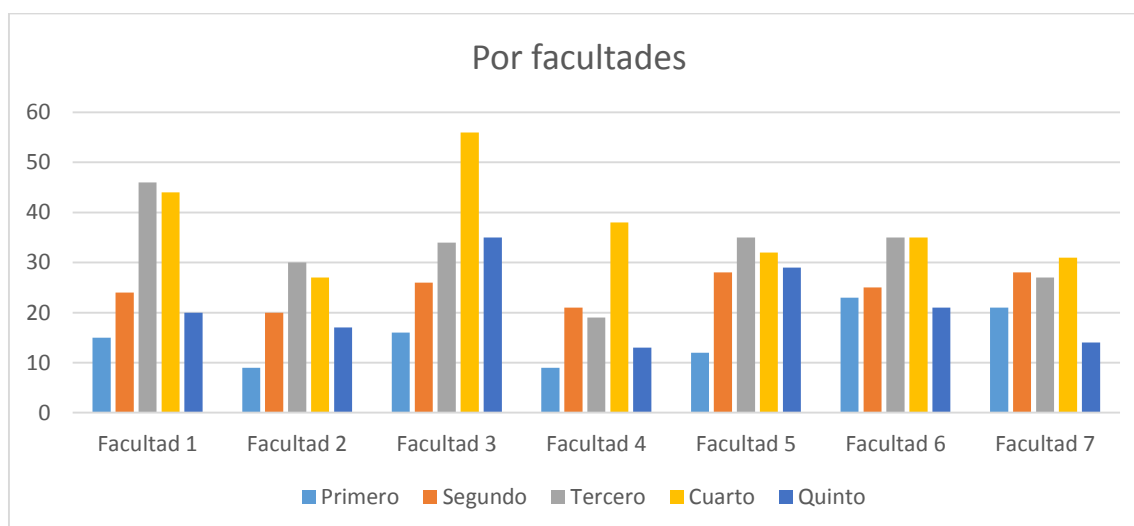
Respuesta abierta.

Muchas gracias por su participación.

## ANEXO 2: Resultados de la encuesta (Anexo 1)

Participaron en la realización un total de 915 estudiantes de todos los años y las facultades lo que representa aproximadamente un 20% del total de la matrícula.

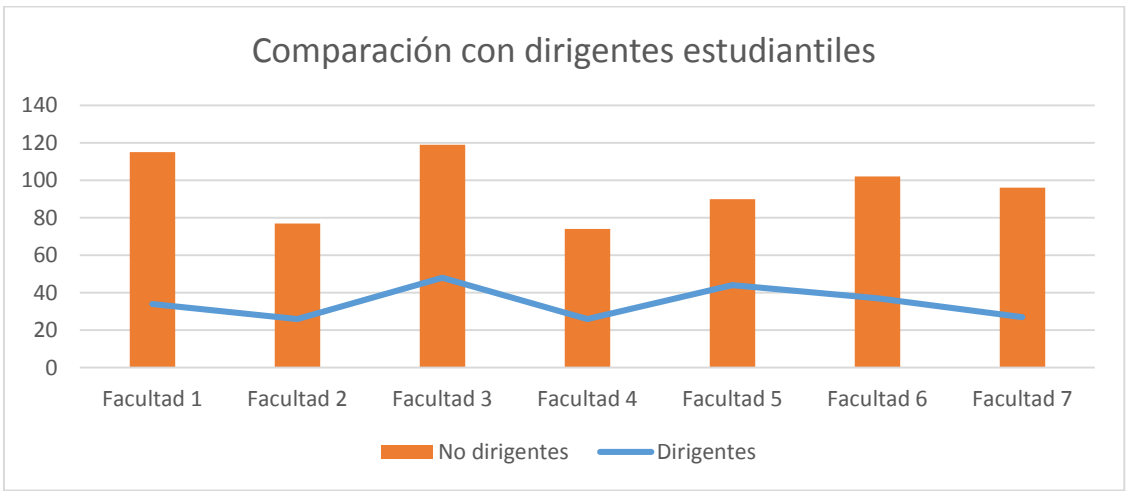
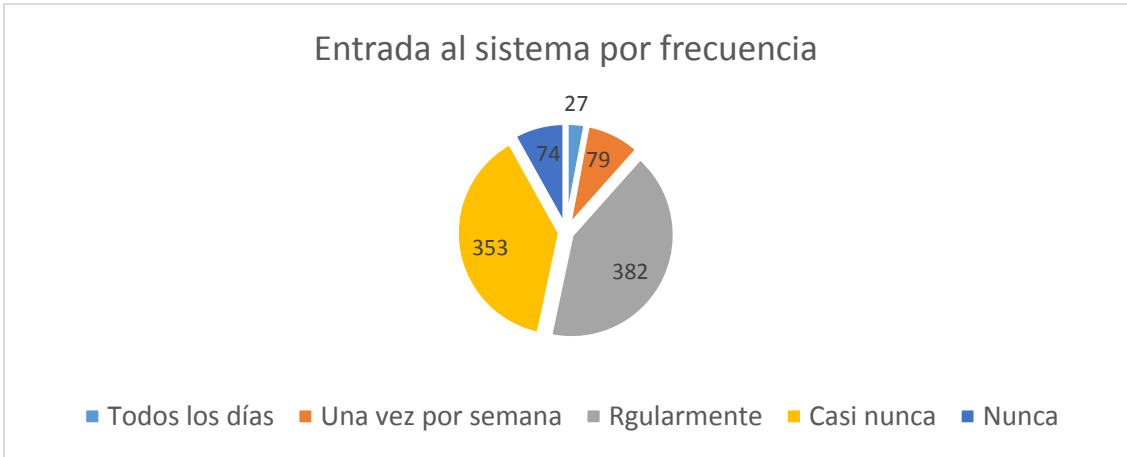
	Primero	Segundo	Tercero	Cuarto	Quinto	TOTAL
<b>Facultad 1</b>	15	24	46	44	20	149
<b>Facultad 2</b>	9	20	30	27	17	103
<b>Facultad 3</b>	16	26	34	56	35	167
<b>Facultad 4</b>	9	21	19	38	13	100
<b>Facultad 5</b>	12	28	35	32	29	136
<b>Facultad 6</b>	23	25	35	35	21	139
<b>Facultad 7</b>	21	28	27	31	14	121
<b>TOTAL</b>	105	172	226	263	149	915



Las facultades que mayor número de participantes tiene en la encuesta corresponden a la 3 y 1 mientras que los años son 4 y 3ero.

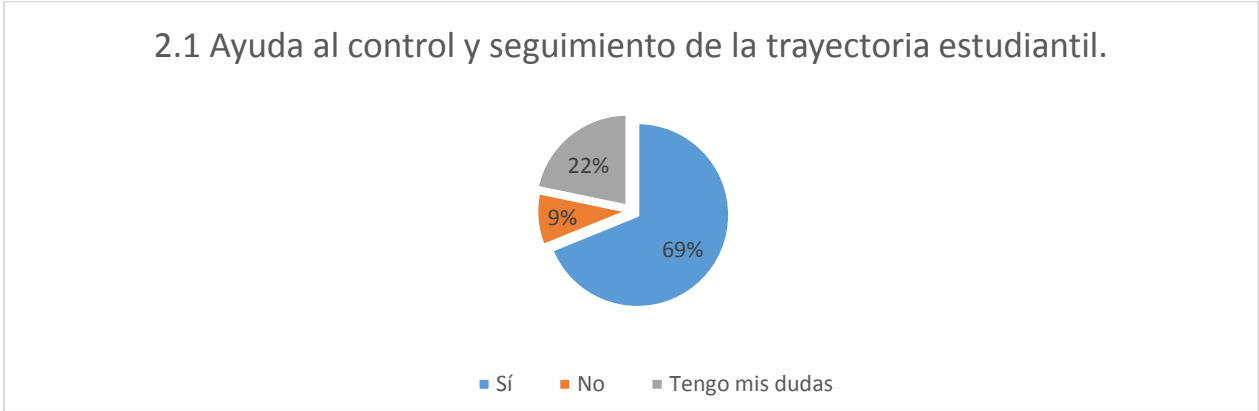
Con respecto a los dirigentes estudiantiles (presidentes de brigadas, secretarios de la UJC, miembros del secretariado de las facultades y la universidad) participaron 242 lo que representa aproximadamente un 51% del total de dirigentes, y el 5,1% con respecto a la matrícula de estudiantes. Las facultades con mayor representación son la 3 y la 4.

1. **¿Con qué frecuencia entras al sistema? (Las posibles respuestas eran: Todos los días, Una vez por semana, Regularmente, Casi nunca, Nunca)**

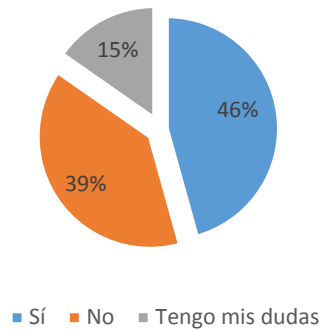


Es de destacar que el 53,3% entra con alguna frecuencia al sistema mientras que el resto lo hace casi nunca o nunca. De los dirigentes estudiantiles el 79,7% lo hace con alguna frecuencia, esto se debe fundamentalmente a que son estos los que tienen que introducir una parte importante de los datos al sistema.

**2. ¿Cómo ves a dataFEU? (posibles respuestas: Sí, No y Tengo dudas)**

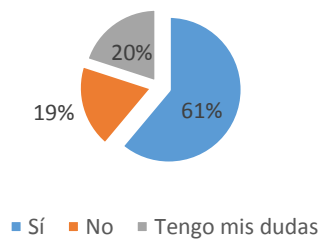


## 2.2 Es un mecanismo para obligarte a ir a las actividades.

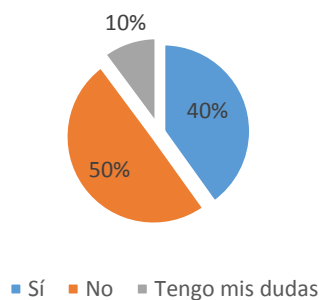


Todavía existe un porcentaje bastante alto que ve a la herramienta como una forma de obligar a ir a los estudiantes a las actividades, esta percepción es porque todavía se utiliza en muchas ocasiones para convocar, no dándose los argumentos correctamente. El 33% de los dirigentes opina que Sí, mientras que un 17% tiene dudas. El 60,5% de los que opinan positivamente se encuentran en los años de 4to y 3ero coincidiendo los que así opinan con el mayor porcentaje de los que entran casi nunca y nunca al sistema.

## 2.3 Permite evaluar cómo marchó durante el curso, además de fijarme metas altas.

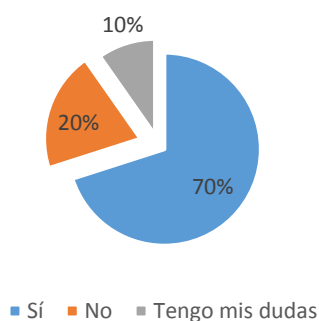


## 2.4 El perfil de los estudiantes debe estar público para que todos lo vean.

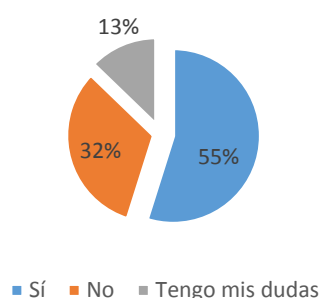


El 50% de los encuestados opina que no debe estar público, mientras que el 37% de los dirigentes se inclinan por esta tendencia. Hay que destacar que debe revisarse la visibilidad de algunos de los datos mostrados, delimitando cuáles deben estar públicos pues esto garantiza transparencia en el proceso de seguimiento y legitimización de los datos introducidos.

### 2.5 Los profesores deben tener acceso para poder seguir la evolución integral del estudiante.

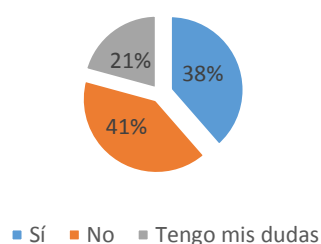


### 2.6 El sistema debe darme la evaluación a partir de las evidencias que tiene registradas.



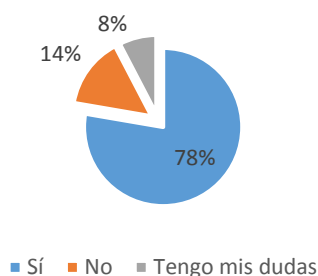
El 55% de los dirigentes coincide que el sistema debe emitir su evaluación. De los encuestados el 13% tiene dudas respecto a esta temática, de los cuales el 15% entra casi nunca o nunca al sistema.

### 2.7 Debe usarse en el proceso de caracterización de 5to año para la ubicación laboral.

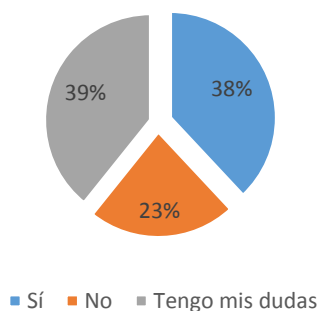


Un porcentaje considerable tiene dudas respecto a este punto, de ellos el 46% entran casi nunca o nunca al sistema, el 23% son dirigentes estudiantiles y el 51% se encuentran en 3ero o 4to año. Esto pudiera darse por el desconocimiento sobre el proceso de caracterización en 5to año.

### 2.8 Todas las actividades por mínimas que sean deben registrarse sin excepción.

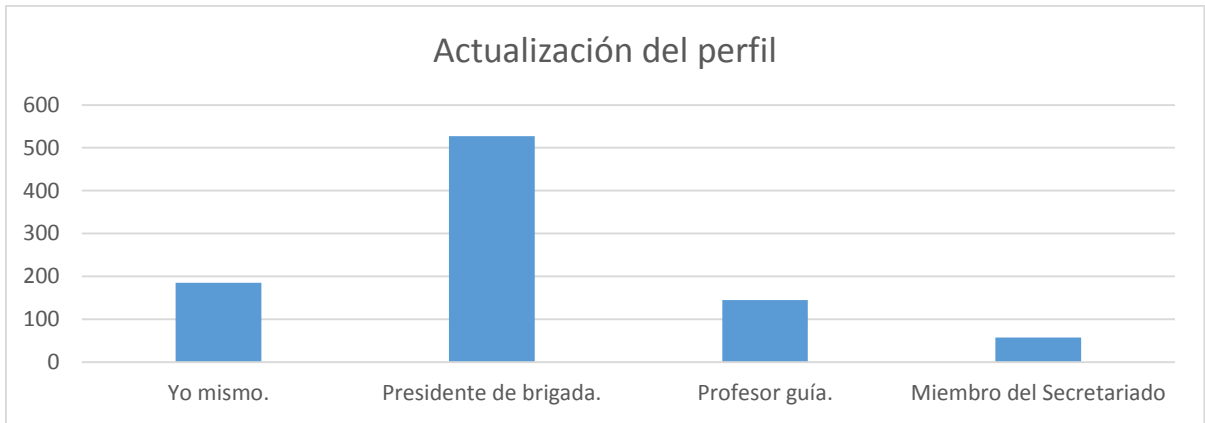


### 2.9 Es un sistema confiable y veraz en la información.



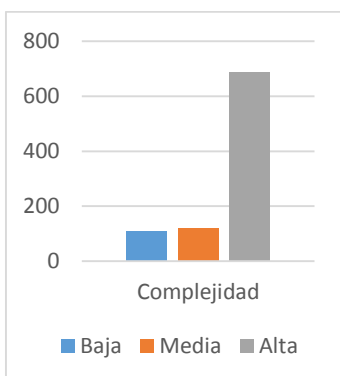
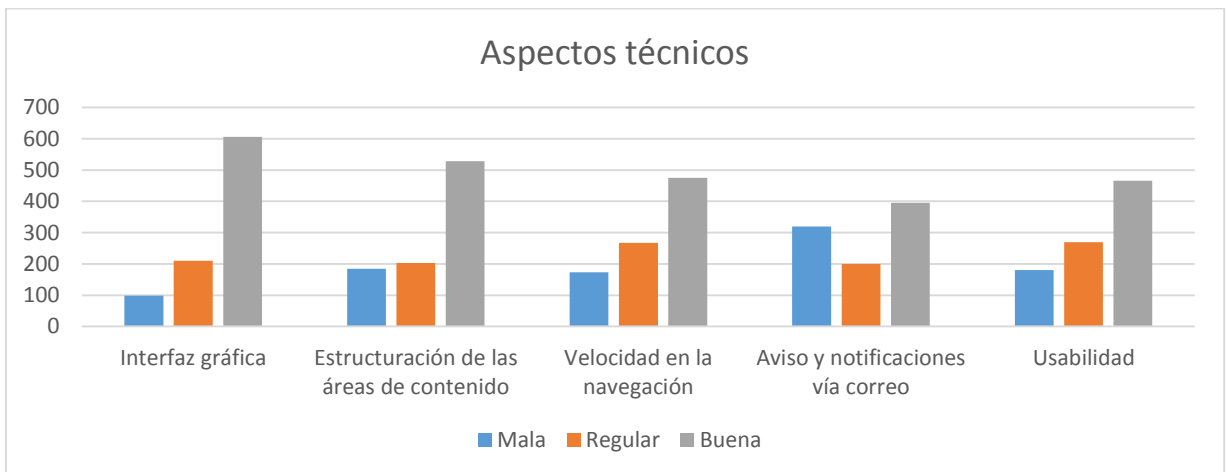
Este es uno de los puntos más importantes de la encuesta, evidenciándose una fragmentación en los criterios, lo que atenta contra la credibilidad hacia los certificadores de la información. De los encuestados que tienen dudas: el 48,1% entra casi nunca o nunca al sistema; 36% son dirigentes estudiantiles y el 52,3% son de 3ero y 4to año. Los que opinan que no: el 22% corresponde a la facultad 3 siendo esta la de mayor por ciento. De manera general el 49% de los dirigentes opinan que es confiable, siendo estos los mayores responsables de la veracidad y confiabilidad de la información que se genera.

### 3. ¿Quién crees deba actualizar tu perfil? (posibles respuestas: Yo mismo, Presidente de brigada, Profesor guía, Miembro del Secretariado de facultad o universidad)



El 65% de los dirigentes estudiantiles opina que también debe ser el presidente de brigada. Debe estudiarse una forma de lograr el equilibrio entre el propio usuario y el presidente de brigada.

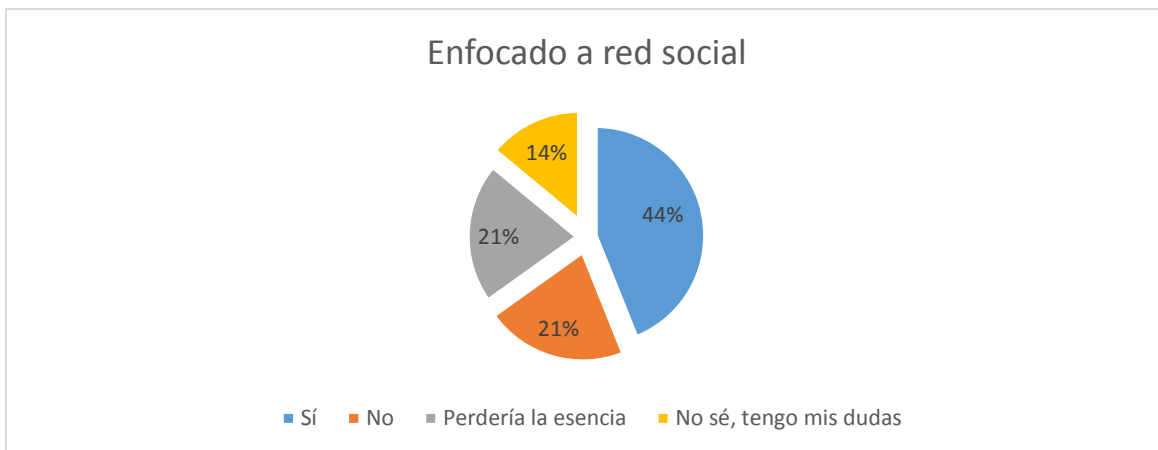
#### 4. Sobre los aspectos técnicos emitir una evaluación de 1 a 5.



De manera general los encuestados opinan que debe revisarse todo lo relacionado con las notificaciones y los avisos vía correo donde un porcentaje elevado opina que este aspecto es Regular o Malo. La complejidad es uno de los indicadores más elevados donde el 75% de los estudiantes y el 79% de los dirigentes opina que es alta.

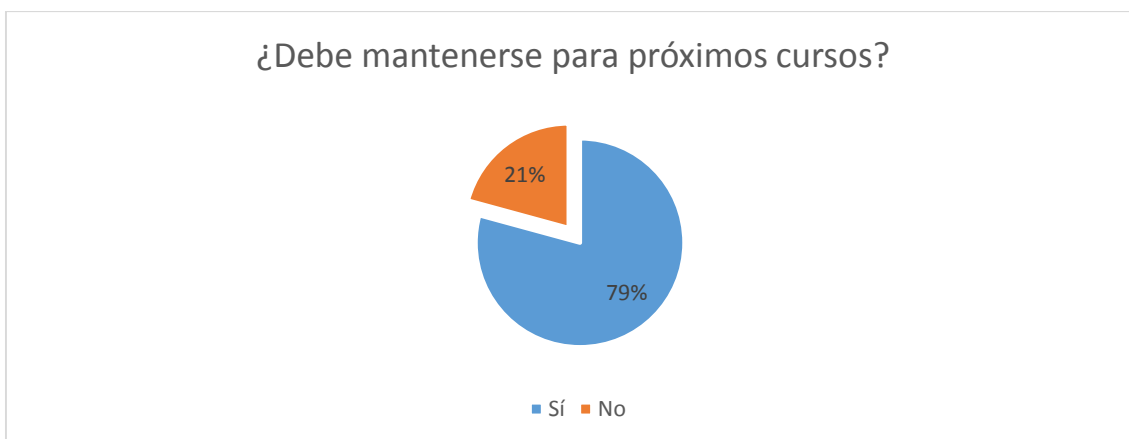


5. **¿Crees que el sistema debe ser más interactivo, enfocado a una red social donde puedas emitir criterios sobre tus compañeros, actividades y demás procesos de la organización y su funcionamiento? (posibles respuestas: Sí, No, Tengo mis dudas, Perdería la esencia)**



Existe una división importante en cuanto a este criterio, a pesar que la mayoría opina que sí. Debe estudiarse este enfoque y aplicarlo al sistema con el objetivo de hacerlo más interactivo para el usuario.

6. **¿Debe mantenerse para próximos cursos? (posibles respuestas: Sí o No)**



Del 21% que opina que no: el 11,5% son dirigentes estudiantiles y el 69,7% entra casi nunca o nunca al sistema.

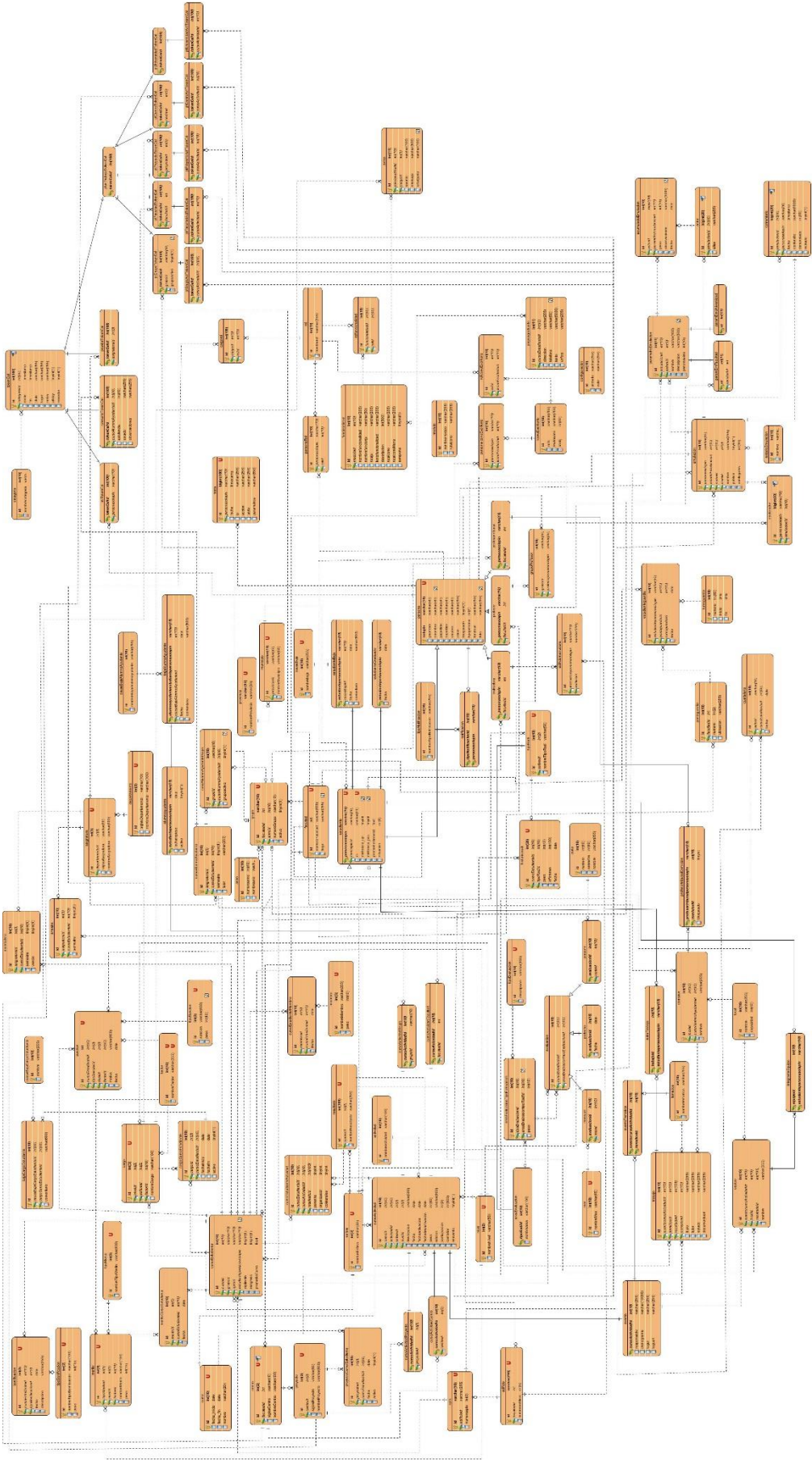
Esto evidencia una confianza en el desarrollo del sistema como herramienta de apoyo a la toma de decisiones para los dirigentes estudiantiles.

## ANEXO 3: Código de Nueva instancia actividad

```
public function executeNuevaInstanciaActividad(sfWebRequest $request) {
    $form = new CursoactividadForm();
    $arrayForm = array('cursoid' => $this->obtenerCursoActual(),
        'actividadid' => $request->getParameter('actividadid'),
        'esferaid' => $request->getParameter('esferaid'),
        'nivelid' => $request->getParameter('nivelid'),
        'descripcion' => $request->getParameter('descripcion'),
        'fecha' => $request->getParameter('fecha'),
        'fechaactivacion' => $request->getParameter('activacion'),
        'fechadesactivacion' => $request->getParameter('desactivacion'),
        'peso' => $request->getParameter('peso'),
        'edicion' => $request->getParameter('edicion'),
        'participacion' => $request->getParameter('participacion'),
        'contador' => 0,
        'esevento' => 0,
        '_csrf_token' => true);
    $this->processForm($form, $arrayForm); //1

    if ($request->getParameter('nivelid') == 1) { //2
        $formBrigada = new CursoactividadgrupoForm();
        $arrayFormBrigada = array('cursoactividadid' => $form->getObject()->getId(),
            'grupoid' => $request->getParameter('grupoid'),
            '_csrf_token' => true);
        $this->processForm($formBrigada, $arrayFormBrigada); //3
    } else if ($request->getParameter('nivelid') == 2) { //4
        $formFacultad = new CursoactividadfacultadForm();
        $arrayFormFacultad = array('cursoactividadid' => $form->getObject()->getId(),
            'facultadid' => $request->getParameter('facultadid'),
            '_csrf_token' => true);
        $this->processForm($formFacultad, $arrayFormFacultad); //5
    }
    return $this->renderText(json_encode(true)); //6
}
```

### ANEXO 4: Modelo de datos



## ANEXO 5: Parámetros para el cálculo del ranking de integralidad para el proceso de evaluación estudiantil de primero a cuarto año

La tabla debajo muestra un ejemplo de cómo son definidos estos parámetros dentro del sistema:

<i>Desglose de puntos</i>	<i>Descripción</i>
<b>+20 (máximo)(*)</b>	Por cada evento a nivel de facultad.
<b>+2</b>	Por obtener mención en un evento a nivel de facultad.
<b>+4</b>	Por obtener destacado en un evento a nivel de facultad.
<b>+5</b>	Por obtener relevante en un evento a nivel de facultad.
<b>+40</b>	Por cada evento a nivel de Universidad.
<b>+5</b>	Por obtener resultados en un evento a nivel de Universidad.
<b>+15 (**)</b>	Por cada evento de primer nivel al que participe a un nivel superior al de Universidad.
<b>+7</b>	Por participar en actividades colaterales.
<b>+5</b>	Por participar como organizador en un evento.
<b>+10</b>	Por cada premio académico que obtenga
<b>+12</b>	Por cada evaluación de B en los cortes trimestrales.
<b>+4</b>	Por cada evaluación de B mensual en los proyectos productivos.
<b>+5</b>	Por cada evaluación de B en la residencia.
<b>+10</b>	Por obtener la evaluación de B en el TSU.
<b>+10</b>	Por cada actividad que clasifique como otras. (Marcha, desfiles).
<b>+2</b>	Por cada asistencia a vespertinos.
<b>+40(máximo)(***)</b>	Por cada mérito o reconocimiento.
<b>-40</b>	Faltar injustificadamente a una reunión de brigada.
<b>-20</b>	Por cada evaluación de Crítico (C) en los cortes trimestrales.
<b>-8</b>	Por cada evaluación de Mal (M) en los cortes trimestrales.
<b>-8</b>	Por cada evaluación de Mal (M) mensual en los proyectos productivos.
<b>-40</b>	Faltar injustificadamente a una guardia estudiantil o a

	una cuartelaría.
<b>-10</b>	Por cada evaluación de Mal (M) en la residencia.
<b>-30</b>	Por cada examen llevado a mundial.
<b>-40</b>	Por obtener evaluación de Mal (M) en el TSU.
<b>-30 (máximo)</b>	Según el tipo de sanción aplicada a nivel de brigada.
<b>-50 (máximo)</b>	Según el tipo de sanción aplicada a nivel de facultad.

(\*) Los miembros del Consejo valoran otorgar una puntuación al resto en dependencia de su magnitud y participación de los estudiantes.

(\*\*) Si los eventos con nivel superior al de universidad siguen el ciclo de vida desde la base entonces tributan 15 puntos, en caso de ser un evento que pasa directo a nivel Provincial, Nacional o Internacional tributa con 45 puntos.

(\*\*\*) El Consejo de la FEU valora la magnitud de cada distinción y en función de ello determinan la ponderación de los reconocimientos de su nivel.

Una vez definidos cada uno de estos indicadores, se realiza una suma comparando el total de evidencias dentro del sistema y las que posee cada estudiante.

## ANEXO 6. Parámetros para el cálculo del ranking de integralidad para el proceso de caracterización

Se les otorgarán valores enteros entre 0 y 5 a los indicadores que a continuación se definen. Según las indicaciones en el Procedimiento del Proceso de Caracterización para quinto año los mismos pueden ser modificados para cada proceso. Se exceptúa solamente el indicador de Formación, cuyo valor es el índice académico.

### Indicadores:

- Formación (Índice académico)
- Producción
- Investigación
- Extensión
  - Residencia
  - Guardia
  - Deporte
  - Cultura
  - TSU
- FEU – UJC

### A los siguientes indicadores se les asigna otros valores, a continuación se detallan:

- Reconocimientos: El valor de este indicador se sumará en función del tipo de reconocimiento que haya obtenido el estudiante.
  - Premio Mella: Se sumarán 8 puntos al total.
  - Título de Oro: Se sumarán 4 puntos al total.
- Agravantes:
  - Medidas disciplinarias: El valor de este indicador se restará al total en función del tipo de medida en la que haya incurrido en el estudiante.
    - ✓ Menos Graves: El valor de este sub-indicador restará 8 puntos al total, por cada medida de este tipo.
    - ✓ Graves: El valor de este sub-indicador restará 12 puntos al total, por cada medida de este tipo.
    - ✓ Muy Graves: El valor de este sub-indicador restará 16 puntos al total, por cada medida de este tipo.
  - Repitente: El valor de este sub-indicador restará 12 puntos al total.
  - Arrastre: El valor de este sub-indicador restará 2 puntos al total, por cada asignatura de arrastre que haya tenido el estudiante en la carrera.

- Mundiales: El valor de este sub-indicador restará 2 puntos al total, por cada mundial que haya llevado el estudiante en la carrera.

### Cálculo del valor final para el Ranking de Integralidad

El valor final de Integralidad para la ubicación en el escalafón de integralidad de la universidad se obtiene sumando el valor del indicador Formación (índice académico: índ\_acad) y el valor obtenido del resto de los indicadores evaluados, al que denotaremos en adelante como: (valor\_otros\_ind).

A continuación se detallan los pasos para calcular el valor de los restantes indicadores evaluados (valor\_otros\_ind):

1. Los valores enteros otorgados entre 0 y 5 a los indicadores: Producción, Investigación y FEU – UJC se suman y el valor resultante se multiplica por 5 y a este resultado se le suma, los valores enteros otorgados entre 0 y 5, a los sub - indicadores del indicador Extensión: Deporte, Guardia, Cultura, Residencia y TSU. El resultado máximo que se puede obtener es 100 puntos y el valor mínimo es 0 puntos.
  - $[(\text{valor\_Producción} + \text{valor\_Investigación} + \text{valor\_FEU-UJC}) * 5] + (\text{valor\_Deporte}) + (\text{valor\_Guardia}) + (\text{valor\_Cultura}) + (\text{valor\_Residencia}) + (\text{valor\_TSU})$
2. A este valor resultante se le suma, el valor asignado al indicador Reconocimientos, donde el Premio Mella aporta un valor de 8 puntos y Título de Oro aporta un valor de 4 puntos. El mínimo valor de este indicador es 0 y el máximo es 12.
  - $(\text{valor\_Premio\_Mella}) + (\text{valor\_Título\_de\_Oro})$
3. Al valor resultante se le resta el valor del indicador Agravantes, el cual está compuesto por los valores de los sub-indicadores: Medidas disciplinarias, Repitencia, Arrastres y Mundiales.
  - $(\text{valor\_Agravantes}) = (\text{valor\_Medidas\_disciplinarias}) + (\text{valor\_Repitencia}) + (\text{valor\_Arrastres}) + (\text{valor\_Mundiales})$
  - $(\text{valor\_Medidas\_disciplinarias}) = [(\text{valor\_Menos\_Graves}) * 8] + [(\text{valor\_Graves}) * 12] + [(\text{valor\_Muy\_Graves}) * 16]$
  - $(\text{valor\_Repitencia}) = 12$  ( si el estudiante ha repetido un año académico)
  - $[(\text{valor\_Arrastres}) = (\text{Cantidad\_asignaturas\_arrastres\_carrera}) * 2$
  - $[(\text{valor\_Mundiales}) = (\text{Cantidad\_de\_mundiales\_carrera}) * 2$
4. El valor resultante se divide entre 100, y se obtiene el valor de los restantes indicadores: (valor\_otros\_ind)

## ANEXO 7: Criterios de complejidad

La complejidad de los requisitos se utiliza para estimar el esfuerzo de implementación de este y planificar en qué iteración se implementará. Un requisito puede tener complejidad alta, media o baja, determinándose a partir de los criterios que se describen a continuación:

- Complejidad por número de transacciones: una transacción es un intercambio de datos a nivel de procesos elementales entre las fronteras del sistema, que fuerzan a la lógica de la aplicación a realizar un procesamiento para entrar y/o mostrar información. La complejidad viene representada en el esfuerzo de implementación de un número elevado de procedimientos reflejados en una transacción.

Alta: El requisito contiene más de 8 transacciones.

Media: El requisito contiene de 5 a 8 transacciones.

Baja: El requisito contiene de 1 a 4 transacciones.

- Complejidad por entidades candidatas: las entidades candidatas son las futuras clases de la aplicación que modelarán los elementos persistentes de la base de datos. Una clase envuelve atributos, procedimientos y está relacionada a clases de control, por lo que un monto elevado de entidades o de atributos de la entidad refleja una complejidad en la implementación de las operaciones de un Requisito que las contiene.

Alta: más de 5 entidades.

Media: de 3 a 5 entidades.

Baja: de 1 a 2 entidades.

- Complejidad por interfaces de comunicación con actores: las interfaces de comunicación muestran a un actor del sistema la forma de intercambiar información ya sea para la entrada, como para la salida. Un actor puede estar instanciado en una persona o un sistema externo que interactúe con el sistema. La complejidad viene reflejada por el esfuerzo de la implementación de dichas interfaces unida a las restricciones y funcionalidades de su uso.

Alta:

- Una persona que interactúa con el sistema mediante una interfaz gráfica donde se evidencie una de las situaciones siguientes:
- La interfaz posee elementos de animación en tercera dimensión 3D y se vale de ellos para la gestión de la misma.
- La interfaz utiliza gráficos en segunda dimensión 2D para la muestra de información y se vale de ellos para la gestión de la misma.



- La interfaz está formada por más de 15 componentes tradicionales de captura y muestra de información.
- Media:
  - La interfaz contiene desde 8 hasta 15 componentes tradicionales de captura y muestra de información.
- Baja:
  - La interfaz contiene desde 1 hasta 7 componentes tradicionales de captura y muestra de información.
  - Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, DLL)
  - Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto (Servicios Web, XML, Correo electrónico).
- Complejidad por número de requisitos no funcionales: los requisitos no funcionales son características que imponen restricciones al sistema, por lo que condicionan la implementación de los requisitos al agregarle especificidades a las funcionalidades del mismo.
 

Alta: más de 6 requisitos no funcionales asociados.

Media: entre 4 y 6 requisitos no funcionales asociados.

Baja: menos de 4 requisitos no funcionales asociados.
- Complejidad por tipo de tecnología: la utilización de los avances tecnológicos de software o hardware que necesiten de una asimilación de su funcionamiento o comunicación condicionan el esfuerzo de la implementación de Requisitos que los empleen.
 

Alta: la realización del requisito utiliza elementos, plataformas, componentes tecnológicos avanzados de conocimiento externo al equipo de programación que necesiten de capacitación.

Media: la realización del requisito utiliza elementos tecnológicos avanzados donde el conocimiento reside en el equipo de implementación, producto de la selección y preparación inicial en los acuerdos tecnológicos del contrato.

Baja: no se emplean nuevas técnicas, plataformas, componentes ni conceptos de programación.
- Complejidad por reutilización: el uso de patrones, componentes predefinidos del lenguaje o plataformas que generen partes de las funcionalidades del sistema aminoran el esfuerzo en la construcción del software.
 

Alta: no se reutilizan elementos, el requisito se codifica por completo por un programador.

Media: se reutilizan elementos donde es necesario ajustar la codificación antes de la generación de la implementación del caso de uso.

Baja: se utilizan elementos que implementan el requisito y sólo es necesario realizar pequeños ajustes.