

Universidad de las Ciencias Informáticas



Tema: Implementación de la Intranet de la Facultad 1, Módulo de Becado.

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

Autores: Jorge Félix Duque Cortina

Dannys Cristobal Jiménez.

Tutores: Ing. Hiraldo Manuel Anaya González.

15 de Junio 2009.
Año del 50 Aniversario del Triunfo de la Revolución



El ignorante afirma, el sabio duda y reflexiona.

Aristóteles.

Declaración de autoría.

Declaramos que somos los únicos autores del trabajo titulado:

Implementación de la Intranet de la Facultad 1, Módulo de Becado.

Y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Firma del Autor:
Dannys Cristobal Jiménez

Firma del autor:
Jorge Félix Duque Cortina

Firma del Tutor:
Hirald M. Anaya González

Agradecimientos

Queremos agradecer:

A nuestra Revolución y a Fidel.

Al tutor Hiraldo Anaya Rodríguez que con tanto esfuerzo y dedicación supo guiarnos en el desarrollo de este trabajo.

Jorge:

Agradecer a mis tíos, a mi tía Miriam, a mi tío Jesús, a mi tío y padrino Ignacio, a mis tíos Rosa María, Natacha, nene, zurro, Daniel, a mi tío chucho también muchas gracias, a todos que son mis guías que me han visto siempre más que como un sobrino, como a un hijo de ellos gracias educadores de mi vida. A mi hermano Orson por dejarme ser siempre su guía y haberle trazado un camino. A mi madrina querida Yenía y a su esposo Arley. Agradecer especialmente a mis abuelos ARGELIA CAMOT MENDOZA y FRANCISCO CORTINA ANGUVEIRA que para mí son lo más grande que tengo. También a mi abuelo Cecilio, mi bisabuelo Félix duque y mi bisabuela tita. Por último quiero agradecer a mis padres:

Mami y papi: Gracias primero que todo por traerme al mundo y darme todo el amor que pueden, deben saber que los quiero mucho y no sé qué haría si no estuvieran en mis días, les agradezco siempre estar a la altura de lo mejor de este mundo para que yo viva orgulloso de Uds. espero que se sientan orgullosos de tener a un hijo bien preparado y que puedan hacer elogios de lo buena persona que me han enseñado a ser, saben que los amo y que velaré siempre por la felicidad de Uds.

Danny:

A mis padres que confiaron en mí, quienes me han educado y apoyado en todo lo correcto ensañándome a esforzarme y no amedrentarme ante las dificultades. Quienes dieron con amor todo cuanto pudieron para que lograra este éxito que hoy es de nosotros. A mis hermanos que siempre me han apoyado y querido pese a todas las dificultades y la distancia.

A mis tías Tata y Noris y a mis tíos Miguel y Tino que siempre he sido para ellos como un hijo.

A mi primo Abe y a su Maritza.

A los Amigos de la secundaria y el pre, la gente del barrio.

A mis Amigos que han compartido conmigo, que me han ayudado y han estado apoyándome en todo momento. A Eduardo y Mariliennys, Llorete y Yeni, Daríel, Ricardo, Annie, Lorenzo, Tony, Aliesky, Dayron y Osmery. A los que ya no están; Diasta, Friky, Frankiel, Norlis y a muchos más.

A todos los profesores que de una forma u otra supieron enseñarme todo lo que sé.

Le agradezco a la vida por haberme dado tanto.

Dedicatoria

Quiero dedicar este Trabajo de Diploma a mis padres, Angela y Adolfo por guiarme por el buen camino, gracias a ellos he obtenido lo logrado; y a mis hermanos que los quiero con el alma...

Dannys Cristobal Jiménez.

Quiero dedicar este Trabajo de Diploma a todos los profesores que me impartieron clases a lo largo de mi carrera, a mis amigos del piquete kaspersky, a mis primos, a mis tíos, a mi hermano y en especial a mis padres.

Jorge Félix Duque Cortina.

Resumen.

Uno de los principales objetivos de la Facultad 1 es la producción de software, algunos de estos destinados a la gestión y control de todos los procesos que en ella se realizan. El proyecto Intranet es el que da soporte para el cumplimiento de dichas tareas, dividiéndolas en módulos para lograr una mayor organización.

A partir de la necesidad de que en la Residencia Estudiantil de la Facultad 1 el control de las actividades y los datos del estudiante se realizan de forma manual, lo que implica demora y desorganización en la actualización de toda la información referente, se creó el Módulo de Becado con el principal objetivo de agilizar todo el proceso de gestión, y así, dar solución a los problemas planteados anteriormente. Este proceso va a estar definido por las funcionalidades Gestionar datos del estudiante, Controlar cuartelería, Controlar guardia, Controlar TSU, Controlar limpieza, Generar reporte de estudiante y Realizar evaluación mensual.

Con la implementación del módulo, el control de las actividades y la gestión de la información del estudiante es mucho más rápida, todo este proceso se torna mucho más agradable para quien lo utiliza, que por supuesto, le ahorra en tiempo y en recursos.

INDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1 INTRODUCCIÓN.	5
1.2 CONCEPTOS BÁSICOS ASOCIADOS CON EL PROBLEMA PLANTEADO.	5
1.3 LAS ARQUITECTURAS DE LAS INTRANETS.	6
1.3.2 La Intranet de la Facultad 1.....	7
1.3.3 Módulos. Concepto y características.....	8
1.4 ESTADO DEL ARTE.	9
1.4.1 Análisis de sistemas existentes.....	9
1.4.2 Deficiencias de los sistemas estudiados.	13
1.5 TENDENCIAS Y TECNOLOGÍAS ACTUALES.....	13
1.5.1 Metodologías de desarrollo.	13
1.5.2 Lenguajes de programación web.	17
1.5.3 Sistemas Gestores de Bases de datos.....	23
1.5.4 Servidores Web.	28
1.5.5 El uso de los CMS.....	30
1.5.6 Zend Studio como IDE a utilizar.....	34
1.5.7 Visual Paradigm como Herramienta CASE.	34
1.6 CONCLUSIONES DEL CAPÍTULO.	35
CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.	36
2.1 INTRODUCCIÓN.....	36
PRINCIPALES FUNCIONALIDADES A AUTOMATIZAR:	36
2.2 ESTUDIO CRÍTICO Y VALORATIVO DE LA TÉCNICA DE PROGRAMACIÓN, PLATAFORMA Y LIBRERÍAS USADAS PARA EL DESARROLLO DEL MÓDULO DE BECADO.	40
2.2.1 Estudio de los Módulos de Drupal.....	40
2.2.2 Técnicas de programación y creación de módulos con Drupal.....	42
2.2.3 Estándares de codificación.	45
2.2.4 Valoración crítica del diseño propuesto por el analista.	46
2.2.5 Análisis de posibles componentes, módulos e implementaciones ya existentes que puedan ser rehusados. Estrategias de integración.	46
2.2.6 Descripción de los algoritmos no triviales a implementar. Análisis de complejidad de los mismos.	47
2.3 DESCRIPCIÓN DE LAS HISTORIAS DE USUARIOS QUE SE UTILIZAN PARA REPRESENTAR COMPUTACIONALMENTE DICHAS ESTRUCTURAS. DIAGRAMA DE COMPONENTES.	52

2.3.1 Personas relacionadas con el sistema.	52
2.3.2 Historias de usuario del paquete Gestión de Información.....	52
2.3.3 Planificación de Sprint.....	57
2.3.4 Tareas asignadas a cada iteración.....	59
2.4.1 Diagrama de Componentes.	64
2.4.2 Diagrama de Despliegue.....	64
2.5 BENEFICIOS DE LA IMPLEMENTACIÓN DEL MÓDULO DE BECADO.....	65
2.6 CONCLUSIONES DEL CAPÍTULO.....	65
CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	66
3.1 INTRODUCCIÓN.....	66
3.2 PRUEBAS. MÉTODOS DE PRUEBAS.	66
3.2.1 Fases de prueba:	66
3.3 CONCLUSIONES DEL CAPÍTULO.....	74
CONCLUSIONES	75
RECOMENDACIONES.....	76
BIBLIOGRAFÍA REFERENCIADA.....	77
BIBLIOGRAFÍA	79

INDICE DE TABLAS

TABLA 2.1 REQUISITOS DEL SISTEMA.	40
TABLA 2.2 PERSONAS RELACIONADAS AL SISTEMA.	52
TABLA 2.3 HU GESTIONAR INFORMACIÓN DEL ESTUDIANTE.	53
TABLA 2.4 HU CONTROLAR CUARTELERÍA.	54
TABLA 2.5 HU CONTROLAR GUARDIA ESTUDIANTIL.	55
TABLA 2.6 HU CONTROLAR TSU.	55
TABLA 2.7 HU CONTROLAR LIMPIEZA DEL APTO.	56
TABLA 2.8 HU GENERAR REPORTE DE ESTUDIANTE.	57
TABLA 2.9 HU REGISTRAR EVALUACIÓN MENSUAL.	57
TABLA 2.10 PLAN DE DURACIÓN DE LOS SPRINT.	59
TABLA 2.11 DESARROLLAR EL FORMULARIO PARA LA INSERCIÓN DE ESTUDIANTES.	59
TABLA 2.12 DESARROLLAR EL FORMULARIO PARA MODIFICAR LOS DATOS DEL ESTUDIANTE.	60
TABLA 2.13 IMPLEMENTAR LA FUNCIONALIDAD PARA ELIMINAR ESTUDIANTE.	60
TABLA 2.14 IMPLEMENTAR LA FUNCIONALIDAD PARA INSERTAR CUARTELERÍA.	60
TABLA 2.15 IMPLEMENTAR LA FUNCIONALIDAD PARA CAMBIAR CUARTELERÍA.	61
TABLA 2.16 IMPLEMENTAR LA FUNCIONALIDAD PARA CORRER CUARTELERÍA.	61
TABLA 2.17 IMPLEMENTAR LA FUNCIONALIDAD PARA INSERTAR GRUPO A LA GUARDIA.	62
TABLA 2.18 IMPLEMENTAR LA FUNCIONALIDAD PARA INSERTAR TSU A UN GRUPO DOCENTE.	62
TABLA 2.19 IMPLEMENTAR LA FUNCIONALIDAD PARA INSERTAR ESTUDIANTE QUE ESTÁ DE LIMPIEZA.	62
TABLA 2.20 IMPLEMENTAR LA FUNCIONALIDAD PARA ELIMINAR ESTUDIANTE.	63
TABLA 2.21 IMPLEMENTAR LA FUNCIONALIDAD PARA INSERTAR EVALUACIÓN.	63
TABLA 2.22 IMPLEMENTAR LA FUNCIONALIDAD PARA INSERTAR EVALUACIÓN.	63
TABLA 3.1 PRUEBA PARA LA FUNCIONALIDAD INSERTAR ESTUDIANTE.	68
TABLA 3.2 PRUEBA PARA LA FUNCIONALIDAD MODIFICAR ESTUDIANTE.	68

TABLA 3.3 PRUEBA PARA LA FUNCIONALIDAD ELIMINAR ESTUDIANTE.	69
TABLA 3.4 PRUEBA PARA LA FUNCIONALIDAD INSERTAR CUARTELERO.	69
TABLA 3.5 PRUEBA PARA LA FUNCIONALIDAD CAMBIAR CUARTELERÍA.	70
TABLA 3.6 PRUEBA PARA LA FUNCIONALIDAD CORRER CUARTELERÍA.	71
TABLA 3.7 PRUEBA PARA LA FUNCIONALIDAD INSERTAR GUARDIA.	71
TABLA 3.8 PRUEBA PARA LA FUNCIONALIDAD INSERTAR TSU.	72
TABLA 3.9 PRUEBA PARA LA FUNCIONALIDAD INSERTAR LIMPIEZA.	72
TABLA 3.10 PRUEBA PARA LA FUNCIONALIDAD GENERAR REPORTE.	73
TABLA 3.11 PRUEBA PARA LA FUNCIONALIDAD INSERTAR EVALUACIÓN MENSUAL.	73
TABLA 3.12 PRUEBA PARA LA FUNCIONALIDAD MOSTRAR EVALUACIÓN MENSUAL.	74

Introducción

El uso irremplazable de las Tecnologías de la Informática y las Comunicaciones (TIC) se está haciendo más evidente en las últimas décadas. El desarrollo de las mismas permite que los procesos de la humanidad se realicen con más calidad, control y velocidad, siendo posible acortar innumerables situaciones complicadas en la sociedad y aumentar la fiabilidad y efectividad de los bienes y servicios que brindan en la actualidad el uso de las TIC.

La Universidad de las Ciencias Informáticas (UCI), se ha convertido en el principal pilar de la industria cubana del software y ha hecho grandes aportes en la revolución informática, la cual está llamada a convertirse en una gran fuente de ingreso para el país.

Con el fin de mejorar en rapidez y eficiencia en la obtención de la información de los alumnos, para la realización de las evaluaciones de los mismos, se creó el Módulo de Becado de la Intranet de la Facultad 1, con el objetivo de controlar la información de los estudiantes y los datos de las actividades en la Residencia Estudiantil.

Inicialmente se logró un análisis y diseño de la situación actual con la cual en una etapa final se busca realizar un trabajo que cumpla con los requerimientos que los trabajadores necesitan para hacer con calidad el trabajo.

Situación Problemática:

La Residencia Estudiantil es uno de los renglones más importantes de la Universidad en cuanto al proceso formativo educacional y conlleva a que se realicen un conjunto de actividades con los estudiantes que necesitan ser gestionadas.

El proceso de gestión de las actividades y evaluaciones de los estudiantes se lleva de forma manual, de manera que toma mucho tiempo organizar y actualizar la información de dichos alumnos y sus actividades.

Problema Científico:

¿Cómo lograr un sistema automatizado que permita en el menor tiempo y costo posible gestionar los procesos que conciernen a las actividades del becado de la Residencia Estudiantil en la Facultad 1?

Objetivo General:

Realizar la implementación del Módulo de Becado para la Intranet de la Facultad 1, mejorando así la gestión de los procesos de la Residencia de la misma.

Objetivos Específicos:

- Revisar y corregir el análisis y diseño del Módulo.
- Realizar el proceso de implementación del Módulo.

Objeto de Estudio:

Lo constituye la implementación de todos los procesos relacionados con la gestión de las actividades del becado en la Residencia de la Facultad 1 de la UCI.

Campo de Acción:

Son los procesos relacionados con la gestión de las actividades del becado en la Residencia de la Facultad 1 de la UCI.

Idea a Defender:

Si se implementa el Módulo de Becado de la Facultad 1 entonces se podrá gestionar las actividades del becado de forma más eficiente.

Para ello se proponen las siguientes **tareas** para cumplir los **objetivos**:

- Realizar la gestión de las actividades y evaluaciones en la Residencia.

- Analizar otros sistemas con características similares al sistema a realizar.
- Aplicar el uso de la metodología, herramientas y lenguaje del modelado que se utilizarán para el desarrollo de la modelación y especificación del Módulo.
- Realizar la implementación de un Módulo para la gestión de actividades y la evaluación del estudiante.
- Analizar la integración con otros componentes del sistema.
- PSP. Analizar productividad.
- Realizar pruebas.

Métodos Teóricos

- Método Hipotético Deductivo.
- Método Histórico Lógico.

El **Método Hipotético Deductivo** permitió establecer conclusiones y obtener una idea del funcionamiento del sistema partiendo de la información acumulada como conocimiento y la teoría existente sobre el tema.

El **Método Histórico Lógico** permitió realizar el estudio a nivel nacional e internacional de sistemas similares al que se quiere implementar, así como los procesos de gestión de información de las actividades de los becados en la Residencia de la Facultad 1.

Métodos Empíricos

- Análisis de Documentos.
- La entrevista.

El **Método Análisis de Documentos** posibilitó realizar un estudio de los modelos manejados por los trabajadores de la Residencia de la Facultad 1.

El contenido del presente trabajo se estructura en tres capítulos:

Capítulo1: En este capítulo se explican conceptos fundamentales que ayudan a comprender mejor el trabajo. Se hace un estudio de los sistemas similares nacional e internacional de lo que se quiere implementar, así como, de las características de las principales herramientas, tecnologías y metodología a utilizar en el desarrollo del Módulo.

Capítulo2: En este capítulo se realiza la crítica del análisis y diseño, y se generan las HU detallándose cada una de las tareas para su solución. Son descritos los algoritmos utilizados en su implementación y la complejidad de los mismos.

Capítulo3: En este capítulo se realiza la validación del sistema mediante las pruebas unitarias y de aceptación. Se genera la documentación respectiva de las mismas dándose una pequeña explicación de sus características y se evalúan los resultados finales de las pruebas y los resultados obtenidos.

Posibles resultados:

Obtener como resultado de la implementación del Módulo de Becado un software que permita la gestión de las actividades y la evaluación de los becados de la Residencia Estudiantil de la Facultad 1 en la UCI.

Capítulo 1. Fundamentación Teórica

1.1 Introducción.

Al iniciarse el desarrollo de un producto de software los primeros pasos que se dan son en función de definir el tipo de aplicación y su ambiente de desarrollo. El presente capítulo aborda la fundamentación teórica del desarrollo del Módulo de Becado para la Intranet de la Facultad 1, define el marco teórico del tema, el estudio de sistemas similares existentes nacional e internacional vinculados al campo de acción, así como, las tendencias, tecnologías, metodologías y herramientas actuales, realizando una selección de aquellas que serán utilizadas durante el desarrollo del proyecto.

1.2 Conceptos básicos asociados con el problema planteado.

El punto de partida de cualquier conversación, debe ser el establecer el conjunto de términos comunes que serán utilizados en la discusión. Quizás la mayor fuente de discrepancias, se debe a que los términos usados, son entendidos de distintas maneras por los participantes. Si bien inicialmente se propone una definición en función de la experiencia de los autores, se han buscado otras fuentes que den soporte a la acepción dada.

En la actualidad en todas las partes del mundo, en cada lugar donde exista una comunidad informatizada o una empresa corporativa en donde se maneje gran cantidad de información y se brinden diferentes servicios se trata por todos los medios crear una Intranet.

Una Intranet es una red de ordenadores privados que utiliza tecnología Internet para compartir de forma segura cualquier información o programa del sistema operativo, para que todos los usuarios de la organización con acceso los puedan utilizar. O sea es la combinación de Internet con una red de área local. Este nombre a veces solo hace referencia a alguna web interna de alguna institución, pero muchas veces es un componente importante de la comunicación y la colaboración dentro de una organización y es más bien una parte extensa de la infraestructura de los ordenadores que la componen.

Marco Teórico. Conceptos.

1.3 Las arquitecturas de las Intranets.

Arquitectura de parte del cliente y arquitectura de parte del servidor. El software cliente puede ser cualquier computadora local (servidor web), mientras que el software servidor se ejecuta en una Intranet anfitriona. No es necesario que estos dos software, el cliente y el servidor, sean ejecutados en el mismo sistema operativo.

Componentes básicos de la intranet:

Los componentes de las intranets se tomaron prestados inicialmente de las redes tradicionales y de Internet. Según avanza esta área de los servicios de información se desarrollan productos específicos para el entorno de la intranet. Las partes de una intranet pueden dividirse en cuatro grandes áreas:

➤ TCP/IP.

Se trata del protocolo principal de una intranet, de la misma forma que es el protocolo principal para Internet. En una intranet, TCP/IP no tiene por qué ser el único protocolo, pero debe estar totalmente activo y en funcionamiento para cada cliente y servidor de la red, aprovechando los beneficios propios de la utilización de este protocolo

➤ Servicios de información.

Los servicios de información forman el corazón de una intranet. Un servicio de información es cualquier paquete software o aplicación que pueda recibir, almacenar y enviar información a o desde uno o más clientes, en otras palabras, cualquier aplicación que permita la interacción con datos o personas mediante una intranet.

➤ **Clientes.**

Los clientes son las herramientas software necesarias para acceder al mundo de la información que está disponible mediante servidores de información. El cliente más habitual en una intranet siempre será el navegador Web.

➤ **Herramientas de autor.**

El último componente de una intranet son las herramientas de autor, que son aplicaciones y utilidades usadas para crear los datos manejados por el servidor de un servicio de información. Una herramienta de autor crea, edita, altera o manipula los datos, pudiendo variar sus capacidades, desde las bases de cortar y pegar hasta aplicaciones punteras de última tecnología, totalmente interactivas y con capacidades multimedia.

1.3.2 La Intranet de la Facultad 1.

Primeramente hemos hecho alusión a qué es una Intranet, qué beneficios trae consigo y por qué es necesario que se implemente en una comunidad informatizada donde se manejen cantidades grandes de información, para que se entienda la necesidad de enriquecerla y de implementar módulos que faciliten sus servicios y particularidades.

En la Universidad de las Ciencias de la Informáticas (UCI) existe una intranet en la cual están conectadas grandes cantidades de operadores y en la que diariamente navegan usuarios, ya sean trabajadores, profesores o estudiantes. Dentro de la cual, las facultades en que está dividida la Universidad, trabajan en la producción de software de todo tipo y en el enriquecimiento de la Intranet interna de cada facultad.

En la Facultad 1 de la UCI, se desea conformar una Intranet segura y eficiente que permita brindar diferentes servicios y gestionar diferentes actividades pertinentes a la Facultad 1.

Se ha creado un grupo de desarrollo en el cual se trabaja en módulos esenciales para la gestión de actividades de la Facultad 1, algunos ejemplos de estos módulos son:

- Módulo de Producción.
- Módulo de Docencia.

En esta investigación se quiere culminar con la implementación de un módulo que forme parte de la Intranet de la Facultad 1 y que permita la gestión de la información pertinente a las actividades del becado en la Residencia y la cual da nombre a este trabajo de diploma:

- Implementación de la Intranet de la Facultad 1, Módulo de Becado.

1.3.3 Módulos. Concepto y características.

Concepto.

Un módulo es una parte de un programa de ordenador que cumple varias tareas y permite extender la funcionalidad de alguna web. Un componente auto-controlado de un sistema, el cual posee una interfaz bien definida hacia otros componentes. Es un software que une un conjunto de subprogramas y estructuras de datos. Los módulos de contenido son pequeñas secciones que agrupan información. Algo es modular si es construido de manera tal que se facilite su ensamblaje, acomodamiento flexible y reparación de sus componentes.

Características de un Módulo.

Cada uno de los módulos de un programa idealmente debería cumplir las siguientes características:

- **Tamaño pequeño:** Permite que al ocurrir un cambio en el programa, ya sea para corregir un error o para rediseñar un algoritmo bloquear el efecto que pueda tener en el sistema.
- **Independencia modular:** ES la característica que permite trabajar con un módulo sin tener que conocer detalles internos de otros módulos. Mientras más independientes sean los módulos entre sí más fácil es el trabajo con ellos. Como consecuencia de la independencia modular un módulo cumplirá:

- Características de caja negra, es decir **abstracción**.
- Aislamiento de los detalles mediante **encapsulamiento**.

Estas dos características hablando de programación orientada a objetos.

1.4 Estado del arte.

1.4.1 Análisis de sistemas existentes.

Sistemas internacionales.

Internacionalmente existen sistemas que gestionan la información de los internos de algunos centros educacionales.

- **DocCF:** Es un sistema que permite la gestión escolar, es construido por la empresa colombiana **Grupo CF Developer** dedicada a desarrollar productos y servicios informáticos al sector educativo. La gestión del sistema es por módulo permitiendo gestionar sin limitaciones información de alumnos, docentes, salones, grupos, horarios, matrículas, biblioteca, calificaciones, boletines escolares, cobros, pagos, inventario escolar, gestión de docencia, ausencias y envío de email.
- **El Módulo de Gestión de Alumnos y Matrículas:** Permite el registro de información general del alumno, información académica, control de la disciplina, control del pago, informes de matrículas y control de ausencias.
- **CALMECAC:** Es un software diseñado para la administración de información básica escolar, aplicable a escuelas primarias, secundarias y bachillerato (posiblemente hasta nivel profesional), tanto públicas como privadas. Creado por Krunnware que es una consultora de TI (Tecnologías de la Información). El sistema tiene las siguientes funcionalidades:
 - Llevar el control de asistencias y evaluaciones de los alumnos.

- Tener un registro sobre los datos personales del alumno y el personal administrativo.
- Llevar un registro sobre los eventos peculiares (incidencias) del alumno y personal administrativo.
- Emisión por período de las evaluaciones.

CALMECAC emite los siguientes reportes y estadísticas:

- Emisión por período de las evaluaciones.
 - Credenciales de identificación para los estudiantes.
 - Catálogos: empleados, materias, grupos, alumnos, e incidencias.
 - Cuadros de concentración.
- **RESIGES:** Es un programa informático para gestionar Residencias de Mayores y Centros de día. Resiges está desarrolla por el Grupo Gestión de España, Barcelona.

Representa un gran avance en la gestión de Residencias de Mayores, haciendo más ágil el servicio diario de su residencia.

El programa tiene las siguientes funciones:

- Puede llevar la gestión de los residentes, de habitaciones, dispone de una completa ficha médica de cada interno, puede hacer una gestión efectiva del personal mediante el generador de cuadrantes incluido en el programa, tiene a los familiares informados mediante los informes generados por el programa, lleva la gestión de facturación /cobro y dispone de un gran número de informes y estadísticas que permitirán hacer una radiografía continua de la marcha de su residencia.

- Se adapta a las necesidades concretas de su residencia, puede pedir cambios que sean necesarios y que sea el programa ideal para cualquier empresa. Se trata de un programa vivo, no es un programa cerrado en el que no se admiten cambios.

Se ha adaptado al uso de cientos de residencias, con cambios específicos para cada Comunidad Autónoma, tipo de centro, necesidades de gestión, etc.

➤ **CONTROL DE SANIDAD.**

Genera informes para el control por parte de Sanidad en la Residencia:

- Control de normas higiénicas.
- Control de limpieza y desinfección.
- Control de potabilidad del agua.
- Control de desinsectación y desratización.

Sistemas nacionales.

Actualmente en Cuba no existen muchos sistemas que respondan a este tipo de situaciones sólo algunas propuestas hechas en las cuales se trabajan:

- **SAIDO:** (Sistema Automatizado de Información Docente) es un sistema creado por **CECAM** (Centro de Cibernética Aplicada a la Medicina). Este sistema gestiona toda la información docente en las facultades de medicina del país. De este se derivan diferentes paquetes que realizan funcionalidades específicas. Algunos de los paquetes que utiliza **SAIDO** son los siguientes:
 - **SAIDO-F** (Sistema Automatizado de Información Docente, Facultades): Es un sistema que incluye módulos de matrícula, actualización y notas de los alumnos, permitiendo obtener por selección listados sobre los datos de los mismos.

- **SICRE_F** (Sistema de Control de Residentes, Facultad): Mantiene datos sobre residentes activos, bajas y graduados, sus datos generales, evaluaciones, cursos, estancias y otras actividades.

Sistemas en la UCI.

- **El sistema de Gestión de Información de la Facultad 8 (SGIF 8)**, es un sistema para gestionar toda la información referente a la Facultad 8, posee 6 módulos, de estos uno es para gestionar gran parte de la información no docente que se lleva a cabo en esta facultad y vinculadas a la Residencia Estudiantil como son:
 - Ubicar a los estudiantes en los apartamentos de los edificios con los que dispone la facultad.
 - Conocer en qué edificio, paso de escalera y apartamento de la residencia se encuentra ubicado un estudiante de la facultad o un grupo de estudiantes.
 - Emitir partes de guardia.
 - Conocer cómo se ha comportado la guardia estudiantil de una brigada específica hasta el momento.
 - Consultar los partes de la guardia estudiantil de cualquier día.

El sistema ha sido desarrollado utilizando una arquitectura cliente-servidor, además puede ser utilizado en múltiples plataformas.

➤ **Sistema de Gestión de Residentes:**

- Este sistema es creado por el Departamento de Informatización de la UCI. Gestiona la ubicación de los estudiantes en la Residencia Estudiantil, genera reportes de estudiantes por apartamentos, edificios, cuartos, lo que permite saber exactamente la ubicación de los estudiantes y saber el estado de la capacidad de cada apartamento.

- El sistema es una aplicación Web desarrollada con PHP, utiliza PostgreSQL como gestor de base de datos para el almacenamiento de los datos, utiliza el Apache como servidor web y servicio web para intercambiar información con otras aplicaciones.

Al analizar estos sistemas se tomaron algunas características positivas como son: La gestión y control de los datos generales del estudiante, ya que contiene las principales características de estos, que son la entrada, eliminación y actualización de los datos.

Los reportes permiten obtener la información de una forma rápida y oportuna que facilita el intercambio de información así como la toma de decisiones. A pesar que estos sistemas tienen algunos aspectos positivos, no reúnen todas las condiciones y característica necesarias para satisfacer el trabajo de los trabajadores de la Residencia de la Facultad 1.

1.4.2 Deficiencias de los sistemas estudiados.

De acuerdo con las necesidades y objetivos del Módulo de Becado se hizo un estudio exhaustivo de estos sistemas y, aunque brindan apoyo para el desarrollo de la implementación del mismo no dan solución a sus principales objetivos puesto que:

- Algunos de estos sistemas fueron desarrollados sobre software propietario.
- Están enmarcados en las particularidades de sus organizaciones.
- Carecen de un módulo que controle la guardia, la cuartelería y la limpieza de los aptos.

1.5 Tendencias y tecnologías actuales.

1.5.1 Metodologías de desarrollo.

De acuerdo con el análisis de algunos conceptos se pudo concluir que las metodologías de desarrollo son el conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a producir un nuevo producto. Imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo crean ejecutando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería. Tienen el propósito de

garantizar la eficacia a la hora de cumplir los requisitos y de disminuir la pérdida de tiempo empleado durante el desarrollo del mismo.

Metodologías ágiles.

Las metodologías ágiles son un marco de trabajo conceptual de la ingeniería de software permitiendo promover iteraciones a lo largo de todo el ciclo de vida de un proyecto. Minimiza riesgos desarrollando productos en cortos plazos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración y cada una incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación.

➤ Scrum.

Scrum, más que una metodología de desarrollo software, es una forma de auto-gestión de los equipos de programadores. Un grupo de programadores deciden cómo hacer sus tareas y cuánto van a tardar en ello. Scrum ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro.

Permite además, seguir de forma clara el avance de las tareas a realizar, de forma que los jefes puedan ver día a día cómo progresa el trabajo.

Ideal para proyectos con un rápido cambio de requerimientos. Se lleva bien con las metodologías ágiles y en concreto, con la programación extrema.

Artefactos SCRUM.

Sprint.

- Es la base del desarrollo Scrum.
- Su duración máxima es de 30 días.
- Se llevan a cabo las tareas pre-establecidas y no se puede modificar el trabajo acordado en el backlog.
- Sólo el ScrumMaster puede abortar un sprint si lo considera no viable por alguna de las siguientes razones:

- ☐ Las circunstancias del negocio han cambiado.
- ☐ La tecnología acordada no funciona.
- ☐ El equipo ha tenido interferencias.

Product Backlog.

- Crea un listado con los requisitos de los usuarios o propietarios del sistema para planificar el proyecto.
- No es una lista completa y definitiva. Es sólo una estimación inicial de los requisitos.
- Es un documento dinámico que incorpora las constantes necesidades del sistema y se mantiene durante todo el ciclo de vida (hasta la retirada del Sistema.).

Sprint Backlog.

- Especifica la serie de tareas que se van a desarrollar según los requisitos señalados.
- Estas tareas tienen una duración de entre 4 y 16 hrs. de trabajo.
- Las de mayor duración intentar descomponerlas en Sub-Tareas dentro de ese rango de tiempo.
- Al final del sprint se busca un incremento en la funcionalidad. [7]

➤ XP (Programación Extrema).

- Es un proceso ligero, ágil, de bajo riesgo, flexible, predecible, científico y divertido de desarrollar software.
- Está orientado hacia quien produce y usa el software (retroalimentación continua cliente y desarrollador).
- Reduce el costo del cambio en todas las etapas del ciclo de vida del sistema.
- Combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo. [7]

Características XP.

- Metodología creada a base de prueba y error.
- Énfasis en el desarrollo del software más que una buena documentación.
- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- No introduce funcionalidades antes de que sean necesarias.
- El cliente o el usuario se convierte en miembro del mismo equipo.
- Su utilidad es medida en los siguientes valores:
 - ☐ Simplicidad en las soluciones implementadas.
 - ☐ Comunicación.
 - ☐ Retroalimentación.
- Coraje (si funciona... mejóralo)[7]

➤ **Scrum-XP.**

La unión de estas metodologías está dividida en cuatro fases, estas son:

- **Planificación-Definición**
- **Desarrollo**
- **Entrega**
- **Mantenimiento**

Cada una de estas fases está compuesta por una serie de actividades que son las que generan los artefactos que quedan incluidos en el nuevo expediente de proyecto.

- **Historias de Usuarios.**
- **Tareas de Ingeniería.**
- **Pruebas.**

En la generación de los artefactos en cada fase las características de las metodologías ágiles son un elemento fundamental, puesto que tienen como condición la no duplicación de esfuerzos. Esto se debe a que el cliente forma parte del equipo de trabajo y así se evita una documentación extensa.

La unión de estas metodologías estuvo estimulada por las ventajas que crean. Por parte de Scrum se obtiene una planificación y organización excepcional y XP refuerza con sus prácticas todo el proceso de desarrollo, logrando como resultado que este sea más completo y organizado.

1.5.2 Lenguajes de programación web.

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial. Para realizar una página con la programación Web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL (Uniform Resource Locators), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita.

El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una página Web por toda la red de Internet, como es el HTTP (Hypertext Transfer Protocol).

Y por último el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una página Web, este lenguaje es el HTML (Hypertext Markup Language).

En la programación Web, el HTML es el lenguaje que permite codificar o preparar documentos de hipertexto, que viene a ser el lenguaje común para la construcción de una página Web. [12]

PHP v5.0

Concepto.

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero

actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

Características de PHP v5.0.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.
- Un sitio con páginas webs dinámicas es aquel que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para propósitos particulares.[9]

Ventajas de PHP v5.0.

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

Desventajas de PHP v5.0.

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Perl.

Concepto.

Perl es un lenguaje pensado para la manipulación de cadenas de caracteres, archivos y procesos. Esta manipulación se va simplificada por el importante número de operadores a disposición del usuario. El lenguaje Perl se percibe habitualmente como un lenguaje intermedio entre los shell scripts y la programación en C. En efecto, los programas en Perl son una sucesión de instrucciones y son similares a los shell scripts porque no existe un procedimiento principal como la subrutina main en C. Sin embargo, se parece al lenguaje C en su sintaxis y en el número importante de funciones que permiten la manipulación de cadenas de caracteres y archivos. [11]

Características de Perl.

- Gran cantidad de operadores y funciones integradas.
- Muchos módulos externos para casi cualquier cosa.
- Usar programación orientada a objetos.
- Escribir código claro o ilegible.
- Largo contenido de strings.
- Tamaño de listas.
- Usa muchos símbolos raros y variables especiales.[10]

Ventajas de Perl.

- Contiene muchas funciones para manipulación de texto.
- Está soportado en gran cantidad de plataformas por lo que es muy portable.
- Se programa rápido.
- DBMS llamado DBI.

- Manejo de XML.[10]

Desventajas de Perl.

- Empiezan a correr muy lento.
- Trata de hacerse de la “vista gorda” de pequeños errores de programación.
- Difícil de aprender todo porque es muy grande. [10]

Java.

Concepto.

Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la Compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras. [1]

Características de Java.

- Lenguaje simple.
- Orientado a objetos
- Distribuido.
- Interpretado y compilado a la vez.
- Robusto.
- Seguro.
- Indiferente a la arquitectura.
- Portable.
- Alto rendimiento.
- Multihebra.

- Dinámico. (3)

Ventajas de Java.

- Es un lenguaje relativamente sencillo y fácil de aprender.
- Es muy productivo, con poco código se puede hacer mucho más que con la misma cantidad de código de C ó C++
- Es un Lenguaje de Programación Orientada a Objetos lo que permite organizar el código en entidades como las clases compuestas de datos y funciones, y a través de la característica de la herencia se puede organizar las clases en jerarquías. [5]

Desventajas de Java.

- Java es lento comparado con C ó C++, pero en algunas tareas es casi igual de veloz.
- Java no permite el acceso directo a todos los recursos de hardware de la PC. Aunque esto también podría ser una ventaja ya que así es más fácil.
- Actualmente hay tantas bibliotecas de clases para Java que es casi imposible conocer para qué sirven todas y cada una de ellas. [5]

En conclusión:

Al ver y analizar las ventajas que ofrece cada lenguaje de programación, Perl y Java superan a los demás lenguajes. Perl presenta como desventaja que consume muchos recursos en la computadora, y se torna lento en lenguajes de bajo nivel, Java por su parte presenta alta complejidad tecnológica.

Se decide PHP como lenguaje para desarrollar la aplicación Web debido a que viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un e-commerce, XML, creación de PDF), es multiplataforma, o sea, funciona en todas las plataformas que soporten Apache, es software libre. Se puede obtener en la Web y su código está disponible bajo la licencia GPL.

1.5.3 Sistemas Gestores de Bases de datos.

Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Los SGBD deben cumplir:

- Definir una Base de Datos: especificar tipos, estructuras y restricciones de datos.
- Construir la Base de Datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la BD: Realizar consultas, actualizarla y generar informes.

PostgreSQL v8.0.

- PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde la década de 1980.
- El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.
- PostgreSQL es ampliamente considerado como una de las alternativas de sistema de bases de datos de código abierto.[4]

Características de PostgreSQL v8.0.

- **Soporte SQL92/SQL99.**
PostgreSQL implementa un subconjunto extendido de los estándares SQL92 y SQL99.
- **Transacciones.**
Permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.
- **Integridad referencial.**
PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

- **Bloqueos de tabla y filas.**

Postgres ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas. Algunos de estos modos de bloqueo los adquiere PostgreSQL automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones.

- **Constraints y triggers.**

Tienen la función de mantener la integridad y consistencia en la BD.
Ejecución de acciones antes o después de un evento de BD.

- **Múltiples tipos de datos predefinidos.**

Como todos los manejadores de bases de datos, PostgreSQL implementa los tipos de datos definidos para el estándar SQL3 y aumenta algunos otros.

- **Soporte de tipos y funciones de usuario.**

PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario. Incorpora una estructura de datos Array.

- **Conectividad TCP/IP, JDBC y ODBC.**

- **Interfaz con diversos lenguajes.**

C, C++, Java, Delphi, Python, Perl, PHP, Bash. [4]

Ventajas de PostgreSQL v8.0:

- **Instalación ilimitada.**

Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

- **Soporte.**

Además de nuestras ofertas de soporte, tenemos una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir.

- **Ahorros considerables en costos de operación.**

PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.

- **Estabilidad y Confiabilidad Legendarias.**

Es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona.

- **Extensible.**

El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

- **Multiplataforma.**

PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y ahora en versión nativa para Windows.

- **Diseñado para ambientes de alto volumen.**

PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

- **Herramientas gráficas de diseño y administración de BD.**

Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect). [4]

Desventajas de PostgreSQL v8.0:

- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K recopilando, pero con una disminución considerable del rendimiento.
- Es algo lento.

MySQL.

Es un sistema de gestión de bases de datos relacional, y un software de código abierto, licenciado bajo la GPL de la GNU. El lenguaje de programación que utiliza MySQL es Structured Query Language (**SQL**) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.[2]

Características principales de MySQL.

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.

- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas.[2]

Ventajas de MySQL.

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad.[2]

Desventajas de MySQL.

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).[2]

Oracle.

Oracle es un sistema de gestión de base de datos relacional (RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones.

- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

En conclusión:

Una vez analizados los sistemas gestores de bases de datos antes mencionados, viendo sus ventajas y desventajas, y principales características la aplicación utilizará PostgreSQL como SGBD, ya que posee gran escalabilidad, soporta una gran cantidad de peticiones simultáneas de forma correcta, implementa el uso de rollback, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, 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. MySQL carece de soporte para transacciones, rollback's y subconsultas, no posee integridad referencial, no es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad. ORACLE presenta desventajas en la seguridad, además de ser muy caro.

1.5.4 Servidores Web.

Un servidor Web es un programa que sirve datos en forma de páginas Web, hipertextos o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo HTTP.

Con esto, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que solemos conocer como un navegador Web. [8]

Apache v2.2.

Principales características de Apache v2.2:

- Servidor Web de código abierto más utilizado en el mundo.
- Software libre bajo licencia GPL.
- Multiplataforma.
- Rápido y confiable.
- Soporte para bases de datos SQL.

Ventajas de Apache v2.2.

- Su licencia. Esta es de código abierto del tipo BSD que permite el uso comercial y no comercial de Apache.
- Una talentosa comunidad de desarrolladores siguiendo un proceso abierto de desarrollo.
- Arquitectura modular. Los usuarios de Apache pueden adicionar fácilmente funcionalidad a sus ambientes específicos.
- Portabilidad. Apache trabaja sobre todas las versiones recientes de UNIX y Linux, Windows, BeOs, mainframes.
- Es robusto y seguro. [6]

Desventajas de Apache v2.2.

- Es complicada su instalación y configuración.

En conclusión:

Como la Intranet de la Facultad 1 está montada sobre Apache se decidió tomar este servidor web para desarrollar el Módulo de Becado. Los motivos por los cuales la Intranet determinó utilizar esta herramienta son porque es muy confiable, su licencia es de código abierto, es multiplataforma y brinda gran seguridad.

1.5.5 El uso de los CMS.

Un Sistema de Gestión de Contenidos (**Content Management System**, en inglés, abreviado **CMS**) permite la creación y administración de contenidos principalmente en páginas web.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño.

Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público.

Drupal v5.7

Es un sistema de gestión de contenido para sitios Web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos. Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en

el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto. Drupal se distribuye bajo la licencia GNU GPL, y por lo tanto es software libre.

A pesar de que empezó como un pequeño BBS, Drupal ha llegado a ser mucho más gracias a su arquitectura flexible. Drupal se compone de una infraestructura base y un conjunto de módulos que ofrecen un amplio conjunto de funciones, incluyendo sistemas de comercio electrónico, galerías de fotos,

administración de listas de correo electrónico, e integración de CVS. Es posible añadir módulos de terceros para modificar el comportamiento de Drupal u ofrecer nuevas funciones.

Drupal se usa, entre otros, en intranets de compañías, enseñanza en línea, comunidades de arte y administración de proyectos. Muchos piensan que la relevancia de Drupal en las comunidades de usuarios es lo que lo hace destacarse de la competencia.

Características generales de Drupal

- **Ayuda on-line:** Un robusto sistema de ayuda online y páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores.
- **Búsqueda:** Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- **Código abierto:** El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- **Módulos:** La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados, bookmarks, etc.
- **Personalización:** Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.
- **URLs amigables:** Drupal usa el mod_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.
- **Gestión de usuarios:** Autenticación de usuarios Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, Blogger, LiveJournal u otro sitio Drupal. Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP.

- **Permisos basados en roles:** Los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles.
- **Control de versiones:** El sistema de control de versiones de Drupal permite seguir y auditar totalmente las sucesivas actualizaciones del contenido: qué se ha cambiado, la hora y la fecha, quién lo ha cambiado, y más. También permite mantener comentarios sobre los sucesivos cambios o deshacer los cambios recuperando una versión anterior.
- **Enlaces permanentes (Permalinks):** Todo el contenido creado en Drupal tiene un enlace permanente asociado a él para que pueda ser enlazado externamente sin temor de que el enlace falle en el futuro.
- **Objetos de Contenido (Nodos):** El contenido creado en Drupal es, funcionalmente, un objeto (Nodo). Esto permite un tratamiento uniforme de la información, como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios -o no- sobre cada objeto.
- **Plantillas (Templates):** El sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP.
- **Indicación del contenido:** Drupal exporta el contenido en formato RDF/RSS para ser utilizado por otros sitios web. Esto permite que cualquiera con un 'Agregador de Noticias', tal como NetNewsWire o Radio UserLand visualice el contenido publicado en la web desde el escritorio.
- **Agregador de noticias:** Drupal incluye un potente Agregados de Noticias para leer y publicar enlaces a noticias de otros sitios web. Incorpora un sistema de caché en la base de datos, con temporización configurable.

- **Soporte de Blogger API:** La API de Blogger permite que un sitio Drupal sea actualizado utilizando diversas herramientas, que puedan ser 'herramientas Web' o 'herramientas de escritorio' que proporcionen un entorno de edición más manejable.
- **Independencia de la base de datos:** Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una 'capa de abstracción de base de datos' que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.
- **Multiplataforma:** Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor Web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.
- **Múltiples idiomas y Localización:** Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interfaz Web, importando traducciones existentes o integrando otras herramientas de traducción como GNU ettext.
- **Administración vía Web:** La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún software adicional.
- **Análisis, Seguimiento y Estadísticas:** Drupal puede mostrar en las páginas Web de administración informes sobre referrals (enlaces entrantes), popularidad del contenido, o de cómo los usuarios navegan en el sitio.
- **Registros e Informes:** Toda la actividad y los sucesos del sistema son capturados en un 'registro de eventos', que puede ser visualizado por un administrador.
- **Control de congestión:** Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable y ajustable.

- **Sistema de Caché:** El mecanismo de caché elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor.[13]

En conclusión:

Se decidió utilizar Drupal, ya que este es un módulo que se integra al de la Intranet de la Facultad 1 que a su vez utiliza el mismo CMS. Además por sus características, entre las que se destacan que es de código libre y multiplataforma.

1.5.6 Zend Studio como IDE a utilizar

Es un programa construido por Zend, que han sido los impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones Web. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, y Zend ha lanzado varias versiones del producto para Windows, Linux, y MacOS. Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. La parte el cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, pero para aprovechar toda la potencia de la herramienta de depuración, habrá que disponer de la parte del servidor que instala Apache y el módulo PHP.

Zen Studio posee el Zend Framework, que se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP, te brinda soluciones para construir sitios Web modernos, robustos y seguros. Además es Open Source y trabaja con PHP 5. [16]

1.5.7 Visual Paradigm como Herramienta CASE.

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. [17]

1.6 Conclusiones del capítulo.

En este capítulo se trató el estado del arte y otros temas relacionados con el caso que se plantea, así como, la fundamentación de las decisiones tomadas acerca de las herramientas y metodologías a utilizar. Se decidió emplear como lenguaje de programación PHP v5.0, como sistema gestor de base de datos PostgreSQL v8.0, como CMS a Drupal v5.7 y como servidor web Apache v8.0. La metodología seleccionada fue Scrum XP.

Destacar también, que se hizo el estudio de varios sistemas gestores de información y actividades referentes a residencias y becas estudiantiles de diferentes ámbitos, tanto en el extranjero, como en el país y la UCI, que aunque no fueron una solución real, sirvieron de apoyo para el desarrollo del módulo.

Capítulo 2. Descripción y análisis de la solución propuesta.

2.1 Introducción

En este capítulo para la implementación del Módulo de Becado se realiza un estudio crítico de las técnicas de programación, plataformas y librerías usadas. Se critica el diseño propuesto por los analistas así como un análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados. Se seleccionan las estructuras de datos apropiadas y se describen las clases para la implementación de los algoritmos antes mencionados para representar computacionalmente dichas estructuras. Además se describen los algoritmos no triviales a implementar y el análisis de la complejidad de los mismos. Por todo lo antes mencionado queda conformado el Modelo de Objeto.

La propuesta de automatización de este trabajo es la implementación de un módulo, que permita la gestión de las actividades y administración de los datos del estudiante en la Residencia de la Facultad 1. Estas funcionalidades pueden ser usadas por cualquier sistema informático, independientemente de la plataforma en que hayan sido desarrollados.

Principales funcionalidades a automatizar:

- Gestionar los datos del estudiante.
 - Los datos son nombre y apellidos, carnet de identidad, número de solapín, dirección particular y teléfono, correo, año que cursa, si son militantes (UJC y/o PCC), si son dirigentes de la FEU o la UJC, edificio, apto y teléfono, proyecto en los que participan, laboratorio en que trabajan y teléfono, si son alumnos ayudantes, asignatura que imparten, evaluación del becado.(ver anexo1)
- Controlar la cuartería.(ver anexo 2)
- Controlar la limpieza del apto.(ver anexo3)
- Controlar la guardia estudiantil.(ver anexo 4)

- Controlar el TSU.(ver anexo 5)
- Generar reporte de estudiante: reporte por categoría:(militantes, dirigente, evaluación mensual y Alumno ayudante), por año, por edificio y por apto.(ver anexo 6)
- Registrar evaluación mensual. (ver anexo 7)

A continuación se muestra la interfaz de usuario:

Módulo Becado
...Por una Residencia mejor...

Este módulo Permite la Gestión de todos los procesos que conciernen a las actividades de becado en la residencia de la Facultad 1.

JUN 01 **Principal**

Donde podrá solucionar con mayor eficiencia y rapidez la gestión de los diferentes procesos de las actividades de la Residencia de la Facultad 1.

administrador

- ▶ Controlar Cuartelería
- ▶ Controlar evaluación mensual
- ▶ Controlar Guardia
- ▶ Controlar limpieza
- ▶ Controlar Trab. Social. U.
- ▶ Mi cuenta
- ▶ Gestionar Estudiante
- ▶ Reporte del Estudiante
- ▶ Administrar
- ▶ Cerrar sesión

CopyRigth® Facult@d 1.

Powered by [Drupal](#) - Design by [artinet](#)

Requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. A continuación se muestran los requerimientos no funcionales:

Usabilidad: El uso del módulo será permitido a cualquier usuario que esté autenticado en el mismo desde cualquier punto de la UCI ya que es compatible con todas las plataformas. El uso de los módulos es muy práctico ya que pueden aportar gran independencia entre el sistema que usa el módulo y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro.

Software: Se necesita como sistema Operativo Windows 98 o superior, Linux o Unix, en sus versiones de S.O servidores. Los servidores están desarrollados sobre una plataforma Web con lenguaje de programación PHP 5.0. La Base de Datos desarrollada en el Sistema Gestor de BD PostgreSQL. La Intranet está montada sobre CMS Drupal.

Hardware: Se necesitan como requerimientos mínimos una PC con procesador Pentium IV o superior. Requiere como mínimo RAM 128 MB. El disco duro requiere como mínimo 5 GB para almacenar la Base de Datos.

Portabilidad: El módulo podrá ser accedido desde cualquier protocolo de transporte que sea capaz de transportar texto, como por ejemplo (http, http, SMTP, etc.). El diseño del módulo permite que el mismo pueda funcionar tanto en sistemas operativos libres como propietarios.

Seguridad y Privacidad: El acceso será controlado con nombres de usuario y contraseñas. Solo los usuarios con derechos de administrador podrán acceder a las funciones administrativas. Se establecerá una llave privada entre el cliente y el proveedor del módulo, para evitar acceso no autorizado a los servicios.

Confiability: La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores. Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario. La aplicación estará disponible las 24 horas brindando así todas las funcionalidades de la misma.

Importación y exportación de datos: El sistema deberá almacenar todos los datos en una base de datos PostgreSQL Server, donde puedan ser accedidos por el módulo.

Restricciones del diseño y la implementación: Se utiliza UML para lograr una mejor documentación del sistema y como herramienta de apoyo Visual Paradigm. Se utiliza como lenguaje de programación PHP y el gestor de base de datos PostgreSQL v8.0. Se usa la librería ADOdb para la abstracción a datos.

Legales: El módulo y toda la documentación generada pertenecen al proyecto Intranet de la Facultad 1 de la Universidad de las Ciencias Informáticas.

Requisitos del sistema:

	Mínimo	Recomendado	Óptimo
Procesador	233 MHz	450 MHz	900 MHz o más
Memoria	128 MB RAM (funcionamiento limitado)	256 MB RAM	512 MB RAM o superior
Vídeo	Súper VGA (800 x 600) con 2 MB de vídeo	SXGA (1024 x 768) con 8 MB de vídeo	SXGA (1024x768) con 32 MB de video o más
Espacio en Disco Duro	2 GB	20 GB	80 GB o superior
Unidades	CD-ROM	CD-ROM / DVD-ROM	CD-ROM/CD-RW / DVD- R/DVD-RW/DVD-DL

Periféricos	Teclado y mouse	Teclado y mouse	Teclado y mouse
Otros (Multimedia)		Tarjeta de sonido, Altavoces, y/o Auriculares	Tarjeta de sonido, Altavoces, y Auriculares

Tabla 2.1 Requisitos del sistema.

2.2 Estudio crítico y valorativo de la técnica de programación, plataforma y librerías usadas para el desarrollo del módulo de Becado.

2.2.1 Estudio de los Módulos de Drupal.

Un módulo es la asociación de un conjunto de varias funciones en Drupal y ayudan a ofrecerle mayor funcionalidad a la web. Para crear un módulo en Drupal son necesarios tres tipos de ficheros de tipo PHP, de los cuales uno es de extensión .info para portar la información que es pertinente al módulo como tal, información como la versión y el autor del módulo, etc., otro fichero es el de extensión .install para crear las tablas y demás que se van a necesitar para que el módulo trabaje, y por último un fichero de extensión .module que será en donde se escriban funciones que actúen de forma de enganche al ser llamadas por Drupal a la hora de construir una página web y gestionar el contenido. Ejemplo de estas funciones escritas: hook_help, hook_perm, hook_menu. Los módulos se pueden desactivar temporalmente para reducir la carga del servidor.

En Drupal existen varios tipos de módulos importantes como son el módulo de bloque, que da la posibilidad de crear contenido abreviado, que aparecen en la mayoría de los temas en los bloques de la izquierda y la derecha; y están además los módulos de nodo que generan el contenido de la página del sitio, como puede ser la página del foro entre otros.

A continuación se ofrece una descripción de los módulos del núcleo de Drupal y los opcionales:

➤ Módulos del Núcleo de Drupal.

Módulo Filter. (Filtro)

- Permite la entrada de contenido para el sitio.

Módulo Watchdog. (Vigilante o Registro)

- Es una lista de los eventos almacenados, que contiene datos de uso, de rendimiento, de errores, de avisos y de información sobre las operaciones que se realizan. Es vital verificar el informe del vigilante con cierta regularidad, porque a menudo es el único que dice qué está pasando.

Módulo Node.

- Todo el contenido en un sitio Web de Drupal se almacena y se trata como “nodos”. Un nodo es por ejemplo una página, una encuesta, una historia, un texto del foro, o una entrada del blog. Este módulo es el que posibilita que se enumere, clasifique y maneje todo el contenido en el sitio.

Módulo Block.

- Controla los bloques que se muestran alrededor del contenido principal.

Módulo User.

- Permite que los usuarios se registren, ingresen y terminen la sesión.

Módulo System. (Sistema)

- Proporciona opciones predefinidas para todo el sistema, se encarga de acceder a todas las páginas de administración

➤ Módulos Opcionales incluidos en Drupal.

Módulo Path.

- Permite indicar alias a los URLs de Drupal ayudando a mejorar la legibilidad de los mismos para los usuarios. Una página puede tener más de un alias.

Módulo Menu. (Menú)

- Brinda una interfaz para controlar y personalizar el completo sistema de menús que tiene Drupal.

Módulo Upload. (Subir archivos)

- Este módulo permite que los usuarios carguen y adjunten archivos al contenido.

Módulo Taxonomy.

- Activa la categorización del contenido.

Módulo Search. (Búsqueda)

- Este módulo permite la realización de búsquedas en todo el sitio creado.

2.2.2 Técnicas de programación y creación de módulos con Drupal.

Como ya se ha explicado anteriormente para comenzar a crear un módulo es necesario crear un archivo PHP, y guardarlo con nombre_módulo.module, otro con el nombre_módulo.install, este sería para crear las tablas en las base de datos y un último archivo nombr_emódulo.info para ofrecer información acerca

del módulo. Todas las funciones del módulo que son usadas por Drupal siguen la siguiente sintaxis, se nombran como {hook}_{nombre_módulo}, donde "hook" es una función predefinida llamada prefijo. Y Drupal llamará a esas funciones para conseguir datos específicos, teniendo así estos nombres bien definidos.

Lo primero que se debe incluir al archivo .module debe ser el siguiente fragmento de código:

```
<?php  
/* $Id$ */
```

Según el Estándar de Código de Drupal se puede omitir el tag de cierre `?>`, la cadena `$Id$` permitirá mantener un seguimiento del número y fecha del módulo cuando sea convertido en CVS. Esto significa que cada vez que se modifique este módulo el controlador de versiones de Drupal le asignará un valor distinto al módulo y una fecha de modificación, permitiendo tener un control total sobre los cambios realizados al código y la persona que los realizó.

El formato general que debe tener el archivo `.info` es el siguiente:

```
; $Id$  
name = Nombre del módulo  
description = "Una descripción de lo que hace el módulo."  
dependencies = "Módulos de los que depende."
```

A continuación se muestran los hooks más importantes, que se utilizan a la hora de crear un nuevo módulo:

Hook para crear módulos.

Hook_help.

- Define temas de ayuda para que sea más fácil el uso del módulo que lo implemente

Hook_access.

- Limita restricciones de acceso a los nodos.

Hook_perm.

- Se especifican quienes tienen permisos del módulo. Se define quien puede acceder al contenido del sitio o administrar el módulo.

Hook_menu.

- La función `hook_menu` es llamada siempre por Drupal antes de crear las páginas, antes de mandarlas al navegador, y en ella pueden indicarse los puntos concretos en los que el módulo interactuará con la web.

Hook_view.

- Esto es un gancho usado por módulos de nodo. Esto permite a un módulo definir un método de mostrar los nodos.

Hook_form_alter.

- Permite hacer alteraciones a cualquier formulario antes que sea renderizado.

Hook_form.

- Es llamado cuando se necesita crear o editar algún artículo y devuelve el título de nodo, el área de texto, y otro campo específico del tipo de nodo.

Hook_submit.

- Se usa para las acciones que deben ocurrir cuando el módulo es guardado. Y también es utilizado después de que la validación ha tenido éxito y antes de insertar o actualizar.

Hook_validate.

- Esto es un gancho usado por módulos de nodo. Es utilizado para permitir al módulo verificar que el nodo se encuentra en un formato válido para añadirse al sitio.

Hook_delete.

- Es un gancho usado por módulos de nodo. Permite al módulo tomar medidas cuando un nodo es eliminado de la base de datos, por ejemplo, borrando la información de tablas relacionadas.

Hook_update.

- Indica cuando un nodo se actualiza. Es utilizado para permitir al módulo tomar acciones cuando un nodo es actualizado en la base de datos.

Hook_link.

- Define los enlaces internos de Drupal. Este hook posibilita añadir enlaces en diversos lugares en Drupal, como en bloques de navegación o en nodos.

Hook_settings.

- Para que el administrador pueda configurar lo que se va a mostrar, se crea la página de configuración. Casi siempre el administrador es quien tiene acceso a esta página.

Hook_search_item.

- Este módulo permite modificar los resultados de una búsqueda.

Hook_search.

- Permite a un módulo realizar búsquedas sobre el contenido que este define cuando una búsqueda de sitio es realizada.

Hook_nodeapi.

- Actúa sobre nodos definidos por otros módulos. Este hook no es reservado para módulos de nodo. Al contrario, esto permite a módulos reaccionar a acciones que afectan todas las clases de nodos, independientemente de si algún módulo definió el nodo.

Hook_filter.

- Define filtros de contenidos. El contenido en Drupal es pasado por todos los filtros permitidos, antes que sea mostrado. Con este se puede modificar el contenido al gusto del administrador de sitio.

2.2.3 Estándares de codificación.

Los estándares de codificación son las reglas que se van a seguir para escribir el código. A continuación se explican más detalladamente:

- Sangrado:
- Llamadas a funciones.
- Declaración de funciones.
- Arreglos.
- Comentarios.

- Etiquetas PHP.
- Comentario de bloques de cabecera.

2.2.4 Valoración crítica del diseño propuesto por el analista.

En la fase de análisis y diseño se realizó un excelente trabajo, esto ayudó a que la fase de implementación se realizara sin problemas, solo señalar que se decidió efectuar el cambio de la metodología RUP por Scrum XP. Esto requirió que se generaran las HU y las tareas para su desarrollo en cada iteración. Algunos atributos también fueron objetos de cambio asegurando que fuera más flexible, fácil y accesible la implementación utilizando las ventajas que brinda Drupal.

2.2.5 Análisis de posibles componentes, módulos e implementaciones ya existentes que puedan ser rehusados. Estrategias de integración.

Aquí se analizan varios módulos de Drupal que por su importancia y usabilidad son necesarios para el correcto funcionamiento del Módulo de Becado. A continuación se describen sus principales características y su integración con el módulo a implementar.

Taxonomy: Es uno de los módulos más dinámicos y más populares de Drupal por las características y los beneficios que brinda. Permite a los usuarios autorizados etiquetar contenido usando etiquetas personalizadas y clasificar automáticamente nuevos contenidos basados en esta taxonomía. Esto posibilita que la clasificación y la recuperación de información sean verdaderamente flexibles. En el módulo a desarrollar se tendrá en cuenta este módulo para etiquetar el contenido.

Views: Aporta un método flexible a los diseñadores para controlar la forma en que se lista el contenido de un sitio Drupal. Es de gran utilidad ya que permite mostrar listados de contenido y para la obtención de reportes.

Search: Este módulo es usado como complemento al módulo Views, ya que permite filtrar información sobre un determinado listado de contenido, facilitando que las búsquedas y los reportes de contenido sean mucho más fáciles de manejar.

Para integrar los módulos, primeramente se copia el módulo en la carpeta modules, que se encuentra en la raíz del sitio en Drupal que se está confeccionando, luego se va a administrar y se chequea la lista de módulos, es necesario primero incluir o activar los módulos de los cuales se depende para el funcionamiento del que se quiere activar. De esta manera el módulo se instalará y brindará las funcionalidades que se requieran en Drupal con los diferentes enganches escritos que proporcione.

2.2.6 Descripción de los algoritmos no triviales a implementar. Análisis de complejidad de los mismos.

Implementación del hook_help ()

Descripción: Este hook permite crear temas de ayuda al módulo para su mejor comprensión y funcionamiento.

Devuelve: Una cadena de texto con la ayuda.

Complejidad: O (1)

Código:

```
function Modulo_Becado_help($section=""){
    $output = "";
    switch ($section)
    {
        case "admin/system/modules#name":
            $output = t("Modulo de Becado");
            break;
        case "admin/system/modules#description":
            $output = '<p>'. t("Módulo para gestionar actividades de los becados en la Residencia"). '</p>';
            break;
        default:
            $output = "";
            break;
    }
}
```

```

    }
    return $output;
}

```

Implementación del hook_menu ()

Descripción: Este hook permite definir un menú y sus entradas al módulo que lo implemente.

Devuelve: Un array indexado con todas las entradas del menú y sus atributos (path, type, title, callback, access, etc).

Complejidad: O(1)

Código:

```

function Modulo_Becado _menu($may_cache){
    $items = array();
    if ($may_cache) {
        $items[] = array(
            'path' => ' Modulo_Becado ',
            'title' => t('Modulo de Becado'),
            'callback' => ' Modulo_Becado _page',
            'access' => TRUE
        );
    }
    return $items;
}

```

Implementación de Buscar_estudiante_form ()

Descripción: Segmento de código de la función que define el formulario para buscar a un estudiante dado, con todos sus campos pasando los argumentos por la función drupal_get_form() que es la encargada de mostrar el formulario al usuario, además de captar los datos por el post de la page.

Devuelve: Un array indexado con el formulario, todos sus campos y sus atributos

Complejidad: $O(n)$

Código:

```
Function Buscar_estudiante_nameform()
{
    $form = array();
    $form['name'] = array(
        '#title' => t('Insertar Estudiante'),
        '#type' => 'fieldset',
        //'#description' => t('La persona que desea buscar sus datos'),
        '#collapsible' => true,
        '#weight' => 0,
    );

    $form['name']['solapin'] = array(
        '#title' => t('Solapin'),
        '#type' => 'textfield',
        '#required' => 1,
        '#description' => t('Escriba el Solapin del Estudiante a Insertar'),
    );

    $form['#action'] = "?q=modulo_becado/buscarresult";
    $form['#method'] = "POST";
    $form['submit'] = array(
        '#type' => 'submit',
        '#value' => t('Enviar'),
    );
    return $form;
}
```

Implementación de Cuarteleria_Cuartelero_form(\$nombre).

Descripción: Segmento de código de la función que define el formulario para insertar a un estudiante dado su nombre, todos sus campos. Debe encargarse de chequear que inserte correctamente todos los datos y permitir mostrar un mensaje de error si no se obtiene un resultado positivo utilizando la función `drupal_set_message()`, encargada de generar los mensajes de Drupal.

Devuelve: Un array indexado con el formulario, todos sus campos y sus atributos,

Complejidad: $O(n)$

Código:

```
function cuarteleria_cuartelero_form($nombre) {
    $form = array();
    $form['nombre'] = array(
        '#title' => t('Nombre completo'),
        '#type' => 'item',
        '#value' => t($nombre),
    );
    $form['nombre'] = array(
        '#title' => t('Nombre completo'),
        '#type' => 'hidden',
        '#value' => t($nombre),
    );
    $form['cuarteleria']['fecha'] = array(
        '#title' => t('Fecha'),
        '#type' => 'date',
        '#required' => TRUE,
        '#weight' => 3,
    );
    $form['submit'] = array(
        '#type' => 'submit',
```

```

        '#value' => t("Asignar"),
    );
return $form;
}

```

Implementación de Cuarteleria_Cuartelero_form_submit(\$form_id, \$form_values).

Descripción: Segmento de código de la función que permite insertar a un estudiante dado su nombre, todos sus campos cuando se accione el componente del form (submit) correspondiente a la función del mismo nombre. Debe encargarse de chequear que se elabore correctamente la consulta SQL y que los argumentos se tomen de manera correcta en caso de encontrar algún problema, debe mostrarse un mensaje de error utilizando la función `drupal_set_message()`, encargada de generar los mensajes de Drupal.

Devuelve: Un array cuartelero con todos sus atributos para almacenar correctamente en la base de datos.

Complejidad: $O(n)$

Código:

```

function cuarteleria_cuartelero_form_submit($form_id, $form_values) {
    $fecha = $form_values['fecha'];
    $anno=$fecha[year];
    $mes=$fecha[month];
    $dia=$fecha[day];
    $nombre =$form_values['nombre'];
    $incidencia='ninguna';
    $evaluacion='ninguna';
    db_query("INSERT INTO {Tb_cuartelero} (nombre,dia,mes,anno,insidencia,Tipo_Eva) VALUES
    ('$nombre',$dia,$mes,$anno,'$incidencia','$evaluacion')");
    drupal_set_message(t('Asignado correctamente'));
    return 'Cuarteleria/edificio';
}

```

2.3- Descripción de las Historias de usuarios que se utilizan para representar computacionalmente dichas estructuras. Diagrama de Componentes.

2.3.1 Personas relacionadas con el sistema.

Personas	Descripción
Vicedecano Residencia.	Gestiona toda la información referente a los estudiantes, así como el control de la cuartería, la limpieza y la guardia estudiantil.
Psicopedagoga.	Realiza evaluación del estudiante y comprueba la cuartería y la limpieza del apto.
Instructora de Residencia.	Realiza evaluación del estudiante y comprueba la cuartería y la limpieza del apto.
Responsable del TSU.	Realiza evaluación del estudiante en el TSU
Responsable de la guardia.	Realiza evaluación del estudiante en la guardia estudiantil.

Tabla 2.2 Personas relacionadas al sistema.

2.3.2 Historias de usuario del paquete Gestión de Información.

Historias de Usuarios.

Producto al cambio en la metodología del diseño fue necesario describir nuevamente las funcionalidades del sistema pero con la especificación de que estamos haciendo uso de las metodologías ligeras y las

mismas sustituyen los casos de usos por las historias de usuarios las cuales serán descritas a continuación.

Las Historias de Usuarios (HU) son artefactos de la metodología XP, son utilizadas para plasmar de forma esquemática y en un lenguaje claro lo que se va a hacer. Son redactadas de acuerdo a las necesidades del cliente y expresan un estimado del tiempo aproximado que se va a demorar su desarrollo.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Gestionar los datos del estudiante.
Modificación de Historia de Usuario número: (-)	
Usuario(s): Psicopedagoga. Instructora de Residencia.	Iteración Asignada: 1
Prioridad en Negocio: Alta.	Puntos Estimados: 2.
Riesgo en Desarrollo: Alto.	Puntos Reales: 2,14.
Descripción: Insertar estudiante: Se inserta el estudiante con sus datos. Modificar estudiante: Se modifica el estudiante haciendo cambios en los datos del mismo Eliminar estudiante: Se elimina el estudiante insertado previamente.	
Observaciones: Para modificar los datos del estudiante o eliminarlo el estudiante debe existir.	

Tabla 2.3 HU Gestionar información del estudiante.

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Controlar cuartelería.

Modificación de Historia de Usuario número: (-)	
Usuario: Psicopedagoga. Instructora de Residencia.	Iteración Asignada: 1
Prioridad en Negocio: Alta.	Puntos Estimados: 2.
Riesgo en Desarrollo: Alto.	Puntos Reales: 2,14.
Descripción: Insertar cuartelería: Se insertan los cuartereros del mes. Los datos para la inserción son: fecha y nombre del estudiante. Cambiar cuartelería: Se cambia la cuartelería a dos o más estudiantes con respecto a la fecha de su realización. Correr cuartelería: Se adelanta un día antes la fecha de la cuartelería a todos los estudiantes.	
Observaciones: Tienen que existir estudiantes en la BD.	

Tabla 2.4 HU Controlar cuartelería.

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Controlar guardia.
Modificación de Historia de Usuario número: (-)	
Usuario(s): Responsable de la guardia.	Iteración Asignada: 1
Prioridad en Negocio: Alta.	Puntos Estimados: 1
Riesgo en Desarrollo: Alto.	Puntos Reales: 0,7.

Descripción:
Insertar guardia: Se insertan los grupos que están de guardia en el mes. Los datos para la inserción son: fecha y grupo.
Observaciones:
Para insertar un grupo de guardia debe existir en la base de datos.

Tabla 2.5 HU Controlar guardia estudiantil.

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Controlar TSU.
Modificación de Historia de Usuario número: (-)	
Usuario(s): Responsable del TSU.	Iteración Asignada: 2
Prioridad en Negocio: Alta.	Puntos Estimados: 1
Riesgo en Desarrollo: Alto.	Puntos Reales: 1.
Descripción:	
Insertar TSU: Se inserta el grupo que le toca el TSU. Los datos para la inserción son: fecha y grupo.	
Observaciones:	
Para insertar un grupo al TSU debe existir en la base de datos.	

Tabla 2.6 HU Controlar TSU.

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Controlar limpieza.
Modificación de Historia de Usuario número: (-)	

Usuario(s): Psicopedagoga. Instructora de Residencia.	Iteración Asignada: 2
Prioridad en Negocio: Media.	Puntos Estimados: 1.
Riesgo en Desarrollo: Medio.	Puntos Reales: 1.
Descripción: Insertar estudiante: Se inserta el estudiante que está de limpieza. Los datos para la inserción son: nombre del estudiante y fecha.	
Observaciones: Para insertar un estudiante de limpieza tiene que existir en la BD.	

Tabla 2.7 HU Controlar limpieza del apto.

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Generar reporte de estudiante.
Modificación de Historia de Usuario número: (-)	
Usuario(s): Psicopedagoga. Instructora de Residencia.	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1.
Descripción: Se genera un reporte por categoría, año ,edificio y por apto.	
Observaciones:	

Tabla 2.8 HU Generar reporte de estudiante.

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Registrar evaluación mensual
Modificación de Historia de Usuario número: (-)	
Usuario(s): Psicopedagoga. Instructora de Residencia.	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1.
Descripción: Insertar evaluación: Se inserta la evaluación de cada actividad del estudiante. Mostrar evaluación: Se muestra la evaluación de la actividad seleccionada de cada estudiante.	
Observaciones: Tiene que existir un estudiante en la BD.	

Tabla 2.9 HU Registrar evaluación mensual.

2.3.3 Planificación de Sprint.

En Scrum todo el desarrollo ocurre en iteraciones de duración fija. Estas iteraciones de duración fija se denominan Sprint.

La duración del Sprint, de entre 2 y 4 semanas, ó 5 semanas en algunos casos se establece al comienzo del proyecto, y se mantiene durante toda la vida del mismo. Esto es importante para hacer uso de las métricas que provee Scrum.

Durante cada Sprint el equipo desarrolla una porción del sistema. Lo destacable del Sprint es que este desarrollo incremental es con calidad productiva, y cumple algún requerimiento del usuario (por más chico que sea).

De esta manera, al finalizar cada Sprint se tiene una porción del producto con calidad productiva, que el usuario puede decidir poner en producción.

Sprint 1

En este sprint es donde se va a obtener la primera versión del producto. Puede ser el sprint de mayor duración, ya que, al principio cuesta más trabajo obtener un ejecutable. Las HU de alta prioridad serán implementadas ofreciéndole al sistema las primeras funcionalidades.

Sprint 2

En este sprint serán analizadas otras HU de alta y media prioridad respectivamente, que no fueron analizadas en la primera iteración. Con la terminación de la misma se habrán implementado todas las HU de alta prioridad y gran parte de las necesidades del cliente.

Sprint 3

En este sprint serán implementadas las restantes HU y se tendrá finalmente todas las funcionalidades implementadas. Comenzaría entonces la fase de pruebas.

Plan de duración de los sprints.

Sprints	HU a implementar	Duración total de los sprint
Sprint 1	1. Gestionar información del estudiante. 2. Controlar cuartelería. 3. Controlar guardia.	5 semanas
Sprint 2	1. Controlar TSU. 2. Controlar limpieza.	2 semanas

Sprint 3	1. Generar reporte de estudiante.	2 semanas
	2. Registrar evaluación del mes..	

Tabla 2.10 Plan de duración de los Sprint.

Tareas de ingeniería.

Para dar solución a las HU es necesario realizar un grupo de acciones llamadas tareas de ingeniería. Estas son descritas una vez terminada toda la documentación necesaria para generar el código y permiten una mejor organización del proceso de implementación del módulo. Se efectúan al final de cada iteración.

2.3.4 Tareas asignadas a cada iteración.

Tareas asignadas al Sprint 1.

Tarea de Ingeniería	
Número Tarea:1	Número Historia de Usuario: HU1
Nombre Tarea: Desarrollar el formulario para la inserción de estudiantes.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.56
Fecha Inicio: 23 de marzo 2009.	Fecha Fin: 27 de marzo 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción: Se crea el código que permite crear el formulario para la inserción del estudiante en el sistema. El formulario que se crea contiene una serie de datos.	

Tabla 2.11 Desarrollar el formulario para la inserción de estudiantes.

Tarea de Ingeniería	
Número Tarea:2	Número Historia de Usuario: HU1
Nombre Tarea: Desarrollar el formulario para modificar los datos del estudiante.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.84

Fecha Inicio: 28 de marzo 2009.	Fecha Fin: 1 de abril 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción: Es desarrollado el formulario para modificar los datos del estudiante insertado previamente.	

Tabla 2.12 Desarrollar el formulario para modificar los datos del estudiante.

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: HU1
Nombre Tarea: Implementar la funcionalidad para eliminar estudiante.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.56
Fecha Inicio: 2 de abril 2009.	Fecha Fin: 6 de abril 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción : Es desarrollado el formulario para eliminar al estudiante insertado previamente.	

Tabla 2.13 Implementar la funcionalidad para eliminar estudiante.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: HU2
Nombre Tarea: Implementar la funcionalidad para insertar cuartelería.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.56
Fecha Inicio: 7 de abril 2009.	Fecha Fin: 11 de abril 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción : Es desarrollado el formulario para insertar la cuartelería.	

Tabla 2.14 Implementar la funcionalidad para insertar cuartelería.

Tarea de Ingeniería	
Número Tarea:2	Número Historia de Usuario: HU2
Nombre Tarea: Implementar la funcionalidad para cambiar cuartelería.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.56
Fecha Inicio: 12 de abril 2009.	Fecha Fin: 16 abril 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción : Es desarrollado el formulario para cambiar cuartelería.	

Tabla 2.15 Implementar la funcionalidad para cambiar cuartelería.

Tarea de Ingeniería	
Número Tarea:3	Número Historia de Usuario: HU2
Nombre Tarea: Implementar la funcionalidad para correr cuartelería.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.84
Fecha Inicio: 17 abril 2009.	Fecha Fin: 21 de abril 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción : Es desarrollado el formulario para correr la cuartelería.	

Tabla 2.16 Implementar la funcionalidad para correr cuartelería.

Tarea de Ingeniería	
Número Tarea:1	Número Historia de Usuario: HU3
Nombre Tarea: Implementar la funcionalidad para insertar grupo a la guardia.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.7
Fecha Inicio: 22 de abril 2009.	Fecha Fin: 26 de abril 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción : Es desarrollado el formulario para insertar guardia a un grupo docente.	

Tabla 2.17 Implementar la funcionalidad para insertar grupo a la guardia.

Tareas asignadas al Sprint 2.

Tarea de Ingeniería	
Número Tarea:1	Número Historia de Usuario: HU4
Nombre Tarea: Implementar la funcionalidad para insertar TSU a un grupo docente.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 27 de abril 2009	Fecha Fin: 3 de mayo 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción :	
Es desarrollado el formulario para insertar el TSU al grupo.	

Tabla 2.18 Implementar la funcionalidad para insertar TSU a un grupo docente.

Tarea de Ingeniería	
Número Tarea:1	Número Historia de Usuario: HU5
Nombre Tarea: Implementar la funcionalidad para insertar estudiante que está de limpieza.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 4 de mayo 2009.	Fecha Fin: 10 de mayo 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción :	
Es desarrollado el formulario para insertar la limpieza al estudiante.	

Tabla 2.19 Implementar la funcionalidad para insertar estudiante que está de limpieza.

Tareas asignadas al Sprint 3

Tarea de Ingeniería	
Número Tarea:1	Número Historia de Usuario: HU6
Nombre Tarea: Implementar la funcionalidad para eliminar estudiante.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1

Fecha Inicio: 11 de mayo 2009.	Fecha Fin: 17 de mayo 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción : Es desarrollado el formulario para eliminar al estudiante insertado previamente.	

Tabla 2.20 Implementar la funcionalidad para eliminar estudiante.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: HU7
Nombre Tarea: Implementar la funcionalidad para insertar evaluación.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.42
Fecha Inicio: 18 de mayo 2009.	Fecha Fin: 21 de mayo 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción : Es desarrollado el formulario para insertar la evaluación del estudiante.	

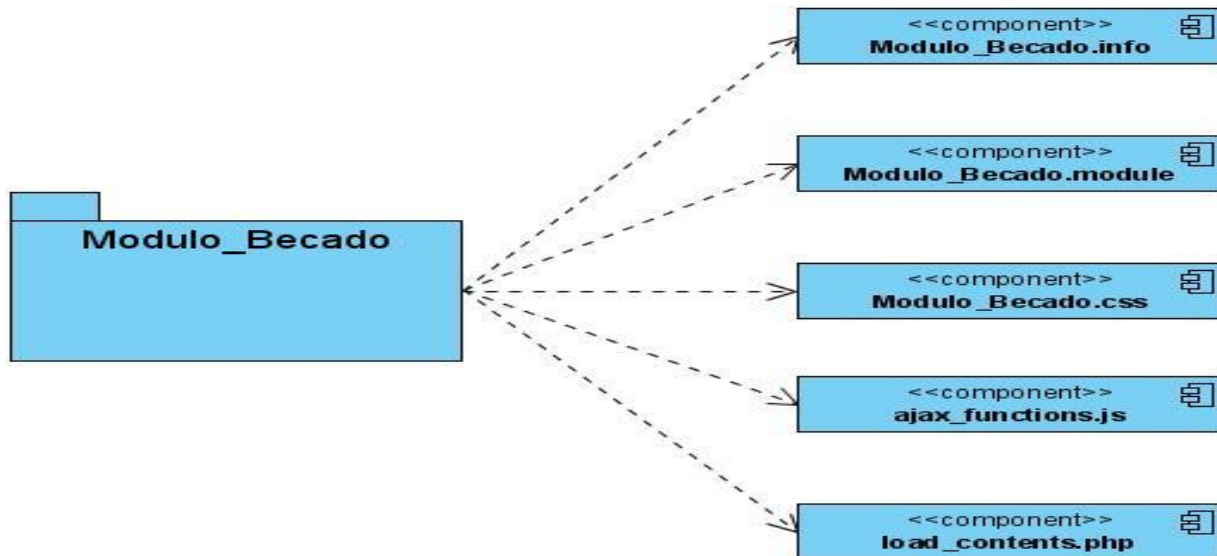
Tabla 2.21 Implementar la funcionalidad para insertar evaluación.

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: HU7
Nombre Tarea: Implementar la funcionalidad para mostrar evaluación.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.56
Fecha Inicio: 22 de mayo 2009.	Fecha Fin: 24 de mayo 2009.
Programadores Responsables: Jorge F. Duque, Dannys Cristobal.	
Descripción : Es desarrollado el formulario para mostrar la evaluación del estudiante.	

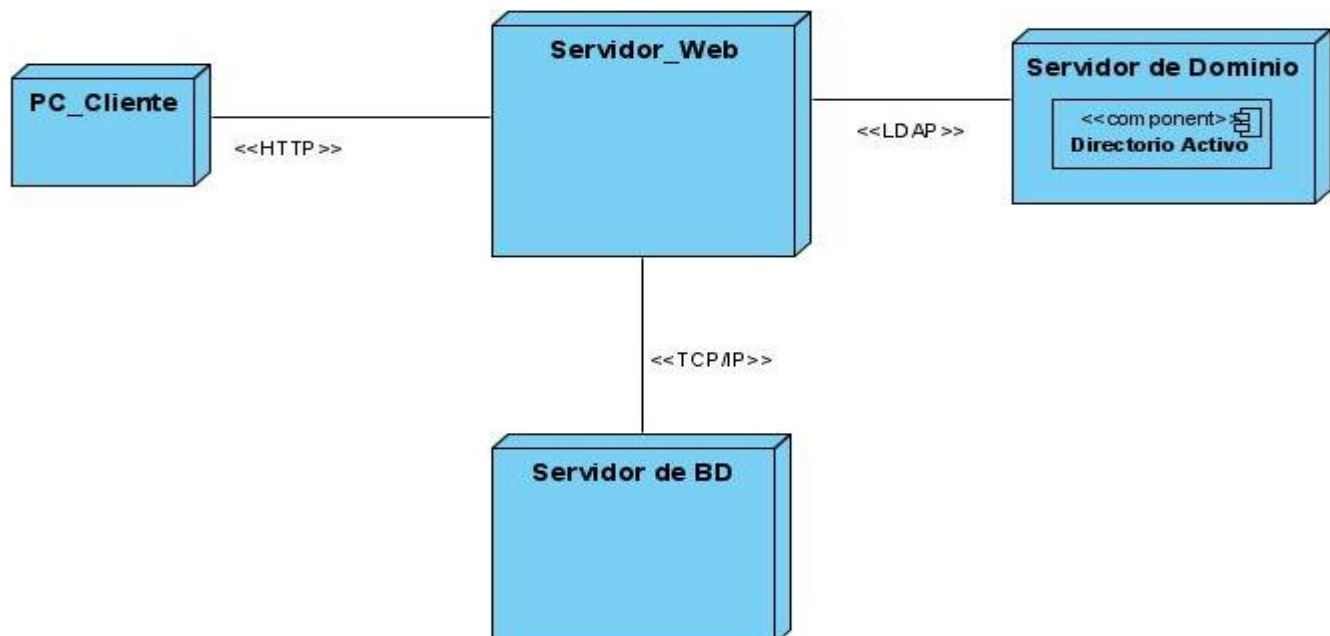
Tabla 2.22 Implementar la funcionalidad para insertar evaluación.

Nota: Un día se tomó como 0.14 unidades aproximadamente.

2.4.1 Diagrama de Componentes.



2.4.2 Diagrama de Despliegue.



2.5 Beneficios de la implementación del Módulo de Becado.

Antes de la implementación del Módulo de Becado se hacía muy difícil la gestión de los procesos de las actividades del estudiante, puesto que todos estos procesos se hacían manualmente y demasiado lento a la hora de consultar algún tipo de información relacionada con la beca, además del aumento en la utilización de recursos materiales. Ahora es mucho más ágil, se cuenta con un sistema automatizado que reduce el tiempo y costo para el control de la información, y es más seguro, aparte de brindar la posibilidad de obtener reportes de los datos gestionados de forma inmediata y reduce el gasto de material considerablemente.

2.6 Conclusiones del capítulo.

En este capítulo se trató el tema de la implementación del módulo. Para el desarrollo del mismo se hizo necesario realizar un análisis de las técnicas de programación refiriéndose exactamente a los módulos y hooks de Drupal.

Se valoró y criticó el diseño propuesto por el analista señalando la nueva metodología a utilizar y se especificaron los nuevos artefactos generados por la misma.

Capítulo 3. Validación de la Solución Propuesta

3.1 Introducción.

Una de las últimas fases del ciclo de vida antes de entregar un programa para su explotación, es la fase de pruebas.

En este capítulo se lleva a cabo la validación de la solución propuesta, mediante las pruebas de unidades. Se realiza la descripción de las pruebas y se evalúan los resultados negativos obtenidos en las mismas para luego corregirlos. Dentro de estas pruebas, se utilizan los métodos de pruebas unitarias y de aceptación, definiéndose el objetivo de cada una de estas.

3.2-Pruebas. Métodos de Pruebas.

Las pruebas son acciones en las cuales el sistema es ejecutado bajo condiciones o requerimientos determinados. Los resultados son chequeados y registrados. Las pruebas verifican los resultados de la implementación del sistema. El modelo de prueba indica cómo han de ser probados aspectos determinados del producto o software.

Hay multitud de conceptos asociadas a las tareas de prueba. Clasificarlas es difícil, pues no son mutuamente disjuntas, sino muy entrelazadas. En lo que sigue se intentará una pequeña organización de acuerdo a sus características.

3.2.1 Fases de prueba:

- UNITARIAS.
- ACEPTACIÓN.

Pruebas unitarias en la metodología Scrum-XP.

Las pruebas unitarias (unit testing) son aquellas pruebas que se les hace a los métodos o procedimientos los cuales deben ser unidades estructurales del programa encargados de una tarea específica. Se

caracterizan por realizarse individualmente, o sea, una prueba hecha a una función es independiente a la realizada a la otra. Se efectúan antes de generar el código y son desarrolladas por los programadores. Se dice que son uno de los métodos de prueba más utilizados en la implementación de software.

Pruebas de aceptación en la metodología Scrum-XP.

Las pruebas de aceptación se crean a partir de las HU. El cliente define los escenarios de prueba para verificar si la HU ha sido correctamente implementada. Una HU puede tener una o varias pruebas de aceptación. El cliente es responsable de verificar el pasaje de las pruebas de aceptación y priorizar la corrección de las pruebas fallidas. Las pruebas de aceptación son pruebas tipo caja negra a nivel del sistema: cada prueba de aceptación corresponde a un resultado producido por el sistema. Las pruebas de aceptación deben ser automáticas, correrse frecuentemente, publicarse sus resultados y programarse su corrección para la próxima iteración. Deben crearse pruebas de aceptación en cada iteración. Si no hay pruebas de aceptación nuevas no se ha hecho nada nuevo. Un relato de usuario no está completo hasta no haber pasado todas sus pruebas de aceptación.

El principal objetivo de estas pruebas es comprobar que las funcionalidades de un sistema requeridas por el cliente se cumplan correctamente.

Caso de Prueba de Aceptación	
Código Caso de Prueba: I-HU1-P1	Nombre Historia de Usuario: Gestionar los datos del estudiante.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad insertar estudiante.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción.	
Entrada/Pasos de ejecución: El usuario inserta los datos necesarios y se ejecuta la acción Aceptar.	

Resultado Esperado: Los estudiantes son insertados correctamente y se muestra un mensaje de aviso.
Evaluación de la Prueba: Satisfactoria.

Tabla 3.1 Prueba para la funcionalidad insertar estudiante.

Caso de Prueba de Aceptación	
Código Caso de Prueba: I-HU1-P2	Nombre Historia de Usuario: Gestionar los datos del estudiante.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad modificar estudiante.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción. El estudiante debe existir en la BD.	
Entrada/Pasos de ejecución: El usuario modifica los datos necesarios y se ejecuta la acción Aceptar.	
Resultado Esperado: Los datos del estudiante son modificados correctamente y se muestra un mensaje de aviso.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3.2 Prueba para la funcionalidad modificar estudiante.

Caso de Prueba de Aceptación	
Código Caso de Prueba: I-HU1-P3	Nombre Historia de Usuario: Gestionar los datos del estudiante.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad eliminar estudiante.	

Condiciones de Ejecución:
El usuario debe estar autenticado.
El usuario debe tener los permisos necesarios para realizar esta acción.
El estudiante debe existir en la BD.
Entrada/Pasos de ejecución:
El usuario busca al estudiante y se ejecuta la acción Eliminar.
Resultado Esperado: El estudiante es eliminado correctamente y es mostrado un mensaje de aviso.
Evaluación de la Prueba: Satisfactoria.

Tabla 3.3 Prueba para la funcionalidad eliminar estudiante.

Caso de Prueba de Aceptación	
Código Caso de Prueba: I-HU2-P1	Nombre Historia de Usuario: Controlar cuartelería.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad insertar cuartelero.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción. El estudiante debe existir en la BD.	
Entrada/Pasos de ejecución: El usuario modifica los datos necesarios y se ejecuta la acción Aceptar.	
Resultado Esperado: El cuartelero es insertado correctamente y es mostrado un mensaje de aviso.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3.4 Prueba para la funcionalidad insertar cuartelero.

Caso de Prueba de Aceptación				
Código Caso de Prueba: I-HU2-P2	Nombre	Historia	de	Usuario: Controlar cuartelería.
Nombre de la persona que realiza la prueba: Dannys Cristobal				
Descripción de la Prueba: Prueba para la funcionalidad cambiar cuartelería.				
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción. El estudiante debe existir en la BD.				
Entrada/Pasos de ejecución: El usuario modifica los datos necesarios y se ejecuta la acción Aceptar.				
Resultado Esperado: Los cuarteros son cambiados correctamente y es mostrado un mensaje de aviso.				
Evaluación de la Prueba: Satisfactoria.				

Tabla 3.5 Prueba para la funcionalidad cambiar cuartelería.

Caso de Prueba de Aceptación				
Código Caso de Prueba: I-HU2-P3	Nombre	Historia	de	Usuario: Controlar cuartelería.
Nombre de la persona que realiza la prueba: Dannys Cristobal				
Descripción de la Prueba: Prueba para la funcionalidad correr cuartelería.				
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción. El estudiante debe existir en la BD.				
Entrada/Pasos de ejecución: El usuario modifica los datos necesarios y se ejecuta la acción Aceptar.				
Resultado Esperado: A los estudiantes se les corre la cuartelería un día correctamente y es mostrado				

un mensaje de aviso.
Evaluación de la Prueba: Satisfactoria.

Tabla 3.6 Prueba para la funcionalidad correr cuartelería.

Caso de Prueba de Aceptación	
Código Caso de Prueba: I-HU3-P1	Nombre Historia de Usuario: Controlar guardia.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad insertar guardia.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción. El grupo debe existir en la BD.	
Entrada/Pasos de ejecución: El usuario inserta los datos necesarios especificando el grupo docente y se ejecuta la acción Aceptar.	
Resultado Esperado: El grupo es insertado correctamente y se muestra un mensaje de aviso.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3.7 Prueba para la funcionalidad insertar guardia.

Caso de Prueba de Aceptación	
Código Caso de Prueba: I-HU4-P1	Nombre Historia de Usuario: Controlar TSU.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad insertar TSU.	

Condiciones de Ejecución:
El usuario debe estar autenticado.
El usuario debe tener los permisos necesarios para realizar esta acción.
El grupo debe existir en la BD.
Entrada/Pasos de ejecución:
El usuario inserta los datos necesarios especificando el grupo docente y se ejecuta la acción Aceptar.
Resultado Esperado: El grupo es insertado correctamente y se muestra un mensaje de aviso.
Evaluación de la Prueba: Satisfactoria.

Tabla 3.8 Prueba para la funcionalidad insertar TSU.

Caso de Prueba de Aceptación	
Código Caso de Prueba: I-HU5-P1	Nombre Historia de Usuario: Controlar limpieza.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad insertar limpieza.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción. El estudiante debe existir en la BD.	
Entrada/Pasos de ejecución: El usuario inserta los datos necesarios y se ejecuta la acción Aceptar.	
Resultado Esperado: El estudiante que está de limpieza es insertado correctamente y se muestra un mensaje de aviso.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3.9 Prueba para la funcionalidad insertar limpieza.

Caso de Prueba de Aceptación

Código Caso de Prueba: I-HU6-P1	Nombre Historia de Usuario: Generar reporte.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad generar reporte.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción.	
Entrada/Pasos de ejecución: El usuario escoge el parámetro por el cual va a generar el reporte y ejecuta la acción Generar.	
Resultado Esperado: El reporte del estudiante es generado correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3.10 Prueba para la funcionalidad generar reporte.

Caso de Prueba de Aceptación	
Código Caso de Prueba: I-HU7-P1	Nombre Historia de Usuario: Registrar evaluación mensual.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad insertar evaluación mensual.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción. El estudiante debe existir en la BD.	
Entrada/Pasos de ejecución: El usuario inserta la evaluación especificando a qué actividad se refiere y se ejecuta la acción Aceptar.	
Resultado Esperado: La evaluación es insertada correctamente y se muestra un mensaje de aviso.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3.11 Prueba para la funcionalidad insertar evaluación mensual.

Caso de Prueba de Aceptación	
Código Caso de Prueba: I-HU7-P2	Nombre Historia de Usuario: Registrar evaluación mensual.
Nombre de la persona que realiza la prueba: Dannys Cristobal	
Descripción de la Prueba: Prueba para la funcionalidad mostrar evaluación mensual.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener los permisos necesarios para realizar esta acción. El estudiante debe existir en la BD.	
Entrada/Pasos de ejecución: El usuario busca al estudiante y muestra la evaluación del mismo según la actividad realizada.	
Resultado Esperado: La evaluación es mostrada correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3.12 Prueba para la funcionalidad mostrar evaluación mensual.

3.3 Conclusiones del capítulo.

En este capítulo se realizaron las pruebas unitarias y las pruebas de aceptación al concluir la implementación. Se explicó en qué consistían y sus objetivos. Como resultado se obtuvo la documentación de las mismas y un producto aceptable para el cumplimiento de su objetivo inicial.

Conclusiones

Con la culminación de la implementación del Módulo de Becado se cumplieron todos los objetivos trazados. Se logró un módulo que permitiera realizar con mayor comodidad, eficiencia y rapidez el control de las actividades y la gestión de los datos del estudiante en la Residencia Estudiantil de la Facultad 1, ya que los estudios realizados a sistemas similares a nivel Internacional, Nacional y de Universidad no satisfacen la necesidad del problema planteado, y se decidieron qué herramientas utilizar.

Para la implementación del módulo se tuvo en cuenta el análisis y diseño del mismo cambiando así la metodología RUP por Scrum XP y los artefactos generados. Se valoraron las técnicas de programación y algunos algoritmos no triviales en su desarrollo y se caracterizaron las tareas de ingeniería para dar solución a las HU.

Al terminar la implementación se efectuaron al sistema las pruebas unitarias y de aceptación generando su documentación y de esta forma se cumplieron todos los objetivos planteados.

Recomendaciones

Se recomienda:

- Adicionar una funcionalidad que registre el mejor y peor edificio, apto y estudiante en la residencia en cuanto al cumplimiento de todas las actividades relacionadas con la beca.
- Adicionar una funcionalidad que muestre la fecha de las actividades adicionales que se realizan en la Residencia, como Paradas de Beca y Elecciones en la Beca; encuentros de dominó y otras actividades recreativas.
- El análisis del módulo para otras posibles implementaciones en la Universidad referente al control de las actividades y gestión de los datos del estudiante.

Bibliografía Referenciada.

- 1- Alvarez Miguel Angel. Descripción y características de este potente y moderno lenguaje de programación. 18 de julio 2001.[Consultada el 2 de marzo 2009]
- 2- Ayala Jesús Maldonado, Toledo Alma Enríquez, Ortega Yunko Nakamuro, Toledo Goretty Nogueron. Artículo MySQL. [Consultada el 5 de marzo 2009].Disponible en:
<http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>
- 3- Bravo C., Bravo J., Redondo M. A., Ortega M.. Intranet: Soporte para entornos de aprendizaje.16 diciembre 2005. [Consultada el 11 de marzo 2009]. Disponible en:
<http://www.adie.es/iecom/index.php/IECom/article/viewFile/102/96>
- 4- Espinoza Humberto. PostgreSQL. Una Alternativa de DBMS Open Source.2005. [Consultada el 9 de abril 2009].
- 5- Gil Elizabeth. Lenguaje JAVA. [Consultada el 26 de marzo 2009]. Disponible en:
http://es.geocities.com/maxisagil/ihai/contenido_1.htm
- 6- Rodas Hinostroza Raul. Usuarios y Grupos en Linux. [Consultada el 2 de mayo 2009]. Disponible en: www.linuxcentro.net
- 7- Márquez Díaz José, Sampedro Leonardo, Vargas Félix. Instalación y configuración de Apache, un servidor Web gratis. [Consultada el 2 de mayo 2009]. Disponible en:
http://ciruelo.uninorte.edu.co/pdf/ingenieria_desarrollo/12/instalacion_y_configuracion_de_apache.pdf
- 8- Moraga Hernán. Curso PROGRAMCION AGIL: SCRUM Y XP. [Consultada el 23 de abril 2009].
- 9- Sarti Paz Luis Pedro. Perl I primera parte. [Consultada el 2 de mayo 2009].
- 10-Peralta Fernández Sagrario. El servidor Web. Arquitectura y funcionamiento. 07 Julio 2006. [Consultada el 30 de marzo 2009]. Disponible en:
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=366>
- 11-Introducción al Perl. [Consultada el 2 de mayo de 2009]. Disponible en:
- 12-Programación Web. [Consultada el 29 de abril].Disponible en:
<http://www.lenguajes-de-programacion.com/programacion-web.shtml>

- 13- Características de Drupal. Comunidad de usuarios de Drupal, 27 de julio 2005.[Consultada el 6 de marzo 2009].Disponible en: <http://drupal.org/es/caracteristicas>
- 14-Windows XP. Taller. 27 de marzo de 2009. [Consultada 29 de abril 2009]. Disponible en : <http://www.slideshare.net/angelssephiroth2005/taller-win-xp>
- 15- Curso de Windows XP. [Consultada el 30 de marzo 2009].Disponible en: <http://www.adrformacion.com/cursos/winxp/leccion1/tutorial1.html>
- 16-Bokly Plana Wendy Rafael. Trabajo de Diploma. Análisis y diseño del Módulo Administración del Becado. Ciudad Habana: s.n., junio 2008.
- 17-Sitio de descargas de software. Visual Paradigm International Ltd. [Consultado el 28 de abril 2009]. Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)

Bibliografía

- 1- Ingeniería de software 1. Scribd. [En línea] [Citado el: 7 de abril de 2009.]
<http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>
- 2- Gracia Joaquin. Patrones de diseño. Diseño de software orientado a objetos. [En línea] 27 de mayo de 2005. [Citado el: 29 de abril de 2009.]
<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
- 3- Kniberg Henrik. Scrum y XP desde las trincheras. 2007.
- 4- Coding standards. Drupal. [En línea] [Citado el: 11 de mayo de 2009.]. Disponible en:
<http://drupal.org/coding-standards>
- 5- Hook. Drupal. [En línea] <http://api.drupal.org/api/group/hooks>
- 6- Letelier Patricio. Pruebas de Aceptación como conductor del Proceso Software. [En línea] 2007.
<http://zweb.iti.upv.es/groups/squac/events/JTS2007/slides/3demayo16.45-PatricioLetelier.pdf/attach/3demayo16.45-PatricioLetelier.pdf>.
- 7- Gracia Joaquin. Patrones de diseño. Diseño de Software Orientado a Objetos. IngenieroSoftware. [En línea] 21 de abril de 2005. [Citado el: 5 de mayo de 2009.]
<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
- 8- Rojas Johanna; Barrios Emilio; Grupo ARQUISOFT. Métodos de prueba de caja negra. [En línea] 2007. [Citado el: 1 de mayo de 2009.]
<http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node28.html>
- 9- Bartle Phil. Información para la gestión y gestión de la información. Potenciación comunitaria. [En línea] <http://www.scn.org/mpfc/modules/mon-miss.htm>.
- 10- Cascante Patricia y Fonseca Mima. Intranet: Una ventana a la comunicacion interna de las organizaciones. [En línea] 2006. <http://www.uned.ac.cr/Redti/cuarta/art2.pdf>.

- 11- Técnicas de programación. Autorneto. [En línea] mayo de 2008.
<http://autorneto.com/tecnologia/software/tecnicas-de-programacion/>.
- 12- Rego Castillo Danis y Chaveco Laurencio Alfonso. Desarrollo de una intranet para el grupo empresarial de transporte por omnibus. Ciudad Habana : s.n., 2007.
- 13- Blanco Cuaresma Sergi. Metodología de desarrollo. Marble Station. [En línea]
<http://www.marblestation.com/?p=644>.
- 14- Canós José H., Letelier Patricio y Penadés M. Carmen. Metodologías Ágiles en el Desarrollo de Software. [En línea] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
- 15- Fases de la Programación Extrema. [En línea] <http://programacionextrema.tripod.com/fases.htm>.
- 16- Escribano, Fernández. Ingeniería de software II. Introduccion a Extreme Programming. [En línea]
<http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
- 17- Rodríguez Villar, Malay. de Diploma Introducción de procedimientos ágiles en la producción de software en la facultad 7 de la Universidad de las Ciencias Informáticas. Ciudad Habana : s.n., 2007.
- 18- Dondo, Agustín. ¿Por qué elegir PHP? PHP en Castellano. [En línea]
<http://www.programacion.com/php/articulo/porquephp/>.
- 19- Programacion en PHP. <http://www.desarrolloweb.com/manuales/12/>. [En línea]
<http://www.desarrolloweb.com/manuales/12/>.
- 20- The Features, Mission, and Principles of the Drupal. Drupal. [En línea] 2008.
<http://drupal.org/getting-started/project-features>.
- 21- Alarcón Medina, José Manuel. Administración SGBD PostgreSQL. Valencia : s.n., 2006.
- 22- Gómez, Carlos. PostgreSQL SGBD de código abierto. TechHerald.com. [En línea] 2008.
<http://techerald.com/view.jsp?postgresql-301220081345.html>

- 23- ¿Qué es un servidor web (Web Servers)? - Definición de servidor web. masadelante.com. [En línea] 2008. <http://www.masadelante.com/faq-servidor.htm>
- 24- Apache. CAMPUS VIRTUAL. [En línea] <http://www.cvirtualuees.edu.sv/mod/book/view.php?id=552>
- 25- Pressman, Roger S. Ingenieria del Software. un enfoque práctico. s.l. : quinta edición.

Glosario de Términos

Cuartelería: Guardia que realiza el estudiante en el edificio donde reside.

Cuartelero: Estudiante que está de cuartelería ese día.

TSU: Conjunto de actividades que realizan los estudiantes como apoyo a diferentes áreas de la facultad como: el docente, comedor, y áreas verdes de la residencia.

Usuario: Persona encargada de utilizar el sistema, obteniendo algún beneficio.

Cliente: Organización, persona o grupo de personas que solicitan la construcción de un software.

HTTP: Es un sistema de comunicación que permite la visualización de páginas Web, desde un navegador.

Apto: Apartamento donde residen los estudiantes.

TCP/IP: Son las siglas de Protocolo de Control de Transmisión/Protocolo de Internet (en inglés Transmission Control Protocol/Internet Protocol).

Software: Es un programa de computador.

Renderizado: Es la acción de asignar y calcular todas las propiedades de un objeto antes de mostrarlo en pantalla.

Algoritmos: Secuencia de código para el desarrollo de un software.

Bucle: En programación se refiere a ciclos dentro de un método, o sea, una serie de instrucciones que se repiten.

Anexos 1

Gestionar Estudiante.



Módulo Becado

...Por una Residencia mejor...

Principal

Gestionar Estudiante

Permite la Gestión de los siguientes datos del Estudiante: (nombre y apellidos, carnet de identidad, grupo, año, foto, número de solapín, dirección particular y teléfono, edificio, apto, Militante (UJC y/o PCC), Dirigente (FEU o UJC), teléfono de apto, proyecto en el que participa, laboratorio en el que trabaja, teléfono del laboratorio, si es alumno ayudante y asignatura que imparte en caso de ser alumno ayudante.

Se inicia cuando el USUARIO invoca el servicio BUSCAR ESTUDIANTE pasándole el solapín. El sistema muestra los datos del estudiante. En caso de querer Insertar Datos, el USUARIO debe invocar el servicio INSERTAR ESTUDIANTE, en caso de Eliminar o Actualizar Datos invoca el servicio ELIMINAR ESTUDIANTE y MODIFICAR ESTUDIANTE, verificando previamente que el estudiante se encuentre en el sistema, y dados estos datos actualiza los datos o elimina.

CopyRigth® Facult@d 1.

Powered by [Drupal](#) - Design by [artinet](#)

Editar enlaces primarios

administrador

- ▶ Buscar estudiante
- ▶ Controlar Cuarteleria
- ▶ Controlar evaluacion mensual
- ▶ Controlar Guardia
- ▶ Controlar limpieza
- ▶ Controlar Trab. Social. U.
- ▶ Mi cuenta
- ▼ Gestionar Estudiante
 - ▶ Buscar Estudiante
 - ▶ Eliminar Estudiante
 - ▶ Insertar Estudiante
 - ▶ Modificar Estudiante
- ▶ Administrar
- ▶ Cerrar sesión

Anexo 2

Controlar Cuartelería



Módulo Becado

...Por una Residencia mejor...

Principal

Controlar Cuartelería

Permite la Gestión de la Cuartelería Estudiantil.

Se inicia cuando el USUARIO desee insertar los cuartereros, Invoca el Servicio INSERTAR CUARTELERO almacenado los datos en la base de datos, en caso de que el USUARIO desee cambiar la cuartelería, invoca el servicio CAMBIAR CUARTELERÍA. El sistema permite hacer cambios entre dos estudiantes. En caso de que el USUARIO desee correr la cuartelería, invoca el servicio CORRER CUARTELERÍA, el sistema adelanta un día la cuartelería. En caso de insertar alguna incidencia o la evaluación del cuarterero se invoca el servicio EVALUAR CUARTELERO para insertar la evaluación y las incidencias de su cuartelería. El sistema almacena los datos en la base de datos. En caso que el estudiante desee obtener la evaluación, invoca el servicio LISTADO CUARTELEROS DEL MES, donde obtendra un reporte de las incidencias y evaluaciones de los mismos .

CopyRigth® Facult@d 1.

Powered by [Drupal](#) - Design by [artinet](#)

Editar enlaces primarios

Anexo 3

Controlar Guardia



Módulo Becado

...Por una Residencia mejor...

Principal

Controlar Guardia

Permite Gestionar la Guardia Estudiantil, ya sea para controlar la asistencia a la misma ó mostrar las incidencias de cada grupo de guardia.

Se inicia cuando el USUARIO invoca el servicio INSERTAR GRUPO A LA GUARDIA. En caso de que el USUARIO desee insertar la asistencia, invoca el servicio TOMAR ASISTENCIA de la guardia. El sistema almacena a los estudiantes que participaron en la guardia. En caso de que el USUARIO desee obtener la asistencia de la guardia, invoca el servicio MOSTRAR ASISTENCIA de la guardia. El sistema muestra a los estudiantes que participaron en la guardia.

[Editar enlaces primarios](#)

administrador

- ▶ Buscar estudiante
- ▶ Controlar Cuarteleria
- ▶ Controlar evaluacion mensual
- ▼ Controlar Guardia
 - Grupos de guardia
 - Insertar grupo a la guardia
 - Mostrar asistencia
 - Tomar asistencia
- ▶ Controlar limpieza
- ▶ Controlar Trab. Social. U.
- Mi cuenta
- ▶ Gestionar Estudiante
- ▶ Administrar
- Cerrar sesión

CopyRigth® *Facult@d* 1.

Powered by [Drupal](#) - Design by [artinet](#)

Anexo 4

Controlar Trab. Social. U.



Módulo Becado

...Por una Residencia mejor...

Principal

Controlar Trab. Social. U.

Permite el control de los Datos del Trabajo Socialmente Útil.

El caso de uso se inicia cuando el USUARIO invoca el servicio INSERTAR GRUPO AL TSU. En caso de que el USUARIO desee insertar la asistencia, invoca el servicio TOMAR ASISTENCIA. El sistema almacena a los estudiantes que participaron en el TSU. En caso de que el USUARIO desee obtener la asistencia del TSU, invoca el servicio MOSTRAR ASISTENCIA. El sistema muestra a los estudiantes que participaron en el TSU. Si el USUARIO desea visualizar los grupos que están de TSU debe invocar el servicio MOSTRAR GRUPOS DE TSU.

Editar enlaces primarios

administrador

- ▶ Buscar estudiante
- ▶ Controlar Cuartelería
- ▶ Controlar evaluación mensual
- ▶ Controlar guardia
- ▶ Controlar limpieza
- ▼ Controlar Trab. Social. U.
 - Insertar grupo al TSU
 - Mostrar asistencia
 - Mostrar grupos de TSU
 - Tomar asistencia
- Mi cuenta
- ▶ Gestionar Estudiante
- ▶ Administrar
- Cerrar sesión

CopyRigth® *Facult@d* 1.

Powered by Drupal - Design by artinet

Anexo 5

Controlar limpieza



Módulo Becado

...Por una Residencia mejor...

Principal

Controlar limpieza

Permite registrar las incidencias de cada estudiante con respecto a la limpieza de su apto.

Se inicia cuando el USUARIO desee controlar la limpieza del apto. En caso de que el USUARIO desee insertar alguna incidencia, invoca el servicio INSERTAR INCIDENCIA. El sistema inserta las incidencias en la base de datos. En caso de que el actor desee eliminar alguna incidencia, invoca el servicio MOSTRAR INCIDENCIAS en donde tendrá un reporte de las incidencias y la opción de eliminarlas en caso de que haya alguna equivocación. El sistema elimina la incidencia de la base de datos.

CopyRigth® Facult@d 1.

Powered by [Drupal](#) - Design by [artinet](#)

Editar enlaces primarios

administrador

- ▶ Buscar estudiante
- ▶ Controlar Cuartelería
- ▶ Controlar evaluación mensual
- ▶ Controlar Guardia
- ▼ Controlar limpieza
 - Insertar Incidencia
 - Mostrar Incidencia
- ▶ Controlar Trab. Social. U.
- Mi cuenta
- ▶ Gestionar Estudiante
- ▶ Administrar
- Cerrar sesión

Anexo 6

Reporte del Estudiante



Módulo Becado

...Por una Residencia mejor...

Principal

Reporte del Estudiante

Permite la búsqueda de Estudiantes por Categoría , obtener reportes evaluativos.

Se inicia cuando el USUARIO desee obtener algún REPORTE DEL ESTUDIANTE. En caso de que el USUARIO desee obtener estudiantes POR CATEGORÍA, invoca el servicio mostrar estudiante POR CATEGORÍA. El sistema muestra un listado de estudiantes según la categoría. En caso de que desee obtener los estudiantes POR APARTAMENTOS o POR EDIFICIO, invoca el servicio obtener estudiantes, pasándole el apartamento o el edificio. En caso de que el actor desee obtener los estudiantes POR AÑO, invoca el servicio mostrar estudiantes POR AÑO. El sistema muestra a los estudiantes.

Editar enlaces primarios

administrador

- ▶ Controlar Cuarteria
- ▶ Controlar evaluacion mensual
- ▶ Controlar Guardia
- ▶ Controlar limpieza
- ▶ Controlar Trab. Social. U.
- Mi cuenta
- ▶ Gestionar Estudiante
- ▼ Reporte del Estudiante
 - Por apto
 - Por categoria
 - Por Curso
 - Por edificio
- ▶ Administrar
- Cerrar sesión

CopyRigth® *Facult@d* 1.

Powered by [Drupal](#) - Design by [artinet](#)

Anexo 7

Controlar evaluación mensual



Módulo Becado

...Por una Residencia mejor...

Principal

Controlar evaluacion mensual

Permite Gestionar la evaluacion mensual de los estudiantes según sus resultados en cuanto a limpieza,guardia,cuartelería y tsu.

Se inicia cuando el USUARIO invoca el servicio INSERTAR EVALUACIÓN mensual, el sistema almacena la evaluación. En caso de que el USUARIO desee obtener la evaluación mensual de algún estudiante, invoca le servicio MOSTRAR EVALUACIÓN mensual, el sistema mostrara un reporte de los estudiantes evaluados por meses.

[Editar enlaces primarios](#)

administrador

- ▶ Controlar Cuarteleria
- ▼ Controlar evaluacion mensual
 - ▶ Insertar evaluacion
 - ▶ Mostrar evaluacion
- ▶ Controlar Guardia
- ▶ Controlar limpieza
- ▶ Controlar Trab. Social. U.
- ▶ Mi cuenta
- ▶ Gestionar Estudiante
- ▶ Reporte del Estudiante
- ▶ Administrar
- ▶ Cerrar sesión

CopyRigth® *Facult@d 1.*

Powered by [Drupal](#) - Design by [artinet](#)