



Universidad de las Ciencias Informáticas

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

Título:

**Sistema de gestión de la información para la bonificación del 50%
del pasaje a estudiantes.**

Autor:

Sucel Daudinot Caricabeur

Tutor:

MSc. Radel Calzada Pando

MSc. Miguel Jaeger Rodriguez Lazo

Consultante:

MSc. William Santana Méndez

La Habana, junio del 2014.

Declaración de autoría

Declaro ser autora del trabajo de diploma titulado “Sistema de gestión de la información para la bonificación del 50% del pasaje a estudiantes” y reconozco a la **Universidad de las Ciencias Informáticas** los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los _ días del mes de _____ del año _____ 2014

Sucel Daudinot Caricabeur

Firma del Autor

MSc. Radel Calzada Pando

Firma del Tutor

MSc. Miguel Jaeger Rodriguez Lazo

Firma del Tutor

MSc. William Santana Méndez

Firma del Consultante

Datos de Contactos

- **Tutores:**

MSc. Radel Calzada Pando

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: rcalzada@uci.cu

MSc. Miguel Jaeger Rodriguez Lazo

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: jaeger@uci.cu

- **Consultante:**

MSc. William Santana Méndez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: wsantana@uci.cu

“Si una persona es perseverante, aunque sea dura de entendimiento, se hará inteligente; y aunque sea débil se transformará en fuerte”.

Leonardo da Vinci



Agradecimientos

Agradezco a todas aquellas personas que me ayudaron en la elaboración de este trabajo y en especial en mi vida como universitaria.

A mis padres: Cristina Caricabeur Vinent y Miguel Angel Daudinot Reve por su apoyo y guía ante cualquier dificultad.

A mis tías: Silvia, Mamita, Lourdes, Irene y Coralia por estar cuando más se le necesita.

A mis amigos: Mora, Yurima, Yudelkis, Arianna, Evelyn, Rosy, Alex, Pedro, Dasiel, Noichel, Manuel, Raciél, Radel, Aliosmi, William, Beylen y Mailen, a todos los compañeros de mi grupo y a aquellos que de una forma u otra me han dado techo por muchos años, me refiero a los profesores del edificio 32 apto 203.

De Su cel

Dedicatoria

A mi mamá y papá por estar siempre ahí para mí.

De Sucel

Resumen

Los sistemas de gestión, cuyo desarrollo está marcado por el avance de las Tecnologías de la Información y las Comunicaciones (TIC), han propiciado la optimización de los procesos en las organizaciones, lo que contribuye al control efectivo de su información.

En la Vicerrectoría Económica de la UCI se lleva a cabo la retribución del 50% del valor del pasaje a estudiantes por concepto de bonificación. Este proceso es realizado de forma manual, provocando deficiencias en su ejecución. En la presente investigación se desarrolla un sistema de gestión automatizado que permite controlar adecuadamente la información relacionada con este proceso.

Con el fin de alcanzar el objetivo trazado, se analizan los principales conceptos que fundamentan la investigación, se realiza una revisión de los antecedentes sobre los sistemas de gestión económica en el ámbito nacional e internacional y se expone la fundamentación de las herramientas utilizadas y los elementos necesarios para la construcción del sistema de gestión de la información para el pago de la bonificación del 50% del pasaje a estudiantes. Se realizan un conjunto de pruebas para validar el correcto funcionamiento de las diferentes funcionalidades propuestas, verificando además que el sistema cumple con las necesidades del cliente.

Palabras claves: sistema de gestión, TIC, bonificación.

Índice de Contenidos

Introducción	12
Capítulo I: Fundamentación teórica.	15
1.1. Introducción	15
1.2. Conceptos asociados al dominio del problema.....	15
1.3. Sistemas de gestión económica.	16
1.3.1. Sistemas de gestión económica en el mundo.	16
1.3.2. Sistemas de gestión económica en Cuba.	16
1.4. Metodologías de desarrollo de software.	17
1.4.1. Metodologías tradicionales.	18
1.4.2. Metodologías ágiles.	18
1.5. Lenguaje de modelado.....	20
1.5.1. Lenguaje Unificado de Modelado.....	20
1.5.2. Notación de Modelado de Procesos del Negocio.	20
1.6. Selección de la herramienta CASE de modelado.....	20
1.7. Selección de la herramienta de desarrollo del software.	21
1.7.1. Selección del CMS.	22
1.8. Lenguajes de programación.....	23
1.8.1. Lenguaje de programación del lado del cliente.	23
1.8.2. Lenguaje interpretado del lado del servidor.	24
1.9. Entorno de desarrollo integrado.	24
1.10. Selección del sistema Gestor de Base de Datos (SGBD).	25
1.11. Servidores de aplicaciones web.....	25
1.12. Conclusiones Parciales.....	26
Capítulo II: Características de la solución propuesta.	27
2.1. Introducción	27
2.2. Descripción general de la propuesta de solución.	27
2.3. Modelado de procesos de negocio.....	27
2.3.1. Mapa de procesos.....	27
2.3.2. Descripción de los procesos del negocio.	28
2.3.3. Actores y trabajadores que intervienen en el proceso del negocio.	31
2.4. Requisitos del sistema.....	32
2.4.1. Listado de requisitos funcionales.....	32
2.4.2. Listado de requisitos no funcionales.	32
2.5. Historias de Usuario (HU).	34

2.6.	Planificación	37
2.6.1	Plan de entregas.	37
2.6.2	Plan de Iteraciones.....	38
2.7	Diseño.....	39
2.7.1	Descripción de la arquitectura.	39
2.7.2	Patrones de diseño.	40
2.7.3	Diagrama de paquetes.	41
2.7.4	Diseño de la base de datos.	43
2.8	Conclusiones parciales.	44
Capítulo III: Implementación y Pruebas.....		45
3.1.	Introducción	45
3.2.	Código fuente.	45
3.2.1.	Estándares de codificación.....	45
3.3.	Implementación.....	48
3.3.1.	Diagrama de componentes.....	48
3.3.2.	Diagrama de despliegue.....	50
3.4.	Pantallas principales de la aplicación.	51
3.5.	Pruebas.....	53
3.5.1.	Pruebas unitarias.	53
3.5.2.	Pruebas de aceptación.	56
3.5.3.	Pruebas de Seguridad.	60
3.5.4.	Pruebas de carga y stress.....	61
3.6.	Conclusiones parciales.	62
Conclusiones		63
Recomendaciones		64
Bibliografía referenciada		65
Bibliografía		69
Glosario de términos.		75
Anexos.....		76
Anexo 1: Tabla de comparación entre metodologías ágiles y tradicionales.....		76
Anexo 2: Modelo de autorizo de pago de bonificación.....		76
Anexo 3: Modelo de bonificación.....		77
Anexo 5: Tareas ingenieriles.		97
Anexo 6: Diseño de casos de prueba.		101

Índice de tablas

Tabla 1: HU Adicionar Bonificación.....	34
Tabla 2: Imprimir bonificación.	35
Tabla 3: HU Registrar fecha de cobro.	36
Tabla 4: Plan de entrega del sistema.....	37
Tabla 5 Plan de iteraciones.	38
Tabla 6: Distribución de tareas de ingeniería por iteración.	39
Tabla 7: Descripción de diagrama de componentes del sistema.....	49
Tabla 8: Método estadísticas_valida_fecha.	55
Tabla 9: Camino básico.....	55
Tabla 10: Casos de prueba unitarias.	55
Tabla 11: Caso de prueba: Adicionar bonificación.	56
Tabla 12: Caso de prueba: Imprimir bonificación.....	58
Tabla 13: Caso de prueba: Registrar fecha de cobro.....	59
Tabla 14: Diferencias entre metodologías ágiles y tradicionales.....	76
Tabla 15: HU buscar estudiante.....	78
Tabla 16: HU buscar estudiante.....	79
Tabla 17: HU eliminar estudiante.....	80
Tabla 18: HU eliminar bonificación.....	81
Tabla 19: HU Buscar Bonificación.....	83
Tabla 20: HU mostrar bonificación.....	84
Tabla 21: HU agregar presupuesto.....	85
Tabla 22: HU buscar presupuesto.	86
Tabla 23: HU editar presupuesto.	87
Tabla 24: HU eliminar presupuesto.	89
Tabla 25: HU mostrar presupuesto.....	90
Tabla 26. HU generar reportes.	91
Tabla 27: HU exportar a Excel.....	92
Tabla 28: HU exportar reportes a Word.....	93
Tabla 29: HU crear rol a usuario.	94
Tabla 30: HU eliminar rol a usuario.	95
Tabla 31: HU Editar rol a usuario.....	96
Tabla 32: Tarea de ingeniería 1. Autenticar con usuario y contraseña de LDAP.	97
Tabla 33: Tarea de ingeniería 2. Buscar estudiante desde el directorio de LDAP.....	97

Tabla 34: Tarea de ingeniería 3. Eliminar estudiante seleccionado de la base de datos del sistema.	97
Tabla 35: Tarea de ingeniería 4. Gestionar bonificación.	98
Tabla 36: Tarea de ingeniería 5. Imprimir modelo.	98
Tabla 37: Tarea de ingeniería 6. Gestionar presupuesto.	98
Tabla 38: Tarea de la ingeniería 7. Generar reporte de la información especificada por el usuario.	99
Tabla 39: Tarea de ingeniería 8. Configurar módulo para exportar a doc.	99
Tabla 40: Tarea de ingeniería 10. Configurar módulo para exportar a xls.	99
Tabla 41: Tarea de ingeniería 11. Consultar resultados de la búsqueda (bonificación).	100
Tabla 42: Tarea de ingeniería 9. Gestionar rol.	100
Tabla 43: Caso de prueba buscar bonificación.	101
Tabla 44: Caso de prueba eliminar bonificación.	101
Tabla 45: Caso de prueba mostrar bonificación.	102
Tabla 46: Caso de prueba buscar estudiante.	102
Tabla 47: Caso de prueba eliminar estudiante.	103
Tabla 48: Caso de prueba agregar presupuesto.	105
Tabla 49: Caso de prueba buscar presupuesto.	106
Tabla 50: Caso de prueba editar presupuesto.	107
Tabla 51: Caso de prueba eliminar presupuesto.	108
Tabla 52: Caso de prueba mostrar presupuesto.	109
Tabla 53: Caso de prueba autenticar usuario.	109
Tabla 54: Caso de prueba exportar reportes a Word.	110
Tabla 55: Caso de prueba generar reportes.	111
Tabla 56: Caso de prueba generar reportes a Excel.	111

Índice de figuras

Figura 1. Mapa de procesos de la bonificación.	28
Figura 2: Proceso de negocio de la bonificación.	28
Figura 3: Proceso de negocio: Autorizar pago de la bonificación.	29
Figura 4: Proceso de negocio: Emitir modelo de pago de la bonificación.	30
Figura 5: Proceso de negocio: Efectuar pago de la bonificación.....	31
Figura 6: Elementos que conforman la arquitectura de Drupal.....	40
Figura 7: Diagrama de paquete del sistema de gestión de bonificación.	42
Figura 8: Modelo físico de la base de datos del sistema de bonificación.	43
Figura 9: Código del módulo estadística del sistema.	48
Figura 10: Diagrama de componentes del sistema.	49
Figura 11: Diagrama de componente del módulo estadísticas.....	49
Figura 12: Diagrama de despliegue del sistema.....	51
Figura 13: Interfaz de inicio del sistema.....	51
Figura 14: Interfaz para acceder a las funcionalidades.....	52
Figura 15: Buscar estudiantes.	52
Figura 16: Interfaz de la funcionalidad gestionar presupuesto.	53
Figura 17: Resultado de las pruebas funcionales.	60
Figura 18: Prueba de carga y stress.....	61
Figura 19: Modelo de autorizo de pago de bonificación.	76
Figura 20: Modelo de Pago de Bonificación.	77

Introducción

La correcta gestión de los procesos es fundamental para el buen funcionamiento de cualquier organización. La conciencia tomada por las organizaciones y su compromiso en lograr la mayor eficiencia en el funcionamiento de sus procesos se manifiesta en la inversión, cada vez mayor, de recursos para su optimización y el empleo de las Tecnologías de la Información y las Comunicaciones (TIC).

La sociedad actual se sustenta en un desarrollo tecnológico sin precedentes. En este contexto debe entenderse que las TIC son un medio para transmitir y gestionar datos, los cuales están presentes en cada aspecto del desarrollo organizacional de las empresas, entidades u organismos (1).

Sobre la base de los aspectos antes expuestos surgen los sistemas de gestión económica, siendo de gran importancia para el desempeño de los procesos económicos en las organizaciones que los utilizan. Estos gestionan y procesan la información con mayor rapidez en comparación al procesamiento y gestión manual de la misma; además inciden en gran medida, en la reducción del número de errores.

En Cuba también se apuesta por el uso de las TIC para gestionar los procesos económicos que se llevan a cabo en las organizaciones, es por ello que actualmente se utilizan en algunas sistemas de gestión económica, mayormente desarrollados sobre plataformas privativas. Ejemplo de los mismos son: SISCONT5, Versat Sarasola, CONDOR, RODAS XXI, SISCOMIP y ASSETS PREMIUS.

En la Universidad de las Ciencias Informáticas (UCI) se utiliza la herramienta ASSETS PREMIUS, una de las áreas que la emplean en el centro es la Vicerrectoría Económica, la cual se encarga de gestionar los procesos contables de la Universidad. Dentro de los procesos contables de la UCI se encuentra el pago de dietas, facturas, salario a trabajadores y estipendio a estudiantes. Otro proceso que se gestiona es el pago del 50% del pasaje a estudiantes por concepto de bonificación, el cual se realiza de diferentes formas en todas las universidades del país según sus características. Este proceso genera un gran volumen de información como resultado del elevado número de estudiantes internos, los que en su mayoría son de provincias lejanas a la Universidad y viajan con cierta frecuencia a su localidad de origen. ASSETS PREMIUS, como sistema contable, ejecuta de los procesos económicos antes expuestos excepto el pago de la bonificación. Esta aplicación carece de una funcionalidad para realizar dicha gestión con las particularidades aplicadas al pago de bonificación en la UCI, trayendo como resultado que la gestión de este proceso se realice de forma manual, surgiendo los siguientes inconvenientes:

- El mecanismo de control de las ejecuciones financieras y contables por cada una de las facultades de aquellos estudiantes que hacen uso de la bonificación, no es eficiente, debido a que se afecta la inmediatez requerida a la hora de dar un reporte. Además se vuelve engorroso tener conocimiento preciso de las bonificaciones realizadas en un período de tiempo determinado y saber el monto que alcanzan las mismas.
- Se tiene dificultad en el control de las bonificaciones que se han autorizado en la Vicerrectoría Económica, también denominadas como bonificaciones en circulación,

debido a que no existe comunicación inmediata entre la caja de la Universidad y la Vicerrectoría Económica, trayendo como consecuencia que no se conozca el día real en que el estudiante cobra la bonificación.

- No se tiene un expediente contable por cada estudiante, que permita conocer la cantidad de veces que hace uso en el año del pago de bonificación del 50% de pasaje.

Teniendo en cuenta los elementos antes mencionados, se enuncia el siguiente **problema de la investigación**: ¿Cómo contribuir a mejorar el proceso de gestión de pago de bonificación del 50% del pasaje a estudiantes en la Universidad de las Ciencias Informáticas?

Ante el problema definido, se establece como **objeto de estudio**: Los sistemas de gestión y como **campo de acción**: Los sistemas de gestión económica.

Para dar solución al problema se plantea como **objetivo general**: Desarrollar un sistema de gestión económica para el proceso de pago de bonificación del 50% del pasaje a estudiantes para la Vicerrectoría Económica de la UCI. Cómo objetivos **específicos** se tiene:

- Elaborar el marco teórico de la investigación identificando las principales tendencias, limitaciones y ventajas de los sistemas de gestión económica existentes, así como las principales tecnologías, metodologías y herramientas para el desarrollo de la investigación.
- Obtener los requisitos funcionales y no funcionales del sistema.
- Diseñar el sistema de gestión, para el control de pago de la bonificación del 50% del pasaje a estudiantes.
- Implementar el sistema de gestión para el control de pago de la bonificación del 50% del pasaje a estudiantes.
- Validar el sistema de gestión desarrollado para el pago de la bonificación del 50% del pasaje a estudiantes, mediante pruebas de software.

Las **tareas de la investigación** planificadas para dar cumplimiento a los objetivos planteados son:

- 1) Caracterización de los aspectos teóricos referentes al proceso de pago de la bonificación del 50% del pasaje a estudiantes por la Vicerrectoría Económica de la UCI.
- 2) Caracterización de los sistemas que realizan gestión económica.
- 3) Selección de las tecnologías y herramientas a utilizar para el desarrollo del sistema.
- 4) Obtención de los requisitos funcionales y no funcionales del sistema.
- 5) Elaboración de la documentación y artefactos necesarios para el análisis, diseño e implementación de la solución propuesta.
- 6) Implementación del sistema de gestión del proceso del pago de la bonificación del 50% del pasaje a estudiantes por la Vicerrectoría Económica de la UCI.
- 7) Selección y puesta en práctica de un método para la evaluación técnica del sistema desarrollado.

Para el desarrollo de la investigación se emplea los siguientes **métodos científicos**:

Métodos Teóricos:

- **Analítico-Sintético:** Se utiliza para analizar los diferentes conceptos, características del proceso de pago de bonificación de la UCI y obtener el conocimiento necesario para la implementación de una solución para el problema planteado.
- **Inductivo-Deductivo:** Permite llegar al planteamiento del objetivo, además de la extracción de las ideas fundamentales para la elaboración y fundamentación teórica del trabajo de diploma.
- **Histórico-Lógico:** Se requiere en el estudio de los antecedentes, la evolución y el desarrollo que han tenido los sistemas de gestión económica.
- **Modelación:** Se utiliza para modelar diagramas que contribuyan a comprender el funcionamiento del objeto de la investigación.

Método Empírico:

- **Entrevista no estructurada:** Se realiza con el objetivo del entendimiento del negocio así como para obtener los requisitos funcionales y no funcionales del proceso de pago de la bonificación del 50% del pasaje a estudiantes por la Vicerrectoría Económica de la UCI.

El presente trabajo de diploma se divide en tres capítulos estructurados de la siguiente forma:

- **Capítulo I: Fundamentación teórica.** El capítulo puntualiza los principales conceptos relacionados con el tema abordado en la investigación, se realiza el estudio del estado del arte a nivel nacional e internacional. Además se definen las herramientas, metodología y tecnologías a utilizar durante el desarrollo del sistema.
- **Capítulo II: Características de la solución propuesta.** Expone una descripción general de la solución propuesta y su funcionamiento. Se capturan los requisitos funcionales y no funcionales del sistema. Se detallan los principales aspectos relacionados con el diseño de la solución a desarrollar. Se define la arquitectura del software, se especifican los patrones del diseño a aplicar y los artefactos derivados de la metodología de desarrollo de software seleccionada.
- **Capítulo III: Implementación y Pruebas.** En este capítulo se abordan aspectos relacionados con la implementación del sistema y con la aplicación de las diferentes pruebas realizadas al mismo, para verificar que cumple con los requisitos y validar sus funcionalidades.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

1.1. Introducción

Durante el desarrollo de este capítulo se expone el marco teórico en el que se desarrolla la investigación, creando una base sólida para los próximos capítulos. Para esto, se realiza un estudio de las tendencias actuales de algunas soluciones informáticas que automatizan la gestión de los procesos económicos en el mundo, lo que sirve como entrada al análisis del proceso que se desea automatizar. Además se analizan las herramientas, metodologías y tecnologías actuales que guían el proceso de desarrollo del software, y se seleccionan las más adecuadas a utilizar para el desarrollo del sistema.

1.2. Conceptos asociados al dominio del problema.

A continuación se presentan los principales conceptos asociados al dominio del problema de la presente investigación, para una mejor comprensión del tema abordado.

- **Bonificación**

La Real Academia de Ciencias Económicas y Financieras (RACEF) define el vocablo bonificación como: “Descuento que se efectúa en el precio a pagar o premio en una cantidad a cobrar” (2).

- **Sistema**

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben entrada de datos, y proveen salida de información. Un grupo de elementos no constituye un sistema si no hay una relación e interacción, que de la idea de un todo con un propósito (3).

La Real Academia de lengua española define como sistema a un: “Conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí” o “Conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto” (4).

Teniendo en cuenta lo anteriormente planteado la autora de la presente investigación define el vocablo sistema como un conjunto de componentes que interactúan entre sí con un objetivo determinado.

- **Sistema de gestión**

Un sistema de gestión ayuda a obtener los objetivos de una organización mediante una serie de estrategias que incluyen la optimización de procesos, el enfoque centrado en la gestión de la información (5).

Se define sistema de gestión como un “sistema para establecer la política y los objetivos y para lograr dichos objetivos”. De igual forma se presenta al sistema como el “conjunto de elementos mutuamente relacionados o que interactúan”. Además se encuentra una forma muy simple para definir sistema de gestión:

Sistema de gestión: Conjunto de procesos de una organización que interactúan para alcanzar objetivos (3).

El desarrollo de un sistema de gestión brinda ventajas como:

- ✓ Control de las actividades de la organización.
- ✓ Disponibilidad de información para los usuarios en tiempo real.
- ✓ Elimina la barrera de la distancia trabajando con un mismo sistema en puntos distantes.
- ✓ Disminuye errores, tiempo y recursos (6).
- **Gestión económica**

En términos teóricos la gestión económica es la encargada de llevar a vía de hecho toda la actividad de una organización con eficiencia y eficacia con vista a que las empresas obtengan resultados que pueden ser positivos o negativos de acuerdo al manejo de dicha gestión realizada (7).

1.3. Sistemas de gestión económica.

A continuación se expone una breve descripción de algunos sistemas de gestión económica, para arribar a conclusiones que permitan guiar la propuesta de solución hacia el cumplimiento de los objetivos propuestos.

1.3.1. Sistemas de gestión económica en el mundo.

- **ContaPyme**

ContaPyme (Sistema de gestión empresarial y contable para PYMES¹), es una herramienta que maneja de forma integrada las áreas económicas de gestión de una organización. Este sistema incorpora un conjunto de herramientas para manipular los datos y la obtención de resultados. Una de las características más relevantes de este, es que utiliza un registro de información muy sencillo, de tal forma que lo convierten en un instrumento fácil de utilizar, ágil en el seguimiento y en la toma de medidas (9).

- **SAP**

SAP es uno de los Sistema de Recursos de Empresas (*Enterprise Resource System*, en inglés abreviado ERP) más utilizado en el mundo. Este sistema tiene muchos módulos integrados que abarca todas las áreas de una organización. Cada módulo realiza una función diferente, pero está diseñado para que la información adquirida se comparta entre todos sus módulos (10).

1.3.2. Sistemas de gestión económica en Cuba.

En Cuba se utilizan varios sistemas que posibilitan a empresas, entidades u organismos del estado llevar un sistema de registro y control de todos los hechos económicos que se evidencian a diario.

- **ASSETS**

ASSETS, es el sistema utilizado en la UCI para la gestión de sus procesos. Es un sistema flexible, con ayuda en línea, tiene pantallas de entradas de datos con opciones fáciles de interpretar y ejecutar, facilita el uso de la parametrización para adaptarse a las exigencias de cada cliente que lo utilice, emite reportes

¹ PYMES: corresponde a pequeñas y medianas empresas (8).

de la forma y el contenido que el usuario defina. También facilita la ejecución de auditorías contables para localizar errores de compatibilidad de datos. Este sistema siempre registra por parte de los clientes que lo trabajan las cifras que se presenten, así como el concepto por el que se utiliza la cifra (11).

- **Versat Sarasola**

Es uno de los primeros sistemas de contabilidad utilizado en Cuba. Se conforma por varios módulos que permiten llevar el control y registro contable individual de todos los hechos económicos que se originan en las entidades, facilitando el análisis y la evaluación de los resultados del negocio o actividad en tiempo real (12).

- **RODAS XXI**

Sistema multiempresa y multiusuario desarrollado en Cuba por CITMATEL para la automatización de la gestión empresarial. Rodas XXI contiene varios módulos de finanzas que contiene el control de la caja, que pueden ser utilizados integrados o separados. Este no permite la gestión de pago de bonificación, además está desarrollado sobre una plataforma propietaria (13).

- **CONDOR**

Sistema automatizado de alta complejidad y seguridad que abarca todos los aspectos del proceso contable de una entidad, tales como la dualidad de moneda y el pago por resultados. Está formado por varios módulos como activos fijos, contabilidad general, nóminas, control de inventarios, recursos humanos, entre otros. Este sistema no realiza pago de bonificación y es privativo (14), por lo que no se tiene acceso al código fuente.

Consideraciones de los sistemas de gestión económica

Después del estudio de las principales características de los sistemas informáticos antes expuestos, se llega a la conclusión de que todos realizan gestión de procesos económicos y que ninguno posee funcionalidad para pago llevar a cabo la bonificación del 50% del pasaje a estudiantes. Estos son propietarios o están desarrollados sobre plataformas privadas, por lo que hay que pagar por su utilización. Además no pueden ser reutilizados, debido a que no se tiene acceso a los códigos fuentes de dichos sistemas. Por lo que se hace necesario desarrollar un sistema de gestión de la información para la bonificación del 50% del pasaje a estudiante para la Vicerrectoría Económica de la UCI y que a su vez propicie crear un expediente económico por cada bonificación que se ejecute.

1.4. Metodologías de desarrollo de software.

La rama de la metodología dentro de la ingeniería de software, se encarga de estructurar, planificar y controlar el proceso de desarrollo del software, a través de un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar un nuevo software. A continuación se mencionan metodologías que fueron estudiadas para el desarrollo de la presente investigación.

1.4.1. Metodologías tradicionales.

Entre las metodologías de desarrollo del software existen las llamadas “tradicionales”, las cuales ayudan a los profesionales a documentar y realizar las tareas de desarrollo. Se aplican fundamentalmente a proyectos de gran tamaño (respecto a tiempo y recursos), donde el cliente interactúa con el equipo de desarrollo mediante reuniones, debiendo existir un contrato prefijado entre éstos que especifique el alcance del proyecto. Estas no resultan ser adecuadas para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad (15). Ante las dificultades de restricciones de tiempo y flexibilidad que poseen estas metodologías, otras más actuales han sido adoptadas para guiar el proceso de desarrollo, denominadas “metodologías modernas”, dentro de las cuales figuran las “metodologías ágiles” (16).

1.4.2. Metodologías ágiles.

Las metodologías ágiles, se fundamentan en la entrega temprana del software con el uso de métodos no formales, ya que enfocan su mayor esfuerzo en la elaboración y entrega del producto. Se nombran ágiles por la capacidad de responder rápida y efectivamente ante los cambios. Se apoyan en las habilidades y experiencias personales del equipo, evitando los extenuantes caminos de las metodologías tradicionales (16). Estas se basan en el “Manifiesto Ágil” el cual plantea los principales aspectos que se valoran en el desarrollo ágil, es decir, se valora (15):

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

Sobre la base de los aspectos antes expuestos se evidencia que las metodologías tradicionales no se ajustan a las particularidades del sistema a desarrollar, debido a que el equipo de trabajo es pequeño, no se cuenta con una definición detallada y exacta de todos los requisitos que debe cumplir el sistema, implicando una alta probabilidad de ocurrencia de cambios de forma gradual, a medida que se vayan obteniendo las versiones del sistema. Por ello, es más conveniente el uso de una de las metodologías ágiles, diseñadas para pequeños equipos de desarrollo y preparadas para enfrentar cambios durante la creación del software. ([Anexo 1](#))

A continuación se describe algunas de las metodologías ágiles existentes:

- **Programación Extrema**

Programación Extrema (*Extreme Programming*, en inglés abreviado XP) es una metodología de desarrollo de software que se basa en la satisfacción del cliente mediante un desarrollo iterativo e incremental, potenciando al máximo el trabajo en grupo, así como la comunicación fluida entre todos los participantes; aportando conocimiento a los desarrolladores, simplicidad en las soluciones implementadas y valor para enfrentar los cambios. Esta metodología es adecuada para proyectos con requisitos borrosos y donde

existe un alto riesgo técnico. Realiza prueba a los principales procesos para predecir errores futuros. Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio y programación en pares, lo que consiste en que la codificación del software siempre se realice por dos personas en una misma estación de trabajo (17).

- **Scrum**

Scrum es una metodología ágil dirigida a la gestión de proyectos. Precisa un proceso empírico e incremental del desarrollo de software. Sus iteraciones son de un proceso corto de 3 o 4 semanas y cada iteración va enfocada a la terminación de una pieza del software ejecutable que incorpora nuevas funcionalidades. Surge como modelo para el desarrollo de productos tecnológicos y actualmente se emplea en los entornos que trabajan con requisitos inconsistentes y requieren rapidez y flexibilidad. Busca entregar un software que realmente resuelva las necesidades del cliente, aumentando su satisfacción. Esta tiene un conjunto de reglas muy pequeño y muy simple y está basado en los principios de inspección continua, adaptación, auto-gestión e innovación (18).

- **Proceso Unificado Abierto**

Proceso Unificado Abierto (en inglés abreviado OpenUp) es una metodología planteada para equipos de desarrollo pequeños, lo que propicia obtener resultados en un corto período de tiempo. Es ágil e incremental y para proyectos de software basados en desarrollo iterativo. Plantea que se debe utilizar sólo los procesos que sean necesarios, sin demasiados artefactos y sobre todo que el proyecto debe acoplarse a las necesidades del usuario. Esta metodología tiene ventajas importantes como que disminuye los riesgos y puede utilizarse tanto en proyectos pequeños como en proyectos grandes, aunque está concebida para proyectos pequeños. Si se hace un correcto uso de esta se puede desarrollar un software de gran calidad, a pesar de que se le diseñe en poco tiempo y con poca documentación (19).

Consideraciones sobre las metodologías ágiles

Después del análisis de las metodologías de desarrollo de software antes expuestas, y en concordancia con las particularidades de la solución a desarrollar en la presente investigación, se elige la metodología XP para realizar el sistema de gestión de la información para la bonificación del 50% del pasaje a estudiantes, por que se ajusta a las características del sistema a implementar; pues el equipo de desarrollo está conformado por una sola persona, y la metodología en sus principios básicos plantea la programación en equipos pequeños con pocos roles, pudiendo los miembros del equipo intercambiar responsabilidades en un momento determinado. Otro argumento es que existe comunicación directa con el cliente, condicionando a que conteste rápido y correctamente a cualquier pregunta, de forma tal que no se atrase la toma de decisiones. Es además una metodología abierta a los cambios y genera poca documentación lo que hace la entrega del software menos complicada y más satisfactoria.

1.5. Lenguaje de modelado.

El primer paso al comenzar a diseñar un software es modelar el sistema, el modelado del mismo es de máxima importancia y tiene una notable repercusión en su etapa de diseño e implementación, por proveer al ingeniero de herramientas y prácticas, que le permiten "visualizar" el sistema a construir.

A continuación se presentan los lenguajes que se utilizarán para modelar el sistema de gestión de la presente investigación:

1.5.1. Lenguaje Unificado de Modelado.

UML (*Unified Modeling Language*, en inglés abreviado UML) es un lenguaje muy conocido y utilizado en el modelado de sistemas de software, permite visualizar, especificar, construir y documentar un proyecto de desarrollo de software. Ofrece un estándar para detallar un plano del sistema (modelo), el cual incluye características conceptuales como procesos de negocio, funciones del sistema, aspectos concretos como expresiones de lenguaje de programación, esquemas de bases de datos, entre otros (20).

1.5.2. Notación de Modelado de Procesos del Negocio.

La Notación de Modelado de Procesos del Negocio (*Business Process Modeling Notation*, en inglés abreviado BPMN) es un estándar para el modelado de procesos de negocio. Este agrupa la planificación y gestión del flujo de trabajo, así como el modelado y la arquitectura. Proporciona un lenguaje gráfico común, con el fin de facilitar su comprensión a los usuarios de negocios, utiliza una arquitectura orientada por servicios para adaptarse rápidamente a los cambios, además de componerse por eventos, actividades, objetos conectores, artefactos, entre otros (21).

1.6. Selección de la herramienta CASE de modelado.

Las herramientas de Ingeniería de Software Asistida por Ordenador (*Computer Aided Software Engineering*, en inglés abreviado CASE), son aplicaciones informáticas que brindan un conjunto de ayudas para el desarrollo de software. Se utilizan para el modelado y brindan a analistas e ingenieros una serie de herramientas y metodologías, además de guiarlos durante el ciclo de vida del desarrollo del software.

Actualmente entre las herramientas CASE se destacan Rational Rose y Visual Paradigm:

- **Rational Rose Enterprise Edition**

Es una herramienta CASE basada en UML, permite crear los diagramas que se generan durante el proceso de desarrollo del software. Se puede usar para producir modelos visuales que sirvan de apoyo y guía para el desarrollo del software.

Las ventajas de esta herramienta residen en su utilidad en aplicaciones grandes y complejas, ya que reduce el tiempo de desarrollo de manera automática para pasar de un esquema a otro y al código, además de que los nombres y los datos se mantienen de manera consistente proporcionando una sincronización para diferentes desarrolladores. Sin embargo tiene como desventajas los costos de la licencia, manuales y capacitación. (22)

- **Visual Paradigm**

Visual Paradigm es una herramienta Lenguaje de Modelado Unificado profesional que soporta el ciclo de vida completo del desarrollo de software. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir todos los tipos de diagramas de clases y crear documentación.

Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación.

Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. Además cumple con las políticas actuales de migración a software libre (23).

Consideraciones sobre las herramientas CASE

Teniendo en cuenta las características y los beneficios de las herramientas CASE antes expuestas, se decidió utilizar para el desarrollo del sistema en cuestión Visual Paradigm for UML, para el modelado del sistema, en su versión 8.0, debido a que genera código para entornos integrados de desarrollo, posee más documentación de ayuda y es multiplataforma a diferencia de Rational Rose Enterprise Edition.

1.7. Selección de la herramienta de desarrollo del software.

En la actualidad los desarrolladores de sistemas no comienzan desde cero, pues existen herramientas que les permite crear y mantener un sistema con facilidad, algunas de estas son los Sistemas de Gestión de Contenidos (*Content Management Systems*, en inglés abreviado CMS) y Marcos de trabajo (en inglés Frameworks).

Los Frameworks permiten simplificar el desarrollo de un sistema mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporcionan estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas (24).

Los CMS permiten la creación y administración de contenidos, facilitan el manejo del contenido y el diseño de forma independiente, es decir, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sistema sin tener que dar formato al contenido de nuevo. Además de permitir controlar una o varias bases de datos (24).

Después de un estudio de las principales características de los Frameworks y los CMS, se escoge para desarrollar el Sistema de gestión de la información para la bonificación del 50% del pasaje a estudiantes a los CMS; debido a que el tiempo de implementación de la solución propuesta es corto y existen CMS que presentan entre sus características parte de los requisitos que se requieren en la aplicación. Además

están orientado en principio al usuario final, es decir, lo que el cliente necesita es un producto con calidad sin importarle el método con el que se desarrolla (25).

1.7.1. Selección del CMS.

Los CMS facilitan la gestión de contenidos, control de acceso y de permisos. Propician disminución de costo y esfuerzo, es decir, minimizan la necesidad de conocimientos técnicos en cuanto a programación se refiere y dan la facilidad de incluir nuevas funcionalidades a través de módulos realizado por terceros. Para la elección del CMS a utilizar se realizó un estudio de los más populares y usados en la actualidad en cuanto a un grupo de criterios que representan las funcionalidades deseadas para el sistema de gestión.

Algunos de estos criterios fueron:

- Licencia: Tipo de licencia bajo la que es distribuido el CMS.
- Base de Datos: El gestor de bases de datos que utiliza el CMS para almacenar sus contenidos.
- Servidor Web: Servidores web con los cuales es compatible.
- Lenguaje: El lenguaje de programación en que está escrito el CMS.
- Sistema operativo: Sistemas operativos con los que es compatible.
- Autenticación LDAP²: ¿Permite el sistema autenticación basada en LDAP?

Los CMS encontrados fueron: Joomla!, WordPress y Drupal.

- **Joomla!**

Joomla! es un sistema de gestión de contenidos que surge en el 2005, permite editar el contenido de manera sencilla, organizar y facilitar la creación de documentos. Es una aplicación de código abierto, bajo una licencia GPL, requiere de una base de datos MySQL, servidor web Apache o IIS³(*Internet Information Services en inglés*), lenguaje de programación PHP. Además es multiplataforma, admite autenticación a través de LDAP, permite granularidad de privilegios, es modular, posee gran volumen de documentación y una comunidad potente (27).

- **WordPress**

WordPress es un CMS que al inicio fue orientado al desarrollo de Blogs, pero ha evolucionado hasta convertirse en unos de los CMS de uso más completo en el mercado. Es libre, está programado en PHP, requiere de base de datos MySQL, utiliza servidor web Apache y posee granularidad de privilegios, pero no permite autenticarse a través de LDAP. También posee una gran comunidad activa, que es el corazón de cualquier proyecto de software de código abierto (27).

² Lightweight Directory Access Protocol, en español Protocolo Ligero de Acceso a Directorios.

³ Servidor web con un conjunto de servicios para el sistema operativo Microsoft Windows (26).

- **Drupal**

Drupal es una plataforma dinámica preferentemente utilizada para la construcción de sitios web, es un sistema modular de gestión de contenidos. Escrito basado en código PHP, utiliza PostgreSQL o MySQL como base de datos y tiene Licencia GNU/GPL⁴. Es de código abierto con una comunidad de usuarios y desarrolladores a su alrededor muy extensa, de variados idiomas y conocimientos, esto lo ha hecho crecer y posicionarse encima de los demás CMS. Integra muchas características populares de los CMS, weblogs, herramientas de colaboración y comunidad de discusión, todo en un solo paquete fácil de utilizar, por lo que se ha ganado por mucho tiempo la condecoración de ser idóneo para cualquier trabajo sobre la web (27)

- **Consideraciones de los CMS**

Después de un estudio de las características de los CMS antes expuestos, se selecciona para desarrollar el sistema de gestión de pago de la bonificación del 50% del pasaje a estudiantes al CMS Drupal en su versión 7.26, pues este ofrece un conjunto de facilidades como la gestión de roles y permisos a dichos roles. Posee en su núcleo varias funcionalidades por defecto a través de módulos. Dicha modularidad facilita la instalación de nuevos módulos los cuales aportan más funcionalidades. Además de poseer todas las características de los CMS antes mencionados y tener más documentación y comunidades de desarrollo.

1.8. Lenguajes de programación.

Los lenguajes de programación son herramientas que permiten crear programas y software. Facilitan la tarea de programación, pues disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de computadora a utilizar (28).

1.8.1. Lenguaje de programación del lado del cliente.

Los lenguajes del lado del cliente son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento, estos son totalmente independientes del servidor. Para el desarrollo del sistema en cuestión se utilizarán:

- **Javascript 1.8**

Javascript es un lenguaje de guiones, que incorpora algunas convenciones del lenguaje Java, orientado a eventos y basado en objetos. Es una tecnología del lado del cliente por lo que permite interactuar con casi todos los navegadores de manera dinámica y eficaz. Es importante conocer que es muy sensible al uso indebido de mayúsculas. Permite dinamismo en las páginas HTML (*HyperText Markup Language*, en inglés) y la programación orientada a objetos. Es soportado por navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox (29).

⁴ Licencia Pública General en inglés *General Public License*, es la licencia más ampliamente usada en el mundo del software y garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

- **CSS 3**

Las Hojas de Estilo en Cascada (*Cascading Style Sheets*, en inglés abreviado CSS) es una tecnología que permite crear páginas web, a través de la cual se le da estilo a una página siendo los desarrolladores los dueños de los resultados finales que se observan, con esta se puede especificar márgenes, tipos de letra, fondos y colores.

1.8.2. Lenguaje interpretado del lado del servidor.

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Estos lenguajes son independientes del cliente por lo que es mucho menos rígido respecto al cambio de un navegador.

Siendo consecuente por la selección del CMS para el desarrollo del sistema y teniendo en cuenta que está concebido con PHP, se utilizará PHP en su versión 5.3 para el desarrollo del sistema de gestión de pago de la bonificación del 50% de pasaje a estudiantes.

- **PHP**

PHP es un lenguaje interpretado del lado del servidor, que tiene mucha utilidad por su rapidez y facilidad de hacer determinadas acciones, su código es embebido en páginas HTML. Es un lenguaje multiplataforma, no es privativo por lo que es de fácil acceso para todos (22).

1.9. Entorno de desarrollo integrado.

El entorno de desarrollo integrado (*Integrated Development Environment*, en inglés abreviado IDE) es un sistema que facilita el trabajo del desarrollador de software, el cual está compuesto por un conjunto de herramientas que permite programar en varios lenguajes de programación, es decir; es un editor de código, compilador, depurador y constructor de interfaces gráficas. El IDE escogido para trabajar en la implementación del sistema por especificación del cliente es PhpStorm en su versión 7.0.

- **PhpStorm 7.0**

PhpStorm es un entorno de desarrollo que les permite a los programadores escribir, compilar, depurar y ejecutar programas. Está basado en Java pero sirve para otros lenguajes de programación. Incluye soporte para PHP 5, integra Vagrant y consola SSH. Posee refactorización e inspecciones de código. Soporta varias tecnologías del front-end como por ejemplo plantillas JavaScript (EJS, Handlebars, Mustache), componentes Web, Stylus, Compass y entre otras. Además esta versión es muy rápida y ligera, por lo que mejora la velocidad de exploración del sistema. Es muy recomendable para el trabajo con CMS como Drupal (30).

1.10. Selección del sistema Gestor de Base de Datos (SGBD).

Los SGBD son un tipo de software dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que los utilizan. El propósito general de los SGBD es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante.

Siendo consecuente por la selección del CMS Drupal para el desarrollo del sistema y teniendo en cuenta que está concebido para trabajar con los SGBD, MySQL y PostgreSQL, solo se analizarán estos, en aras de seleccionar el más adecuado para utilizar en la propuesta de solución.

- **MySQL**

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia. Es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Es muy utilizado en aplicaciones web por su rapidez en la lectura de datos. El lenguaje de programación que utiliza MySQL es *Structured Query Language* (en inglés abreviado SQL). Unas de las principales desventajas que presenta es que; un gran porcentaje de las utilidades de MySQL no están documentadas y no es intuitivo, como otros programas. (31)

- **PostgreSQL**

PostgreSQL es un Sistema de Base de Datos Relacional libre (*Open Source*, su código fuente está disponible). Posee más de 15 años de activo desarrollo y arquitectura probada que se ha ganado una muy buena reputación por su confiabilidad e integridad de datos. Funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX (AIX, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Posee una gran escalabilidad, implementa subconsultas y transacciones. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle. (16)

Consideraciones sobre los sistemas de gestión de base de datos.

Después del análisis de los SGBD antes expuestos, se llega a la conclusión de que ambos se encuentran muy nivelados en cuanto a sus características. No obstante se escoge a PostgreSQL en su versión 9.1, ya que es de licencia libre y posee mayor documentación a diferencia de MySQL en sus versiones más actuales.

1.11. Servidores de aplicaciones web.

Un servidor de aplicaciones web es un software encargado de aceptar las peticiones de las páginas que provienen de los usuarios que acceden a un sitio. Este implementa el Protocolo de Transferencia de Hipertexto (*Hyper Text Transfer Protocol*, en inglés abreviado HTTP) y se ejecuta constantemente manteniéndose a la espera de peticiones.

Algunos ejemplos de servidores web son: Cherokee, Nginx, Lighttpd y Apache.

Para el desarrollo del sistema de gestión de la información para la bonificación del 50% del pasaje a estudiantes, se escoge como servidor web Apache ya que este es un servidor de código abierto, tiene entre sus ventajas arquitectura modular, multiplataforma, extensible y altamente configurable. Permite la creación de ficheros de archivos de registro a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor (32).

1.12. Conclusiones Parciales.

Luego de haberse realizado un estudio de diferentes sistemas contables y sus características así como del proceso del pago de la bonificación del 50% del pasaje a estudiantes, se llega a la conclusión que la selección de Drupal como CMS, Visual Paradigm como herramienta CASE, para modelar UML y BPMN, como lenguaje de programación PHP, Javascript y CSS, PhpStorm como IDE de desarrollo, Apache como servidor web, PostgreSQL como SGBD y para apoyar el ciclo de vida del software, la metodología XP, propiciaran la creación del sistema propuesto.

CAPÍTULO II: CARACTERÍSTICAS DE LA SOLUCIÓN PROPUESTA.

2.1. Introducción

En el capítulo se exponen las principales características del sistema a implementar, mediante el modelado del negocio, los requisitos funcionales y no funcionales y las historias de usuarios relacionadas a estos. En correspondencia con la etapa de planificación del sistema se puntualiza el plan de entregas, el de iteraciones y las tareas ingenieriles. Se describe la arquitectura por la cual se regirá la construcción de la solución, los patrones de diseño y los artefactos que serán utilizados para cumplir con el objetivo que dará solución al problema que dio origen a la presente investigación.

2.2. Descripción general de la propuesta de solución.

El proceso de pago de la bonificación del 50% de pasaje a estudiantes en la Vicerrectoría Económica de la UCI, se realiza de forma manual, lo que implica algunos inconvenientes como falta de inmediatez requerida a la hora de dar un reporte. No se tiene conocimiento preciso de las bonificaciones realizadas en un periodo de tiempo y del monto que alcanzan las mismas. Otra dificultad es que no existe comunicación inmediata entre la caja de la Universidad y la Vicerrectoría Económica, trayendo desconocimiento del día real en que el estudiante cobra el 50% del pasaje. Además no se posee un expediente contable por cada estudiante, por lo que se desconoce la cantidad de veces que este hace uso del pago de la bonificación del 50% de pasaje. El sistema de gestión a desarrollar automatizaría este proceso, permitiendo a la Vicerrectoría Económica de la UCI mejorar la gestión de la información relacionada con la bonificación.

2.3. Modelado de procesos de negocio.

El modelado de procesos del negocio facilita entender y organizar algunos procesos que están conformados de múltiples actividades, personas y departamentos en una organización. Su uso trae como beneficios conocer de qué trata el proceso de principio a fin, revelar las fallas y problemas, así como generar soluciones a los problemas detectados.

El modelado de los procesos del negocio se realizará mediante la notación BPMN, a través de la cual se puede realizar diferentes niveles de procesos. Los procesos de negocio identificados se muestran en los siguientes diagramas:

2.3.1. Mapa de procesos.

Para el modelado de los procesos se utilizan mapas de procesos, estos son un simple diagrama de flujo con pocos detalles, es decir, solo con los nombres de las actividades y tal vez las condiciones de decisión más generales (30)

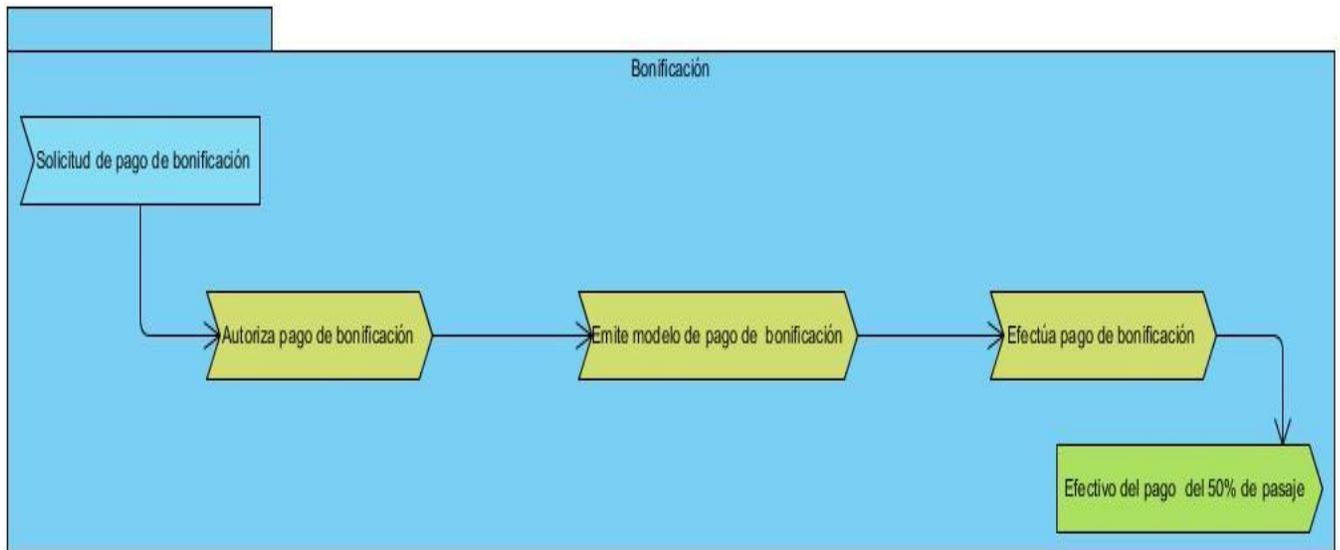


Figura 1. Mapa de procesos de la bonificación.

2.3.2. Descripción de los procesos del negocio.

Los modelos de procesos son diagramas de flujo más detallados con suficiente información, para poder estudiar el proceso y simularlo como se muestra a continuación.

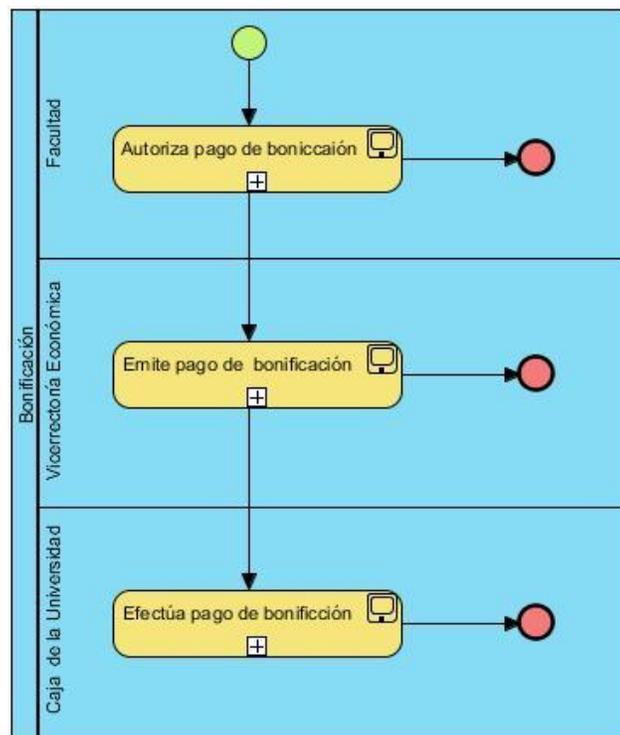


Figura 2: Proceso de negocio de la bonificación.

- **Proceso de pago de bonificación del 50% del pasaje a estudiantes en la UCI.**

En el VII Congreso de la FEU celebrado en diciembre de 2007, se decide bonificar a los estudiantes pertenecientes al Ministerio de Educación Superior (MES), con el pago del 50% del pasajes por concepto de bonificación, en dependencia del tipo de transporte (barco, guagua, tren) que utilice. En la UCI el pago de la bonificación se realiza como proceso de liquidación de dieta, a partir de un mecanismo expuesto en la Resolución No. 53/2011 emitida por el rector de la universidad la cual plantea que:

- El estudiante sólo tiene derecho de disfrutar por cada mes del curso una bonificación, la cual consiste en reintegrarle al estudiante el 50% del pasaje que consuma.
- Antes de que el estudiante proceda a cobrar la bonificación, debe dirigirse a la dirección de su Facultad con los comprobantes de viajes para solicitar dicho proceso dentro de las 72 horas posterior a la fecha del pasaje.
- El estudiante debe entregar el modelo que emite el decano de su facultad y los comprobantes de viajes dentro del mismo pazo de tiempo antes mencionado en la Vicerrectoría Económica.
- Después de generado el modelo de pago de bonificación en la Vicerrectoría Económica y recogido por el estudiante, este tiene 15 días para cobrar el efectivo en caja.

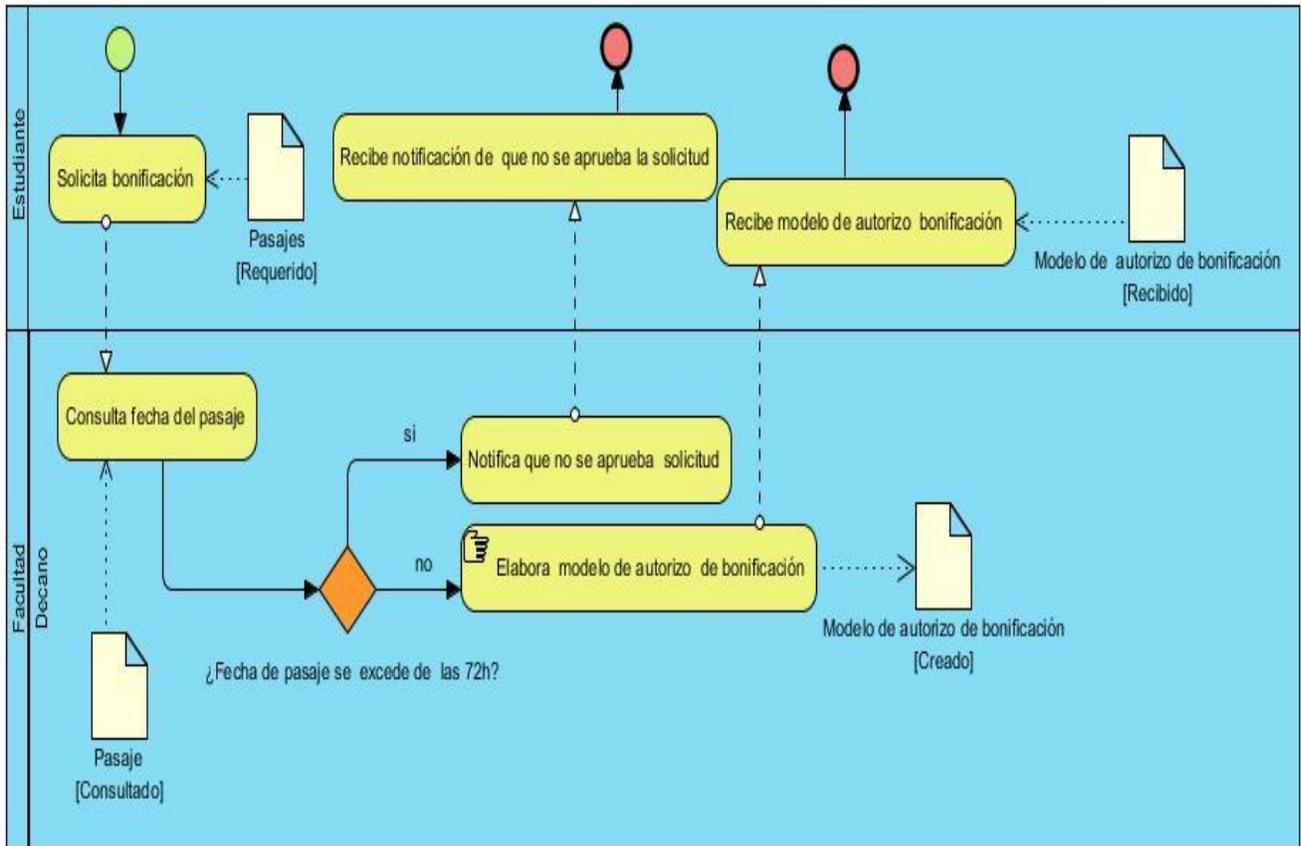


Figura 3: Proceso de negocio: Autorizar pago de la bonificación.

Autoriza pago de bonificación: El estudiante se presenta en la facultad para solicitar el pago del 50% de pasaje por concepto de bonificación, para ello, requiere llevar consigo el comprobante de pasaje. En la

Facultad es atendido por el Decano o Vicedecano Administrativo, los cuales consultan la fecha de dicho comprobante y verifica que no se exceda de las 72 horas llegada al centro por parte del estudiante, si no se excede, el Decano o Vicedecano Administrativo confeccionan el modelo de autorizo de pago de bonificación (Anexo 2) y se lo hace llegar al estudiante; si no, notifica al estudiante que su solicitud no procede.

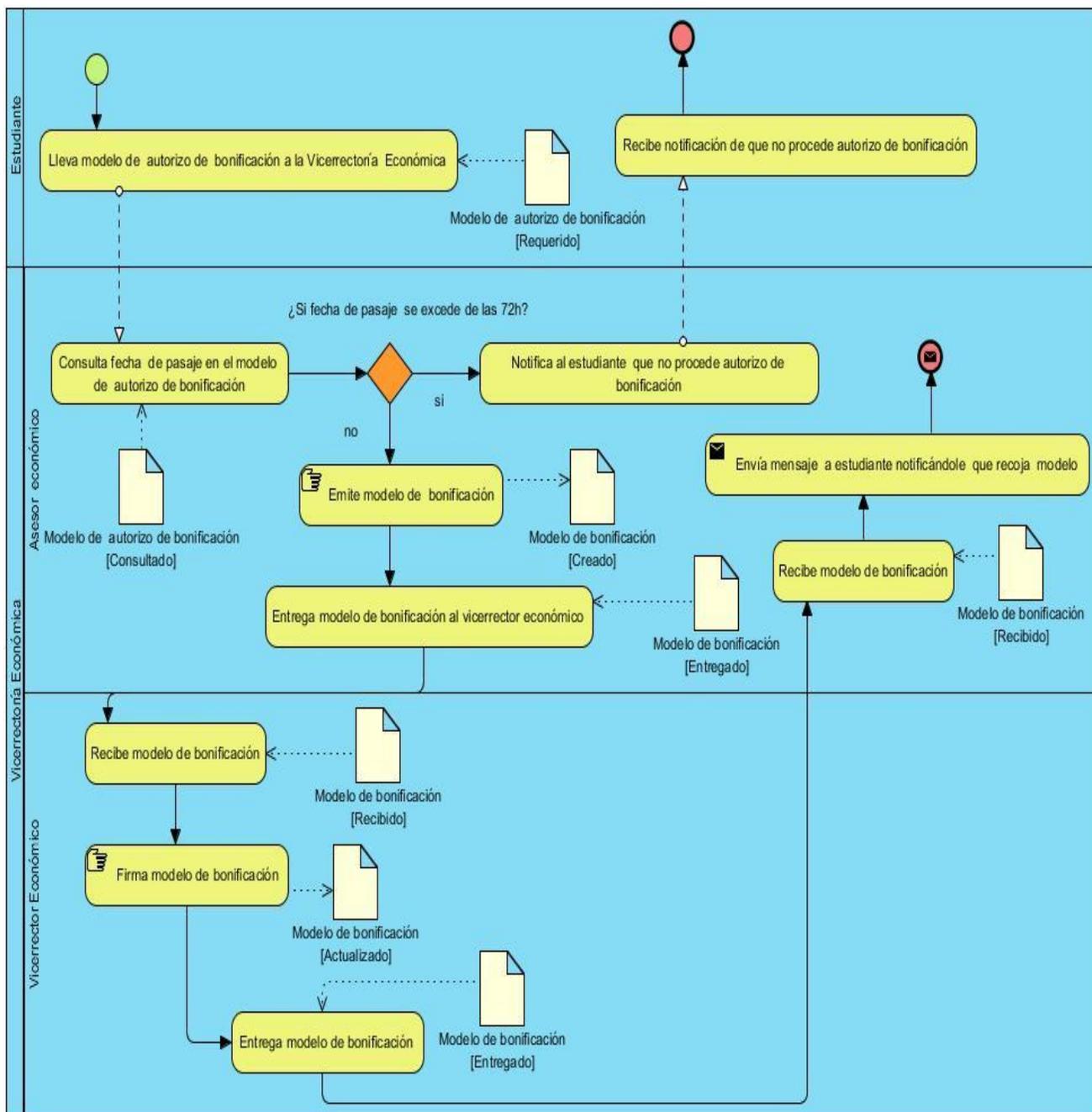


Figura 4: Proceso de negocio: Emitir modelo de pago de la bonificación.

Emitir modelo de pago de la bonificación: El estudiante lleva el modelo de autorizo de pago de la bonificación a la Vicerrectoría Económica, el asesor del Vicerrector Económico consulta dicho autorizo y verifica que la fecha del pasaje que está adjunto al modelo no se exceda de las 72 horas de la llegada al

centro del estudiante, si no se excede, elabora el modelo de bonificación (Anexo 3) y se lo hace llegar al Vicerrector Económico. Este último firma el modelo y se lo entrega al asesor, el cual envía un correo al estudiante notificándole que la bonificación se ha firmado y que puede pasar a recogerla para posteriormente la cobre en la caja de la Universidad.

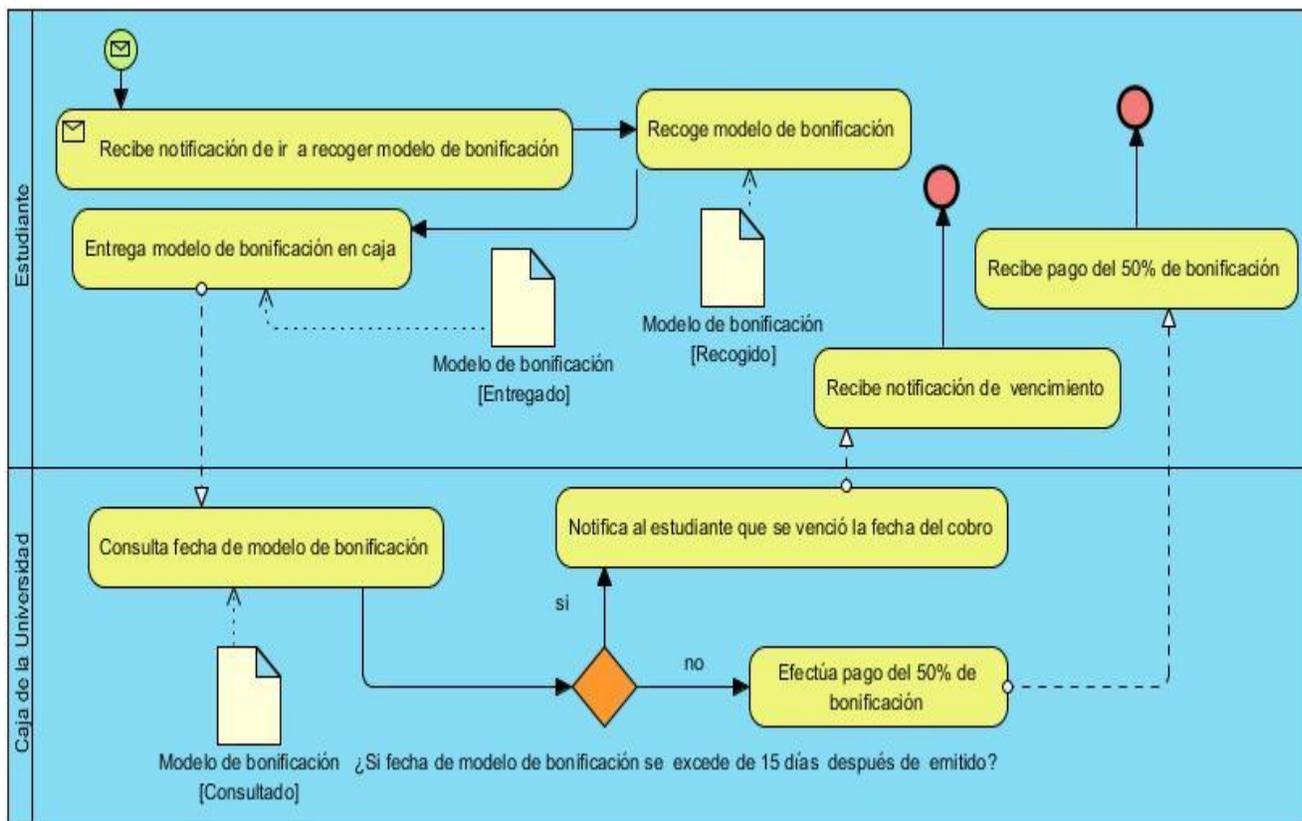


Figura 5: Proceso de negocio: Efectuar pago de la bonificación.

Efectúa pago de bonificación: El estudiante recibe un correo notificándole por parte del asesor del Vicerrector Económico que puede realizar el cobro de la bonificación en la caja del centro. Este recoge el modelo de bonificación y se dirige a la caja de la Universidad. En la caja, la cajera consulta la fecha de creación del modelo, y verifica que no se exceda de los 15 días de emitido por el asesor y formado por el Vicerrector Económico, si este no se excede efectúa el pago del 50% del pasaje al estudiante y en caso de tener problema le notifica al estudiante que no puede efectuar pago y que la bonificación solicitada queda anulada.

2.3.3. Actores y trabajadores que intervienen en el proceso del negocio.

- **Estudiante:** persona que interviene en el proceso de solicitud de la bonificación del 50% del pasaje así como del cobro de la misma.
- **Decano:** persona que interviene en el proceso de autorizar el pago de la bonificación.
- **Vicedecano Administrativo:** persona que al igual que el Decano interviene en el proceso de autorizar el pago de la bonificación.

- **Asesor Económico:** persona que interviene en el proceso de emisión del modelo de pago de la bonificación. Realiza la gestión del modelo de pago de la bonificación.
- **Vicerrector Económico:** persona que interviene en el proceso emitir el modelo de pago de la bonificación. Se encarga de autorizar el modelo de pago de la bonificación.
- **Cajera:** persona que interviene en el proceso de efectuar el pago de la bonificación. Se encarga de pagar el 50% de pasaje a estudiantes por concepto de bonificación.

2.4. Requisitos del sistema.

Los requisitos son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos se clasifican en requisitos funcionales o no funcionales. Los funcionales declaran los servicios que debe brindar el sistema, de la manera que éste debe reaccionar y funcionar ante una entrada o situación en particular y los no funcionales son las restricciones de los servicios o funciones ofrecidas por el sistema (33).

El flujo de trabajo de requisitos es uno de los más importantes para el desarrollo de un software, ya que su propósito fundamental es guiar el desarrollo hacia el sistema correcto (34). Para la adquisición de requisitos se emplean algunas técnicas como entrevistas, cuestionarios y tormenta de ideas. Las técnicas seleccionadas para el levantamiento de los requisitos del sistema a desarrollar fueron entrevistas y tormenta de ideas y se llevaron a cabo a: Asesor del Vicerrector Económico, cajera así como a los administrativos de las facultades del centro universitario.

A continuación se mencionan los requisitos identificados:

2.4.1. Listado de requisitos funcionales.

Los requisitos funcionales (RF) que describen las funcionalidades que el sistema debe cumplir son:

RF-1. Autenticar usuario.	RF-8. Eliminar bonificación.	RF-15. Editar presupuesto.
RF-2. Crear rol a usuario.	RF-9. Imprimir bonificación.	RF-16. Eliminar presupuesto.
RF-3. Eliminar rol a usuario.	RF-10. Registrar fecha de cobro.	RF-17. Mostrar presupuesto.
RF-4. Editar rol a usuario.	RF-11. Buscar bonificación.	RF-18. Generar reportes.
RF-5. Buscar estudiante.	RF-12. Mostrar bonificación.	RF-19. Exportar reportes a Excel.
RF-6. Eliminar estudiante.	RF-13. Buscar presupuesto.	RF-20. Exportar reportes a Word.
RF-7. Adicionar bonificación.	RF-14. Agregar presupuesto.	

2.4.2. Listado de requisitos no funcionales.

Se definieron como requisitos no funcionales (RNF):

- **Usabilidad:**

RNF-1 El sistema debe ser intuitivo y fácil de navegar, pues se utilizará nombres y mensajes sugerentes para que el usuario realice la acción que desea.

RNF-2 El sistema debe posibilitar a los usuarios llegar al contenido que desea en un corto tiempo, siempre que no fuerce la estructura del sistema, ninguna página debe encontrarse a más de tres clic de la página de inicio.

- **Confiabilidad:**

RNF-3 El sistema validará la captación de datos para evitar entradas inadecuadas.

- **Fiabilidad**

RNF-4 El sistema debe registrar cualquier fallo en su funcionamiento para que los administradores puedan resolverlo y de la misma forma mostrarle al usuario, un mensaje indicándole que ha ocurrido un fallo en la operación que realice.

- **Seguridad:**

RNF-5 Se garantizará la integridad y confidencialidad de la información mediante mecanismos de control de acceso no autorizados utilizando: usuario, contraseña y definiendo niveles de acceso para cada usuario, de manera que cada usuario pueda tener disponible solamente las opciones relacionadas con su actividad y tenga datos de acceso propios.

RNF-6 Se podrá acceder a las páginas de administración del sitio web y las páginas de usuarios a través del protocolo https.

RNF-7 El sistema debe permitir que cuando se borre cualquier información pueda existir una opción de advertencia antes realizar la acción.

RNF-8 Los mensajes de error mostrados a los usuarios deben ser genéricos sin dar detalles de información, para no comprometer la seguridad e integridad de los datos.

RNF 9 Se realizarán salvallas periódicas de la información en otros dispositivos.

- **Soporte:**

RNF-10 El sistema será instalado y probado donde lo necesite el cliente.

- **Apariencia o Interfaz externa**

RNF-11 La interfaz de usuario debe diseñarse para una resolución de 1024 x 768.

RNF-12 El diseño de la interfaz debe ser sencillo e intuitivo, es decir fácil de comprender.

RNF-13 Se debe garantizar una correcta organización de la información para permitir una adecuada interpretación.

RNF-14 Debe ser interactiva y con uso de colores adecuados.

- **Interfaces de Hardware**

PC cliente:

RNF-15 Los requerimientos mínimos para las PC clientes son el uso de una computadora Core 2duo, con una velocidad del procesador de 2.20 GHz, 512 MB de RAM.

Servidor de BD:

RNF-16 El servidor de BD debe ser Core 2duo, con 1GB de RAM y 80 GB de disco duro.

Servidor de Aplicaciones:

RNF-17 El servidor de aplicaciones debe ser Core 2duo, con 1GB de RAM y 80 GB de disco duro.

- **Interfaces Software**

RNF-18 Sistema Operativo: Ubuntu 12.04 LTS o cualquier versión LTS superior.

RNF-19 Servidor de aplicaciones: Apache 2.0 o superior. Lenguaje de programación: PHP 5.3 o superior.

RNF-20 Servidor de Base de Datos: Sistema Operativo: Ubuntu 12.04 LTS o superior. Sistema Gestor de Base de Datos PostgreSQL versión 9.1.

2.5. Historias de Usuario (HU).

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Estas describen brevemente las características que el sistema debe poseer. Cada HU debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarla.

Estas se usan para estimar el tiempo y el plan de lanzamiento los cuales dirigen la creación de las pruebas de aceptación. Se caracterizan por ser independientes una de otras, negociables, valoradas por los clientes o usuarios, estimables y verificables (17).

A continuación se muestra unas de las HU preliminares con alta prioridad para el negocio. Esta es solo una planificación inicial, el proceso es cambiante para ir adecuándolo a las necesidades y nuevas propuestas. Todas las decisiones se toman de conjunto con el cliente que es parte del equipo de desarrollo. [\(Anexo 4\)](#)

Tabla 1: HU Adicionar Bonificación.

Historia de Usuario	
Número: HU_4	Nombre Historia de Usuario: Adicionar bonificación
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Media	Puntos Reales: 1 semana
Descripción: La HU inicia el usuario autenticado en el sistema con privilegios de administración accede al módulo: “Buscar estudiantes” , que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema le muestra al usuario un listado con los estudiantes existentes en el sistema y sus respectivos datos, seguida de las opciones: “Eliminar” y “Adicionar bonificación” . El usuario selecciona la opción: “Adicionar bonificación” y el sistema le muestra un formulario con los siguientes campos: <ul style="list-style-type: none">• Fecha de ida (input de tipo date)• Fecha de regreso(input de tipo date)• Valor de pasaje de ida (campo de texto)• Valor de pasaje de regreso (campo de texto)	

- Destino (combobox)
- Regreso (combobox)

El usuario registra la información y presiona el botón: **“Guardar”**. El sistema verifica y actualiza la información, finalizando así la HU.

Observaciones: Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.

Prototipo de interfaz:

Tabla 2: Imprimir bonificación.

Historia de Usuario	
Número: HU_6	Nombre Historia de Usuario: Imprimir bonificación
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana
Descripción:	
<p>La HU inicia el cuando el usuario autenticado en el sistema con privilegios de administración accede al módulo: “Gestionar bonificaciones”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema le muestra al usuario una interfaz con un listado de las bonificaciones existentes en el sistema que se encuentran las “Cobradas” y las “Pendientes” por cobrar. El usuario selecciona la opción de “Cobradas”, luego busca la bonificación que desea imprimir y selecciona la opción: “Imprimir”. El sistema muestra un formulario con las opciones de impresión seguida de las opciones: “Cancelar” e “Imprimir”.</p> <p>El usuario selecciona la opción “Cancelar”, el sistema deshace la operación de impresión.</p> <p>El usuario selecciona la opción “Imprimir”, el sistema imprime el modelo de la bonificación, finalizando así la HU.</p>	
Observaciones:	

Prototipo de interfaz:

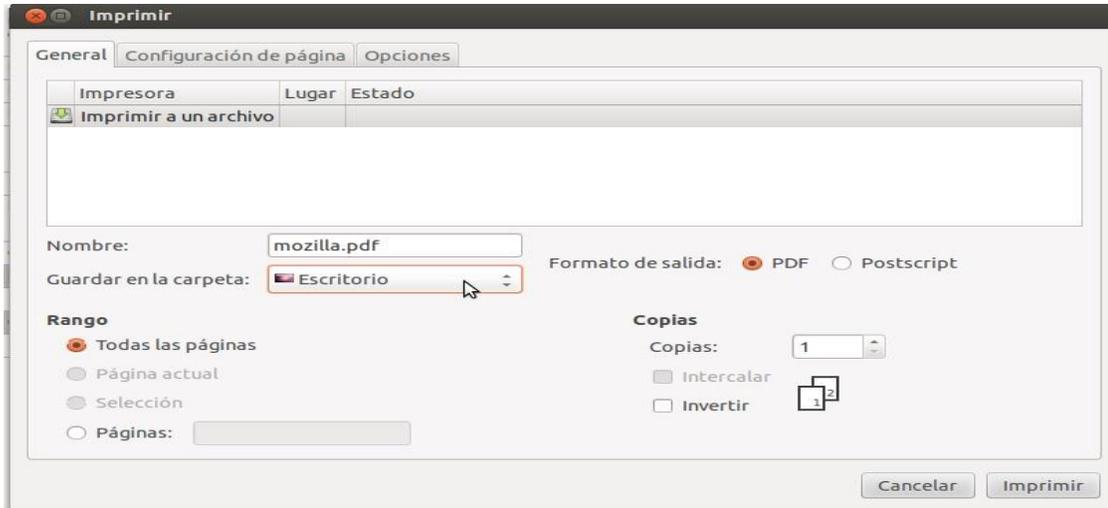


Tabla 3: HU Registrar fecha de cobro.

Historia de Usuario	
Número: HU_7	Nombre Historia de Usuario: Registrar fecha de cobro
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana
Descripción:	
<p>La HU inicia el cuando el usuario autenticado en el sistema con permiso de acceso al módulo: “Gestionar bonificaciones”, que se encuentra ubicado en la parte lateral izquierda del sistema accede a este. El sistema le muestra al usuario una interfaz con un listado de las bonificaciones existentes en el sistema “Pendientes” por cobrar, seguida de la opción: “Registrar cobro”. El usuario selecciona la opción: “Registrar cobro”. El sistema muestra un formulario con el siguientes campo:</p> <ul style="list-style-type: none"> • Fecha de cobro (input de tipo date) <p>El usuario registra la información y presiona el botón: “Guardar”. El sistema verifica y almacena la información, finalizando así la HU.</p>	
Observaciones:	
<ul style="list-style-type: none"> • Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. 	

Prototipo de interfaz:

FECHA COBRO

Fecha

E.g., 29/03/2014

2.6. Planificación

2.6.1 Plan de entregas.

En esta fase el cliente establece la prioridad de cada HU, el programador realiza una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

Se propone el siguiente plan de entregas para la solución propuesta:

Tabla 4: Plan de entrega del sistema.

Entregables	Iteración	Fin de la iteración
Autenticar usuarios	1	Marzo 2014
Buscar estudiante	1	Marzo 2014
Eliminar estudiante	1	Marzo 2014
Adicionar bonificación	1	Marzo 2014
Eliminar bonificación	1	Marzo 2014
Imprimir bonificación	1	Marzo 2014
Registrar fecha de cobro	1	Marzo 2014
Buscar bonificación	1	Abril 2014
Mostrar bonificación	1	Abril 2014
Buscar presupuesto	2	Abril 2014
Agregar presupuesto	2	Abril 2014
Editar presupuesto	2	Abril 2014
Eliminar presupuesto	2	Abril 2014
Mostrar presupuesto	2	Abril 2014
Generar reportes	2	Abril 2014
Exportar reportes a Excel	2	Abril 2014
Exportar reportes a Word	2	Abril 2014

Crear rol a usuario	2	Abril 2014
Eliminar rol a usuario	2	Abril 2014
Editar rol a usuario	2	Abril 2014

2.6.2 Plan de Iteraciones.

El ciclo de desarrollo de software guiado por XP se caracteriza por ser iterativo e incremental, por lo que se realizan varias iteraciones sobre el sistema antes de su fase de producción. Para elaborar este plan se deben tomar en cuenta las HU no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas.

El sistema de gestión de la información para la bonificación del 50% del pasaje a estudiante se realiza en dos iteraciones, en la primera iteración se desarrollan las HU principales para el funcionamiento básico del sistema.

Tabla 5 Plan de iteraciones.

Iteración	Historia de Usuario	Semanas estimadas
1	Autenticar usuarios	1
	Buscar estudiante	
	Eliminar estudiante	
	Adicionar bonificación	
	Eliminar bonificación	
	Imprimir bonificación	
	Registrar fecha de cobro	
	Buscar bonificación	
	Mostrar bonificación	
2	Buscar presupuesto	1
	Agregar presupuesto	
	Editar presupuesto	
	Eliminar presupuesto	
	Mostrar presupuesto	
	Generar reportes	
	Exportar reportes a Excel	
	Exportar reportes a Word	
	Crear rol a usuarios	
	Eliminar rol a usuarios	
	Editar rol a usuarios	

Las tareas de ingeniería son tablas donde se especifican las tareas a realizar por cada HU descritas. Las tareas de programación o ingenieriles a realizar en cada una de estas iteraciones se especifican en la próxima tabla y en los anexos se detallan los elementos y descripción de cada una de estas tareas. (Anexo 5)

Tabla 6: Distribución de tareas de ingeniería por iteración.

Iteración	Historia de Usuario	Tareas de Ingeniería
1	Autenticar usuarios	Autenticar usuario con usuario y contraseña de LDAP
	Buscar estudiante	Buscar estudiante desde el directorio de LDAP
	Eliminar estudiante	Eliminar estudiante seleccionado, de la base de datos del sistema.
	Adicionar bonificación	Gestionar bonificación
	Eliminar bonificación	
	Registrar fecha de cobro	
	Mostrar bonificación	
	Buscar bonificación	Consultar resultados de la búsqueda
	Imprimir bonificación	Imprimir modelo.
2	Buscar presupuesto	Gestionar Presupuesto.
	Agregar presupuesto	
	Editar presupuesto	
	Eliminar presupuesto	
	Mostrar presupuesto	
	Generar reportes	Generar reporte de la información especificada por el usuario.
	Exportar reportes a Excel	Configurar módulo para exportar a xls
	Exportar reportes a Word	Configurar módulo para exportar a doc
	Crear rol a usuarios.	Gestionar roles
	Eliminar rol a usuarios.	
	Modificar rol a usuarios.	
	Modificar permisos a rol.	

2.7 Diseño.

El papel del diseño en el ciclo de vida del software es adquirir conocimiento de su funcionamiento, constituye el punto de partida para las actividades de implementación, dando soporte a los requisitos funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables y sistemas operativos, que debe poseer la aplicación (16).

2.7.1 Descripción de la arquitectura.

La arquitectura del software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución. La arquitectura entre sus objetivos define los módulos principales del sistema, la

responsabilidad que tendrá cada uno de los módulos, la interacción entre los mismos y controlar el flujo de los datos en el sistema. Cada sistema posee una arquitectura propia, la cual esta es definida según el tipo de sistema.

Los CMS poseen una arquitectura diferente de los demás sistemas, pues proporcionan una estructura de soporte para el desarrollo, principalmente, en páginas web. El sistema del pago del 50% de pasaje a estudiantes por concepto de bonificación está implementado en el CMS Drupal, por ello la arquitectura que se hereda de este, que es “una arquitectura modular que permite extender sus funcionalidades a través de unos métodos uniformes de desarrollo e integración de nuevos módulos”(35). La cual está conformada por los siguientes elementos como se muestra en la figura siguiente:

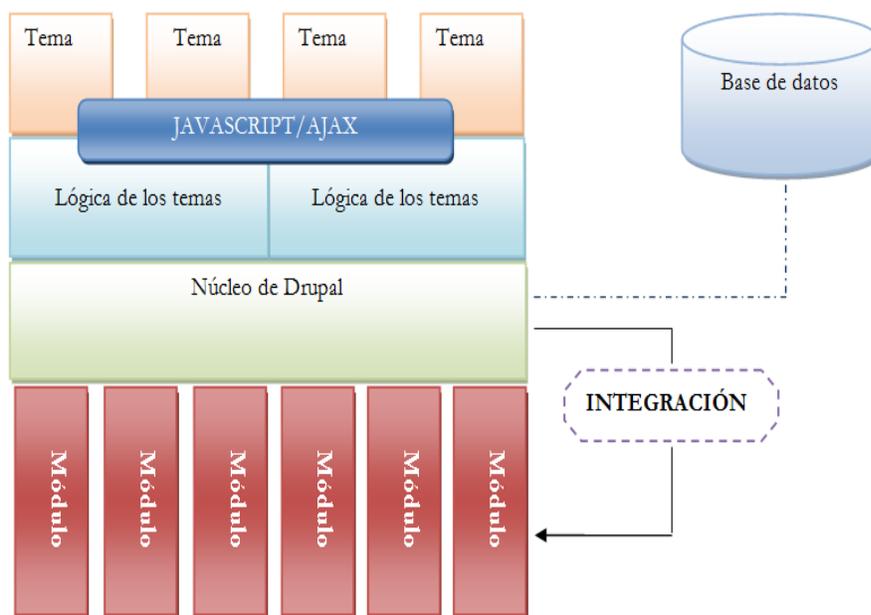


Figura 6: Elementos que conforman la arquitectura de Drupal.

El núcleo de Drupal presenta todos los módulos que se instalan y es el encargado de unificarlos. Este envía la información que procesa de los módulos al componente de la lógica del tema, el cual se encarga de visualizar dicha información en las vistas del sistema. Los temas utilizan los datos, permitiendo procesar o afectar los mismos antes de ser mostrados. A su vez estos pueden utilizar las funciones del API⁵ de JavaScript y AJAX para brindar una interfaz más intuitiva a los usuarios (35).

2.7.2 Patrones de diseño.

Los patrones del 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. El CMS Drupal, se encuentra concebido de manera tal que se puede usar varios patrones. Entre estos patrones se encuentran:

⁵ *Application Program Interface* o Interfaz de programación de aplicaciones

- **Bridge** (Puente): La capa de abstracción de bases de datos de Drupal se aplica de una forma similar al patrón de diseño bridge. En el desarrollo de la solución propuesta los módulos fueron escritos de forma independiente al sistema de base de datos utilizado. La información almacenada en la base de datos puede modificarse sin la necesidad de modificar el código de algún módulo.
- **Chain of Responsibility** (Cadena de responsabilidad): en la propuesta de solución, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y qué función se llama para hacer el trabajo, siguiendo así el flujo de trabajo de este patrón hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario o la cadena se ha agotado.
- **Command** (Comando): En el desarrollo de la solución propuesta fue empleado el patrón Command a fin de que los módulos implementados no tengan que definir cada hook, sino sólo los que vayan a aplicar.
- **Observer** (Observador): Es el encargado de definir una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Al igual que en el caso del patrón de diseño Decorator, este módulo es utilizado para llevar a cabo la extensión de los componentes internos de Drupal a través de los correspondientes hooks.
- **Decorator** (Decorador): El CMS Drupal hace amplio uso del patrón decorador. Mediante el empleo de diferentes hook lo que permite, que módulos arbitrarios extiendan el comportamiento de otros nodos. Esta característica posibilita añadir comportamientos a los nodos sin la necesidad de implementar una nueva subclase. En el desarrollo de la solución propuesta se emplearon hooks como el hook_node_menu (), hook_node_update (), hook_node_init (), entre otros.
- **Singleton** (Instancia única): Este patrón se evidencia si se asocian los módulos y los temas como objetos, cada uno de ellos en sí, no almacenan datos, sino que representan un conjunto de funciones que extienden el núcleo para añadirle funcionalidades al mismo, cada uno por separado. De esta forma se pueden considerar estos como una clase con una instancia Singleton.

2.7.3 Diagrama de paquetes.

Un diagrama de paquete muestra cómo está estructurado el sistema. Cada paquete puede contener otros paquetes o clases, que tienen interfaces y realizan cierta funcionalidad. Permiten dividir un modelo para agrupar y encapsular sus elementos en unidades lógicas individuales. Los paquetes pueden ser simples estructuras conceptuales o pueden estar reflejados en la implementación. Un paquete puede contener

paquetes subordinados, así como otros tipos de elementos del modelo. Todo tipo de elementos del modelo UML pueden ser organizados en paquetes (36).

La siguiente imagen muestra el diagrama de paquetes que genera el CMS utilizado:

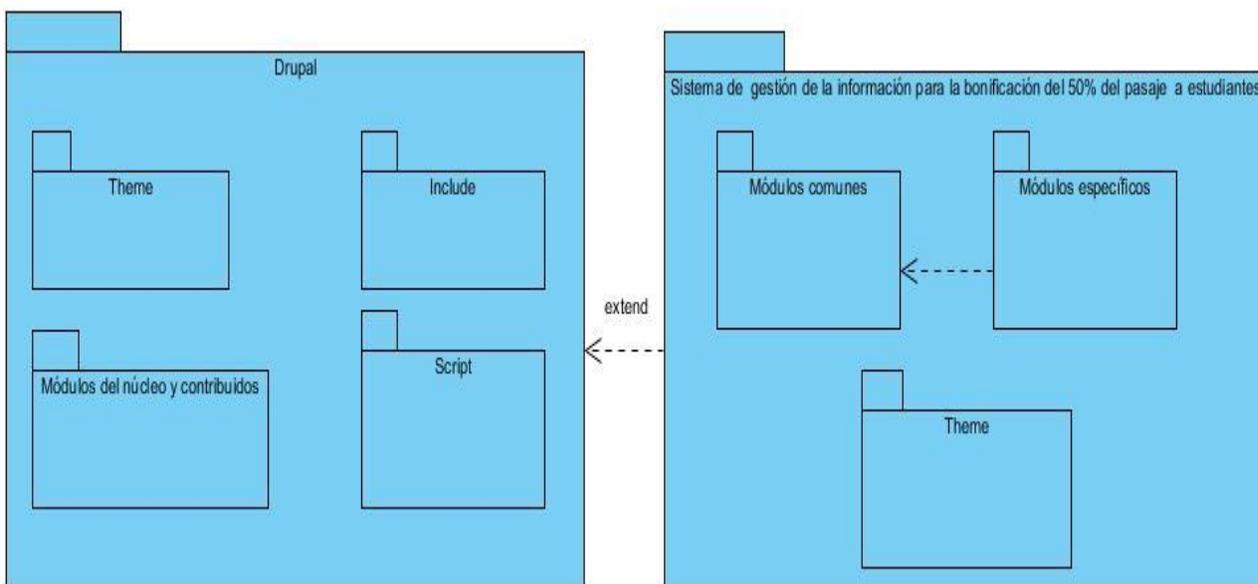


Figura 7: Diagrama de paquete del sistema de gestión de bonificación.

En el diagrama antes expuesto se muestra un paquete nombrado “Drupal”, que incluye los paquetes contenedores de los componentes del CMS y otro paquete de los más generales es el denominado “Gestión de bonificación”, que incluye los elementos que conforman la aplicación. A continuación se describe el contenido de cada uno:

Paquete CMS Drupal:

Includes: concentra un conjunto de librerías en forma de archivos PHP con extensión.inc, que incluyen funciones comunes del sistema indispensable para su funcionamiento, como son las conexiones a la base de datos.

Módulos: incluye todos los módulos, que permiten las distintas funcionalidades del CMS, cuando desee incorporar un nuevo módulo sólo se tiene que copiar en dicha carpeta.

Theme: incluye los temas que incluye de forma predefinida Drupal, permiten cambiar el estilo de la vista que se le muestra al usuario, cuando desee incorporar un nuevo tema sólo se tiene que copiar en dicha carpeta.

Perfiles: es un conjunto de instrucciones que permite definir qué módulos y opciones se quieren activar durante el proceso de instalación de Drupal. Paquete Plataforma:

Desarrollado: contiene el módulo que fue necesario implementar para el desarrollo de la aplicación. Estos dos paquetes generales (CMS Drupal y Gestión de bonificación) lógicamente se encuentran relacionados mediante un extend, pues el sistema se implementa haciendo uso de dicho CMS y por tanto utiliza las características y funcionalidades que este brinda.

2.7.4 Diseño de la base de datos.

El diseño de la base de datos representa el tratamiento de la información con carácter persistente dentro del sistema. Para este trabajo fue construido el Modelo Físico de Datos, cual proporciona una flexibilidad óptima para el soporte de la informatización.

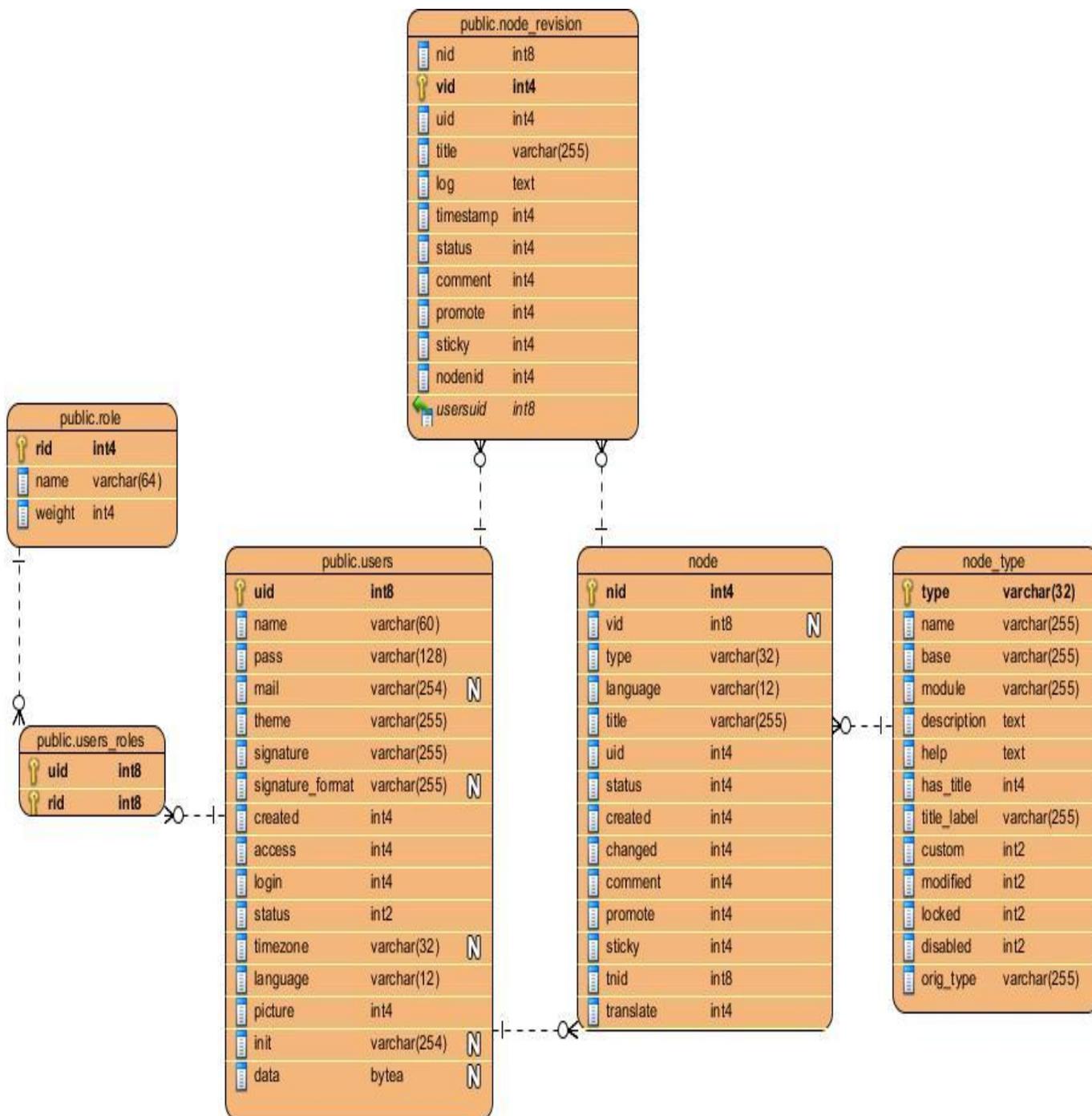


Figura 8: Modelo físico de la base de datos del sistema de bonificación.

2.8 Conclusiones parciales.

El modelado de los procesos del negocio y la explicación detallada de estos, permitieron identificar con mayor facilidad los requisitos funcionales que el sistema debe satisfacer, propiciando así la comprensión de su funcionamiento. Las historias de usuario especificadas para cada uno de estos requisitos sin emplear lenguaje técnico y sí términos del cliente, detallando entre otros elementos el tiempo que requieren para su codificación, favorecieron la estimación del plan de entregas del producto delimitando el propósito y duración de cada iteración. El desarrollo de la propuesta de solución basado en la arquitectura modular y los patrones de diseño con que cuenta Drupal, brindaron la flexibilidad y adaptabilidad presentes en el sistema ante posibles cambios en la implementación de las funcionalidades.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS.

3.1. Introducción

El capítulo presenta las características de la implementación del sistema, con el objetivo de adquirir un producto final que responda a los requisitos definidos por el cliente, lo cual se valida empleando diferentes tipos de pruebas que propone la metodología de desarrollo de software utilizada. Para ello se especifican los estándares de codificación que debe cumplir el equipo de desarrollo, se crea el diagrama de despliegue respresentando los recursos físicos necesarios para el despliegue de la aplicación, y se presentan las interfaces de pruebas creadas para comprobar el funcionamiento de la solución. A partir del código resultante y su funcionamiento se ejecutan las pruebas unitarias y de aceptación del módulo.

3.2. Código fuente.

Una vez definidas las historias de usuario y acabado el diseño se transita a la etapa de codificación de la solución propuesta, cuyo objetivo es desarrollar de forma iterativa e incremental un producto que esté apto para transferirse a su comunidad de usuarios, obteniéndose versiones útiles de forma rápida y práctica, que paulatinamente completen la planeación, diseño, desarrollo y prueba de toda la funcionalidad necesaria.

3.2.1. Estándares de codificación.

XP propone varias prácticas a aplicar durante el proceso de desarrollo de software, entre estas se encuentran la refactorización, programación en parejas y la propiedad colectiva del código, las cuales insisten en la simplicidad en la codificación, trabajo en dúo de programadores y que cualquier desarrollador pueda corregir e implementar cualquier parte del producto. Para consolidar estas prácticas, la metodología insiste que la comunicación de los programadores es a través del código, por lo cual es necesario que se sigan ciertos estándares de programación (17).

En un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada y así obtener mejor comprensión del código fuente. Los estándares de codificación sugieren normas o pautas a seguir para generar el código, permitiendo que la programación sea fácil de entender y modificar, independientemente de quien haya sido el autor (37). Para la implementación del sistema de gestión de información para la bonificación del 50% del pasaje a estudiantes se estandarizó el código, garantizando que exista legibilidad y organización para su mantenimiento. A continuación se especifican los estándares de código a utilizar en la construcción del sistema, propuesto por Drupal.

- **Etiquetas de apertura y cierre de PHP**

Siempre se debe utilizar las etiquetas de apertura y cierre `<?php y?>`.

- **Operadores**

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio.

- **Uso de comillas**

Se pueden usar tanto las comillas simples, como las comillas dobles para delimitar las cadenas de caracteres.

- **Uso de punto y coma (;) en código PHP**

Es obligatorio el uso del (;) al escribir líneas de código individuales en PHP.

- **Estructuras de control**

Debe haber un espacio entre el comando que define la estructura (**if**, **while**, **for**, **switch**) y el paréntesis de apertura. La llave de apertura { se situará en la misma línea que la definición de la estructura, separada por un espacio. Se recomienda usar siempre las llaves {} aún en los casos en que no sea obligatorio su uso (una sola "línea" de código dentro de la estructura de control). Las estructuras **else** y **elseif** se escribirán en la línea siguiente al cierre de la sentencia anterior.

- **Funciones**

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo. Además, se debe incluir siempre como prefijo el nombre del módulo, tema, para evitar así duplicidad de funciones. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior. En la llamada a la función se aplican las mismas reglas anteriores con respecto a los parámetros. Como excepción, es posible usar más de un espacio antes de una asignación (=) para mejorar la presentación, cuando se estén realizando varias asignaciones en bloque.

- **Arrays**

Los valores dentro de un **array** (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados. Cuando la línea de declaración del array supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

- **Constantes**

Los nombres de las constantes deben escribirse en mayúsculas, con guiones bajos para separar palabras. Al igual que ocurre con las funciones, los nombres de las constantes deben tener como prefijo el nombre del módulo (o tema) en el que se utilizan, para evitar errores de duplicidad de constantes. Este prefijo también se escribirá en mayúsculas.

- **Variables globales**

Se declaran utilizando un guión bajo inicial, seguido del nombre del módulo o tema y otro guión bajo antes del nombre de la variable.

- **Nombres de módulos**

El nombre de un módulo nunca debería incluir guiones bajos, aunque se componga de varias palabras. De esta forma será más fácil identificar el módulo al que pertenece una función, ya que el prefijo o nombre del módulo será todo aquello que esté antes del primer guión bajo.

- **Nombres de archivos**

Los nombres de archivos deben escribirse siempre en minúsculas. La única excepción son los archivos de documentación, que tendrás extensión `.txt` y el nombre es mayúsculas.

- **URLs de ejemplo**

Por convención, siempre que haya que indicar una URL de ejemplo.

- **Idioma**

Drupal utiliza el idioma inglés como base para todo el sistema. Es por ello que todos los módulos deben escribirse en inglés, tanto su código (nombres de funciones, variables.) como las cadenas de texto que se muestran al usuario. Para que estas cadenas sean traducibles se debe utilizar la función de traducción `t()`.

- **Comentar el código**

Se pueden utilizar las etiquetas `/* */` para comentarios en varias líneas y `//` para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, ésta debe escribirse en mayúsculas (por ejemplo `TRUE` o `FALSE`).

- **Documentando archivos**

Se incluye al principio de cada archivo una descripción general del mismo, utilizando la directiva `@file`.

- **Documentando funciones**

Todas las funciones que pueden ser llamadas desde otros archivos deben estar comentadas (es recomendable comentar todas las funciones, independientemente del ámbito en que vayan a ser utilizadas). El bloque de comentario debe ir justo encima de la declaración de la función, sin dejar líneas en blanco.

- **Documentando hooks**

Cuando una función es la implementación de un hook, puede no ser necesario dar más detalles sobre la misma, ya que los comentarios se habrán definido en la función hook original.

A continuación se presentan ejemplo de código del Sistema de gestión de la información para la bonificación del 50% de pasaje a estudiantes.

```

estadisticas.module ✕
32 // funcion a ejecutar siempre que se inserte un nodo
33 function estadisticas_node_presave($node) {
34
35     if ($node->type == 'bonificacion' && arg(1) == 'add') {
36         $folio = variable_get('Folio');
37         $folio+=1;
38         $node->field_folio['und'][0]['value'] = $folio;
39         variable_set('Folio', $folio);
40     }
41 }
42
43 //funcion a ejecutar siempre que se actualice un nodo
44 function estadisticas_node_update($node) {
45
46     if ($node->type == 'bonificacion') {
47         if (@$node->field_fecha_cobro['und'][0]['value'] != '') {
48             $fecha = date('d-m-Y', $node->created);
49             $res = estadisticas_devuelve_presupuesto($fecha);
50             $pasaje_regreso = $node->field_valor_pasaje_regreso['und'][0]['value'];
51             $pasaje_id = $node->field_valor_pasaje_ida['und'][0]['value'];
52             $pasaje = ($pasaje_id + $pasaje_regreso) / 2;
53             $node = node_load($res->nid);
54             $node->field_presupuesto['und'][0]['value'] += $pasaje;
55             node_save($node);
56         }
57     }
58 }

```

Figura 9: Código del módulo estadística del sistema.

3.3. Implementación.

La etapa de implementación inicia con los resultados obtenidos de la fase de diseño, esta describe como los elementos del modelo de diseño se implementan en términos de componentes y se organizan de acuerdo a los nodos específicos del modelo de despliegue. El modelo de implementación contiene los diagramas de componentes y despliegue. Estos describen los componentes a construir y su organización y la dependencia entre nodos físicos en los que funcionará el sistema, los cuales comienzan a desarrollarse en el flujo de trabajo de diseño y que se perfeccionan en la implementación (38).

3.3.1. Diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Los componentes son las piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas (16). Estos representan los elementos del software resultantes del proceso de implementación. A continuación se ilustra el diagrama definido para la solución propuesta.

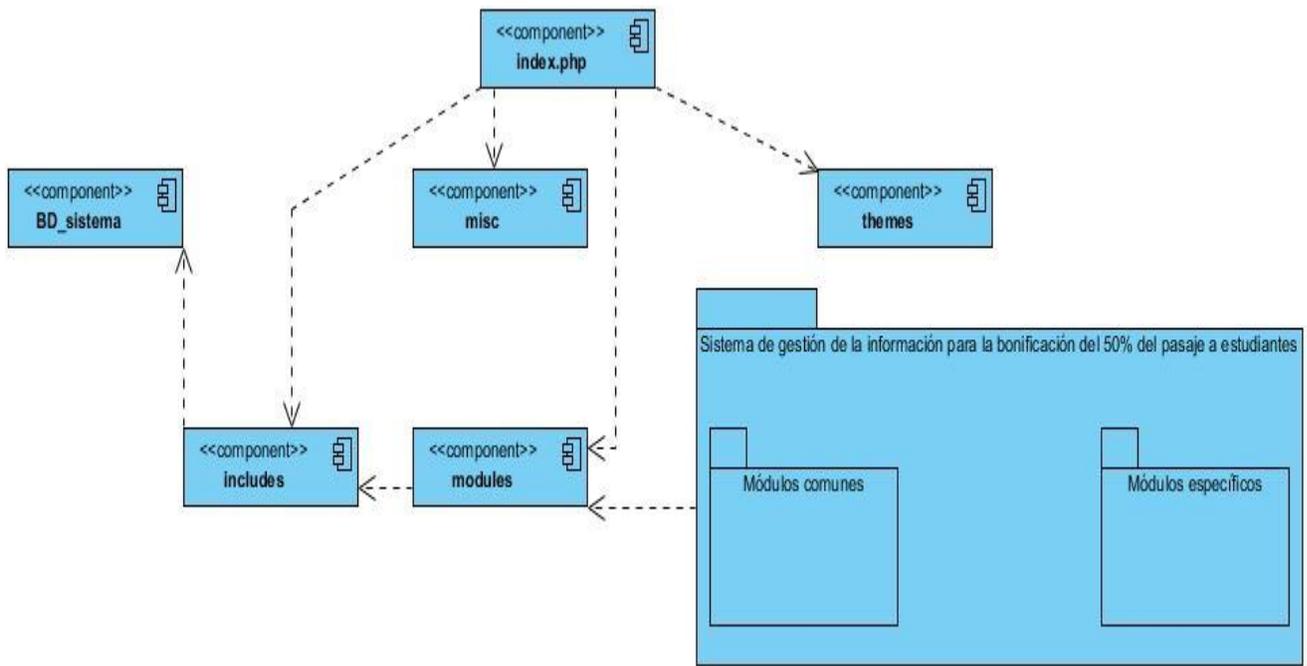


Figura 10: Diagrama de componentes del sistema.

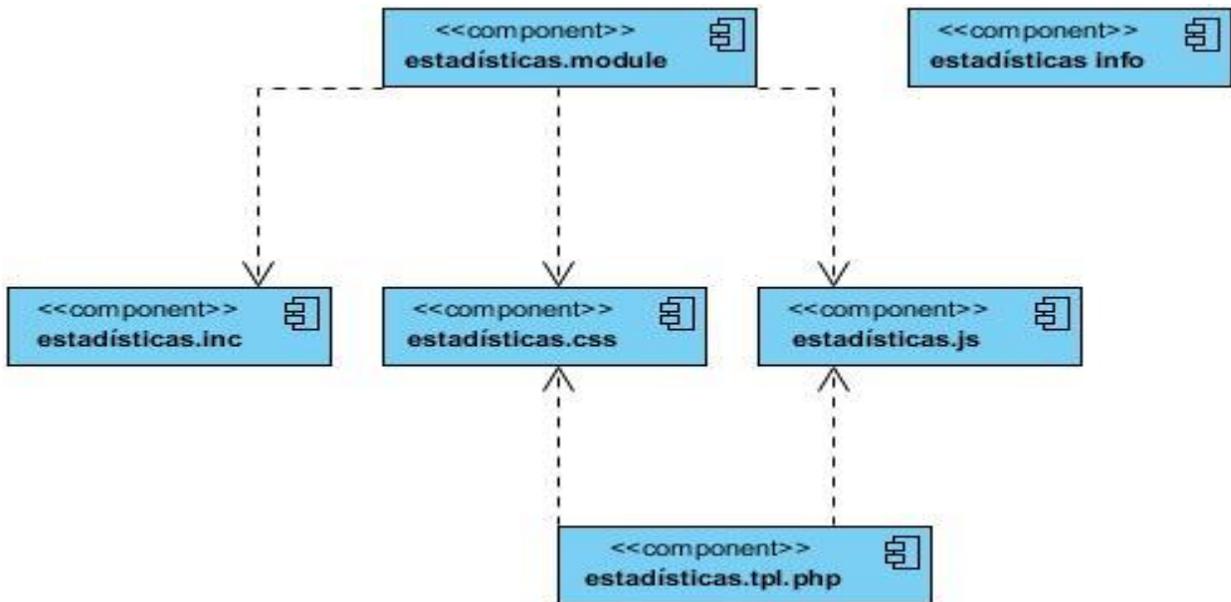


Figura 11: Diagrama de componente del módulo estadísticas.

Para un mejor entendimiento del diagrama de componentes del sistema se describe en la tabla siguiente los aspectos del mismo:

Tabla 7: Descripción de diagrama de componentes del sistema.

index.php	Es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS.
BD_sistema	Es la base de datos de Drupal, dentro de esta se alojan las tablas generadas por el Sistema de gestión de la información para la bonificación del 50% del pasaje a

	estudiantes.
<i>includes</i>	Componente que contiene un conjunto de funciones y scripts necesarios para el funcionamiento de los módulos y el sistema Drupal, incluyendo los APIs de conexión a base de datos.
<i>modules</i>	Componente que contiene todos los módulos, que permiten las distintas funciones del CMS.
Sistema de gestión de la información para la bonificación del 50% de pasaje a estudiantes.	Paquete que contiene los módulos que permiten la gestión de la información de la binificación.
<i>themes</i>	Componente que contiene los temas que se pueden utilizar en la aplicación, los cuales permiten separar el contenido de la presentación.
<i>Misc</i>	Componente que contiene contenidos que tienen que ver con el diseño y funcionamiento.

3.3.2. Diagrama de despliegue.

El diagrama de despliegue es uno de los artefactos que se genera durante el proceso de desarrollo del software, modela la arquitectura en tiempo de ejecución de un sistema, por lo que sitúa el software en el hardware que lo contiene. Está compuesto por nodos, dispositivos y conectores que representan un medio computacional. Estos sirven para modelar la topología del hardware sobre el que se ejecuta el sistema. La figura que se muestra a continuación representa el diagrama de despliegue del Sistema de gestión de la información para la bonificación del 50% del pasaje a estudiantes y está compuesto por:

- PC_Clientes: representa una computadora desde la cual el usuario podrá acceder a la aplicación.
- Impresora: para que el usuario imprima los modelos y reportes, está conectada con la PC_Cliente por puerto USB
- Servidor Web (apache): establece comunicación con la PC-Cliente utilizando el protocolo HTTPS y realiza peticiones mediante TCP/IP al Servidor de Base de Datos.
- Servidor de base de datos: representa el servidor donde estará el Sistema Gestor de Base de Datos que dará respuesta a las peticiones hechas por la aplicación.

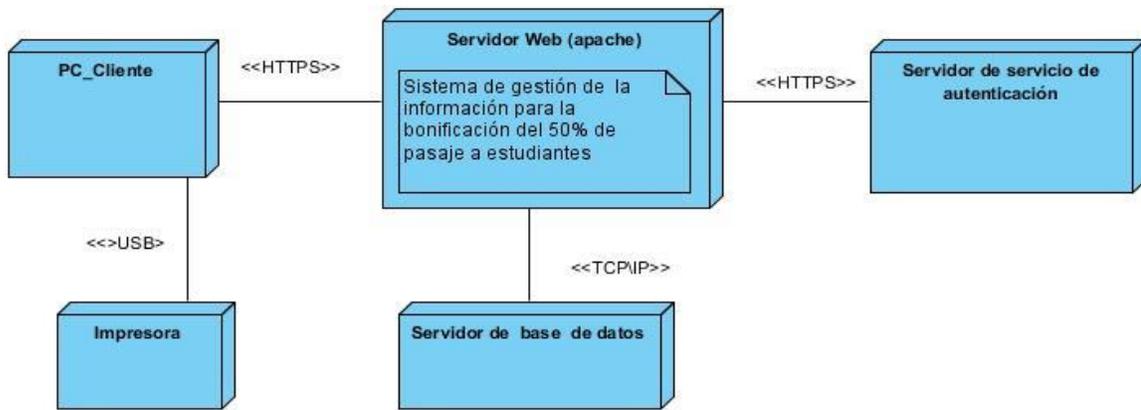


Figura 12: Diagrama de despliegue del sistema.

3.4. Pantallas principales de la aplicación.

A continuación se exponen las principales interfaces del Sistema de gestión de la información para la bonificación del 50% de pasaje.

- La Figura 13 muestra la interfaz de inicio del sistema, esta da paso a las principales funcionalidades del sistema como: buscar estudiantes, gestionar bonificaciones, gestionar presupuesto y generar reportes.



Figura 13: Interfaz de inicio del sistema.

- La Figura 14 muestra la interfaz de acceso a las funcionalidades del sistema.



Figura 14: Interfaz para acceder a las funcionalidades.

- La Figura 15 muestra la interfaz de la funcionalidad buscar estudiantes del sistema.

Listado de estudiantes

Solapín

Usuario



- Nombres y Apellidos: Sucel Daudinot Caricabeur
Usuario: sdaudinot
Solapín: E103903
Facultad: [Facultad 1](#)
CI: 88022935219 [Editar CI](#)
Acciones: [Eliminar](#) | [Adicionar bonificación](#)

Figura 15: Buscar estudiantes.

- La Figura 16 muestra la interfaz de la funcionalidad gestionar bonificaciones del sistema.



Figura 16: Interfaz de la funcionalidad gestionar presupuesto.

3.5. Pruebas.

Las pruebas de software son un proceso de gran importancia para la elaboración de un programa, tienen como propósito velar por la garantía de la calidad de los sistemas. Su objetivo es descubrir los errores de los programas en ejecución y solo así tienen éxito (39). Estas verifican que el software implemente una función específica de forma correcta y validan que las acciones que realiza respondan a los requisitos del cliente (40).

Para la validación y verificación del Sistema de gestión de la información para la bonificación del 50% del pasaje a estudiantes se realizaron pruebas unitarias, de aceptación, seguridad y carga y estrés.

3.5.1. Pruebas unitarias.

La metodología XP se basa en pruebas unitarias para comprobar el funcionamiento de los códigos que se vayan implementando. Las pruebas unitarias se utilizan para detectar errores en el código, pues se centran en la menor unidad de diseño del software (módulos, método de una clase, clases). Tiene como objetivo principal detectar errores en los principales procesos al ser ejecutado de manera independiente. Estas son diseñadas por los programadores y realizadas ante cualquier modificación del sistema para detectar fallas que pudieran ocurrir (41).

Las pruebas de unitarias proporcionan algunas ventajas como que impulsa al desarrollador a crear una estructura de código más simple (lo que se conoce como refactorización). Simplifica la integración, pues permite llegar a la fase de integración con un código que funcione correctamente y facilita un mejor

entendimiento del código ya que lo documenta a través de las pruebas (42). Estas siempre están orientadas a las técnicas de prueba de caja blanca, denominadas también pruebas estructurales.

- **Pruebas de caja blanca.**

Las pruebas de caja blanca también llamadas pruebas de cristal es un método de diseño que se basa en un examen de los datos procedimentales del código a evaluar. Esta prueba garantiza que al menos se recorran todos los caminos independientes de cada módulo (método, clases), se ejerciten todas las decisiones lógicas, se ejecuten todos los bucles y las estructuras de datos internos para asegurar su validez (43).

Para el sistema desarrollado, se aplicó la técnica de pruebas de caja blanca de la ruta básica o camino básico como también es conocida. Esta permite obtener la complejidad lógica de un diseño y con esta medida definir un conjunto básico de recorridos de ejecución, es decir obtener casos de prueba a partir del conjunto de caminos independientes. Para obtener un conjunto de caminos independientes se construirá un grafo de flujo asociado y se calculará su complejidad ciclomática.

Un Grafo de Flujo está formado por 3 componentes fundamentales que ayudan a su elaboración, comprensión y brinda información para corroborar que el trabajo se está haciendo adecuadamente.

Los componentes son:

- **Nodo.**

Cada círculo representado se denomina nodo del Grafo de Flujo, el cual representa una o más secuencias procedimentales. Un solo nodo puede corresponder a una secuencia de procesos o a una sentencia de decisión. Puede ser también que hallan nodos que no se asocian, se utilizan principalmente al inicio y final del grafo.

- **Aristas.**

Las flechas del grafo se denominan aristas y representan el flujo de control, son análogas a las representadas en un diagrama de flujo. Una arista debe terminar en un nodo, incluso aunque el nodo no represente ninguna sentencia procedimental.

- **Regiones.**

Las regiones son las áreas delimitadas por las aristas y nodos. También se incluye el área exterior del grafo, contando como una región más. Las regiones se enumeran y la cantidad de regiones es equivalente a la cantidad de caminos independientes del conjunto básico de un programa.

El método diseñado para validar la fecha, cuyo código se muestra a continuación, la aplicación de la prueba del camino básico quedaría de la siguiente manera:

Tabla 8: Método estadísticas_valida_fecha.

Función estadísticas_valida_fecha	
function estadísticas_valida_fecha(\$form, &\$form_state){	
\$fecha_ini=strtotime(\$form['field_fecha_de_inicio']['und'][0]['#value']['value']['date']);	(1)
\$fecha_fin=strtotime(\$form['field_fecha_fin']['und'][0]['#value']['value']['date']);	(1)
\$presupuesto=@\$form['field_presupuesto_asignado']['und'][0]['value']['#entity']->field_presupuesto_asignado['und'][0]['value'];	(1)
if(\$presupuesto<0){	(1)
form_set_error('field_presupuesto', 'El presupuesto no puede ser negativo.');	(2)
}	(2)
if (\$fecha_fin < \$fecha_ini) {	(3)
form_set_error('error', 'La fecha de fin no puede ser menor que la fecha de inicio.');	(4)
}	(5)
}	(5)

Tabla 9: Camino básico.

Construcción del grafo de complejidad ciclomática correspondiente al código	
<pre> graph TD 1((1)) --> 2((2)) 1((1)) --> 3((3)) 2((2)) --> 3((3)) 3((3)) --> 4((4)) 3((3)) --> 5((5)) 4((4)) --> 5((5)) </pre>	Complejidad Ciclomática: A = 6 (Aristas) N = 5 (Nodos) $V(G) = A - N + 2$ $V(G) = 6 - 5 + 2 = 3$ P = 3 (Nodos predicados) $V(G) = P + 1$ $V(G) = 2 + 1$ $V(G) = 3$ R (Regiones) $R = V(G)$ $R = 3$ Caminos 1) 1-3-5 2) 1-2-3-5 3) 1-3-4-5

Tabla 10: Casos de prueba unitarias.

Caso de prueba
Camino: 1-3-5
Entrada: Recibe números positivos, se introduce la fecha actual y fecha fin.

Resultado: Muestra un mensaje notificando la acción.
Condiciones:
Caso de prueba
Camino: 1-2-3-5
Entrada: Recibe número negativo, se introduce fecha actual y fecha fin.
Resultado: Muestra un mensaje notificando el error.
Condiciones:
Caso de prueba
Camino: 1-3-4-5
Entrada: Recibe número positivo, se introduce fecha actual mayor que la fecha fin.
Resultado: Muestra un mensaje notificando que existe error.
Condiciones:

3.5.2. Pruebas de aceptación.

Las pruebas de aceptación o pruebas funcionales como también son conocidas, son las pruebas finales antes de la liberación del sistema. Se realizan con el objetivo de confirmar si el software está listo y puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales fue implementado. En XP las pruebas de aceptación son creadas a través de las HU. El cliente o usuario detalla los diversos aspectos a comprobar para verificar que una HU ha sido correctamente implementada. Las HU pueden tener varias pruebas de aceptación, para garantizar su buen funcionamiento. Las pruebas de aceptación son consideradas como “pruebas de caja negra”, las cuales están orientadas a los requisitos funcionales del software, buscan que se hayan ingresado toda clase de entrada y que la salida obtenida es igual a la esperada (44).

A continuación se muestra algunas de las pruebas realizada a la HU; que constituye una de las funcionalidades primordiales para el cliente. Las pruebas restantes se encuentran en los anexos. ([Anexo 6](#))

Tabla 11: Caso de prueba: Adicionar bonificación.

SC Adicionar bonificación.			
EC 1.1 Adicionar bonificación de forma correcta.	El sistema adiciona la bonificación forma correcta.	El sistema almacena la información y muestra un mensaje de confirmación de la acción: "Bonificación para Sucel Daudinot Caricabeur se ha creado".	El usuario se autentica en el sistema con privilegios de administración. 1.-El usuario accede al módulo: “Buscar estudiantes” , que se encuentra ubicado en la parte lateral izquierda del sistema.
EC 1.2 Adicionar bonificación de forma incorrecta.	El sistema adiciona la bonificación forma incorrecta.	El sistema no adiciona la bonificación y emite el siguiente mensaje de error: "La fecha de regreso no puede	2.-El sistema le muestra una interfaz con un listado de estudiantes con sus correspondientes datos. 3.-El usuario selecciona la opción

		ser menor que la ida".	<p>"Adicionar bonificación" que se encuentra ubicada después de los datos de cada estudiante existente en el sistema.</p> <p>4.-El sistema muestra un formulario para que el usuario introduzca la información.</p> <p>5.- El usuario introduce la información y presiona el botón: "Guardar"</p>
		El sistema no adiciona la bonificación y emite el siguiente mensaje de error: "El valor de la fecha no coincide con el formato esperado".	
		El sistema no adiciona la bonificación y emite el siguiente mensaje de error: "El valor de la fecha no coincide con el formato esperado".	
		El sistema no adiciona la bonificación y emite el siguiente mensaje de error: "El estudiante ha solicitado una bonificación en este mes".	
		El sistema no adiciona la bonificación y emite el siguiente mensaje de error: "Debe seleccionar al menos destino o regreso".	
		El sistema no adiciona la bonificación y emite el siguiente mensaje de error: " El valor del pasaje de ida no puede estar vacío si se ha seleccionado un Destino y viceversa".	
		El sistema no adiciona la bonificación y emite el siguiente mensaje de error: " El valor del pasaje de ida no puede estar vacío si se ha seleccionado un Destino y	

		viceversa".	
		El sistema no adiciona la bonificación y emite los siguientes mensajes de error: "El valor del pasaje de regreso no puede estar vacío si se ha seleccionado un Regreso y viceversa".	
		El sistema no adiciona la bonificación y emite el siguiente mensaje de error: "El valor del pasaje de regreso no puede estar vacío si se ha seleccionado un Regreso y viceversa".	

Tabla 12: Caso de prueba: Imprimir bonificación.

SC Imprimir bonificación.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Imprimir bonificación de forma correcta.	El sistema imprime la bonificación seleccionada de forma correcta.	El sistema muestra un formulario con las opciones de impresión seguida de las opciones: "Cancelar" e "Imprimir" . El usuario selecciona la opción: "Imprimir" , el sistema imprime el modelo de la bonificación seleccionada.	El usuario autenticado en el sistema con privilegios de administración accede al módulo: "Gestionar bonificaciones" , que se encuentra ubicado en la parte lateral izquierda del sistema. 1.- El sistema le muestra al usuario una interfaz con un listado de las bonificaciones existentes en el sistema, donde se encuentran las "Cobradas" y las "Pendientes" de cobro. 2.- El
EC 1.2 Cancelar operación.	El sistema cancela la operación.	El sistema muestra un formulario con las opciones de impresión seguida de las opciones: "Cancelar" e "Imprimir" . El usuario selecciona la opción "Cancelar" , el sistema deshace la operación de impresión.	usuario realiza clic en la opción de "Cobradas" , luego busca la bonificación que desea imprimir dándole un clic a la opción: "Imprimir" .

Tabla 13: Caso de prueba: Registrar fecha de cobro.

SC Registrar fecha de cobro.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar fecha de cobro de forma correcta.	El sistema registra fecha de cobro de manera correcta.	El sistema registra la fecha de cobro introducida por el usuario	El usuario autenticado en el sistema con permiso de acceso al módulo: “Gestionar bonificaciones” , que se encuentra ubicado en la parte lateral izquierda del sistema accede a este.
EC 1.2 Registrar fecha de cobro de forma incorrecta.	El sistema no registra fecha de cobro de manera incorrecta.	El sistema no registra la fecha de cobro introducida por el usuario y emite el siguiente mensaje de error: "La entrada de valor para el campo no es válido: El valor de la fecha no coincide con el formato esperado".	1.- El sistema le muestra al usuario una interfaz con un listado de las bonificaciones existentes en el sistema “Pendientes” por cobrar.
		El sistema no registra la fecha de cobro introducida por el usuario y emite el siguiente mensaje de error: "La fecha de cobro no puede ser menor que la ida o el regreso."	2.- El usuario realiza un clic en la opción: “Registrar cobro” . El sistema muestra un formulario con el siguientes campo: • Fecha de Cobro (input de tipo date). El usuario registra la información y presiona el botón: “Guardar” .

Durante el desarrollo del sistema se realizaron pruebas de aceptación a las HU desarrolladas por cada iteración, verificando que las funcionalidades que estas contienen, se desempeñen correctamente.

Resultados de las pruebas:

- En la primera iteración se detectaron 20 no conformidades, de ellas 3 significativas, 17 no significativas y se realizaron 4 recomendaciones
- En la segunda iteración se detectaron 5 no conformidades, de ellas 2 significativas, 3 no significativas y se realizaron 3 recomendaciones.
- En la tercera iteración fueron detectadas 2 no conformidades no significativas y se realizaron 5 recomendaciones.

El gráfico que se muestra a continuación establece una relación entre las iteraciones de pruebas realizadas y el número de no conformidades detectadas en las mismas.

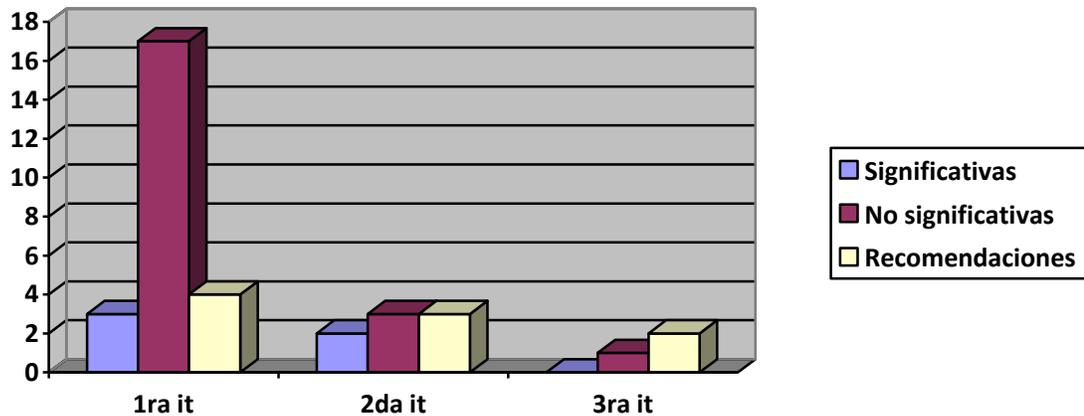


Figura 17: Resultado de las pruebas funcionales.

En la figura mostrada anteriormente se puede observar el comportamiento del proceso de pruebas de aceptación definido por el cliente, donde se detectaron un total de 27 no conformidades durante las tres iteraciones de codificaciones planificadas. Las mismas fueron de dos tipos fundamentales de documentación y de codificación, las cuales no incidían significativamente en las respuestas esperadas. Estas dificultades fueron corregidas en un corto plazo, ayudando a que la aplicación alcanzara una mejor calidad y cumpliera con los requisitos planteados. Una vez obtenido el producto final se realizó una iteración con el propósito de verificar el correcto funcionamiento de todo el sistema, en la que se detectaron nuevas no conformidades, las cuales no eran significativas para los requisitos planteados por el cliente.

3.5.3. Pruebas de Seguridad.

Las pruebas de seguridad son mecanismos de protección dentro del mismo sistema, que permiten verificar los niveles de acceso al mismo. Tienen como propósito evaluar la confidencialidad, integridad y disponibilidad de los datos. En el Sistema de gestión de la información para la bonificación del 50% del pasaje a estudiantes para garantizar la protección de los datos, se implementa la seguridad a través de la autenticación de los usuarios utilizando sesiones y estableciendo un sistema de control de acceso basado en roles definidos por Drupal. La asignación de permisos se realiza a través de los roles del sistema, y de esta forma, el administrador no tiene que establecer los permisos para cada usuario, sino, que se asignan los permisos a un determinado rol y se agrupan los usuarios por roles. Esta asignación por roles permite controlar a qué funcionalidades el usuario tiene acceso, de manera que la información accedida esté restringida y protegida en dependencia de los distintos permisos y roles de usuarios, obteniendo como resultado un sistema más seguro. El sistema define los roles de usuario autenticado y el administrador. Además Drupal posee un módulo *Database Logging* el cual se encarga de guardar registro de la actividad de los usuarios en el sistema, lo que permite auditar los fallos que ocurran en el sistema.

3.5.4. Pruebas de carga y stress.

Este tipo de prueba que se realiza para comprobar la solidez de la aplicación en los momentos de carga extrema, ayuda a determinar si la aplicación rinde lo suficiente en caso de que la carga real supere a la carga esperada. Busca validar el correcto funcionamiento del sistema bajo las condiciones de carga normales para la operación, para ultimar sobre variables como: el tiempo de respuesta, carga de procesamiento, trabajo por unidad de tiempo y utilización de recursos.

- **Resultados pruebas de carga y stress.**

Las pruebas de carga y stress fueron realizadas en la herramienta JMeter propuesta por la UCI. Esta aplicación realiza pruebas del rendimiento de la interfaz a través de la construcción de peticiones HTTP/HTTPS con cantidad de usuarios concurrentes variables, lo que posibilita obtener un resumen de los niveles de estrés del sistema y los límites de trabajo del mismo en condiciones extremas. El entorno en que estas fueron ejecutadas posee las siguientes características:

- microprocesador Pentium 4 a 300 GHz
- memoria RAM de 2Gb
- sistema operativo Ubuntu 12.04

Se muestra a continuación los resultados arrojados realizados a las principales funcionalidades del sistema.

Informe Agregado										
Nombre: Informe Agregado										
Comentarios										
Escribir todos los datos a Archivo										
Nombre de archivo		Navegar...		Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos			Configurar			
Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec	
/app/sites/a...	50	98	85	198	4	320	0,00%	56,8/min	,8	
/app/sites/a...	50	55	51	112	3	175	0,00%	56,9/min	,5	
/app/sites/a...	50	44	45	78	5	123	0,00%	56,8/min	1,6	
/app/sites/a...	50	64	47	133	2	412	0,00%	56,6/min	,4	
/app/sites/a...	50	49	40	92	3	204	0,00%	56,7/min	,7	
/app/sites/a...	50	52	42	99	4	216	0,00%	56,7/min	,6	
/app/reporte	50	13943	14524	16041	7194	16850	100,00%	47,2/min	2,1	
/app/sites/a...	50	88	66	202	3	272	0,00%	1,1/sec	,5	
/app/misc/u...	50	283	248	469	10	767	0,00%	1,0/sec	11,2	
/app/sites/a...	50	99	79	232	5	456	0,00%	1,0/sec	6,2	
/app/modul...	50	55	48	101	3	214	0,00%	1,0/sec	1,0	
/app/sites/a...	50	53	54	88	3	151	0,00%	1,0/sec	,9	
/app/misc/u...	50	58	49	119	3	165	0,00%	1,0/sec	1,4	
/app/sites/a...	50	55	45	115	3	185	0,00%	1,0/sec	3,3	
/app/sites/a...	50	48	43	78	3	227	0,00%	1,0/sec	3,7	
/app/facult...	50	17192	18158	20020	9171	20681	0,00%	44,9/min	2,4	
/app/modul...	50	60	54	118	2	160	0,00%	1,0/sec	,5	
/app/user	50	14643	15010	17006	11056	18158	0,00%	50,0/min	2,3	
/app/user/l...	50	14872	15187	16436	10138	17337	100,00%	49,0/min	1,8	
Total	27350	641	57	208	2	78282	1,28%	75,5/sec	206,9	

¿Incluir el nombre del grupo en la etiqueta? Guardar la cabecera de la tabla

Figura 18: Prueba de carga y stress.

Del análisis de los resultados de las pruebas se puede concluir que el sistema con más de 27350 usuarios realizando más de 641 peticiones que equivalen a 57 conexiones concurrentes, en un tiempo promedio de 2 milisegundos observa un 1.28 % de peticiones fallidas incrementándose este valor a medida que se

eleva el número de usuarios. Partiendo de estos resultados y teniendo en cuenta las características del servidor utilizado en las mismas se demuestra que las interfaces cumplen satisfactoriamente con los requerimientos establecidos

3.6. Conclusiones parciales.

El uso de los estándares de codificación definidos por Drupal permitió desarrollar un código reutilizable y simple. La representación del diagrama de componentes, el diagrama de despliegue y las interfaces de pruebas creadas, visualizaron la estructura y funcionamiento del sistema. Además con la ejecución de varias iteraciones cuyas entregas eran inmediatamente probadas se detectaron no conformidades, las cuales fueron corregidas, surgiendo otras no significativas para los requisitos definidos por el cliente, validándose el correcto funcionamiento del sistema.

Conclusiones

Para dar cumplimiento al objetivo de la presente investigación se arrojaron a las siguientes conclusiones:

- El estudio de los sistemas de gestión económica permitió comprender y aplicar las teorías existentes al sistema propuesto en la investigación.
- El análisis del negocio y las entrevistas realizadas al cliente permitió identificar las principales deficiencias del proceso del pago del 50% de la bonificación del pasaje a estudiantes, así como definir los requisitos funcionales y no funcionales de la solución propuesta y guiar el proceso de desarrollo de la implementación.
- La implementación de las funcionalidades permitió contribuir a la gestión de la información que maneja el proceso de la bonificación del 50% del pasaje a estudiantes en la Vicerrectoría Económica de la UCI.
- La aplicación de pruebas de software contribuyó a identificar y solucionar las vulnerabilidades detectadas y así lograr el funcionamiento del sistema y la satisfacción del usuario final.

Recomendaciones

- Aplicar la propuesta realizada en todas las facultades de la Universidad y capacitar al personal con cursos o seminarios.
- Mantener actualizada toda la documentación archivada en el sistema.
- Implementar en la solución otros servicios de pagos como dietas y facturas que se desarrollen en la Universidad.
- Por la utilidad práctica del sistema, se considera que puede ser usado por otras Universidades cubanas.

Bibliografía referenciada

1. LOURDES AJA QUIROGA. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. En: *ACIMED*. 2002, Vol. 10, no. 5, pp. 7–8.
2. REAL ACADEMIA DE CIENCIAS ECONÓMICAS Y FINANCIERAS. *Diccionario de términos económicos y financieros de la Real Academia de Ciencias Económicas y Financieras (RACEF)* [en línea]. S.l.: s.n., 2009. Disponible desde: <http://www.diccionarioracef.com/index.php?a=search&texto=BONIFICACI%F3N&exacta=Y>.
3. ROSA NATIVIDAD SEGUENCIA AGUAYO. *Desarrollo de sistema de gestión de calidad aplicando la norma ISO 9001:2008 para la empresa Mesón del Valle Lojano*. Tesis de grado previo de la obtención al título de Magíster en Sistemas Integrados de Gestión. España: Universidad Técnica Particular de Loja y Universidad de Huelva, 2011.
4. REAL ACADEMIA ESPAÑOLA. *Diccionario de la lengua española (DRAE)* [en línea]. España: s.n., 2001. [Accedido 5 February 2014]. Disponible desde: <http://lema.rae.es/drae/?val=sistema>.
5. ¿Qué son los sistemas de gestión? En: *The British Standards Institution* [en línea]. 2013. [Accedido 1 May 2014]. Disponible desde: <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>.
5. ORQUIDIA FERRER RODRÍGUEZ; RAYNER BESTARD DÍAZ. *Sistema de Gestión de la Información para las Ópticas de Cuba*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
6. CARIDAD DARROMÁN SAVIGNE; REYNERIO VELÁZQUEZ LEYVA. El proceso de gestión y la gestión económica en las empresas. En: *Observatorio de la Economía Latinoamericana* [en línea]. 2011. [Accedido 21 January 2014]. Disponible desde: <http://www.eumed.net/cursecon/ecolat/cu/2011/>.
8. ¿Qué es PYME? En: *Empresa e Industria* [en línea]. 2014. [Accedido 5 May 2014]. Disponible desde: http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/sme-definition/index_es.htm.
9. Que es ContaPyme. En: *CONTAPYME* [en línea]. 2014. [Accedido 25 January 2014]. Disponible desde: <http://www.contapyme.com/software-contable>.
10. Que es SAP. En: *Informática-Hoy* [en línea]. 2012. [Accedido 3 March 2014]. Disponible desde: <http://www.informatica-hoy.com.ar/sap/Que-es-SAP.php>.
11. ASSETS: Sistema de Gestión Integral. En: *ASSETS* [en línea]. 2004. [Accedido 8 May 2014]. Disponible desde: <http://www.assets.co.cu/contabilidad.asp>.
12. MARISEL SOSA PORTEIRO and PEDRO H. COBO MORALES. El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. En: *Consultoría Informacional*. Casa Consultora DISAIC [en línea]. 2008. [Accedido 25 February 2014]. Disponible desde: <http://www.disaic.cu/modules.php?name=content&pa=showpage&pid=818>.
13. TAMARA RODRÍGUEZ SÁNCHEZ, MAIRELYS FERNÁNDEZ GONZÁLEZ and ELIECER CABRERA CASAS. La gestión empresarial de las entidades cubanas. Cedrux a la vuelta de la esquina. En: *Revista Infociencia*. 2011, Vol. 15, no. 2, pp. 4–6.
14. CONDOR INTERPRISE. En: *OPEN Solutions Argentina S.A* [en línea]. 2009. [Accedido 15 February 2014]. Disponible desde: <http://www.open-sol.com.ar/php/condorenterprise.php>.

15. PATRICIO LETELIER and M^a CARMEN PENADÉS. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. España. Universidad Politécnica de Valencia, 2012.
16. LIUBA TORRES ALMIRA. *Intranet del Ministerio de la Informática y las Comunicaciones*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2012.
15. JOSÉ H. CANÓS, PATRICIO LETELIER, M^a CARMEN PENADÉS. *Metodologías Ágiles en el Desarrollo de Software*. Valencia. DSIC -Universidad Politécnica de Valencia.
16. ROBERTH G. FIGUEROA, CAMILO J. SOLÍS and ARMANDO A. CABRERA. *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*. España. Universidad Técnica Particular de Loja.
19. ELISA LISBETH VÁZQUEZ CHACÓN. *Plugin generador de StoryBoards para el Sistema Gestor de Procesos de Medía*. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
20. JOSÉ ENRIQUE GONZÁLEZ CORNEJO. El Lenguaje de Modelado Unificado. En: *Doc/RS* [en línea]. 2008. [Accedido 21 March 2014]. Disponible desde: <http://www.docirs.cl/uml.htm>.
19. STEPHEN A. WHITE. *Guía de Referencia y Modelado BP MN. Comprendiendo y utilizando BPMN* [en línea]. Florida: Future Strategies Inc., Book Division, 2009. [Accedido 11 marzo 2014]. Disponible desde: http://www.futstrat.com/books/book_images/Guia%20de%20Referencia%20y%20Modelado%20BPMN%20chap1-2.pdf.
22. EVELYN MENÉNDEZ ALONSO. Herramientas CASE para el proceso de desarrollo de Software. En: *herramientas-case-proceso-desarrollo-software* [en línea]. [Accedido 2 February 2014]. Disponible desde: <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
23. LIANET CABRERA GONZÁLEZ and ENRIQUE ROBERTO POMPA TORRES. Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información. En: *Serie Científica de la Universidad de las Ciencias Informáticas*. 2012, Vol. 10, no. 5, pp. 6.
24. LEOVER ARMANDO GONZÁLEZ RODRÍGUEZ. *Alternativas para el desarrollo de Aplicaciones Web*. Granma. Facultad Regional de Granma de la Universidad de las Ciencias Informáticas, 2011.
25. YANELIS PÉREZ LA ROSA and LUIS ÁNGEL DEL ROSARIO GUEVARA. *Aplicación informática para la gestión de la información asociada al grupo Gestión de la Continuidad de los Servicios Tecnológicos y de la Información*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
26. WINDOWS SERVER. Servidor web (IIS). En: [en línea]. 2014. [Accedido 5 May 2014]. Disponible desde: <http://technet.microsoft.com/es-es/library/cc753433%28v=ws.10%29.aspx>.
27. CENTRO DE APOYO TECNOLÓGICO A EMPRENDEDORES, FUNDACIÓN PARQUE CIENTÍFICO Y and TECNOLÓGICO DE ALBACETE. *Estudio de los Sistemas de Gestión de Contenidos web*. S.l.: Publicado bajo licencia Creative Commons By - Sa, 2012.
28. PEDRO LUIS ROMERO LONDRES. *Sistema para la gestión del Plan de Resultados Anual del profesor y el Currículum VITAE del profesional en el CESIM*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2012.

29. MIGUEL ÁNGEL SÁNCHEZ MAZA. Internet Javascript. En: *Javascript* [en línea]. 2012. [Accedido 11 March 2014]. Disponible desde: <http://books.google.com.cu/books?id=3x09sewjaHIC&printsec=frontcover&dq=javascript+books&hl=es&sa=X&ei=D8dvU7PZHenksASu-YGQAQ&sqi=2&ved=0CC4Q6AEwAA#v=onepage&q&f=false>.
30. PHPSTORM. PHPSTORM DEVELOP Smarte. En: [en línea]. 2013. [Accedido 5 February 2014]. Disponible desde: http://www.jetbrains.com/phpstorm/documentation/phpstorm_web.pdf.
31. MATIAS FOSSATI. Todo sobre MySQL: Libro ideal para ingresar en el mundo de la base de datos MySQL. En: [en línea]. 24 February 2014. [Accedido 5 March 2014]. Disponible desde: <http://books.google.com.cu/books?id=GS3kAgAAQBAJ&pg=PT4&dq=que+es+MySQL&hl=es&sa=X&ei=qxBrU4yoE9KryAS9t4KgAQ&ved=0CEMQ6AEwBA#v=onepage&q=que%20es%20MySQL&f=false>.
32. THE APACHE SOFTWARE FOUNDATION. Apache.org. En: [en línea]. 2012. [Accedido 1 March 2014]. Disponible desde: <http://www.apache.org>.
33. IAN SOMMERVILLE. *Ingeniería de Software*. Séptima edición. Madrid: Pearson Educación. S.A, 2005.
34. IVAR JACOBSON, GRADY BOOCH and JAMES RUMBAUGH. *El Proceso Unificado de Desarrollo de software*. Pearson Educación.S.A. Madrid: s.n., 2000.
35. FRAN GIL RODRÍGUEZ. *Experto en Drupal 7*. S.l.: Forcontu S.L., 2012.
36. ARIANNA GALÁN QUIALA. *Desarrollo de las funcionalidades de solicitud, instalación y configuración de eventos en la plataforma electrónica GRECIA de gestión de eventos científicos*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
37. FRAN GIL RODRÍGUEZ. *Experto en Drupal 7*. S.l.: Forcontu S.L., 2012.
38. MERLYN VÁZQUEZ VENTURA and DANIEL RAÚL RODRÍGUEZ QUINTANA. *Sistema de gestión de información de las redes sociales Facebook y Twitter*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
39. YOLANDA GONZÁLEZ ARECHAVALA and FERNANDO DE CUADRA GARCÍA. *Calidad de software* [en línea]. S.l. [sin fecha]. Disponible desde: http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_2/Comun/Calidad_del_software_I_Yolanda_Gonzalez_Fernando_de_Cuadra_.pdf.
40. ROGER PRESSMAN. *Estrategia Prueba* [en línea]. Sexta edición. S.l.: Mc Graw Hill, 2005. Disponible desde: http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/Ediciones_del_Pressman/Pressman_6ta_edicion/Pressman_Cap_13_Estrategia_Prueba.pdf.
41. JAVIER TUYA, ISABEL RAMOS ROMÁN and JOSÉ JAVIER DOLADO COSÍN. *Técnicas cuantitativas para la gestión de la ingeniería del software*. S.l.: Gesbiblo, S.L, 2007.
42. ERICK CÓRDOVA DORTA. *Procedimiento para la ejecución de las pruebas de unidad a los componentes de la Capa de presentación del sistema CEDRUX*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2011.
43. ROGER PRESSMAN. *Estrategia Prueba* [en línea]. Sexta edición. S.l.: Mc Graw Hill, 2005. Disponible desde:

http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/Ediciones_del_Pressman/Pressman_6ta_edicion/Pressman_Cap_14_Tecnicas_de_Prueba_Parte_1.pdf.

44. ROGER PRESSMAN. *Estrategia Prueba* [en línea]. Sexta edición. S.l.: Mc Graw Hill, 2005. Disponible desde: http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/Ediciones_del_Pressman/Pressman_6ta_edicion/Pressman_Cap_14_Tecnicas_de_Prueba_Parte_1.pdf.

Bibliografía

- ¿Qué es PYME? En: *Empresa e Industria* [en línea]. 2014. [Accedido 5 May 2014]. Disponible desde: http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/sme-definition/index_es.htm.
- ¿Qué son los sistemas de gestión? En: *The British Standards Institution* [en línea]. 2013. [Accedido 1 May 2014]. Disponible desde: <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>.
- ABILEIDY CARDENAS LANZ. and ISMAEL ERNESTO ÁVILA MARTIN. *Desarrollo del Componente Análisis de la Calidad Energética*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2009.
- ADRIÁN PÉREZ ALVAREZ. *Guía básica de Drupal 7*. La Habana. Universidad de las Ciencias Informáticas, 2012.
- ALICIA DELGADO HERNÁNDEZ. *Componente de clasificación de huellas dactilares*. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
- ARIANNA GALÁN QUIALA. *Desarrollo de las funcionalidades de solicitud, instalación y configuración de eventos en la plataforma electrónica GRECIA de gestión de eventos científicos*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
- ASSETS: Sistema de Gestión Integral. En: *ASSETS* [en línea]. 2004. [Accedido 8 May 2014]. Disponible desde: <http://www.assets.co.cu/contabilidad.asp>.
- BEATRIZ PÉREZ LAMANCHA. *Gestión de las Pruebas Funcionales*. Uruguay. Universidad de la República, Montevideo, [sin fecha].
- BENJAMÍN HERRERO HURTADO. *DESARROLLO WEB CON DRUPAL*. Proyecto fin de carrera. S.I.: ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA, 2010.
- Buscador de términos de la Real Academia de Ciencias Económicas y Financieras. En: *Diccionario de Términos Económicos y Financieros* [en línea]. 2009. [Accedido 8 February 2014]. Disponible desde: <http://www.diccionarioracef.com/index.php?a=search&texto=BONIFICACI%F3N&exacta=Y>.
- CARIDAD DARROMÁN SAVIGNE and REYNERIO VELÁZQUEZ LEYVA. El proceso de gestión y la gestión económica en las empresas. En: *Observatorio de la Economía Latinoamericana* [en

- línea]. 2011. [Accedido 21 January 2014]. Disponible desde: <http://www.eumed.net/cursecon/ecolat/cu/2011/>.
- CASA DESARROLLADORA DE AGROWIN Y CONTAPYME. Que es ContaPyme. En: *Programa Contable | Contapyme* [en línea]. 2014. [Accedido 25 January 2014]. Disponible desde: <http://www.contapyme.com/software-contable>.
 - CENTRO DE APOYO TECNOLÓGICO A EMPRENDEDORES, FUNDACIÓN PARQUE CIENTÍFICO Y and TECNOLÓGICO DE ALBACETE. *Estudio de los Sistemas de Gestión de Contenidos web*. S.I.: Publicado bajo licencia Creative Commons By - Sa, 2012.
 - CONDOR INTERPRISE. En: *OPEN Solutions Argentina S.A* [en línea]. 2009. [Accedido 15 February 2014]. Disponible desde: <http://www.open-sol.com.ar/php/condorenterprise.php>.
 - ELISA LISBETH VÁZQUEZ CHACÓN. *Plugin generador de StoryBoards para el Sistema Gestor de Procesos de Media*. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
 - ERICK CÓRDOVA DORTA. *Procedimiento para la ejecución de las pruebas de unidad a los componentes de la Capa de presentación del sistema CEDRUX*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2011.
 - EVELYN MENÉNDEZ ALONSO. Herramientas CASE para el proceso de desarrollo de Software. En: *herramientas-case-proceso-desarrollo-software* [en línea]. [Accedido 2 February 2014]. Disponible desde: <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
 - FRAN GIL RODRÍGUEZ. *Experto en Drupal 7, Nivel avanzado*. S.I.: Forcontu S.L., 2012.
 - FRAN GIL RODRÍGUEZ. *Experto en Drupal 7, Nivel intermediol*. S.I.: Forcontu S.L., 2011.
 - FRAN GIL RODRÍGUEZ. *Experto en Drupal 7. Nivel avanzado*. S.I.: Forcontu S.L., 2012.
 - FRAN GIL RODRÍGUEZ. *Experto en Drupal 7. Nivel inicial*. S.I.: Forcontu S.L., 2011.
 - IAN SOMMERVILLE. *Ingeniería de Software*. Séptima edición. Madrid: Pearson Educación. S.A, 2005.
 - IBM - Rational Rose Modeler. En: *Reatuinal Rose* [en línea]. [Accedido 11 May 2014]. Disponible desde: <http://www-03.ibm.com/software/products/es/rosemod>.

- ISABEL BLANK, LARISSA HERRERA and MIGUEL ORTIZ. Pruebas de Funcionalidad. En: S.I. 2005.
- IVAR JACOBSON, GRADY BOOCH and JAMES RUMBAUGH. *El Proceso Unificado de Desarrollo de software*. Pearson Educación.S.A. Madrid: s.n., 2000.
- JAVIER TUYA, ISABEL RAMOS ROMÁN and JOSÉ JAVIER DOLADO COSÍN. *Técnicas cuantitativas para la gestión de la ingeniería del software*. S.I.: Gesbiblo, S.L, 2007.
- JOSÉ ENRIQUE GONZÁLEZ CORNEJO. ¿Qué es UML? En: *El Lenguaje de Modelado Unificado (UML)* [en línea]. 2008. [Accedido 3 December 2014]. Disponible desde: <http://www.docirs.cl/uml.htm>.
- JOSÉ H. CANÓS, PATRICIO LETELIER and M^a CARMEN PENADÉS. *Métodologías Ágiles en el Desarrollo de Software*. Valencia. DSIC -Universidad Politécnica de Valencia, [sin fecha].
- LEOVER ARMANDO GONZÁLEZ RODRÍGUEZ. *Alternativas para el desarrollo de Aplicaciones Web*. Granma. Facultad Regional de Granma de la Universidad de las Ciencias Informáticas, 2011.
- LEYNER RODRÍGUEZ BORREGO. *Sistema de adquisición de datos para mediciones Multiconteo*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2010.
- LIANET CABRERA GONZÁLEZ and ENRIQUE ROBERTO POMPA TORRES. Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información. En: *Serie Científica de la Universidad de las Ciencias Informáticas*. 2012, Vol. 10, no. 5, pp. 6.
- LIANNY HERNÁNDEZ RAMOS. *Módulo para la gestión de la información del Grupo de Investigación de Automática Aplicada*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
- LISUÉ CAPÓ MARRERO and DAILY GARCÍA DORTA. *Sistema de gestión de solicitudes de descargas para la red de la Universidad Agraria de La Habana*. S.I. Universidad Agraria de La Habana (UNAH) Cuba, 2014.
- LIUBA TORRES ALMIRA. *Intranet del Ministerio de la Informática y las Comunicaciones*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2012.
- LOURDES AJA QUIROGA. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. En: *ACIMED*. 2002, Vol. 10, no. 5, pp. 7–8.

- LUIS ALFREDO BORRERO TORRES. *Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
- MARISEL SOSA PORTEIRO and PEDRO H. COBO MORALES. EI VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. En: *Casa Consultora DISAIC - content* [en línea]. 2008. [Accedido 8 February 2014]. Disponible desde: <http://www.disaic.cu/modules.php?name=content&pa=showpage&pid=818>.
- MATIAS FOSSATI. Todo sobre MySQL: Libro ideal para ingresar en el mundo de la base de datos MySQL. En: [en línea]. 24 February 2014. [Accedido 5 March 2014]. Disponible desde: <http://books.google.com.cu/books?id=GS3kAgAAQBAJ&pg=PT4&dq=que+es+MySQL&hl=es&sa=X&ei=qxBru4yoE9KryAS9t4KgAQ&ved=0CEMQ6AEwBA#v=onepage&q=que%20es%20MySQL&f=false>.
- MERLYN VÁZQUEZ VENTURA and DANIEL RAÚL RODRÍGUEZ QUINTANA. *Sistema de gestión de información de las redes sociales Facebook y Twitter*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
- MIGUEL ÁNGEL SÁNCHEZ MAZA. Internet Javascript. En: *Javascript* [en línea]. 2012. [Accedido 11 March 2014]. Disponible desde: <http://books.google.com.cu/books?id=3x09sewjaHIC&printsec=frontcover&dq=javascript+books&hl=es&sa=X&ei=D8dvU7PZHenksASu-YGQAQ&sqi=2&ved=0CC4Q6AEwAA#v=onepage&q&f=false>.
- ORQUIDIA FERRER RODRÍGUEZ and RAYNER BESTARD DÍAZ. *Sistema de Gestión de la Información para las Ópticas de Cuba*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
- PATRICIO LETELIER and M^a CARMEN PENADÉS. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. España. Universidad Politécnica de Valencia, 2012.
- PHPSTORM. PHPSTORM DEVELOP Smarte. En: [en línea]. 2013. [Accedido 5 February 2014]. Disponible desde: http://www.jetbrains.com/phpstorm/documentation/phpstorm_web.pdf.
- Que es SAP. En: *Informática-Hoy* [en línea]. 2012. [Accedido 3 March 2014]. Disponible desde: <http://www.informatica-hoy.com.ar/sap/Que-es-SAP.php>.
- REAL ACADEMIA ESPAÑOLA. *Diccionario de la lengua española (DRAE)* [en línea]. España: s.n., 2001. [Accedido 5 February 2014]. Disponible desde: <http://lema.rae.es/drae/?val=sistema>.

- REAL ACADEMIA ESPAÑOLA. *Diccionario de la lengua española (DRAE)* [en línea]. España: s.n., 2001. [Accedido 5 February 2014]. Disponible desde: <http://lema.rae.es/drae/?val=gesti%C3%B3n>.
- RICARDO GONZÁLEZ CHANG. *Empleo de las Técnicas Star para la Recuperación de la Información*. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
- ROBERTH G. FIGUEROA, CAMILO J. SOLÍS and ARMANDO A. CABRERA. *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*. España. Universidad Técnica Particular de Loja, [sin fecha].
- ROGER PRESSMAN. *Estrategia Prueba* [en línea]. Sexta edición. S.I.: Mc Graw Hill, 2005. Disponible desde: http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/Ediciones_del_Pressman/Pressman_6ta_edicion/Pressman_Cap_13_Estrategia_Prueba.pdf.
- ROGER PRESSMAN. *Estrategia Prueba* [en línea]. Sexta edición. S.I.: Mc Graw Hill, 2005. Disponible desde: http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/Ediciones_del_Pressman/Pressman_6ta_edicion/Pressman_Cap_14_Tecnicas_de_Prueba_Parte_1.pdf.
- ROGER PRESSMAN. *Estrategia Prueba* [en línea]. Sexta edición. S.I.: Mc Graw Hill, 2005. Disponible desde: http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/Ediciones_del_Pressman/Pressman_6ta_edicion/Pressman_Cap_14_Tecnicas_de_Prueba_Parte_1.pdf.
- ROLANDO GUTIÉRREZ ROSALES and DAYANA CABRERA FLEITES. *Adaptación de OpenUp/Basic para el Polo Productivo de BioInformática*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2009.
- ROSA NATIVIDAD SEGUENCIA AGUAYO. *Desarrollo de sistema de gestión de calidad aplicando la norma ISO 9001:2008 para la empresa Mesón del Valle Lojano*. Tesis de grado previo de la obtención al título de Magíster en Sistemas Integrados de Gestión. España: Universidad Técnica Particular de Loja y Universidad de Huelva, 2011.
- STEPHEN A. WHITE. *Guía de Referencia y Modelado BP MN. Comprendiendo y utilizando BPMN* [en línea]. florida: Future Strategies Inc., Book Division, 2009. [Accedido 11 March 2014]. Disponible desde: http://www.futstrat.com/books/book_images/Guia%20de%20Referencia%20y%20Modelado%20BP%20MN%20chap1-2.pdf.

- TAMARA RODRÍGUEZ SÁNCHEZ, MAIRELYS FERNÁNDEZ GONZÁLEZ and ELIECER CABRERA CASAS. La gestión empresarial de las entidades cubanas. Cedrux a la vuelta de la esquina. En: *Revista Infociencia*. 2011, Vol. 15, no. 2, pp. 4–6.
- THE APACHE SOFTWARE FOUNDATION. Apache.org. En: [en línea]. 2012. [Accedido 1 March 2014]. Disponible desde: <http://www.apache.org>.
- WINDOWS SERVER. Servidor web (IIS). En: [en línea]. 2014. [Accedido 5 May 2014]. Disponible desde: <http://technet.microsoft.com/es-es/library/cc753433%28v=ws.10%29.aspx>).
- YADIRA MACHADO PEÑA. *Estrategia de Pruebas de función*. S.I. Universidad de las Ciencias Informáticas, [sin fecha].
- YANELIS PÉREZ LA ROSA and LUIS ÁNGEL DEL ROSARIO GUEVARA. *Aplicación informática para la gestión de la información asociada al grupo Gestión de la Continuidad de los Servicios Tecnológicos y de la Información*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
- YILIAN BEATRIZ GUERRA ÁVILA and JAVIER SEVILLA MENDOZA. *Portal web Soy Cuba*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2013.
- YISEL PACHECO DUQUE. *Diseño de un sistema de control de pago de dietas y facturas para el Departamento de Transportaciones Nacionales de la Universidad de las Ciencias Informáticas*. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2010.
- YOLANDA GONZÁLEZ ARECHAVALA and FERNANDO DE CUADRA GARCÍA. *Calidad de software* [en línea]. S.I. [sin fecha]. Disponible desde: http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_2/Comun/Calidad_del_software_I_Yolanda_Gonzalez_Fernando_de_Cuadra_.pdf.

Glosario de términos.

Actores del Negocio: Es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

LDAP: es un protocolo a nivel de aplicación, que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversas informaciones en el entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas. Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contactos del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc.).

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

Scripts: Guión o conjunto de instrucciones

Trabajadores del Negocio: Es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol. El trabajador no se beneficia con el negocio, solo trabaja en él.

Anexos

Anexo 1: Tabla de comparación entre metodologías ágiles y tradicionales.

Tabla 14: Diferencias entre metodologías ágiles y tradicionales.

Metodologías Tradicionales	Metodologías Ágiles
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Impuestas externamente	Impuestas internamente (por el equipo de desarrollo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Más artefactos	Pocos artefactos
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software

Anexo 2: Modelo de autorizo de pago de bonificación.



Universidad de las Ciencias Informáticas
Modelo de Bonificación

La Habana, ____ de ____ de ____

Se autoriza al estudiante: _____ con CI: _____ de la Facultad _____ a viajar y hacer uso de la Bonificación del 50% del pasaje aprobado por el VII Congreso de la FEU.

	Origen	Destino
Ida:		
Fecha __/__/__	Provincia _____	_____
	Municipio _____	_____
Regreso:		
Fecha __/__/__	Provincia _____	_____
	Municipio _____	_____
Firma Autorizada _____		Estudiante _____

Figura 19: Modelo de autorizo de pago de bonificación.

Anexo 3: Modelo de bonificación

UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS				LIQUIDACION DE GASTOS DE VIAJES BONIFICACION 50% DEL PASAJE				MODELO		
Dirección:				Código:				D	M	A
Nombre y Apellidos:										
Motivo del Viaje:				Fecha				D	M	Hora
Bonificación 50%				Fecha de Salida						
				Fecha de Regreso						
AUTORIZADO										
Liquidación				D	M	A	# Modelo B:			
C.I:				Concepto	Total	Alimentación	Hospedaje	Desayuno	Transporte	
Liquidado	D	M	A	Utilizado						
					A Entregar					
Custodio	D	M	A	Solicitud No.	Anotado			Número		

Figura 20: Modelo de Pago de Bonificación.

Anexo 4: Historias de usuarios.

Tabla 15: HU buscar estudiante.

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Autenticar usuario
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 día
<p>Descripción:</p> <p>La HU inicia cuando el usuario se dirige a la opción “Inicio de sesión” que se encuentra situada en la parte lateral izquierda del sistema. Para que el usuario pueda autenticarse en el sistema debe entrar los siguientes datos:</p> <ul style="list-style-type: none"> • Usuario (obligatorio, campo de texto) • Contraseña (obligatorio, campo de texto) <p>Luego dar clic en la opción: “Iniciar Sesión”, finalizando así la HU.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse. 2. El usuario debe autenticarse con usuario y contraseña de dominio de (LDAP). 3. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. 	
<p>Prototipo de interfaz:</p> <div style="text-align: center;">  </div>	

Tabla 17: HU eliminar estudiante.

Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: Eliminar estudiante.
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 1
Prioridad en Negocio: Bajo	Puntos Estimados: 2 días.
Riesgo en Desarrollo: Medio	Puntos Reales: 2 días.
<p>Descripción:</p> <p>La HU inicia cuando el usuario autenticado en el sistema con privilegios de administración accede al módulo: “Buscar estudiantes”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema muestra al usuario una interfaz con un listado de los estudiantes existentes en el sistema, seguida de las opciones: “Eliminar” y “Adicionar bonificación”. El usuario selecciona el estudiante que desea eliminar, dando clic en la opción: “Eliminar”.</p> <p>El sistema muestra un mensaje de confirmación de la acción seguido de las opciones:</p> <ul style="list-style-type: none"> • Eliminar • Cancelar <p>El usuario selecciona la opción “Cancelar”, el sistema deshace la operación y muestra el listado de estudiantes existentes en el sistema.</p> <p>El usuario selecciona la opción “Eliminar”, el sistema elimina el usuario de la base de datos, finalizando así la HU.</p>	
Observaciones:	
<p>Prototipo de interfaz:</p> <div style="text-align: center; padding: 20px;"> <p>¿Seguro que quiere eliminar <i>Richard Luciano García</i>?</p> <p>Esta acción no se puede deshacer.</p> <p>Eliminar Cancelar</p> </div>	

Tabla 18: HU eliminar bonificación.

Historia de Usuario	
Número: HU_5	Nombre Historia de Usuario: Eliminar bonificación
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana
<p>Descripción:</p> <p>La HU inicia cuando el usuario autenticado en el sistema con privilegios de administración accede al módulo: “Gestionar bonificaciones”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema le muestra al usuario una interfaz con un listado de las bonificaciones existentes en el sistema, las pendientes por cobrar y las cobradas. El usuario busca en las pendientes por cobrar la bonificación que desea eliminar y le da clic a la opción: “Registrar cobro”. El sistema muestra un formulario con los siguientes campos:</p> <ul style="list-style-type: none"> • Fecha de ida (input de tipo date) • Fecha de regreso (input de tipo date) • Valor de pasaje de ida (campo de texto) • Valor de pasaje de regreso (campo de texto) • Destino (combobox) • Regreso (combobox) • Fecha de cobro (input de tipo date) • Modelo (campo de texto) <p>Seguida de las opciones: “Guardar” y “Eliminar”. El usuario selecciona la opción: “Eliminar”. El sistema muestra un mensaje de confirmación de la acción seguido de las opciones:</p> <ul style="list-style-type: none"> • Eliminar • Cancelar <p>El usuario selecciona la opción “Cancelar”, el sistema deshace la operación y muestra el listado de bonificaciones existentes en el sistema.</p> <p>El usuario selecciona la opción “Eliminar”, el sistema elimina la bonificación de la base de datos, finalizando así la HU.</p>	
Observaciones:	

Prototipo de interfaz:

¿Seguro que quiere eliminar para *Richard Luciano García*?

Esta acción no se puede deshacer.

Eliminar

Cancelar

Tabla 19: HU Buscar Bonificación.

Historia de Usuario	
Número: HU_8	Nombre Historia de Usuario: Buscar bonificación
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2 días.
Riesgo en Desarrollo: Medio	Puntos Reales: 2 días.
<p>Descripción:</p> <p>La HU inicia cuando el usuario autenticado en el sistema con permiso de acceso al módulo: “Gestionar bonificaciones”, accede a este. El sistema muestra una interfaz para introducir la información referente a la búsqueda con los siguientes campos:</p> <ul style="list-style-type: none"> • Usuario (campo de texto) • Solapín (campo de texto) • CI (campo de texto) • Folio (campo de texto) <p>El usuario introduce cualquiera de los cuatros criterios de búsqueda y presiona el botón: “Filtrar”. El sistema valida los datos y muestra los resultados existentes, finalizando así la HU.</p>	
<p>Observaciones:</p> <p>2- Si el usuario introduce la información de forma incorrecta, el sistema debe emitir un mensaje notificando el error.</p>	
<p>Prototipo de interfaz:</p> <div style="text-align: center; margin-top: 20px;"> <h2>Listado de bonificaciones</h2> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="text-align: center;"> <p>CI</p> <input style="width: 150px; height: 25px;" type="text"/> </div> <div style="text-align: center;"> <p>Usuario</p> <input style="width: 150px; height: 25px;" type="text"/> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="text-align: center;"> <p>Solapín</p> <input style="width: 150px; height: 25px;" type="text"/> </div> <div style="text-align: center;"> <p>Folio</p> <input style="width: 150px; height: 25px;" type="text"/> </div> </div> <div style="text-align: right; margin-top: 20px;"> <input style="background-color: #ccc; border: 1px solid #ccc; padding: 5px 15px; border-radius: 5px;" type="button" value="Filtrar"/> </div> </div>	

Tabla 20: HU mostrar bonificación.

Historia de Usuario	
Número: HU_9	Nombre Historia de Usuario: Mostrar bonificación
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 2 días.
Riesgo en Desarrollo: Bajo	Puntos Reales: 2 días.
<p>Descripción:</p> <p>La HU inicia cuando el usuario autenticado en el sistema con permiso de acceso al módulo: “Gestionar bonificaciones”, accede a este. El sistema le muestra al usuario un listado con las bonificaciones “Cobradas” y “Pendientes” de cobro existentes en el sistema, finalizando así la HU.</p>	
<p>Observaciones:</p> <p>1- Si no existen bonificaciones el sistema, no se mostrará ninguna información</p>	
<p>Prototipo de interfaz:</p> <div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <p style="text-align: center;">Listado de bonificaciones</p> <p style="text-align: center;"> <input type="button" value="Pendientes"/> <input type="button" value="Cobradas"/> </p> <hr/> <p> <input type="text" value="CI"/> <input type="text" value="Usuario"/> </p> <p> <input type="text" value="Solapín"/> <input type="text" value="Folio"/> <input type="button" value="Filtrar"/> </p> <p>Estudiante: Dannelis Melgarejo Pozo</p> <ul style="list-style-type: none"> • Folio: 1000030 Fecha ida: 11-03-2014 Destino: Artemisa»Alquizar Fecha regreso: 15-03-2014 Regreso: Camaguey»Camaguey Total a cobrar: 95 Acciones: Ya ha realizado el cobro. Imprimir </div> <div style="border: 1px solid #ccc; padding: 10px;"> <p style="text-align: center;">Listado de bonificaciones</p> <p style="text-align: center;"> <input type="button" value="Pendientes"/> <input type="button" value="Cobradas"/> </p> <hr/> <p> <input type="text" value="CI"/> <input type="text" value="Usuario"/> </p> <p> <input type="text" value="Solapín"/> <input type="text" value="Folio"/> <input type="button" value="Filtrar"/> </p> <p>Estudiante: Abel Salas López</p> <ul style="list-style-type: none"> • Folio: 1 000 029 Fecha ida: 04-03-2014 Destino: Artemisa»Alquizar Fecha regreso: 10-03-2014 Regreso: Camaguey»Camaguey Total a cobrar: 62.5 Acciones: Registrar cobro </div>	

Tabla 21: HU agregar presupuesto.

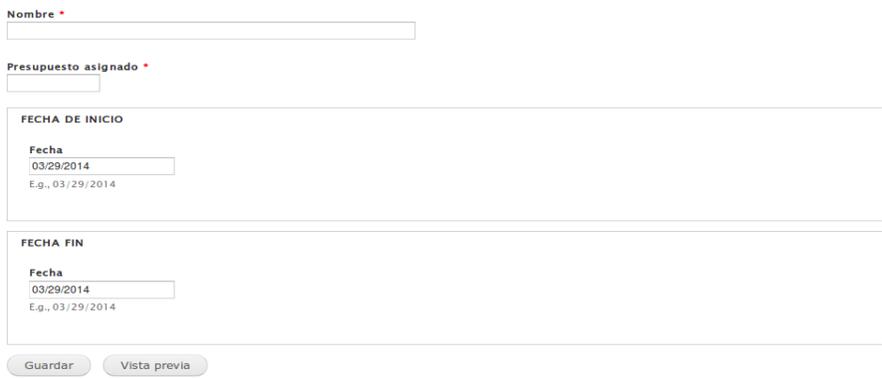
Historia de Usuario	
Número: HU_10	Nombre Historia de Usuario: Agregar presupuesto
Modificación de Historia de Usuario Número: 1	
Usuario: Sucl Daudinot Caricabeur	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Media	Puntos Reales: 1 semana
<p>Descripción:</p> <p>La HU inicia el usuario autenticado en el sistema con privilegios de administración accede al módulo “Gestionar presupuesto”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema le muestra al usuario una interfaz con un listado de los presupuestos existentes en el sistema y la opción: “Agregar nuevo presupuesto”. El usuario selecciona la opción y el sistema le muestra al usuario un formulario con los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre (campo de texto) • Presupuesto asignado (campo de texto) • Fecha de inicio (input de tipo date) • Fecha fin (input de tipo date) <p>El usuario registra la información y presiona el botón: “Guardar”. El sistema verifica y guarda la información, finalizando así la HU.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> • Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. • Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse. 	
<p>Prototipo de interfaz:</p> 	

Tabla 22: HU buscar presupuesto.

Historia de Usuario	
Número: HU_11	Nombre Historia de Usuario: Buscar presupuesto
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 2 días.
Riesgo en Desarrollo: Bajo	Puntos Reales: 2 días.
<p>Descripción:</p> <p>La HU inicia el usuario autenticado en el sistema con privilegios de administración accede a la opción: “Gestionar presupuesto”. El sistema muestra una interfaz para introducir la información referente a la búsqueda de presupuesto con los siguientes campos:</p> <ul style="list-style-type: none"> • Fecha de inicio (input de tipo date). • Fecha fin (input de tipo date) <p>El usuario introduce cualquiera los datos y presiona el botón: “Filtrar”. El sistema valida los datos y muestra los resultados existentes, finalizando así la HU.</p>	
<p>Observaciones:</p> <p>3- Si el usuario introduce la información de forma incorrecta, el sistema debe emitir un mensaje notificando el error.</p>	
<p>Prototipo de interfaz:</p> <div style="text-align: center;"> <h2>Listado presupuesto</h2> <p>Agregar nuevo presupuesto</p> <div style="display: flex; justify-content: space-around; align-items: flex-end;"> <div style="text-align: center;"> <p>Fecha de inicio</p> <input type="text"/> <small>E.g., 2014-03-29</small> </div> <div style="text-align: center;"> <p>Fecha fin</p> <input type="text"/> <small>E.g., 2014-03-29</small> </div> <div style="text-align: center;"> <input type="button" value="Filtrar"/> </div> </div> </div>	

Tabla 23: HU editar presupuesto.

Historia de Usuario	
Número: HU_12	Nombre Historia de Usuario: Editar presupuesto
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana
<p>Descripción:</p> <p>La HU inicia cuando el usuario autenticado en el sistema con privilegios de administración accede al módulo: “Gestionar presupuesto”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema le muestra al usuario un listado de los presupuestos existentes en el sistema, seguido de las opciones: “editar” y “eliminar”. El usuario selecciona la opción: “editar”. El sistema muestra un formulario con los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre (obligatorio, campo de texto) • Presupuesto asignado (obligatorio, campo de texto) • Fecha de inicio (input de tipo date) • Fecha fin (input de tipo date) <p>El usuario registra la información y presiona el botón: “Guardar”. El sistema verifica y actualiza la información, finalizando así la HU.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Si el usuario modifica la información de forma incorrecta, el sistema emite un mensaje notificando el error. 2. Si el usuario modifica la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse. 	

Prototipo de interfaz:

Nombre *

Presupuesto 2014

Presupuesto asignado *

1000

FECHA DE INICIO

Fecha

01/01/2014

E.g., 03/29/2014

FECHA FIN

Fecha

12/31/2014

E.g., 03/29/2014

Guardar

Vista previa

Eliminar

Tabla 24: HU eliminar presupuesto.

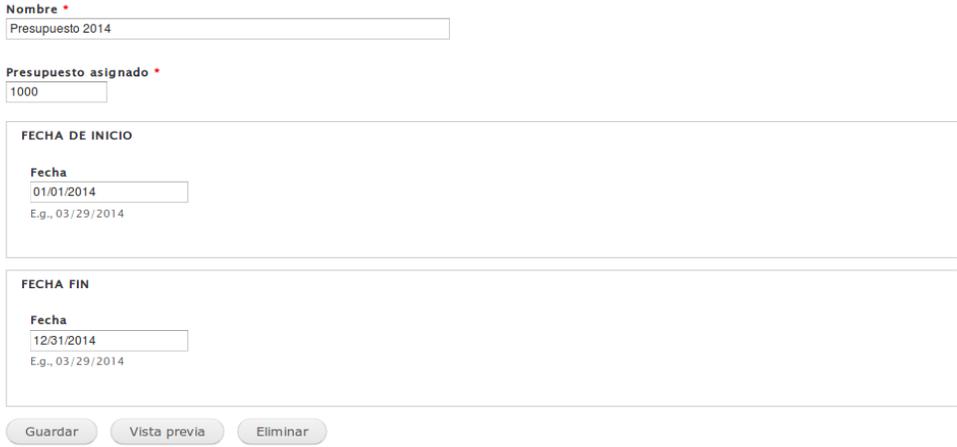
Historia de Usuario	
Número: HU_13	Nombre Historia de Usuario: Eliminar presupuesto
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana
<p>Descripción:</p> <p>La HU inicia cuando el usuario autenticado en el sistema con privilegios de administración accede al módulo: “Gestionar presupuesto”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema le muestra al usuario un listado de los presupuestos existentes en el sistema, seguido de las opciones: “editar” y “eliminar”. El usuario selecciona la opción: “eliminar”. El sistema muestra un mensaje de confirmación de la acción seguido de las opciones:</p> <ul style="list-style-type: none"> • Eliminar. • Cancelar. <p>El usuario selecciona la opción “Cancelar”, el sistema deshace la operación y muestra el listado de bonificaciones existentes en el sistema.</p> <p>El usuario selecciona la opción “Eliminar”, el sistema elimina la bonificación de la base de datos, finalizando así la HU.</p>	
Observaciones:	
<p>Prototipo de interfaz:</p> 	

Tabla 25: HU mostrar presupuesto

Historia de Usuario																
Número: HU_14	Nombre Historia de Usuario: Mostrar presupuestos															
Modificación de Historia de Usuario Número: 1																
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 2															
Prioridad en Negocio: Media	Puntos Estimados: 1 semana															
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana															
Descripción: La HU inicia cuando el usuario autenticado en el sistema con privilegios de administración accede al módulo: “ Gestionar presupuesto ”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema le muestra al usuario un listado con los presupuestos existentes en el sistema, finalizando así la HU.																
Observaciones: 1.-Si no existen páginas el sistema, no se mostrará ninguna información.																
Prototipo de interfaz: <div style="text-align: center;"> <h3>Listado presupuesto</h3> <p>Agregar nuevo presupuesto</p> <p> Fecha de inicio <input type="text"/> Fecha fin <input type="text"/> <input type="button" value="Filtrar"/> </p> <p> <small>E.g., 2014-03-29</small> <small>E.g., 2014-03-29</small> </p> <table border="1"> <thead> <tr> <th>Presupuesto</th> <th>Presupuesto asignado</th> <th>Presupuesto real</th> <th>Fecha de inicio</th> <th>Fecha fin</th> </tr> </thead> <tbody> <tr> <td>Presupuesto 2014</td> <td>1 000</td> <td>655</td> <td>01-01-2014</td> <td>31-12-2014</td> </tr> <tr> <td>Presupuesto 2013</td> <td>1 000 000</td> <td>1 000 000</td> <td>01-01-2013</td> <td>31-12-2013</td> </tr> </tbody> </table> </div>		Presupuesto	Presupuesto asignado	Presupuesto real	Fecha de inicio	Fecha fin	Presupuesto 2014	1 000	655	01-01-2014	31-12-2014	Presupuesto 2013	1 000 000	1 000 000	01-01-2013	31-12-2013
Presupuesto	Presupuesto asignado	Presupuesto real	Fecha de inicio	Fecha fin												
Presupuesto 2014	1 000	655	01-01-2014	31-12-2014												
Presupuesto 2013	1 000 000	1 000 000	01-01-2013	31-12-2013												

Tabla 26. HU generar reportes.

Historia de Usuario									
Número: HU_15	Nombre Historia de Usuario: Generar reportes								
Modificación de Historia de Usuario Número: 1									
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 2								
Prioridad en Negocio: Alta	Puntos Estimados: 1 semana								
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana								
<p>Descripción:</p> <p>La HU inicia cuando el usuario autenticado en el sistema con privilegios de administración accede al módulo: “Reportes”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema le muestra al usuario una interfaz con los siguientes campos:</p> <ul style="list-style-type: none"> • Facultad(combobox) • Fecha de inicio (input de tipo date) • Fecha final (input de tipo date) • Elementos por página (combobox) <p>El usuario registra la información y presiona el botón: “Filtrar”. El sistema muestra al usuario un listado de la información solicitada existente en el sistema.</p>									
<p>Observaciones:</p> <p>1.-Si no existe la información en el sistema, no se mostrará ninguna información.</p>									
<p>Prototipo de interfaz:</p> <div style="border: 1px solid black; padding: 10px;"> <p style="text-align: center;">Reporte</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; vertical-align: top;"> <p>Facultad</p> <div style="border: 1px solid gray; padding: 2px;">- Cualquiera - ▾</div> </td> <td style="width: 25%; vertical-align: top;"> <p>Fecha de inicio</p> <div style="border: 1px solid gray; width: 100%; height: 20px;"></div> <p style="font-size: small;">E.g., 2014-03-30</p> </td> <td style="width: 25%; vertical-align: top;"> <p>Elementos por página</p> <div style="border: 1px solid gray; padding: 2px;">5 ▾</div> </td> <td style="width: 25%; vertical-align: middle; text-align: center;"> <div style="border: 1px solid gray; padding: 5px; background-color: #f0f0f0;">Filtrar</div> </td> </tr> <tr> <td></td> <td style="vertical-align: top;"> <p>Fecha final</p> <div style="border: 1px solid gray; width: 100%; height: 20px;"></div> <p style="font-size: small;">E.g., 2014-03-30</p> </td> <td></td> <td></td> </tr> </table> </div>		<p>Facultad</p> <div style="border: 1px solid gray; padding: 2px;">- Cualquiera - ▾</div>	<p>Fecha de inicio</p> <div style="border: 1px solid gray; width: 100%; height: 20px;"></div> <p style="font-size: small;">E.g., 2014-03-30</p>	<p>Elementos por página</p> <div style="border: 1px solid gray; padding: 2px;">5 ▾</div>	<div style="border: 1px solid gray; padding: 5px; background-color: #f0f0f0;">Filtrar</div>		<p>Fecha final</p> <div style="border: 1px solid gray; width: 100%; height: 20px;"></div> <p style="font-size: small;">E.g., 2014-03-30</p>		
<p>Facultad</p> <div style="border: 1px solid gray; padding: 2px;">- Cualquiera - ▾</div>	<p>Fecha de inicio</p> <div style="border: 1px solid gray; width: 100%; height: 20px;"></div> <p style="font-size: small;">E.g., 2014-03-30</p>	<p>Elementos por página</p> <div style="border: 1px solid gray; padding: 2px;">5 ▾</div>	<div style="border: 1px solid gray; padding: 5px; background-color: #f0f0f0;">Filtrar</div>						
	<p>Fecha final</p> <div style="border: 1px solid gray; width: 100%; height: 20px;"></div> <p style="font-size: small;">E.g., 2014-03-30</p>								

Tabla 27: HU exportar a Excel.

Historia de Usuario																																	
Número: HU_16	Nombre Historia de Usuario: Exportar reportes a Excel																																
Modificación de Historia de Usuario Número: 1																																	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 2																																
Prioridad en Negocio: Media	Puntos Estimados: 1 semana																																
Riesgo en Desarrollo: Medio	Puntos Reales: 1 día																																
<p>Descripción:</p> <p>La HU inicia cuando el usuario autenticado en el sistema con privilegios de administración accede al módulo: “Reportes”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema muestra al usuario un listado con la información solicitada, seguida de las opciones: “Exportar a Excel” y “Exportar a Word”. El usuario selecciona la opción: “Exportar a Excel”. El sistema muestra una ventana con los siguientes campos:</p> <ul style="list-style-type: none"> • Abrir con (input de tipo radio) • Guardar archivo (input de tipo radio) <p>El usuario marca la opción deseada y presiona el botón: “Cancelar”, el sistema deshace la operación y muestra el listado de la información solicitada existente en el sistema.</p> <p>El usuario selecciona la opción “Guardar archivo”, el sistema guarda la información, finalizando así la HU</p>																																	
Observaciones:																																	
<p>Prototipo de interfaz:</p> <p>Reporte</p> <p>Facultad <input type="text" value="- Cualquiera -"/> Fecha de inicio <input type="text"/> Elementos por página <input type="text" value="5"/> <input type="button" value="Filtrar"/></p> <p>E.g., 2014-03-30</p> <p>Fecha final <input type="text"/></p> <p>E.g., 2014-03-30</p> <table border="1"> <thead> <tr> <th>Nombres y Apellidos</th> <th>Facultad</th> <th>CI</th> <th>Solapín</th> <th>Destino</th> <th>Regreso</th> <th>Total a cobrar</th> <th>Fecha cobro</th> </tr> </thead> <tbody> <tr> <td>Dannelis Melgarejo Pozo</td> <td>Facultad 3</td> <td>5555555555</td> <td>E112769</td> <td>Artemisa»Alquizar</td> <td>Camaguey»Camaguey</td> <td>95</td> <td>04-04-2014</td> </tr> <tr> <td>Abel Salas López</td> <td>Facultad 1</td> <td>5555555555</td> <td>E101244</td> <td>Artemisa»Alquizar</td> <td>Camaguey»Camaguey</td> <td>62.5</td> <td>17-03-2014</td> </tr> <tr> <td>Adalid Cecilia Martínez Álvarez</td> <td>Facultad 1</td> <td>2222222213</td> <td>E100007</td> <td>Artemisa»Alquizar</td> <td>Artemisa»Artemiza</td> <td>17.5</td> <td>31-03-2014</td> </tr> </tbody> </table> <p>Total a pagar: \$175</p> <p style="text-align: right;"> </p>		Nombres y Apellidos	Facultad	CI	Solapín	Destino	Regreso	Total a cobrar	Fecha cobro	Dannelis Melgarejo Pozo	Facultad 3	5555555555	E112769	Artemisa»Alquizar	Camaguey»Camaguey	95	04-04-2014	Abel Salas López	Facultad 1	5555555555	E101244	Artemisa»Alquizar	Camaguey»Camaguey	62.5	17-03-2014	Adalid Cecilia Martínez Álvarez	Facultad 1	2222222213	E100007	Artemisa»Alquizar	Artemisa»Artemiza	17.5	31-03-2014
Nombres y Apellidos	Facultad	CI	Solapín	Destino	Regreso	Total a cobrar	Fecha cobro																										
Dannelis Melgarejo Pozo	Facultad 3	5555555555	E112769	Artemisa»Alquizar	Camaguey»Camaguey	95	04-04-2014																										
Abel Salas López	Facultad 1	5555555555	E101244	Artemisa»Alquizar	Camaguey»Camaguey	62.5	17-03-2014																										
Adalid Cecilia Martínez Álvarez	Facultad 1	2222222213	E100007	Artemisa»Alquizar	Artemisa»Artemiza	17.5	31-03-2014																										

Tabla 28: HU exportar reportes a Word.

Historia de Usuario																																	
Número: HU_17	Nombre Historia de Usuario: Exportar reportes a Word																																
Modificación de Historia de Usuario Número: 1																																	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 2																																
Prioridad en Negocio: Media	Puntos Estimados: 1 semana																																
Riesgo en Desarrollo: Medio	Puntos Reales: 1 día																																
<p>Descripción:</p> <p>La HU inicia cuando el usuario autenticado en el sistema con privilegios de administración accede al módulo: “Reportes”, que se encuentra ubicado en la parte lateral izquierda del sistema. El sistema muestra al usuario un listado con la información solicitada, seguida de las opciones: “Exportar a Excel” y “Exportar a Word”. El usuario selecciona la opción: “Exportar a Word”. El sistema muestra una ventana con los siguientes campos:</p> <ul style="list-style-type: none"> • Abrir con (input de tipo radio) • Guardar archivo (input de tipo radio) <p>El usuario marca la opción deseada y presiona el botón: “Cancelar”, el sistema deshace la operación y muestra el listado de la información solicitada existente en el sistema.</p> <p>El usuario selecciona la opción “Guardar archivo”, el sistema guarda la información, finalizando así la HU</p>																																	
Observaciones:																																	
<p>Prototipo de interfaz:</p> <p>Reporte</p> <p>Facultad: <input type="text" value="- Cualquiera -"/> Fecha de inicio: <input type="text"/> E.g., 2014-03-30 Fecha final: <input type="text"/> E.g., 2014-03-30 Elementos por página: <input type="text" value="5"/> <input type="button" value="Filtrar"/></p> <table border="1"> <thead> <tr> <th>Nombres y Apellidos</th> <th>Facultad</th> <th>CI</th> <th>Solapín</th> <th>Destino</th> <th>Regreso</th> <th>Total a cobrar</th> <th>Fecha cobro</th> </tr> </thead> <tbody> <tr> <td>Dannelis Melgarejo Pozo</td> <td>Facultad 3</td> <td>5555555555</td> <td>E112769</td> <td>Artemisa»Alquizar</td> <td>Camaguey»Camaguey</td> <td>95</td> <td>04-04-2014</td> </tr> <tr> <td>Abel Salas López</td> <td>Facultad 1</td> <td>5555555555</td> <td>E101244</td> <td>Artemisa»Alquizar</td> <td>Camaguey»Camaguey</td> <td>62.5</td> <td>17-03-2014</td> </tr> <tr> <td>Adalid Cecilia Martínez Álvarez</td> <td>Facultad 1</td> <td>2222222213</td> <td>E100007</td> <td>Artemisa»Alquizar</td> <td>Artemisa»Artemiza</td> <td>17.5</td> <td>31-03-2014</td> </tr> </tbody> </table> <p>Total a pagar: \$175</p> <p style="text-align: right;"> </p>		Nombres y Apellidos	Facultad	CI	Solapín	Destino	Regreso	Total a cobrar	Fecha cobro	Dannelis Melgarejo Pozo	Facultad 3	5555555555	E112769	Artemisa»Alquizar	Camaguey»Camaguey	95	04-04-2014	Abel Salas López	Facultad 1	5555555555	E101244	Artemisa»Alquizar	Camaguey»Camaguey	62.5	17-03-2014	Adalid Cecilia Martínez Álvarez	Facultad 1	2222222213	E100007	Artemisa»Alquizar	Artemisa»Artemiza	17.5	31-03-2014
Nombres y Apellidos	Facultad	CI	Solapín	Destino	Regreso	Total a cobrar	Fecha cobro																										
Dannelis Melgarejo Pozo	Facultad 3	5555555555	E112769	Artemisa»Alquizar	Camaguey»Camaguey	95	04-04-2014																										
Abel Salas López	Facultad 1	5555555555	E101244	Artemisa»Alquizar	Camaguey»Camaguey	62.5	17-03-2014																										
Adalid Cecilia Martínez Álvarez	Facultad 1	2222222213	E100007	Artemisa»Alquizar	Artemisa»Artemiza	17.5	31-03-2014																										

Tabla 29: HU crear rol a usuario.

Historia de Usuario	
Número: HU_18	Nombre Historia de Usuario: Crear rol a usuarios
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1 día
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 día
<p>Descripción:</p> <p>El usuario autenticado en el sistema con privilegios de administración accede a la opción: “Personas” ubicado en el panel superior del sistema. El sistema le muestra al usuario una interfaz donde este accede a la opción de “Permisos” y luego a la opción de “Roles”. El sistema le muestra al usuario una interfaz con los siguientes campos para introducir información:</p> <ul style="list-style-type: none"> • Añadir rol (campo de texto) <p>El usuario registra la información y presiona el botón: “Añadir rol”. El sistema verifica y almacena la información, finalizando así la HU.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Si el usuario no introduce la información y presiona el botón “Añadir rol”, el sistema emite un mensaje indicándole el error. 2. El sistema notifica la acción del usuario. 	
<p>Prototipo de interfaz:</p>  <p>El prototipo de interfaz muestra un campo de texto rectangular con un borde gris claro. A la derecha del campo de texto hay un botón redondeado con el texto "Añadir rol". Debajo de estos elementos, centrado, hay otro botón redondeado con el texto "Guardar pedido".</p>	

Tabla 30: HU eliminar rol a usuario.

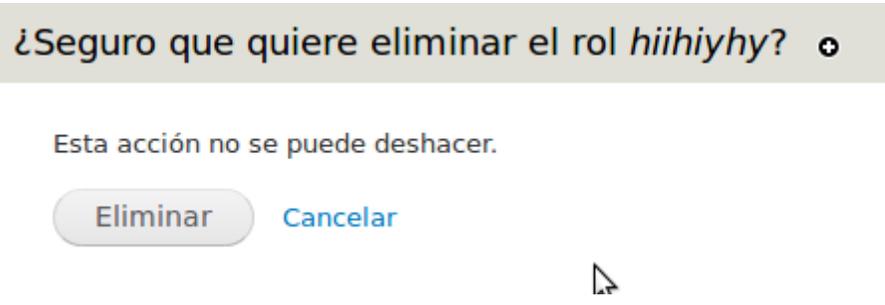
Historia de Usuario	
Número: HU_19	Nombre Historia de Usuario: Eliminar rol de usuario
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 días.
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 días.
<p>Descripción:</p> <p>El usuario autenticado en el sistema con privilegios de administración accede a la opción: “Personas” ubicado en el panel superior del sistema. El sistema le muestra al usuario una interfaz donde este accede a la opción de “Permisos” y luego a la opción de “Roles”. El sistema le muestra al usuario un listado de los roles existentes en el sistema, seguido de las opciones: “editar rol” y “editar permisos”. El usuario selecciona un rol y le da a la opción: “editar rol”. El sistema muestra un campo de texto seguido de las opciones:</p> <ul style="list-style-type: none"> • Eliminar rol. • Guardar rol. <p>El usuario selecciona la opción “Eliminar rol”, el sistema muestra un mensaje de confirmación de la acción seguido de las opciones:</p> <ul style="list-style-type: none"> • Eliminar. • Cancelar. <p>El usuario selecciona la opción “Cancelar”, el sistema deshace la operación y muestra el listado de roles existentes en el sistema.</p> <p>El usuario selecciona la opción “Eliminar”, el sistema elimina el rol, finalizando así la HU.</p>	
Observaciones:	
<p>Prototipo de interfaz:</p> 	

Tabla 31: HU Editar rol a usuario.

Historia de Usuario	
Número: HU_20	Nombre Historia de Usuario: Editar rol de usuario
Modificación de Historia de Usuario Número: 1	
Usuario: Sucel Daudinot Caricabeur	Iteración Asignada: 3
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 días.
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 días.
<p>Descripción:</p> <p>El usuario autenticado en el sistema con privilegios de administración accede a la opción: “Personas” ubicado en el panel superior del sistema. El sistema le muestra al usuario una interfaz donde este accede a la opción de “Permisos” y luego a la opción de “Roles”. El sistema le muestra al usuario un listado de los roles existentes en el sistema, seguido de las opciones: “editar rol” y “editar permisos”. El usuario selecciona un rol y le da a la opción: “editar rol”. El sistema un campo de texto seguido de las opciones:</p> <ul style="list-style-type: none"> • Eliminar rol. • Guardar rol. <p>El usuario selecciona la opción “Guardar rol”, el sistema valida la información y muestra el listado de roles existentes en el sistema, finalizando así la HU.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. El sistema emite un mensaje notificando el cambio del rol. 	
<p>Prototipo de interfaz:</p> <p>Nombre del rol *</p> <input type="text" value="hiihiyh"/> <p>El nombre de este rol. Ejemplos: «moderador», «grupo editorial», «arquitecto del sitio».</p> <p> <input type="button" value="Guardar rol"/> <input type="button" value="Eliminar rol"/> </p>	

Anexo 5: Tareas ingenieriles.

Tabla 32: Tarea de ingeniería 1. Autenticar con usuario y contraseña de LDAP.

Tarea de ingeniería	
Número de tarea: 1	Número de Historia de Usuario: HU_1
Nombre tarea: Autenticar usuario con usuario y contraseña de LDAP	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/04/2014	Fecha fin: 30/04/2014
Programador responsable: Sucel Daudinot Caricabeur	
Descripción: Se muestra una interfaz que permite entrar los datos de usuario (usuario y contraseña) para acceder a la aplicación.	

Tabla 33: Tarea de ingeniería 2. Buscar estudiante desde el directorio de LDAP.

Tarea de ingeniería	
Número de tarea: 2	Número de Historia de Usuario: HU_2
Nombre tarea: Buscar estudiante desde el directorio de LDAP	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/04/2014	Fecha fin: 30/04/2014
Programador responsable: Sucel Daudinot Caricabeur	
Descripción: Se muestra un cuadro de texto, donde se introduce el criterio de búsqueda, y luego se muestra el resultado de la misma	

Tabla 34: Tarea de ingeniería 3. Eliminar estudiante seleccionado de la base de datos del sistema.

Tarea de ingeniería	
Número de tarea: 3	Número de Historia de Usuario: HU_3
Nombre tarea: Eliminar estudiante seleccionado, de la base de datos del sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/04/2014	Fecha fin: 30/04/2014
Programador responsable: Sucel Daudinot Caricabeur	
Descripción: Se crean los contenidos y las vistas correspondientes a esta HU, permitiendo las operaciones de eliminar el usuario.	

Tabla 35: Tarea de ingeniería 4. Gestionar bonificación.

Tarea de ingeniería	
Número de tarea: 4	Número de Historias de Usuarios: HU_4, HU_5, HU_7, HU_9
Nombre tarea: Gestionar bonificación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/04/2014	Fecha fin: 30/04/2014
Programador responsable: Sucel Daudinot Caricabeur	
Descripción: Se crean los contenidos y las vistas correspondientes a esta HU, permitiendo las operaciones de eliminar, crear, editar, la bonificación.	

Tabla 36: Tarea de ingeniería 5. Imprimir modelo.

Tarea de ingeniería	
Número de tarea: 5	Número de Historias de Usuarios: HU_6
Nombre tarea: Imprimir modelo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/04/2014	Fecha fin: 30/04/2014
Programador responsable: Sucel Daudinot Caricabeur	
Descripción: Se crean los contenidos y las vistas correspondientes a esta HU, permitiendo las operaciones de imprimir.	

Tabla 37: Tarea de ingeniería 6. Gestionar presupuesto.

Tarea de ingeniería	
Número de tarea: 6	Número de Historias de Usuarios: HU_10, HU_11, HU_12, HU_13, HU_14
Nombre tarea: Gestionar presupuesto	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/05/2014	Fecha fin: 31/05/2014
Programador responsable: Sucel Daudinot Caricabeur	
Descripción: Se crean los contenidos y las vistas correspondientes a esta HU, permitiendo las operaciones de crear, eliminar, editar y consultar.	

Tabla 38: Tarea de la ingeniería 7. Generar reporte de la información especificada por el usuario.

Tarea de ingeniería	
Número de tarea: 7	Número de Historias de Usuarios: HU_15
Nombre tarea: Generar reporte de la información especificada por el usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/05/2014	Fecha fin: 31/05/2014
Programador responsable: Sucel Daudinot Caricabeur	
Descripción: Se crean los contenidos y las vistas correspondientes a esta HU, permitiendo las operaciones de consultar y crear los contenidos pertinentes.	

Tabla 39: Tarea de ingeniería 8. Configurar módulo para exportar a doc.

Tarea de ingeniería	
Número de tarea: 8	Número de Historias de Usuarios: HU_17
Nombre tarea: Configurar módulo para exportar a doc.	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/05/2014	Fecha fin: 31/05/2014
Programador responsable: Sucel Daudinot Caricabeur	
Descripción: Se configurará el módulo Exportar a XLS el cual permitirá al usuario salvar el contenido correspondiente a una vista determinada en formato doc.	

Tabla 40: Tarea de ingeniería 10. Configurar módulo para exportar a xls.

Tarea de ingeniería	
Número de tarea: 10	Número de Historias de Usuarios: HU_16
Nombre tarea: Configurar módulo para exportar a xls	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/05/2014	Fecha fin: 31/05/2014
Programador responsable: Sucel Daudinot Caricabeur	
Descripción: Se configurará el módulo Exportar a XLS el cual permitirá al usuario salvar el contenido correspondiente a una vista determinada en formato .xls.	

Tabla 41: Tarea de ingeniería 11. Consultar resultados de la búsqueda (bonificación).

Tarea de ingeniería	
Número de tarea: 11	Número de Historias de Usuarios: HU_8
Nombre tarea: Consultar resultados de la búsqueda (bonificación)	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/05/2014	Fecha fin: 31/05/2014
Programador responsable: Sucl Daudinot Caricabeur	
Descripción: Se muestra un cuadro de texto, donde se introduce el criterio de búsqueda, y luego se muestra el resultado de la misma	

Tabla 42: Tarea de ingeniería 9. Gestionar rol.

Tarea de ingeniería	
Número de tarea: 9	Número de Historias de Usuarios: HU_18, HU_19, HU_20
Nombre tarea: Gestionar rol	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha inicio: 1/05/2014	Fecha fin: 31/05/2014
Programador responsable: Sucl Daudinot Caricabeur	
Descripción: Se crean los contenidos y las vistas correspondientes a esta HU, permitiendo las operaciones de crear, editar, eliminar y consultar los usuarios, así como establecer los roles.	

Anexo 6: Diseño de casos de prueba.

Tabla 43: Caso de prueba buscar bonificación.

SC Buscar bonificación.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Buscar bonificación de forma correcta.	Se busca la información de manera correcta.	El sistema muestra la bonificación solicitada por el usuario.	El usuario autenticado en el sistema posee permiso de acceso al módulo "Gestionar bonificación" .
EC 1.2 Buscar bonificación de forma correcta sin resultados.	El sistema busca la bonificación y no encuentra información que cumpla con los criterios de búsqueda.	El sistema no muestra resultado.	1.-El sistema le muestra la interfaz correspondiente al módulo. 2.-El usuario introduce los datos correspondientes y realiza clic en el botón "Filtrar" .

Tabla 44: Caso de prueba eliminar bonificación.

SC Eliminar bonificación.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar bonificación de forma correcta.	El sistema elimina la bonificación seleccionada.	El sistema le muestra un mensaje al usuario de confirmación de la acción: "¿Seguro que quiere eliminar para Sucel Daudinot Caricabeur?". Si el usuario selecciona la opción: "Eliminar" . El sistema elimina la bonificación y muestra un mensaje de confirmación de la acción: "Se ha eliminado Bonificación para Sucel Daudinot Caricabeur".	El usuario se autentica en el sistema con privilegios de administración. 1.- El usuario accede al módulo "Gestionar bonificaciones" y el sistema le muestra un listado de las bonificaciones pendientes de cobro. 2.- El usuario selecciona una bonificación y realiza clic en la opción: "Registrar cobro" . 4.- El sistema permite eliminar la bonificación al seleccionar la opción: "Eliminar" .
EC 1.2 Cancelar operación.	El sistema cancela la operación.	El sistema le muestra un mensaje al usuario de confirmación de la acción: "¿Seguro que quiere eliminar para Sucel Daudinot Caricabeur?". Si el usuario selecciona la opción: "Cancelar" . El sistema deshace	

		la operación y muestra el listado de las bonificaciones sin cobrar existentes en el sistema.	
--	--	--	--

Tabla 45: Caso de prueba mostrar bonificación.

SC Mostrar bonificación.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar bonificación.	El sistema muestra bonificación.	El sistema muestra un listado de las bonificaciones existente en el sistema pendiente de cobro. Y las cobradas.	El usuario autenticado en el sistema con permiso de acceso al módulo: “Gestionar bonificaciones” , accede a este. 1.-El sistema le muestra al usuario un listado con las bonificaciones “Cobradas” y “Pendientes” de cobro existentes en el sistema.
EC 1.2 Mostar bonificación sin resultados.	El sistema no muestra bonificación.	El sistema no muestra un listado de las bonificaciones existente en el sistema Pendientes de cobro.	

Tabla 46: Caso de prueba buscar estudiante.

SC Buscar estudiante.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Buscar estudiante de forma correcta.	El sistema busca estudiantes de manera correcta.	El sistema le muestra al usuario la información del estudiante obtenido a través del directorio LDAP.	El usuario se autentica en el sistema con privilegios de administración. 1.-El usuario accede al módulo “Buscar estudiantes” , que se encuentra ubicado en
EC 1.2 Buscar estudiante de forma incorrecta.	El sistema no busca estudiante de manera incorrecta.	El sistema no muestra ningún resultado y muestra el mensaje de error “No existen resultados a mostrar.”	

		<p>El sistema no muestra ningún resultado y muestra el mensaje de error: "El usuario solo puede contener caracteres".</p> <p>"Solo se pueden buscar estudiantes".</p> <p>"No existen resultados a mostrar."</p>	<p>la parte lateral izquierda del sistema.</p> <p>2.-El sistema muestra una interfaz con campos para introducir la información referente a la búsqueda.</p>
		<p>El sistema no muestra ningún resultado y muestra el mensaje de error: "El solapín insertado no corresponde a un estudiante o está incorrecto."</p> <p>"Solo se puede buscar estudiantes."</p> <p>"No existen resultados a mostrar."</p>	<p>3.- El usuario introduce el usuario o el solapín del estudiante y presiona el botón: "Buscar"</p>
		<p>El sistema no muestra ningún resultado y muestra el mensaje de error: "El usuario solo puede contener caracteres"</p> <p>"El solapín insertado no corresponde a un estudiante o está incorrecto."</p> <p>"Solo se puede buscar estudiantes."</p> <p>"No existen resultados a mostrar."</p>	

Tabla 47: Caso de prueba eliminar estudiante.

SC Eliminar estudiante.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar estudiante de	El sistema elimina el estudiante	El sistema le muestra un mensaje al usuario de	El usuario se autentica en el sistema con privilegios de

forma correcta.	seleccionado.	confirmación de la acción: "¿Seguro que quiere eliminar Sucel Daudinot Caricabeur?". Si el usuario selecciona la opción: "Eliminar" . El sistema elimina el estudiante seleccionado y muestra un mensaje de confirmación de la acción: "Se ha eliminado Estudiante Sucel Daudinot Caricabeur".	administración. 1.- El usuario accede al módulo "Buscar estudiantes" y el sistema le muestra una interfaz con un listado de los estudiantes existente en el sistema. 2.- El usuario selecciona un estudiante y realiza clic en la opción: "Eliminar" . 4.- El sistema elimina al estudiante.
EC 1.2 Cancelar operación.	El sistema cancela la operación.	El sistema no elimina el estudiante y le muestra al usuario un mensaje de confirmación de la acción: "¿Seguro que quiere eliminar Sucel Daudinot Caricabeur?". Si el usuario selecciona la opción: "Cancelar" . El sistema deshace la operación y muestra el listado de los estudiantes existentes en el sistema.	

Tabla 48: Caso de prueba agregar presupuesto.

SC Agregar presupuesto.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Agregar presupuesto de forma correcta.	El sistema agrega presupuesto de forma correcta	El sistema almacena la información y muestra un mensaje de confirmación de la acción: "Presupuesto 2014 se ha creado."	El usuario autenticado en el sistema con privilegios de administración accede al módulo " Gestionar presupuesto ", que se encuentra ubicado en la parte lateral izquierda del sistema.
EC 1.2 Agregar presupuesto de forma incorrecta.	El sistema agrega presupuesto de forma incorrecta	El sistema no agrega el presupuesto y emite el siguiente mensaje de error: "El campo Nombre es obligatorio". "El campo Presupuesto asignado es obligatorio". "Hace falta una fecha válida para Fecha fin". "Hace falta una fecha válida para Fecha de inicio"	1.-. El sistema le muestra al usuario una interfaz con un listado de los presupuestos existentes en el sistema. 2.-El usuario realiza un clic en la opción " Agregar presupuesto ". 3.-El sistema muestra un formulario para que el usuario introduzca la información. 4.- El usuario introduce la información y presiona el botón: " Guardar ".
		El sistema no agrega el presupuesto y emite el siguiente mensaje de error: "Solo se permiten números en Presupuesto asignado".	
		El sistema no agrega el presupuesto y emite el siguiente mensaje de error:	
		El sistema no agrega el presupuesto y emite el siguiente mensaje de error: "El valor de la fecha no coincide con el formato esperado".	
		El sistema no agrega el presupuesto y emite el siguiente mensaje de error: "El valor de la fecha no	

		coincide con el formato esperado".	
--	--	------------------------------------	--

Tabla 49: Caso de prueba buscar presupuesto.

SC Buscar presupuesto.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Buscar presupuesto de forma correcta.	El sistema busca el presupuesto de forma correcta.	El sistema muestra la información solicitada por el usuario existente en el sistema.	El usuario autenticado en el sistema con privilegios de administración accede a la opción: “Gestionar presupuesto” .
EC 1.2 Buscar bonificación de forma correcta sin resultados.	El sistema busca presupuesto y no encuentra información que cumpla con los criterios de búsqueda.	El sistema no muestra resultado.	1.-El sistema muestra una interfaz para introducir la información referente a la búsqueda de presupuesto. 2.- El usuario introduce los datos y presiona el botón: “Filtrar” .
EC 1.3 Buscar presupuesto de forma incorrecta.	El sistema no busca el presupuesto de forma incorrecta.	El sistema no muestra la información solicitada por el usuario y muestra mensaje de error: "La fecha de inicio no debe ser mayor que la fecha fin"	

Tabla 50: Caso de prueba editar presupuesto.

SC Editar presupuesto.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Editar presupuesto de forma correcta.	El sistema edita presupuesto de forma correcta	El sistema almacena la información y muestra un mensaje de confirmación de la acción: "Presupuesto 2014 se ha creado."	El usuario autenticado en el sistema con privilegios de administración accede al módulo "Gestionar presupuesto" , que se encuentra ubicado en la parte lateral izquierda del sistema.
EC 1.2 Editar presupuesto de forma incorrecta.	El sistema edita presupuesto de forma incorrecta	El sistema no agrega el presupuesto y emite el siguiente mensaje de error: "El campo Nombre es obligatorio". "El campo Presupuesto asignado es obligatorio". "Hace falta una fecha válida para Fecha fin". "Hace falta una fecha válida para Fecha de inicio".	1.- El sistema le muestra al usuario una interfaz con un listado de los presupuestos existentes en el sistema, seguido de las opciones: "editar" y "eliminar" . 3.- El usuario accede la opción: "editar" y el sistema muestra un formulario con los siguientes campos: <ul style="list-style-type: none"> • Nombre (obligatorio, campo de texto) • Presupuesto asignado (obligatorio, campo de texto) • Fecha de inicio (date). • Fecha fin (date).
		El sistema no agrega el presupuesto y emite el siguiente mensaje de error: "Solo se permiten números en Presupuesto asignado".	
		El sistema no agrega el presupuesto y emite el siguiente mensaje de error: "Fecha de inicio no debe ser mayor que fecha fin"	
		El sistema no agrega el presupuesto y emite el siguiente mensaje de error: "El valor de la fecha no coincide con el formato esperado".	

		El sistema no agrega el presupuesto y emite el siguiente mensaje de error: "El valor de la fecha no coincide con el formato esperado".	
--	--	---	--

Tabla 51: Caso de prueba eliminar presupuesto.

SC Eliminar presupuesto.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar presupuesto de forma correcta.	El sistema elimina el presupuesto seleccionado.	El sistema le muestra un mensaje al usuario de confirmación de la acción: "¿Seguro que quiere eliminar presupuesto 2014?". El usuario selecciona la opción: "Eliminar". El sistema elimina el presupuesto y muestra un mensaje de confirmación de la acción: "Se ha eliminado presupuesto2014."	El usuario autenticado en el sistema posee privilegios de administración. 1.- El usuario accede al módulo " Gestionar presupuesto ". 2.- El sistema le muestra una interfaz con un listado de los presupuestos existente en el sistema. 3.- El usuario selecciona un presupuesto y realiza clic en la opción: " Eliminar ". 4.- El sistema permite eliminar el presupuesto.
EC 1.2 Cancelar operación.	El sistema cancela la operación.	El sistema no elimina el presupuesto y le muestra al usuario un mensaje de confirmación de la acción: "¿Seguro que quiere eliminar presupuesto 2014?". El usuario selecciona la opción: "Cancelar". El sistema deshace la operación y muestra el listado de los presupuestos existentes en el sistema.	

Tabla 52: Caso de prueba mostrar presupuesto.

SC Mostrar presupuesto.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar presupuesto.	El sistema muestra presupuesto.	El sistema muestra un listado de los presupuestos existente en el sistema.	El usuario autenticado en el sistema con privilegios de administración accede al módulo: " Gestionar presupuesto ", que se encuentra ubicado en la parte lateral izquierda del sistema. 1.- El sistema le muestra al usuario un listado con los presupuestos existentes en el sistema.

Tabla 53: Caso de prueba autenticar usuario.

SC Autenticar usuario.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Autenticar usuarios de forma correcta.	El usuario se autentica de manera correcta.	El sistema verifica las credenciales del usuario, luego autentica al usuario y le permite el acceso a las funcionalidades del sistema, según su rol.	El usuario accede a la página principal del sistema. 1.-El sistema habilita un formulario para introducir la información.
EC 1.2 Autenticar usuarios de forma incorrecta.	El usuario no se autentica de manera correcta.	El sistema verifica las credenciales del usuario, luego no autentica al usuario y emite el siguiente mensaje de error: "Lo sentimos. No reconocemos el nombre de usuario o la contraseña. ¿Olvidó su contraseña?".	2.-El usuario introduce la información y presiona el botón: " Iniciar Sesión ".
EC 1.3 Autenticar usuario dejando campos vacíos.	El sistema autentica al usuario dejando los campos obligatorios vacíos.	El sistema verifica las credenciales del usuario, luego no autentica al usuario y emite los siguientes mensajes de error: "El campo Nombre de usuario es obligatorio." "Lo sentimos. No reconocemos el nombre de usuario o la contraseña."	

		<p>El sistema verifica las credenciales del usuario, luego no autentica al usuario y emite los siguientes mensajes de error:</p> <p>"El campo Contraseña es obligatorio".</p> <p>"Lo sentimos. No reconocemos el nombre de usuario o la contraseña."</p>	
		<p>El sistema verifica las credenciales del usuario, luego no autentica al usuario y emite los siguientes mensajes de error:</p> <p>"El campo Nombre de usuario es obligatorio".</p> <p>"El campo Contraseña es obligatorio".</p>	

Tabla 54: Caso de prueba exportar reportes a Word.

SC Exportar reportes a Word.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Exportar reportes a Word.	El sistema descarga a formato .doc los reportes del sistema.	El sistema construye el documento .doc y le permite al usuario exportar el documento.	<p>El usuario se autentica en el sistema con privilegios de administración.</p> <p>1.- El usuario accede al módulo de "Reportes".</p> <p>2.- El sistema muestra una interfaz con las bonificaciones cobradas existente en el sistema.</p> <p>3.- El usuario selecciona la opción: "Exportar a documento de texto", ubicada en el lateral derecho después del contenido de la vista.</p> <p>3.- El sistema exporta la vista en formato .doc.</p>

Tabla 55: Caso de prueba generar reportes.

SC Generar reportes.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Generar reportes de forma correcta.	El sistema genera reportes de manera correcta.	El sistema genera la información de las bonificaciones cobradas dependiendo del criterio de búsqueda.	<p>El usuario autenticado en el sistema con privilegios de administración accede al módulo: “Reportes”, que se encuentra ubicado en la parte lateral izquierda del sistema.</p> <p>1.- El sistema le muestra al usuario una interfaz con los siguientes campos:</p> <ul style="list-style-type: none"> • Facultad • Fecha de inicio. • Fecha final • Elementos por página. <p>2.- El usuario registra la información y presiona el botón: “Filtrar”.</p>
EC 1.2 Generar reportes de forma correcta sin resultados.	El sistema no encuentra información que cumpla con los criterios de búsqueda	El sistema no muestra resultado.	
EC 1.3 Generar reportes de forma incorrecta.	El sistema no genera reportes de manera incorrecta.	El sistema no genera la información y emite el siguiente mensaje de error: "La fecha de regreso no puede ser menor que la ida".	

Tabla 56: Caso de prueba generar reportes a Excel.

SC Exportar reportes a Excel.			
Escenario	Descripción	Respuesta del Sistema	Flujo Central
EC 1.1 Exportar reportes a Excel	El sistema descarga a formato .xls los reportes del sistema.	El sistema construye el documento .xls y le permite al usuario exportar el documento.	<p>El usuario se autentica en el sistema con privilegios de administración.</p> <p>1.- El usuario accede al módulo de "Reportes".</p> <p>2.- El sistema muestra una interfaz con las bonificaciones cobradas existente en el sistema.</p>

			<p>3.- El usuario selecciona la opción: "Exportar a hoja de cálculo", ubicada en el lateral derecho después del contenido de la vista.</p> <p>3.- El sistema exporta la vista en formato .xls.</p>
--	--	--	--