



“Facultad 5”

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

**Sistema para la gestión de matrícula de cursos
optativos y electivos de la facultad 5 de la
Universidad de las Ciencias Informáticas.**

Autor: Yuleisy Galloso Soler.

Tutora: Ing. Yadira Ramírez Rodríguez.

La Habana, 2014.

Año 55 del Triunfo de la Revolución.

DATOS DE CONTACTO

Tutora: Ing. Yadira Ramírez Rodríguez, graduada en el año 2007 de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesora con categoría docente asistente, jefa de colectivo de ingeniería de software y asesora de relaciones internacionales de la facultad 5.

Correo electrónico: yramirez@uci.cu

Agradecimientos

A mi mamá y a mi papa, que han sido fieles acompañantes de mi causa, sin dejar de alentarme, felicitarme o requerirme en los momentos precisos y por acompañarme siempre que los he necesitado.

A mi novio Rudelquis porque sin su aliento, comprensión y amor no hubiera sido posible llegar hasta aquí.

A mi tutora Yadira y profesora de la carrera, muchas gracias por todo lo que hizo por mí y por la ayuda que me brindo.

A Marin y a Dagmar dos amigos que les voy a agradecer toda la vida todo lo que hicieron por mí.

A mis profesores de la carrera como Zoraida, Villar, Julián, Zenaida, Yadira, Olga, Cacha a todos en general muchas gracias por extenderme la mano y ser eternamente tan excelentes profesores.

A mis amigos Yusiél y Yindra por estar ahí en los momentos malos y buenos.

Dedicatoria

A mi mamá y a mi papa porque sin su apoyo y comprensión no hubiera sido posible llegar hasta aquí.

A mi novio Rudelquis por su amor incondicional.

A Marin por estar siempre ahí en los momentos malos y buenos.

Declaración de autoría

Declaro ser autora del presente Trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

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

Firma del Autor

Yuleisy Galloso Soler

Firma del Tutor

Ing. Yadira Ramírez Rodríguez

Resumen

La gestión de la información, dígase almacenaje, proceso, control y disponibilidad, ha devenido en una ocupación de interés para el desarrollo de la sociedad mundial. Lo cual se ha convertido en un proceso de perfeccionamiento constante, para llevar a cabo el control, la planificación, la dirección y la toma de decisiones, contribuyendo esto al buen desarrollo y funcionamiento de las entidades en general. En la Universidad de las Ciencias Informáticas (UCI), se hace necesaria la aplicación de los conceptos expuestos anteriormente, pues en los vicedecanatos de formación que existen en las diferentes facultades de la universidad, se maneja un volumen de información considerable, entre los cuales está el referente a la matrícula de los estudiantes para cursos optativos y electivos. El tratamiento de este proceso se realiza manualmente mediante el tabulador electrónico Microsoft Excel, lo que conlleva a errores que inciden en el almacenamiento, control, prontitud, seguridad y disponibilidad de la información. A esto se adiciona la misión por parte del vicedecano de formación, del envío a profesores y estudiantes, de toda la información inherente a los cursos, a través del correo electrónico. A partir de la situación planteada anteriormente y las consideraciones basadas en las gestiones actuales del proceso de matrícula, el trabajo de tesis que se propone, permite la captación automática descentralizada de datos, evitando largas filas de esperas e impidiendo el envío y reenvío de correos de dicha información, además del acceso a la información por parte de los profesores. Finalmente, con la implantación del sistema que se propone, será posible obtener reportes a partir de toda la información almacenada, lo que permitirá a todos los usuarios del sistema conocer cuantitativa y cualitativamente todos los atributos del fondo informativo.

Palabras claves: gestión de la información, cursos optativos, cursos electivos.

Índice de contenidos

| | |
|--|-----------|
| Introducción | 14 |
| Capítulo 1: Fundamentación Teórica | 19 |
| 1.1 Introducción..... | 19 |
| 1.2 Cursos Optativos y Electivos..... | 19 |
| 1.3 Gestión de la información..... | 20 |
| 1.4 Sistemas de Gestión de la Información | 20 |
| 1.5 Análisis de soluciones existentes en Cuba y el mundo..... | 21 |
| 1.5.1 Soluciones existentes a nivel internacional..... | 21 |
| 1.5.2 Soluciones existentes a nivel nacional..... | 22 |
| 1.6 Aplicaciones Web..... | 24 |
| 1.7 Descripción de las tecnologías y herramientas | 24 |
| 1.7.1 Entorno de Desarrollo Integrado (IDE)..... | 25 |
| 1.7.2 Lenguaje de Programación | 25 |
| 1.7.2.1 Selección del lenguaje de programación..... | 26 |
| 1.7.3 Servidores de Aplicaciones y Contenedores Web | 27 |
| 1.7.3.1 Selección del entorno de ejecución..... | 28 |
| 1.7.4 Gestores de Base de Datos..... | 28 |
| 1.7.4.1 Selección del Gestor de Base de Datos | 30 |
| 1.7.5 Herramienta de administración de Bases de Datos PgAdmin III | 30 |
| 1.7.6 Framework Vaadin | 30 |
| 1.8 Metodología de Desarrollo de Software..... | 31 |
| 1.8.1 Selección de la metodología de desarrollo de software..... | 33 |
| 1.9 Visual Paradigm for UML 8.0..... | 33 |
| 1.10 Consideraciones parciales | 33 |
| CAPÍTULO 2: Propuesta de Solución | 34 |
| 2.1 Introducción..... | 34 |
| 2.2 Reglas de la propuesta de solución | 34 |
| 2.3 Propuesta de solución | 34 |
| 2.4 Roles del sistema | 35 |
| 2.5 Historias de usuario..... | 35 |

| | | |
|---|---|-----------|
| 2.6 | Estimación de esfuerzo por historia de usuario..... | 44 |
| 2.7 | Plan de entregas..... | 45 |
| 2.8 | Arquitectura propuesta | 46 |
| 2.9 | Patrones de diseño | 47 |
| 2.10 | Tarjetas Clases-Responsabilidades-Colaboradores(CRC) | 47 |
| 2.11 | Modelo de la base de datos..... | 52 |
| 2.11.1 | Diagrama entidad relación de la Base de Datos | 53 |
| 2.12 | Consideraciones parciales | 53 |
| CAPÍTULO 3: Validación de la Propuesta de Solución | | 54 |
| 3.1 | Introducción..... | 54 |
| 3.2 | Iteraciones | 54 |
| 3.2.1 | Primera iteración..... | 54 |
| 3.2.1.1 | Descripción de las tareas de ingeniería de la primera iteración | 55 |
| 3.2.1.2 | Pruebas de aceptación de la primera iteración | 60 |
| 3.2.2 | Segunda Iteración | 64 |
| 3.2.2.1 | Descripción de las tareas de ingeniería de la segunda iteración..... | 65 |
| 3.2.2.2 | Pruebas de aceptación de la segunda iteración..... | 69 |
| 3.2.3 | Tercera iteración..... | 72 |
| 3.2.3.1 | Descripción de las tareas de ingeniería de la tercera iteración | 73 |
| 3.2.3.2 | Pruebas de aceptación de la tercera iteración..... | 74 |
| 3.3 | Resumen del resultado de las iteraciones..... | 77 |
| 3.4 | Conclusiones parciales..... | 78 |
| Conclusiones generales..... | | 79 |
| Recomendaciones | | 80 |
| Referencias Bibliográficas | | 81 |
| Bibliografía..... | | 84 |
| Glosario de Términos | | 85 |
| Anexos..... | | 89 |

Índice de Ilustraciones

| | |
|--|----|
| Ilustración 1 Arquitectura en capas del sistema. | 47 |
| Ilustración 2 Diagrama entidad relación de la Base de Datos | 53 |
| Ilustración 3 Resultado de las pruebas realizadas. | 78 |
| Ilustración 4 Autenticar Usuario | 95 |
| Ilustración 5 Adicionar Curso | 96 |
| Ilustración 6 Modificar Curso | 96 |
| Ilustración 7 Eliminar Curso 1 | 97 |
| Ilustración 8 Eliminar Curso 2 | 97 |
| Ilustración 9 Matricular curso | 97 |
| Ilustración 10 Listado de matricula | 97 |

Índice de Tabla

| | |
|---|----|
| Tabla 1 Roles del sistema | 35 |
| Tabla 2 Historia de usuario No1 | 36 |
| Tabla 3 Historia de usuario No2 | 36 |
| Tabla 4 Historia de usuario No3 | 36 |
| Tabla 5 Historia de usuario No4 | 37 |
| Tabla 6 Historia de usuario No5 | 38 |
| Tabla 7 Historia de usuario No6 | 38 |
| Tabla 8 Historia de usuario No7 | 39 |
| Tabla 9 Historia de usuario No8 | 39 |
| Tabla 10 Historia de usuario No9 | 40 |
| Tabla 11 Historia de usuario No10 | 40 |
| Tabla 12 Historia de usuario No11 | 41 |
| Tabla 13 Historia de usuario No12 | 41 |
| Tabla 14 Historia de usuario No13 | 42 |
| Tabla 15 Historia de usuario No14 | 42 |
| Tabla 16 Historia de usuario No15 | 42 |
| Tabla 17 Historia de usuario No16 | 43 |
| Tabla 18 Historia de usuario No17 | 43 |
| Tabla 19 Historia de usuario No18 | 43 |
| Tabla 20 Historia de usuario No19 | 44 |
| Tabla 21 Estimación de esfuerzo por historia de usuario | 44 |
| Tabla 22 Plan de Entregas | 45 |
| Tabla 23 Tarjeta CRC 1 | 48 |
| Tabla 24 Tarjeta CRC 2 | 48 |
| Tabla 25 Tarjeta CRC 3 | 48 |
| Tabla 26 Tarjeta CRC 4 | 49 |
| Tabla 27 Tarjeta CRC 5 | 49 |
| Tabla 28 Tarjeta CRC 6 | 49 |
| Tabla 29 Tarjeta CRC 7 | 50 |
| Tabla 30 Tarjeta CRC 8 | 50 |
| Tabla 31 Tarjeta CRC 9 | 50 |
| Tabla 32 Tarjeta CRC10 | 51 |
| Tabla 33 Tarjeta CRC11 | 51 |
| Tabla 34 Tarjeta CRC12 | 51 |
| Tabla 35 Tarjeta CRC13 | 52 |
| Tabla 36 Tarjeta CRC14 | 52 |
| Tabla 37 Planificación de las Tareas de Ingeniería de la primera iteración. | 54 |
| Tabla 38 Diseño del prototipo de la interfaz principal de la aplicación para autenticar usuario. | 55 |
| Tabla 39 Implementar código para autenticar usuario. | 55 |

| | |
|--|-----------|
| Tabla 40 Implementación de código para la asignación de permisos a cada usuario según su rol..... | 56 |
| Tabla 41 Implementación de la clase FabricaCampos..... | 56 |
| Tabla 42 Iniciar la implementación de la clase Ventana..... | 57 |
| Tabla 43 Implementación de la clase UploadTest..... | 57 |
| Tabla 44 Implementación del método adicionar de la clase ContenedorVertical. | 58 |
| Tabla 45 Culminar la implementación de la clase Ventana. | 58 |
| Tabla 46 Implementación del método modificar de la clase ContenedorVertical. | 59 |
| Tabla 47 Implementación del método eliminar de la clase ContenedorVertical. .. | 59 |
| Tabla 48 Probar la funcionalidad Autenticar usuario..... | 60 |
| Tabla 49 Probar la funcionalidad Asignar permisos. | 61 |
| Tabla 50 Probar la funcionalidad Adicionar curso..... | 61 |
| Tabla 51 Probar la funcionalidad Modificar curso..... | 62 |
| Tabla 52 Probar la funcionalidad Eliminar curso..... | 63 |
| Tabla 53 Planificación de las Tareas de Ingeniería de la segunda iteración..... | 64 |
| Tabla 54 Implementación del método Matricular de la clase ContenedorVerticalE3. | 65 |
| Tabla 55 Implementación del método Matricular de la clase ContenedorVerticalE4. | 66 |
| Tabla 56 Implementación del método Matricular de la clase ContenedorVerticalE5. | 67 |
| Tabla 57 Implementación del método ListarEstudiantes de la clase ContenedorVerticalP..... | 68 |
| Tabla 58 Probar la funcionalidad Matricular cursos para tercero..... | 69 |
| Tabla 59 Probar la funcionalidad Matricular cursos para cuarto. | 70 |
| Tabla 60 Probar la funcionalidad Matricular cursos para quinto. | 70 |
| Tabla 61 Probar la funcionalidad Mostrar el listado de matrícula de un curso..... | 71 |
| Tabla 62 Planificación de las Tareas de Ingeniería de la tercera iteración..... | 72 |
| Tabla 63 Implementación de la clase Utilidades para reporte de matrícula..... | 73 |
| Tabla 64 Implementación de la clase Utilidades para reporte de cursos optativos. | 73 |
| Tabla 65 Implementación de la clase Utilidades para reporte de cursos electivos. | 74 |
| Tabla 66 Probar la funcionalidad Generar reporte con el listado de los estudiantes matriculados en un curso. | 74 |
| Tabla 67 Probar la funcionalidad Generar reporte con los nombres de todos los cursos optativos..... | 75 |
| Tabla 68 Probar la funcionalidad Generar reporte con los nombres de todos los cursos electivos. | 76 |
| Tabla 69 Probar la funcionalidad Autenticar usuario Alternativa 1..... | 89 |
| Tabla 70 Probar la funcionalidad Autenticar usuario Alternativa 2..... | 90 |
| Tabla 71 Probar la funcionalidad Adicionar curso Alternativa 1..... | 91 |
| Tabla 72 Probar la funcionalidad Modificar curso Alternativa 1. | 92 |
| Tabla 73 Probar la funcionalidad Eliminar curso Alternativa 1. | 93 |

Tabla 74 Probar la funcionalidad Matricular cursos para tercero Alternativa 1.... 93

Tabla 74 Probar la funcionalidad Matricular cursos para tercero Alternativa 2.... 94

Introducción

En Cuba existe una extensa gama de entidades, entre las que se encuentran los Organismos Centrales, Empresas y Universidades cubanas. En la mayoría actualmente se están haciendo uso de las nuevas tecnologías informáticas para una mejor gestión de la información, a partir de las múltiples ventajas que estas brindan.

Una de estas entidades es la Universidad de las Ciencias Informáticas (UCI), en la misma existen varias facultades. Cada facultad está estructurada en su dirección por áreas y una de estas es el Vicedecanato de Formación donde se lleva a cabo una planificación, organización y control de la información docente. Debido a la cantidad de información docente que se procesa en dicho lugar, se precisa mejorar la gestión de la información de algunos de estos procesos, como es en el caso del proceso de matrícula de los cursos optativos y electivos.

Estos cursos contribuyen al desempeño del estudiante en las asignaturas que recibirá en años siguientes, como en su vida profesional y cotidiana, ya que le aportan múltiples conocimientos. En el caso de los cursos optativos le brinda la oportunidad al estudiante de aprender sobre algún tema en específico que sea de su interés y que no esté contemplado dentro del plan de estudio de la facultad a la que pertenece. En el caso de los cursos electivos, sus temáticas están más orientadas a contribuir y mejorar la cultura general integral de cada estudiante, ya que estos cursos pueden ser de idioma, cultura física y extensión universitaria.

El proceso de matrícula de cursos optativos y electivos se realiza actualmente de forma manual, por lo que resulta un proceso bastante tedioso. Este proceso es realizado por una sola persona, lo que trae como consecuencia la acumulación de estudiantes en el lugar donde se realiza la matrícula, así como que el estudiante tenga que presentarse varias veces ya que tal vez no dispone de tiempo para poder esperar su turno de matrícula. Otra de las dificultades que se presenta, es que los profesores no tienen un lugar donde publicar el Programa Analítico (P1) o un resumen de la temática de cada curso, en el cual el estudiante pueda acceder y consultar esta información para una mejor selección de los cursos de acuerdo a su interés. La información referente a la matrícula de los cursos queda reflejada en un documento de Microsoft Excel, donde el mismo queda archivado en una máquina, por lo que esta información pudiera perderse por diferentes factores como: rotura de la PC o violación de la seguridad de la misma, lo que trae como consecuencia

volver hacer el trabajo y emplear más tiempo en este proceso. Además partiendo del documento (Excel), que se utiliza para guardar los listados de matrícula se hace muy engorrosa la gestión de la información de la matrícula de los cursos.

A partir de la problemática anterior surge el **problema de investigación** que da origen al presente trabajo de diploma: ¿Cómo mejorar la gestión de la información de la matrícula de cursos optativos y electivos que se efectúa en el vicedecanato de formación de la facultad 5?

Este problema se enmarca en el **objeto de estudio**: Sistemas de gestión de información académica.

Delimitándose como **campo de acción**: Sistemas de gestión de la información de la matrícula de cursos optativos y electivos.

Definiéndose en este trabajo como **objetivo general**: Desarrollar una aplicación web para apoyar la gestión de la información de la matrícula de cursos optativos y electivos que se efectúa en el vicedecanato de formación de la facultad 5 de la Universidad de las Ciencias Informáticas.

Para darle cumplimiento al objetivo general se plantean las siguientes **tareas de investigación**:

- Estudio teórico metodológico del estado actual de los sistemas de gestión de la información.

- Entrevista a los vice-decano de formación para conocer en detalles como es que se realiza el proceso de matrícula de cursos optativos y electivos en cada facultad.

- Estudio de soluciones existentes.

- Selección de la metodología, herramientas de software a utilizar y lenguajes de programación para el desarrollo de la aplicación web.

- Análisis y diseño de la estructura de la aplicación web a desarrollar para una mejor guía en el proceso de implementación.

- Implementación de la aplicación web.

- Realizarle pruebas a la aplicación web para confirmar su correcto funcionamiento.

Métodos de investigación:

Para el desarrollo de la investigación fue necesario utilizar algunos métodos teóricos y empíricos que se listan a continuación:

Métodos Teóricos:

Analítico-Sintético: Para arribar a conclusiones especialmente en el capítulo referente a la fundamentación teórica de la investigación, así como llegar a la síntesis de los datos recopilados del presente trabajo.

Histórico-lógico: Para realizar el análisis y estudio de la información referente al estado del arte de los sistemas de gestión de la información, esencialmente el desarrollo de aplicaciones web para la gestión de la información.

Métodos empíricos:

Entrevista: Para entrevistar esencialmente a los Vicedecano de Formación para poder conocer cómo funciona el proceso de matrícula de cursos optativos y electivos.

El presente documento está compuesto por 3 capítulos:

Capítulo 1: Fundamentación Teórica. Se realiza un estudio del estado del arte y se exponen las bases que sustentan la investigación, donde se describen los conceptos esenciales para una mejor comprensión del trabajo a realizar. Así como la selección de las tecnologías a utilizar para la solución del problema y se fundamenta la metodología de desarrollo de software utilizada para guiar el proceso de desarrollo de la propuesta de solución.

Capítulo 2: Propuesta de Solución. Se elabora una pequeña descripción de la solución propuesta. Se realizan las historias de usuario que se utilizan para especificar los requisitos funcionales y no funcionales, reflejando una breve descripción de cada uno de estos. Se elaboran además las tarjetas Clase-Responsabilidades-Colaboradores (CRC), para describir las clases que conforman el sistema. Se especifican los roles del sistema con sus responsabilidades. Se definen los patrones de diseño a utilizar y la arquitectura del sistema. Concluyendo con el diseño del diagrama entidad relación de la base de datos.

Capítulo 3: Validación de la Propuesta de Solución. Se realiza una planificación por iteraciones correspondiente a la implementación de la aplicación donde se describen las tareas de ingeniería correspondiente a cada historia de usuario y se realizan un conjunto

de pruebas de aceptación a cada una de estas, concluyendo con un gráfico resumen de resultado de las pruebas aplicadas.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo, se describe una especificación de los principales conceptos enmarcados en la investigación. Se realiza el estudio del actual estado del arte acerca de los Sistemas de Gestión de la Información académica existentes, así como su importancia y se fundamenta la utilización de un conjunto de herramientas, lenguaje de programación y metodología a seguir para realizar el proceso de desarrollo de software.

1.2 Cursos Optativos y Electivos

Los cursos optativos y electivos pueden variar según las necesidades educativas, territoriales o las impuestas por el desarrollo científico técnico. A partir del currículo base, de los objetivos generales formulados para la carrera y de los programas de las disciplinas, el colectivo de la carrera de cada centro de educación superior decide cómo completar su plan de estudio particular, en correspondencia con las características del centro de educación superior. Partiendo de aquí surgen los cursos optativos y electivos como complemento de la formación integral de cada estudiante universitario. (1)

Estos cursos se imparten en el período de tiempo correspondiente a un semestre de un curso escolar, tienen como objetivos: 1) Desarrollar en los estudiantes los hábitos de estudio individual que se requieren para poder enfrentarse a la solución de problemas en el mundo de la computación. 2) Contribuir a la formación de profesionales con una alta calificación en el uso de las nuevas tecnologías. 3) Fomentar en los estudiantes un estilo de trabajo en el que siempre este presente la atención a la calidad del software. 4) Posibilitar que los estudiantes conozcan diversas técnicas, herramientas y tecnologías.

En la UCI cada estudiante debe cursar no menos de ocho cursos optativos y uno electivo para poder graduarse. El estudiante está obligado a aprobar estos cursos posteriormente de haber realizado la matrícula, y la nota correspondiente a los mismos se toma en cuenta en el cálculo del índice académico general. A continuación se expone una descripción que define en que consiste cada uno de estos cursos. (1)

Cursos Optativos: Son aquellos que se incluyen en el plan de estudio, entre los cuales el estudiante selecciona una cantidad determinada para cursar en forma obligatoria. Los contenidos de estos cursos tienen como propósito ampliar y actualizar a los estudiantes sobre temas científicos relacionados con la profesión. Además tienen un perfil: obligatorio

u opcional, dentro del primer perfil se agrupan aquellos cursos relacionados con el perfil de estudio de la facultad a la que pertenece el estudiante, mientras que el segundo perfil no guarda relación con el mismo. (2)

Cursos Electivos: Son aquellos que el estudiante elige libremente de acuerdo con sus gustos e intereses personales, a partir de un grupo de ofertas que se brindan y que pueden, inclusive, pertenecer a otras carreras. (3)

1.3 Gestión de la información

La gestión de información es un proceso que ha alcanzado un gran auge e importancia a nivel mundial. Debido a esta repercusión diversos autores e instituciones han dedicado un espacio dentro de sus investigaciones para dejar reflejado definiciones como las que se muestran a continuación:

- **Para Lynda Woodman (1985), la gestión de información se define como:** Todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado para tomar la acción precisa. (4)
- **Gestión de información:** Se trataría de la explotación de la información para la consecución de los objetivos de la entidad. Su creación, adquisición, procesamiento y difusión. (5)
- **Se define gestión de información como:** El proceso que se encarga de gestionar la información necesaria para la toma de decisiones y un mejor funcionamiento de los procesos, productos y servicios de la organización. (6)
- **Según Fairer Wessels, la gestión de información:** Es vista como la planificación, organización, dirección y control de la información dentro de un sistema abierto. (4)

Teniendo en cuenta las definiciones anteriores se puede llegar a la conclusión de definir gestión de información como: El proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, permitiendo mejorar los procesos, productos y servicios de la organización. Brindando la posibilidad de identificar, organizar, representar, recuperar información dispersas en áreas.

1.4 Sistemas de Gestión de la Información

Los sistemas de gestión de la información actualmente ocupan un lugar dentro de los principales sistemas informáticos de diversas instituciones. A lo largo del desarrollo y uso de las tecnologías de la información diversos autores han definidos algunos conceptos como los enunciados a continuación:

- **Según Davis y Olsón (1985) los Sistemas de gestión de la información son:** Sistemas integrados y automatizados para proveer la información que sostenga las funciones de operatividad, gestión y toma de decisiones en una organización. (7)
- **Para Moreiro González (1998) define los Sistemas de Gestión de la Información como:** El conjunto de políticas y normas relacionadas entre sí que se establecen para el acceso y tratamiento de los recursos de información, incluye los registros administrativos y los archivos, el soporte tecnológico de los recursos y el público a que se destina. (8)

Por lo que se puede llegar a la conclusión que un Sistema de Gestión de la Información: Es un sistema orientado a realizar una mejor organización, control y procesamiento de la información, además de brindar una mayor disponibilidad de la misma, lo que influye en un mejor desempeño a la hora de consultar dicha información para una posterior toma de decisiones.

1.5 Análisis de soluciones existentes en Cuba y el mundo

Diversas instituciones educativas poseen Sistemas de Gestión de la Información para un mejor procesamiento de la información académica como los que se mencionan a continuación.

1.5.1 Soluciones existentes a nivel internacional

- **Sistema de Gestión de la Información de la Universidad Alberto Hurtado, Chile:** En este sistema dentro de la información que gestiona se encuentra la de cursos optativos de formación general, donde en este sitio los estudiantes pueden informarse en cuanto a la necesidad y propósito de impartir estos cursos, también respecto a las áreas temáticas generales que son deporte y salud, desarrollo personal, arte y cultura y social, especificando los cursos por cada área. También en este sitio se informa de cuantos cursos se le acreditaran al estudiante, en que docente, días, horario y lugar se impartirán cada curso, además de la dirección

donde se encuentra el docente, la cantidad de estudiantes por curso y el crédito de cada uno de estos. (9)

- **Sistema de Gestión de la Información Búho:** Está diseñado específicamente para resolver los problemas cotidianos de control de estudios de una institución educativa. Mediante el uso de tecnologías Web y arquitectura cliente servidor. Permite además generar todos los reportes requeridos por las autoridades educativas tanto en formato digital como en formato impreso. Además cuenta con una interfaz web donde puede permitirse a los representantes de los estudiantes realizar preinscripciones directamente desde la página web de la institución, ahorrando así tiempo y esfuerzo en esta delicada labor cada inicio de año escolar. También cuenta con una intranet, la cual da soporte web para actividades internas de la organización como lo son la creación de los planes de estudio y el ingreso de las evaluaciones por parte del personal docente de la institución. (10)

Después de analizadas las características de cada uno de estos sistemas, se llega a la conclusión de que ninguno de los dos sistemas puede ser la solución a la problemática del presente trabajo, ya que en el caso del sistema de la Universidad Alberto Hurtado es un sistema donde solamente se publica información académica y de la universidad en general, por lo que es un sistema que no está orientado para interactuar, así como no cuenta con un módulo específico para la gestión de la información de los cursos. En el caso del sistema Búho no cuenta como solución ya que solamente los únicos autorizados para interactuar con este sistema es el personal docente es decir excluye a los estudiantes, además de que tampoco cuenta con un módulo para la gestión de información de los cursos.

1.5.2 Soluciones existentes a nivel nacional

- **Sistema de Gestión de la Información, Sistema de Gestión de la Nueva Universidad (SIGENU), Cuba:** En el 2004 a solicitud de la dirección del Ministerio de Educación Superior (MES) de Cuba, y como necesidad de automatizar los procesos fundamentales de la gestión académica de un Centro de Educación Superior en sus modalidades de estudio surge el proyecto SIGENU. El sistema está compuesto por varios módulos que gestionan la información de un estudiante desde que se matricula hasta que se gradúa o causa baja definitiva. Entre las funciones del sistema de gestión se encuentra la inscripción de un estudiante, registro de asignaturas a cursar, evaluaciones, control de bajas, emisión de reportes oficiales, entre otros. En correspondencia con su carácter nacional y la

gran diversidad de sistemas de enseñanza superior con que cuenta la universidad cubana, este sistema ha sido concebido de manera tal que sea capaz de brindar gran seguridad e integridad de la información. Así como ser tan flexible que permita ser adaptado a todos los centros de educación superior del país. (11)

- **Sistema de Gestión de la Información del Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE), Sistema Docente Integral (SDI), Cuba:** Este sistema se creó en el Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE) y surgió debido a la vital importancia que tiene para este centro el control, planificación y resultado de los procesos docentes. El SDI permite registrar, procesar y crear mecanismos de recuperación de la información relacionada con el proceso docente, tanto de pregrado como de postgrado de forma automatizada. Esto constituye una ventaja ya que con su uso se pueden realizar estos procesos de forma eficiente permitiendo optimizar los recursos del centro secretaría de facultades. (12)

- **Sistema de Gestión de la Información, Sistema de Gestión Universitaria (SGU) de la Universidad de las Ciencias Informáticas, Cuba:** El Sistema de Gestión Universitaria es una solución integral para la gestión de los procesos sustantivos de la universidad. Tiene como objetivo informatizar estos procesos partiendo de las debilidades de los sistemas existentes. Para ello se identificaron doce áreas de procesos. Una de ellas es Pregrado o Subsistema de Gestión Académica de Pregrado, el cual es el encargado de gestionar lo relacionado con la formación de estudiantes. Pregrado posee las funcionalidades del anterior Sistema de Gestión Académica “Akademos”, solución implementada con herramientas propietarias, e incorpora nuevos módulos y funcionalidades, entre los que se encuentra el módulo Gestión de Tesis. Este módulo surge a partir de la necesidad de informatizar el proceso de gestión de trabajos de diplomas en la universidad. (13)

Después del estudio y análisis de las características de los sistemas anteriormente expuestos, se puede concluir que ninguno de estos dan solución a la problemática del presente trabajo ya que los mismos de forma general están diseñados para una mejor gestión, organización y control de los procesos docentes específicamente para apoyar el trabajo que se lleva a cabo en las secretarías docentes de las instituciones mencionadas anteriormente. Además los sistemas (SIGENU), (SDI), (SGU) son sistema orientado para controlar toda la información docente de un estudiante y solamente tienen acceso a los

mismos el personal docente autorizado y no cuenta con un módulo orientado a la gestión de información de cursos y tampoco los estudiantes tienen acceso a estos sistemas.

1.6 Aplicaciones Web

Es un sistema informático distribuido cuya interfaz de usuario es accesible desde un cliente web, normalmente un navegador web o por una red como internet o una intranet. (14)

Características:

- El usuario puede acceder fácilmente a estas aplicaciones empleando un navegador web (cliente) o similar.
- Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No ocupan espacio en nuestro disco duro.
- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil) porque se accede a través de una página web, sólo es necesario disponer de acceso a Internet.
- No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas. (15)
- Gestionan datos almacenados en un servidor web, y que utilizan como interfaces páginas en formato Lenguaje de Marcas Hipertextuales (HTML por sus siglas en ingles), conteniendo datos hipermedia. (16)

Los tres niveles típicos de las aplicaciones web son:

- **El nivel de interfaz de usuario:** Está compuesto por las páginas HTML que el usuario solicita a un servidor web y que visualiza en un cliente web (normalmente, un navegador web).
- **El nivel de lógica de negocio:** Está compuesto por los módulos que implementan la lógica de la aplicación y que se ejecutan en un servidor de aplicaciones.
- **El nivel de datos:** Está compuesto por los datos, normalmente gestionados por un sistema de gestión de bases de datos (servidor de datos), que maneja la aplicación web. (14)

1.7 Descripción de las tecnologías y herramientas

Las tecnologías y herramientas para el desarrollo de aplicaciones web son muy importantes, ya que las mismas poseen diversas características y ventajas que ayudan en el proceso de desarrollo. Por eso a continuación se hace una breve descripción de cada una de estas que son las que se utilizarán en la presente investigación.

1.7.1 Entorno de Desarrollo Integrado (IDE)

Eclipse: Es un IDE de código abierto multiplataforma y extensible que permite a los desarrolladores crear rápidamente aplicaciones web, empresariales, de escritorio y móviles. Dispone de un editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene asistentes para creación de proyectos, clases, enumerativos, interfaces y refactorización. A través de plugins es posible añadir control de versiones con Subversión. Es una plataforma gratuita, creada inicialmente por IBM y desarrollada en la actualidad por el Proyecto Eclipse. Es totalmente extensible con módulos que aumentan su funcionalidad y que permiten que el entorno de desarrollo soporte otros lenguajes además de Java. (17)

Provee un conjunto de herramientas para administrar espacios de trabajo, construir, correr y depurar aplicaciones y compartir artefactos con un equipo. Permite el control del editor de código, del compilador y del depurador desde una única interfaz de usuario. Su misión consiste en evitar tareas repetitivas, facilitar la escritura de código correcto, disminuir el tiempo de depuración e incrementar la productividad del desarrollador. (18)

La plataforma Eclipse está habilitada para afrontar las siguientes necesidades:

- Soportar la construcción de gran variedad de herramientas de desarrollo.
- Soportar las herramientas proporcionadas por diferentes fabricantes de software independientes.
- Soportar herramientas que permitan manipular diferentes contenidos (HTML, Java, C, JSP, EJB, XML, y GIF).
- Facilitar una integración transparente entre todas las herramientas y tipos de contenidos sin tener en cuenta al proveedor.
- Proporcionar entornos de desarrollo gráfico (GUI) o no gráficos.
- Ejecutarse en una gran variedad de sistemas operativos, incluyendo Windows y Linux. (18)

1.7.2 Lenguaje de Programación

Java: Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma gran parte de su

sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Este lenguaje tiene la ventaja de ser multiplataforma. (19) La compañía Sun Microsystems lo describe como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”. (20) Como características presenta que, los programas ejecutables creados por el compilador Java son independientes de la arquitectura, es fácil de aprender, está bien estructurado, sus aplicaciones son fiables, funciona perfectamente en red, aprovecha características de la mayoría de los lenguajes modernos evitando sus inconvenientes y tiene una gran funcionalidad gracias a sus librerías. (21)

PHP: Es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Permite la conexión a diferentes tipos de servidores de base de datos tales como: MySQL, Postgre, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite; lo cual facilita la creación de aplicaciones Web muy robustas. Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Linux, Windows y Mac OS X, y puede interactuar con los servidores de Web más populares. 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. Permite las técnicas de Programación Orientada a Objetos. (22)

1.7.2.1 Selección del lenguaje de programación

Después de haber expuesto las características de los lenguajes mencionados se ha decidido utilizar Java, ya que es el lenguaje que es totalmente compatible con el resto de las tecnologías y herramientas a utilizar en el desarrollo del sistema informático de la presente investigación. Además brinda múltiples facilidades en la utilización de servicios web para el desarrollo de sistemas. En el caso del lenguaje PHP presenta algunas desventajas como: difícil de optimizar, todo el trabajo lo realiza el servidor y no delega al cliente por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número, la legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP y dificultad la organización por capas de la aplicación.

Ventajas de Java

- **Simple:** Elimina la complejidad de los lenguajes como "C" y da paso al contexto de los lenguajes modernos orientados a objetos.
- **Familiar:** Como la mayoría de los programadores están acostumbrados a programar en C o en C++, la sintaxis de Java es muy similar al de estos.
- **Robusto:** El sistema de Java maneja la memoria de la computadora por las personas. No te tienen que preocuparse por apuntadores o memoria que no se esté utilizando.
- **Seguro:** El sistema de Java tiene ciertas políticas que evitan que se puedan codificar virus con este lenguaje.
- **Dinámico:** Java no requiere que compile todas las clases de un programa para que este funcione.
- **Lenguaje Multiplataforma:** El código que escrito en java es leído por un intérprete, por lo que su programa andará en cualquier plataforma. Puede correr en el explorador y en dispositivos móviles. (23)

1.7.3 Servidores de Aplicaciones y Contenedores Web

Tomcat: Es un motor de ejecución de las tecnologías Servlets y Java Server Page (JSP), que incluye un pequeño servidor de HTTP para atender las peticiones de contenido estático y redirigir el resto al motor de ejecución de Servlets (los Servlets y los JSP son mecanismos de desarrollo de aplicaciones web), pero a pesar de incluir un servidor web, Apache es el servidor web usado por excelencia con Tomcat. Para ello se dispone de un módulo de Apache que realiza el enlace con Tomcat. Dicho módulo se denomina mod_jk2. Es un proyecto de software libre de la Apache Software Foundation. El mismo es desarrollado en un entorno abierto y participativo. Está liberado bajo Apache Software Licence, siendo totalmente libre y gratis su utilización. Es un producto muy robusto, altamente eficiente y uno de los más potentes contenedores de Servlets existentes. Al ser un contenedor web consume menos recursos que los servidores de aplicaciones, pudiendo adaptarse en máquinas con poca capacidad de rendimiento. (24)

Entre sus características principales están:

- **Administrador de seguridad:** ayuda a mantener un control sobre el comportamiento de las aplicaciones web mediante el uso y la configuración de un Java Security Manager.

- **SSL:** permite configurar el SSL dentro del servidor para que este responda peticiones a través del protocolo seguro https.
- **Clusterización:** puede habilitar sesiones de replicación en un entorno de ejecución de Apache Tomcat.
- **Balanceador:** permite configurar, usar y extender el balanceador de carga. (18)

JBoss: Es una plataforma certificada por la JEE para desarrollar y desplegar aplicaciones empresariales, aplicaciones web y portales hechos en Java. El servidor de aplicaciones JBoss provee de todas las ventajas de J2EE 1.4, así como servicios empresariales extendidos incluyendo la clusterización, manejo de memoria caché y persistencia. Además incluye soporte para EJB 3.0, el cual es diseñado para simplificar el modo de programación de las aplicaciones empresariales en Java. (25) **Algunas de las características son:**

- Completo soporte J2EE 1.4 (EJB, JCA, JSP, JMX, HTTP).
- Completa implementación de la seguridad e integración con Java Autenticación y Autorización de Servicios (JAAS por sus siglas en ingles).
- Clusterización de objetos Java (EJB, HTTP, POJO).
- Dentro de su arquitectura cuenta con una capa de aspectos basada en el modelo de programación orientada a aspectos. Esto le permite adicionar transparentemente el comportamiento suministrado por un servicio a cualquier objeto. (25)

1.7.3.1 Selección del entorno de ejecución

Después de un análisis de las características expuestas anteriormente se llega a la conclusión de utilizar Tomcat, ya que es un contenedor web fácil y sencillo de administrar y configurar. El mismo consume menos recursos que un servidor de aplicaciones como JBoss. Es gratis y posee un gran número de usuarios y soporte en la comunidad mundial. Es compatible con la mayoría de los API de JEE. Permite lograr escalabilidad mediante clúster. Este contenedor presenta un manejador de reserva de conexiones que se configura de manera que desde la aplicación se acceden a estas conexiones, este mecanismo es más eficiente que el de realizar la conexión en el momento que requiera. Provee también una consola Web para manipular las características básicas de las aplicaciones desplegadas en él, además de información del estado y monitoreo requerido.

1.7.4 Gestores de Base de Datos

PostgreSQL: Es un sistema gestor de bases de datos objeto-relacional, desarrollado en la Universidad de California y liberado bajo la licencia BSD 1; soporta casi toda la sintaxis SQL (incluyendo subconsultas, transacciones y funciones definidas por el usuario), posee características avanzadas como consultas complejas, llaves foráneas, disparadores, vistas, integridad transaccional, control de concurrencia multiversión y puede ser extendido por el usuario añadiéndole tipos de datos, funciones, operadores, funciones agregadas, métodos de indexado y lenguajes procedurales. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. (26)

El mismo, funciona sobre 34 plataformas incluyendo Windows XP, Linux, FreeBSD, Solaris, Unix; y además soporta gran cantidad de lenguajes dentro de su API para el desarrollo de aplicaciones, como por ejemplo SQL, Java, Perl, Python, C, C++, Ruby y PHP. Posee una documentación muy bien organizada, pública y libre. PostgreSQL es el gestor de código abierto más avanzado del mundo; lo que se refleja en que lo utilicen empresas como Yahoo, Greenplum, MyYearbook, Hi.5 y Facebook. (26)

MySQL: Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Funciona en diferentes plataformas. Posee un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Soporta a grandes bases de datos. Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT. MySQL Server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. (27)

Características:

- La velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está. (28)

1.7.4.1 Selección del Gestor de Base de Datos

El gestor de base de datos MySQL presenta algunas desventajas como: un gran porcentaje de sus utilidades no están documentadas, no tiene integridad referencial, no tiene grandes capacidades como otros gestores, es lento en la gestión de grandes base de datos y no es intuitivo. Por tanto después de haber analizado las características de los gestores de base de datos antes mencionados se seleccionó PostgreSQL. El mismo cuenta con múltiples ventajas como las que se mencionan a continuación.

Ventajas de PostgreSQL

- Cuenta con interfaz para diversos lenguajes como es en el caso del lenguaje Java seleccionado anteriormente como lenguaje de programación a utilizar.
- Es uno de los gestores de bases de datos relacional orientado a objetos de software libre más utilizado en el mundo.
- Es capaz de manejar una enorme cantidad de datos, permitiendo el acceso simultáneo de un conjunto de usuarios. Además brinda seguridad y estabilidad a los mismos. Facilita el trabajo con procedimientos almacenados y consultas.
- Es compatible con las herramientas y tecnologías seleccionadas para trabajar en el desarrollo de la aplicación informática.

1.7.5 Herramienta de administración de Bases de Datos PgAdmin III

PgAdmin III es una herramienta para la administración gráfica de PostgreSQL. Funciona sobre casi todas las plataformas. Fue diseñado para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL hasta la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita la administración.

La aplicación también incluye un editor de la sintaxis SQL, un editor de código del lado del servidor, un agente para la programación de tareas «SQL/batch/shell» y soporte para el motor de replicación Slony-I. (29) Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris. El mismo está desarrollado por una comunidad de expertos de PostgreSQL en todo el mundo y está disponible en más de una docena de idiomas. Es una herramienta libre y se desarrolla bajo la licencia PostgreSQL Licence a partir de la versión v1.10. (30)

1.7.6 Framework Vaadin

Vaadin es un framework Java, para el desarrollo de aplicaciones web vistosas e interactivas sin requerir para ello de plugins en el servidor web, ya que es un framework que trabaja en el lado del servidor. Funciona perfectamente con los siguientes navegadores: Mozilla Firefox 3+, Internet Explorer 6, 7, y 8 y Google Chrome. Para el programador la ventaja es que no se trabaja con HTML, XMLs ni JavaScript. Directamente se programa en Java basándose en las librerías de Google Web Toolkit (GWT). Su licencia es OpenSource (licencia Apache2.0). En el corazón de Vaadin existen, principalmente, tres tecnologías: Ajax para la interacción entre el usuario y la aplicación posibilitando la actualización de contenido mediante peticiones y respuestas asíncronas, Google Web Toolkit aísla al programador de trabajar con JavaScript y otras tecnologías dependientes del tipo de navegador utilizando llamadas a procedimiento remoto (RPC) hacia el servidor y serializando toda la información y utiliza notación de objetos de JavaScript (JSON) que permite crear y parsear información sobre los componentes para trabajar directamente con GWT. Tiene un modelo basado en eventos y listener y permite desplegar portlets de Vaadin en portales comoiferay. La documentación es bastante extensa y sobre todo bien explicada. Cada parte de su arquitectura está explicada desde lo más básico hasta lo más complejo, además de estar acompañado de ejemplos. Posee un plugins que le permite integrarse con varios entornos de desarrollo como NetBeans y Eclipse. (32)

1.8 Metodología de Desarrollo de Software

Es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a la producción de software. (33) Entre las metodologías ágiles más comunes se pueden mencionar Programación Extrema (XP por sus siglas en inglés) y Proceso Unificado Ágil (AUP por sus siglas en inglés) que son dos de las más conocidas y usadas.

Proceso Unificado Ágil (AUP): Es una versión simplificada del Proceso Unificado del Rational (RUP) desarrollada por Scott Ambler. Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (TDD por sus siglas en inglés), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad. La misma se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. El proceso AUP establece un Modelo

más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de las disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión de proyectos y Entorno) coinciden con las restantes de RUP.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados: (34)

- **Inicio**
- **Elaboración**
- **Construcción**
- **Transición**

Principios de la AUP:

- **El personal sabe lo que está haciendo:** No obliga a conocer detalles. La gente no va a leer documentación del proceso detallada, pero quieren algo de orientación a alto nivel y/o formación de vez en cuando.
- **Simplicidad:** Esta descrito de forma concisa.
- **Agilidad:** Procesos simplificados del RUP.
- **Centrarse en actividades de alto valor:** Esenciales para el desarrollo.
- **Independencia de herramientas:** A disposición del usuario. La recomendación es que usen las herramientas que mejor se adapten al trabajo.
- **Fácil de adaptación:** La AUP producto es de fácil acomodo común a través de cualquier herramienta de edición del HTML. (34)

Programación Extrema (XP): Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (35)

Ventajas:

- Programación organizada.

- Menor tasa de errores.
- Satisfacción del programador.
- El cliente tiene el control sobre las prioridades.
- Se hacen pruebas continuas durante el proyecto. (28)

1.8.1 Selección de la metodología de desarrollo de software

En el caso de la metodología (AUP) presenta algunas desventajas, no posee muchas fuentes de documentación, no especifica explícitamente que artefactos genera, deja a elección del desarrollador generar la documentación que le sea más importante. Partiendo de las características de las metodologías antes expuestas se llega a la conclusión de seleccionar la metodología (XP), ya que está diseñada para grupos de desarrollo pequeños, genera poca documentación, está diseñada para proyectos donde la cantidad de requisitos sea pequeña, especifica claramente los artefactos que genera, además de que es la que más se adapta al tiempo disponible para el desarrollo del software de la presente investigación.

1.9 Visual Paradigm for UML 8.0

Visual Paradigm para Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es una herramienta que ayuda a una rápida construcción de aplicaciones de mejor calidad y a un menor costo. Permite construir diagramas de diversos tipos, 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. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones. Visual Paradigm 8.0 ha sido la herramienta seleccionada para soportar el ciclo de desarrollo de la aplicación web a elaborar. (29)

1.10 Consideraciones parciales

En este capítulo quedaron reflejados todos los conceptos relacionados con la presente investigación, llegando a conclusiones en el análisis del actual estado del arte. Se definieron un conjunto de elementos a utilizar para llevar a cabo el desarrollo de la propuesta de solución como entorno de desarrollo integrado Eclipse, lenguaje de programación Java, de este mismo lenguaje el framework Vaadin, contenedor web Tomcat, gestor de base de datos PostgreSQL, herramienta de administración de base de datos PgAdmin III, metodología de desarrollo Programación Extrema (XP) y como herramienta de modelado Visual Paradigm quedando reflejado una breve descripción de cada uno de estos elementos.

CAPÍTULO 2: Propuesta de Solución

2.1 Introducción

En el presente capítulo se dan a conocer cuáles son los procesos que serán automatizados. Donde se expone una breve descripción de la solución propuesta. Se identifican cuáles son los requisitos funcionales y no funcionales reflejando su descripción en la elaboración de historias de usuario. Se definen los roles del sistema y sus responsabilidades. Además se confeccionan tarjetas CRC para describir las clases del sistema. Se identifican los patrones de diseño a utilizar y se define la arquitectura del sistema. Concluyendo con la elaboración del diagrama entidad relación de la base de datos.

2.2 Reglas de la propuesta de solución

En el despliegue del sistema de gestión de matrícula de curso se deben cumplir determinadas reglas preestablecidas a partir del levantamiento de requisitos como son las siguientes que se mencionan a continuación: el vicedecano de formación es la única persona que posee permisos para acceder a las funcionalidades adicionar curso, modificar curso, eliminar curso, emitir reporte optativo y emitir reporte electivo. En el caso de los profesores solamente tienen permiso para ver los listados de matrícula de los cursos que imparten y emitir un reporte de los mismos, y en caso del vicedecano de formación posee los permisos para ver todos los listados de matrícula de cada curso disponible y emitir un reporte de cada listado. En el caso de los estudiantes solamente poseen permisos para ver los cursos disponibles y para acceder a la funcionalidad matricular curso.

2.3 Propuesta de solución

El sistema gestionará toda la información referente a la matrícula de cursos optativos y electivos. El sistema debe ser capaz de permitir crear, eliminar y modificar los cursos. Los estudiantes podrán matricular de forma "online". Generará reportes con los listados de los estudiantes matriculados en cada curso y con los nombres de los cursos optativos y electivos a matricular. También debe dar la posibilidad de poder ver el listado de los estudiantes matriculados en un curso. Además le debe permitir al estudiante consultar la información referente a cualquier curso. Debe contar con las siguientes características: 1) Rapidez para acceder a la información de la aplicación, ya que el sistema será solamente para gestionar la información de la matrícula de estos cursos. 2) Los usuarios contarán

con una aplicación sencilla y de fácil uso que estará disponible en todo momento determinado para efectuar la matrícula de cursos, 3) La información estará más centralizada lo que permite una mejor búsqueda o consulta de la misma.

2.4 Roles del sistema

| Roles | Responsabilidades |
|---------------|--|
| Administrador | Es la persona encargada de: <ul style="list-style-type: none"> • Adicionar curso • Modificar curso • Eliminar curso • Emitir reporte optativo • Emitir reporte electivo |
| Profesor | Es la persona encargada de: <ul style="list-style-type: none"> • Obtener el listado de matrícula de cada curso que imparte. • Emitir un reporte con cada uno de estos listados de matrícula. |
| Estudiante | Es la persona encargada de: <ul style="list-style-type: none"> • Matricular en un curso. |

Tabla 1 Roles del sistema

2.5 Historias de usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. (35)

Se delimitaron un grupo de historias de usuario que corresponden con las funcionalidades que debe contener el sistema, así como con los requisitos no funcionales que delimitan herramientas fundamentales en el desarrollo de la aplicación y características que debe cumplir la misma.

Tabla 2 Historia de usuario No1

| Historia de usuario | |
|--|--------------------------------|
| Numero: 1 | Usuario: Usuario |
| Nombre de Historia de usuario: Autenticar usuario. | |
| Prioridad en el negocio: Alta | Riesgo en el desarrollo: Media |
| Iteración asignada: 1 | |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: Para que los usuarios puedan acceder al sistema deben autenticarse insertando su usuario y contraseña. | |

Tabla 3 Historia de usuario No2

| Historia de usuario | |
|---|--------------------------------|
| Numero: 2 | Usuario: Usuario |
| Nombre de Historia de usuario: Asignar permisos. | |
| Prioridad en el negocio: Alta | Riesgo en el desarrollo: Media |
| Iteración asignada: 1 | |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: Para los roles del sistema de gestión de matrícula de cursos se asignaron los siguientes permisos: Administrador se le asignaron permisos para: <ul style="list-style-type: none"> • Adicionar curso • Modificar curso • Eliminar curso • Emitir reporte optativo • Emitir reporte electivo • Obtener el listado de matrícula de cada uno de los cursos disponibles. • Emitir un reporte con cada uno de estos listados de matrícula. Profesor se le asignaron permisos para: <ul style="list-style-type: none"> • Obtener el listado de matrícula de cada curso que imparte. • Emitir un reporte con cada uno de estos listados de matrícula. Estudiante se le asignaron permisos para: <ul style="list-style-type: none"> • Matricular en un curso | |

Tabla 4 Historia de usuario No3

| Historia de usuario | |
|--|--------------------------------|
| Numero: 3 | Usuario: Usuario |
| Nombre de Historia de usuario: Adicionar curso. | |
| Prioridad en el negocio: Alta | Riesgo en el desarrollo: Media |
| Iteración asignada: 1 | |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción: Una vez que este autenticado el administrador del sistema, al mismo se le debe mostrar un botón que se llama (Listado de cursos), al presionar este botón se le muestra una interfaz que contiene el botón (Adicionar Curso), al presionar este botón se muestra una ventana que contiene un formulario donde debe insertar los siguientes elementos: nombre del curso, cantidad de estudiantes a matricular, año de los estudiantes que les corresponde el curso, semestre en que se impartirá, disciplina a la que pertenece, curso escolar, tipo de curso, descripción del curso, usuario del profesor que impartirá el curso y las fechas de la convocatoria de matrícula del curso. A continuación se debe acceder a guardar los datos insertados y en esta misma interfaz donde se puede acceder al adicionar se podrá ver una tabla que muestra la información de cada curso registrado.</p> | |

Tabla 5 Historia de usuario No4

| Historia de usuario | |
|---|--------------------------------|
| Numero: 4 | Usuario: Usuario |
| Nombre de Historia de usuario: Modificar curso. | |
| Prioridad en el negocio: Alta | Riesgo en el desarrollo: Media |
| Iteración asignada: 1 | |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción: Una vez que este autenticado el administrador del sistema, al mismo se le debe mostrar un botón que se llama (Listado de cursos), al presionar este botón se le muestra una interfaz, que contiene la tabla que tiene la funcionalidad modificar de cada curso, al presionar la opción modificar se muestra una ventana que contiene un formulario con los datos del curso seleccionado , se deben seleccionar el o los elementos que se mencionan a continuación: nombre del curso, cantidad de estudiantes a matricular, año de los estudiantes que les corresponde el curso, semestre en que se impartirá, disciplina a la que pertenece, curso escolar, tipo de curso, descripción del curso, usuario del profesor que impartirá el curso y las fechas</p> | |

de la convocatoria de matrícula del curso. Después de seleccionar los elementos que desea modificar posteriormente debe acceder a guardar los cambios realizados.

Tabla 6 Historia de usuario No5

| Historia de usuario | |
|--|---------------------------------------|
| Numero: 5 | Usuario: Usuario |
| Nombre de Historia de usuario: Eliminar curso. | |
| Prioridad en el negocio: Alta | Riesgo en el desarrollo: Media |
| Iteración asignada: 1 | |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción: Una vez que este autenticado el administrador del sistema, al mismo se le debe mostrar un botón que se llama (Listado de cursos), al presionar este botón se le muestra una interfaz, que contiene la tabla que tiene la funcionalidad eliminar correspondiente a la fila de cada curso, se le muestra un mensaje (¿Esta seguro que desea eliminar el curso seleccionado?) presiona (si) y se elimina el curso.</p> | |

Tabla 7 Historia de usuario No6

| Historia de usuario | |
|--|---------------------------------------|
| Numero: 6 | Usuario: Usuario |
| Nombre de Historia de usuario: Matricular cursos para tercero. | |
| Prioridad en el negocio: Alta | Riesgo en el desarrollo: Media |
| Iteración asignada: 2 | |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción: Una vez que este autenticado un estudiante en el sistema, al mismo se le debe mostrar un botón que se llama (Cursos disponibles) al presionar este botón se le muestra un ComboBox donde debe seleccionar el año que cursa en este caso tercero y automáticamente se le muestran los cursos correspondientes a su año y la información correspondiente a cada uno de estos. Cuando el estudiante accede a la opción matricular de la fila de cada curso se le mostrarían uno de los tres mensaje que se muestran al matricular que son:</p> <p>Mensaje 1: Se ha matriculado correctamente</p> <p>Mensaje 2: No puede matricular en este curso porque ya no quedan capacidades de matrícula.</p> <p>Mensaje 3: No puede matricular en este curso porque ya usted está matriculado en este curso.</p> | |

Tabla 8 Historia de usuario No7

| Historia de usuario | |
|---|---------------------------------------|
| Numero: 7 | Usuario: Usuario |
| Nombre de Historia de usuario: Matricular cursos para cuarto. | |
| Prioridad en el negocio: Alta | Riesgo en el desarrollo: Media |
| Iteración asignada: 2 | |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción: Una vez que este autenticado un estudiante en el sistema, al mismo se le debe mostrar un botón que se llama (Cursos disponibles) al presionar este botón se le muestra un ComboBox donde debe seleccionar el año que cursa en este caso cuarto y automáticamente se le muestran los cursos correspondientes a su año y la información correspondiente a cada uno de estos. Cuando el estudiante accede a la opción matricular de la fila de cada curso se le mostrarían uno de los tres mensaje que se muestran al matricular que son:</p> <p>Mensaje 1: Se ha matriculado correctamente</p> <p>Mensaje 2: No puede matricular en este curso porque ya no quedan capacidades de matrícula.</p> <p>Mensaje 3: No puede matricular en este curso porque ya usted está matriculado en este curso.</p> | |

Tabla 9 Historia de usuario No8

| Historia de usuario | |
|---|---------------------------------------|
| Numero: 8 | Usuario: Usuario |
| Nombre de Historia de usuario: Matricular cursos para quinto. | |
| Prioridad en el negocio: Alta | Riesgo en el desarrollo: Media |
| Iteración asignada: 2 | |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción: Una vez que este autenticado un estudiante en el sistema, al mismo se le debe mostrar un botón que se llama (Cursos disponibles) al presionar este botón se le muestra un ComboBox donde debe seleccionar el año que cursa en este caso quinto y automáticamente se le muestran los cursos correspondientes a su año y la información correspondiente a cada uno de estos. Cuando el estudiante accede a la opción matricular de la fila de cada curso se le mostrarían uno de los tres mensaje que se muestran al matricular que son:</p> | |

Mensaje 1: Se ha matriculado correctamente

Mensaje 2: No puede matricular en este curso porque ya no quedan capacidades de matrícula.

Mensaje 3: No puede matricular en este curso porque ya usted está matriculado en

Tabla 10 Historia de usuario No9

| Historia de usuario | |
|---|---------------------------------------|
| Numero: 9 | Usuario: Usuario |
| Nombre de Historia de usuario: Mostrar el listado de matrícula de un curso. | |
| Prioridad en el negocio: Media | Riesgo en el desarrollo: Media |
| Iteración asignada: 3 | |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción: Una vez que este autenticado un profesor en el sistema, al mismo se le debe mostrar un botón que se llama (Datos de mi curso) al presionar este botón se le muestra un ComboBox con los cursos que el imparte, donde debe seleccionar el curso del cual quiere obtener el listado de estudiantes matriculados en ese curso. A continuacion se le muestra una tabla con el listado de estudiantes. Mostrándose nombre y apellidos de cada estudiante.</p> <p>Una vez que este autenticado el administrador del sistema, al mismo se le debe mostrar un botón que se llama (Datos de mi curso) al presionar este botón se le muestra un ComboBox con todos los cursos registrados, donde debe seleccionar el curso del cual quiere obtener el listado de estudiantes matriculados en ese curso. A continuacion se le muestra una tabla con el listado de estudiantes. Mostrándose nombre y apellidos de cada estudiante.</p> | |

Tabla 11 Historia de usuario No10

| Historia de usuario | |
|---|---------------------------------------|
| Numero: 10 | Usuario: Usuario |
| Nombre de Historia de usuario: Generar reporte con el listado de los estudiantes matriculados en un curso. | |
| Prioridad en el negocio: Media | Riesgo en el desarrollo: Media |
| Iteración asignada: 3 | |
| Programador responsable: Yuleisy Galloso Soler | |

Descripción: Una vez que este autenticado un profesor en el sistema, al mismo se le debe mostrar un botón que se llama (Datos de mi curso) al presionar este botón se le muestra un ComboBox con los cursos que el imparte, donde debe seleccionar el curso del cual quiere obtener el listado de estudiantes matriculados en ese curso. A continuación se le muestra una tabla con el listado de estudiantes. Mostrándose nombre y apellidos de cada estudiante. Después presiona en el botón (Reporte de Matricula) automáticamente se emite un PDF con el listado de estudiantes matriculados en el curso seleccionado.

Una vez que este autenticado el administrador del sistema, al mismo se le debe mostrar un botón que se llama (Datos de mi curso) al presionar este botón se le muestra un ComboBox con todos los cursos registrados, donde debe seleccionar el curso del cual quiere obtener el listado de estudiantes matriculados en ese curso. A continuación se le muestra una tabla con el listado de estudiantes. Mostrándose nombre y apellidos de cada estudiante. Después presiona en el botón (Reporte de Matricula) automáticamente se emite un PDF con el listado de estudiantes matriculados en el curso seleccionado.

Tabla 12 Historia de usuario No11

| Historia de usuario | |
|--|--------------------------------------|
| Numero: 11 | Usuario: Usuario |
| Nombre de Historia de usuario: Generar reporte con los nombres de todos los cursos optativos. | |
| Prioridad en el negocio: Media | Riesgo en el desarrollo: Baja |
| Iteración asignada: 3 | |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: Una vez que este autenticado el administrador del sistema, al mismo se le debe mostrar un botón que se llama (Listado de cursos) al presionar este botón se le muestra una interfaz que contiene el botón (Reporte Optativo) al presionar este botón automáticamente se emite un PDF con el listado de cursos optativos. | |

Tabla 13 Historia de usuario No12

| Historia de usuario | |
|--|-------------------------|
| Numero: 12 | Usuario: Usuario |
| Nombre de Historia de usuario: Generar reporte con los nombres de todos los | |

| | |
|--|--------------------------------------|
| cursos electivos. | |
| Prioridad en el negocio: Media | Riesgo en el desarrollo: Baja |
| Iteración asignada: 3 | |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: Una vez que este autenticado el administrador del sistema, al mismo se le debe mostrar un botón que se llama (Listado de cursos) al presionar este botón se le muestra una interfaz que contiene el botón (Reporte Electivo) al presionar este botón automáticamente se emite un PDF con el listado de cursos electivos. | |

Tabla 14 Historia de usuario No13

| | |
|--|-------------------------|
| Historia de usuario | |
| Numero:13 | Usuario: Usuario |
| Nombre de Historia de usuario: Requisito de apariencia o interfaz externa | |
| Descripción: El sistema debe tener una interfaz que sea entendible y manejable para los usuarios. Con colores y letras donde generalmente deben predominar colores como el verde, gris y azul con un tamaño de letra estándar. Predominando el nombre de la aplicación Sistema de Gestión de Matricula de Cursos (GEXMAC) en el banner de la misma, manteniendo un mismo diseño uniforme y colores en todas las interfaces. | |

Tabla 15 Historia de usuario No14

| | |
|---|-------------------------|
| Historia de usuario | |
| Numero:14 | Usuario: Usuario |
| Nombre de Historia de usuario: Requisito de usabilidad | |
| Descripción: El sistema debe ser de fácil uso, de tal manera que el usuario pueda acceder a las distintas interfaces de manera intuitiva sin necesidad de un previo conocimiento acerca del mismo. | |

Tabla 16 Historia de usuario No15

| | |
|---|-------------------------|
| Historia de usuario | |
| Numero:15 | Usuario: Usuario |
| Nombre de Historia de usuario: Requisito de seguridad | |
| Descripción: El sistema deberá garantizar que las funcionalidades se muestren de | |

acuerdo al nivel de usuario que este activo. El sistema deberá estar protegido contra acciones que no estén autorizadas o que puedan afectar la integridad de los datos. El sistema deberá verificar acciones irreversibles (eliminaciones).

Tabla 17 Historia de usuario No16

| Historia de usuario | |
|--|-------------------------|
| Numero:16 | Usuario: Usuario |
| Nombre de Historia de usuario: Requisito de confiabilidad | |
| Descripción: El sistema debe validar todas las posibles acciones del usuario en el proceso, teniendo en cuenta los privilegios de cada uno. | |

Tabla 18 Historia de usuario No17

| Historia de usuario | |
|--|-------------------------|
| Numero:17 | Usuario: Usuario |
| Nombre de Historia de usuario: Restricciones en el diseño y la implementación | |
| Descripción: Para el desarrollo de la solución se definen un conjunto de restricciones: -Lenguaje de programación: Java -Entorno integrado de desarrollo: Eclipse -Gestor de base de datos: PostgreSQL 8.2 -Herramienta de administración de Bases de Datos: PgAdmin III. -Gestor de Portales: Liferay versión 6.1.1 -Framework: Vaadin versión 6.8.3 | |

Tabla 19 Historia de usuario No18

| Historia de usuario | |
|---|-------------------------|
| Numero:18 | Usuario: Usuario |
| Nombre de Historia de usuario: Requisito de software | |
| Descripción: Para que el sistema pueda funcionar es necesario: -Sistema Operativo: Windows. -Contenedor web: Tomcat 7.0 -Para la PC del cliente solo se requiere Mozilla Firefox 1.5 o versión superior e Internet Explorer 6.0 o versión superior. | |

Tabla 20 Historia de usuario No19

| Historia de usuario | |
|---|-------------------------|
| Numero:19 | Usuario: Usuario |
| Nombre de Historia de usuario: Requisito de hardware | |
| Descripción: -La PC donde se ubique el servidor requiere como mínimo, un procesador Pentium IV, 1.6 GHz y 1 GB de memoria RAM -En el caso de la PC cliente, deberá ser como mínimo una PC Pentium 2 o superior, con 512 MB de memoria RAM. | |

2.6 Estimación de esfuerzo por historia de usuario

La siguiente tabla contiene la estimación de esfuerzo por historia de usuario. Cada estimación incluye el esfuerzo asociado a la implementación de la HU, la misma se expresa utilizando como medida una puntuación, que es directamente proporcional al esfuerzo. Un punto se considera como una semana ideal de trabajo, donde se trabaje el tiempo planeado sin ningún tipo de interrupción.

Tabla 21 Estimación de esfuerzo por historia de usuario

| Historias de usuario a implementar en la entrega | | Puntos Estimados |
|--|---|------------------|
| 1 | Autenticar usuario | 2 |
| 2 | Asignar permisos | 2 |
| 3 | Adicionar curso | 2 |
| 4 | Modificar curso | 1 |
| 5 | Eliminar curso | 1 |
| 6 | Matricular cursos para tercero | 1 |
| 7 | Matricular cursos para cuarto | 1 |
| 8 | Matricular cursos para quinto | 1 |
| 9 | Mostrar el listado de matrícula de un curso. | 1 |
| 10 | Generar reporte con el listado de los estudiantes matriculados en | 2 |

| | | |
|-----------|--|---|
| | un curso. | |
| 11 | Generar reporte con los nombres de todos los cursos optativos. | 1 |
| 12 | Generar reporte con los nombres de todos los cursos electivos. | 1 |

2.7 Plan de entregas

El plan de entregas especifica exactamente que historias de usuario serán implementadas en cada entrega del sistema, en qué fecha deben ser entregadas y la prioridad que tiene cada una de estas.

Tabla 22 Plan de Entregas

| Plan de Entregas | | | | |
|---|--|------------------------------|-------------------------|--|
| Sistema de Gestión de Matricula de Cursos (SGMC) | | | | |
| Nombre del Documentador: | | Yuleisy Galloso Soler | | |
| Entrega No: | | 3 | | |
| Historias de usuario a implementar en la Entrega | | | | |
| No HU | Título | Prioridad | Fecha de entrega | Liberación en la que se incluye |
| 1 | Autenticar usuario | Alta | 28/10/13 | 1 |
| 2 | Asignar permisos | Alta | 11/11/13 | 1 |
| 3 | Adicionar curso | Alta | 25/11/13 | 1 |
| 4 | Modificar curso | Alta | 9/12/13 | 1 |
| 5 | Eliminar curso | Alta | 15/12/13 | 1 |
| 6 | Matricular cursos para tercero | Alta | 13/01/14 | 2 |
| 7 | Matricular cursos para cuarto | Alta | 24/01/14 | 2 |
| 8 | Matricular cursos para quinto | Alta | 8/02/14 | 2 |
| 9 | Mostrar el listado de matrícula de un curso. | Media | 20/02/14 | 2 |
| 10 | Generar reporte con el listado de los | Media | 17/03/14 | 3 |

| | | | | |
|----|--|-------|----------|---|
| | estudiantes matriculados en un curso. | | | |
| 11 | Generar reporte con los nombres de todos los cursos optativos. | Media | 3/04/14 | 3 |
| 12 | Generar reporte con los nombres de todos los cursos electivos. | Media | 10/04/14 | 3 |

2.8 Arquitectura propuesta

Los patrones arquitectónicos son plantillas que describen los principios estructurales globales que construyen las distintas arquitecturas de software viables. Plantean una organización estructural fundamental para un sistema de software, expresando un conjunto de subsistemas predefinidos, especificando responsabilidades y organizando las relaciones entre ellos. (36)

El patrón arquitectónico seleccionado para el desarrollo de la solución es el que se describe a continuación:

Patrón Arquitectónico tres capas

En el caso del desarrollo del sistema de la presente investigación se ha definido el diseño de una arquitectura distribuida en tres capas, el dividir una aplicación en capas separadas que desempeñan diferentes roles y funcionalidades, ayuda a mejorar el mantenimiento del código, permite también diferentes tipos de despliegue y contribuye a organizar, reutilizar y rediseñar mejor los componentes que integran el sistema.

Diseño de capas:

- **Presentación:** Es la capa que se encarga de presentar información e interactuar con la capa externa (usuario). Se define en general como Interfaz Gráfica de Usuario (GUI).
- **Lógica de aplicación:** Es la capa que se encarga de implementar las operaciones solicitadas por los usuarios a través de la capa de presentación.
- **Acceso a datos:** Es la que permite el acceso a la base de datos del sistema.

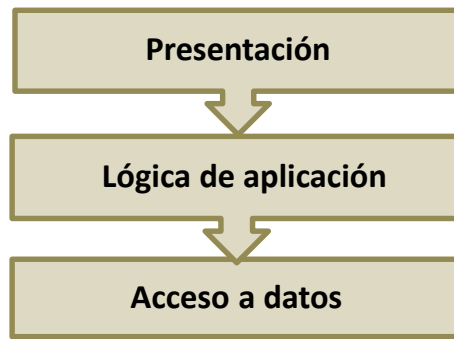


Ilustración 1 Arquitectura en capas del sistema.

2.9 Patrones de diseño

Un patrón de diseño provee un esquema para refinar componentes de un sistema de software y la forma en que se relacionan entre sí. Describe una estructura generalmente recurrente de comunicación de componentes que resuelve un problema de diseño general dentro de un contexto particular. (37)

A continuación se definen algunos patrones a utilizar:

Patrones GRASP

Experto: Este patrón se utiliza para asignar la responsabilidad que tendrá cada clase, en dependencia de la información que maneja. Se evidencia en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información manejada dentro del sistema, como por ejemplo las clases controladoras.

Creador: Este patrón se utiliza para asignar a determinadas clases la responsabilidad de crear instancias de otras. En las clases controladoras se encuentran definidas las acciones del sistema y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase controladora es "creador" de dichas entidades.

Bajo Acoplamiento: Este patrón es evidenciado en la distribución del sistema, cada clase controladora maneja la información de las entidades que están relacionadas con las funcionalidades que se especifican en dicha clase controladora para lograr un bajo acoplamiento de clases.

2.10 Tarjetas Clases-Responsabilidades-Colaboradores(CRC)

Las Tarjetas CRC sirven para diseñar el sistema en conjunto entre todo el equipo, permiten reducir el modo de pensar procedural y apreciar la tecnología de objetos. Además de contribuir en una mejor construcción de las clases que integran el sistema.
(38)

Las siguientes tarjetas CRC describen las clases que conforman el sistema:

Tabla 23 Tarjeta CRC 1

| Clase: ContenedorVertical | |
|---|--|
| Descripción: Es la clase que se encarga de gestionar toda la información referente a los cursos y los reportes de dicha información. | |
| Responsabilidades: <ul style="list-style-type: none"> • Adiciona los cursos. • Modifica los cursos. • Elimina los cursos. • Emite reporte con la información de los cursos optativos. • Emite reporte con la información de los cursos electivos. | Colaboradores: <ul style="list-style-type: none"> • Ventana • FabricaCampos • Utilidades • UploadTest |

Tabla 24 Tarjeta CRC 2

| Clase: ContenedorVerticalP | |
|--|--|
| Descripción: Es la clase que controla la información de los estudiantes matriculados en los distintos cursos y el reporte correspondiente a dicha información. | |
| Responsabilidades: <ul style="list-style-type: none"> • Listar los estudiantes matriculados según el curso seleccionado. • Emite reporte con la información de los estudiantes matriculados en el curso seleccionado. | Colaboradores: <ul style="list-style-type: none"> • Utilidades • ContenedorVerticalE3 • ContenedorVerticalE4 • ContenedorVerticalE5 |

Tabla 25 Tarjeta CRC 3

| Clase: ContenedorVerticalE3 | |
|--|--|
| Descripción: Es la clase que contiene la información de los cursos correspondientes a los estudiantes de tercer año y controla además la matrícula de cursos. | |
| Responsabilidades: <ul style="list-style-type: none"> • Muestra la información de los cursos correspondiente a los estudiantes de tercer año. • Permite matricular en los cursos. | Colaboradores: <ul style="list-style-type: none"> • ContenedorVertical |

Tabla 26 Tarjeta CRC 4

| Clase: ContenedorVerticalE4 | |
|--|--|
| Descripción: Es la clase que contiene la información de los cursos correspondientes a los estudiantes de cuarto año y controla además la matrícula de cursos. | |
| Responsabilidades: <ul style="list-style-type: none"> • Muestra la información de los cursos correspondiente a los estudiantes de cuarto año. • Permite matricular en los cursos. | Colaboradores: <ul style="list-style-type: none"> • ContenedorVertical |

Tabla 27 Tarjeta CRC 5

| Clase: ContenedorVerticalE5 | |
|--|--|
| Descripción: Es la clase que contiene la información de los cursos correspondientes a los estudiantes de quinto año y controla además la matrícula de cursos. | |
| Responsabilidades: <ul style="list-style-type: none"> • Muestra la información de los cursos correspondiente a los estudiantes de quinto año. • Permite matricular en los cursos. | Colaboradores: <ul style="list-style-type: none"> • ContenedorVertical |

Tabla 28 Tarjeta CRC 6

| | |
|---|-----------------------|
| Clase: FabricaCampos | |
| Descripción: Es la clase que define el tipo de interfaz que recibe la información de los cursos. | |
| Responsabilidades: <ul style="list-style-type: none"> Definir interfaz de recepción para cada uno de los datos correspondientes a los cursos. | Colaboradores: |

Tabla 29 Tarjeta CRC 7

| | |
|---|---|
| Clase: Ventana | |
| Descripción: Es la clase que controla la interfaz que permite adicionar o modificar los cursos. | |
| Responsabilidades: <ul style="list-style-type: none"> Mostrar un formulario que permite adicionar o modificar los cursos. | Colaboradores: <ul style="list-style-type: none"> FabricaCampos |

Tabla 30 Tarjeta CRC 8

| | |
|--|--|
| Clase: Utilidades | |
| Descripción: Es la clase que se encarga de confeccionar documentos.pdf, para emitir los distintos reportes a una dirección previamente especificada. | |
| Responsabilidades: <ul style="list-style-type: none"> Confeccionar reportes con la información de los cursos. Confeccionar reportes con la información de los estudiantes matriculados en los cursos. | Colaboradores: <ul style="list-style-type: none"> ContenedorVertical |

Tabla 31 Tarjeta CRC 9

| | |
|---|-----------------------|
| Clase: UploadTest | |
| Descripción: Es la clase que permite subir al servidor de la aplicación documentos.pdf, en este caso la descripción de la temática de cada curso. | |
| Responsabilidades: <ul style="list-style-type: none"> Subir al servidor de la aplicación la descripción de la temática de cada curso. | Colaboradores: |

Tabla 32 Tarjeta CRC10

| | |
|---|---|
| Clase: GexmacApplication | |
| Descripción: Es la clase que se encarga de compilar cada una de las clases principales. | |
| Responsabilidades: <ul style="list-style-type: none"> Compila las clases principales. Implementa la interfaz principal de la aplicación. | Colaboradores: <ul style="list-style-type: none"> ContenedorVertical ContenedorVerticalP ContenedorVerticalE3 ContenedorVerticalE4 ContenedorVerticalE5 |

Tabla 33 Tarjeta CRC11

| | |
|--|---|
| Clase: Controladora | |
| Descripción: Es la clase que se encarga de obtener la persona autenticada en el sistema a partir de la implementación del autenticar usuario. | |
| Responsabilidades: <ul style="list-style-type: none"> Obtener la persona autenticada | Colaboradores: <ul style="list-style-type: none"> autenticacion2.v7.AutenticarUsuario autenticacion2.v7.AutenticarUsuarioResponse autenticacion2.v7.PasarelaAutenticacionWSSStub autenticacion2.v7.Persona |

Tabla 34 Tarjeta CRC12

| Clase: VLEstudiantes | |
|---|--|
| Descripción: Es la clase que se encarga de la implementación de la interfaz que se le muestra al estudiante para que seleccione su año y así mostrarle los cursos que le corresponden. | |
| Responsabilidades: <ul style="list-style-type: none"> • Es la clase intermedia para mostrarle los cursos a los estudiantes. | Colaboradores: <ul style="list-style-type: none"> • ContenedorVerticalE3 • ContenedorVerticalE4 • ContenedorVerticalE5 |

Tabla 35 Tarjeta CRC13

| Clase: DatabaseHelper | |
|--|-----------------------|
| Descripción: Es la clase a través de la cual se accede a la base de datos. | |
| Responsabilidades: <ul style="list-style-type: none"> • Con su conexión a la base de datos permite mostrar los cursos. | Colaboradores: |

Tabla 36 Tarjeta CRC14

| Clase: LoginLayout | |
|---|-----------------------|
| Descripción: Es la clase que implementa el autenticar usuario. | |
| Responsabilidades: <ul style="list-style-type: none"> • Verificar las validaciones para autenticarse. | Colaboradores: |

2.11 Modelo de la base de datos

2.11.1 Diagrama entidad relación de la Base de Datos

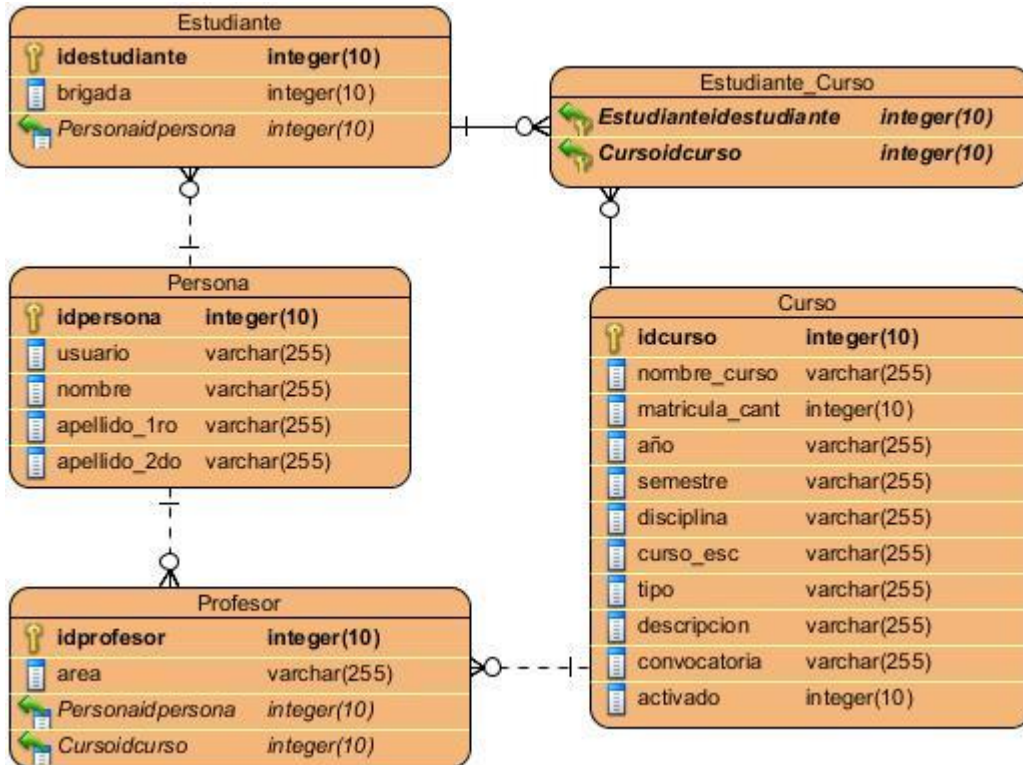


Ilustración 2 Diagrama entidad relación de la Base de Datos

2.12 Consideraciones parciales

En este capítulo se expone brevemente una descripción de la propuesta de solución.. Se especifican los requisitos funcionales y no funcionales a través de la descripción de historias de usuario. Se conforma el plan de entregas correspondiente a la evolución y realización de la implementación. Se define la arquitectura del sistema a desarrollar así como los patrones de diseño a utilizar. Además se elaboran un conjunto de tarjetas CRC dirigidas a la descripción de las clases que conforman la aplicación. Concluyendo con el diseño del diagrama entidad relación de la base de datos. Todo esto proporciona las bases o esquema para la implementación del sistema propuesto.

CAPÍTULO 3: Validación de la Propuesta de Solución

3.1 Introducción

En el presente capítulo se lleva a cabo la culminación de la implementación del sistema. Se planifica un conjunto de iteraciones de acuerdo al proceso de implementación, donde se involucra la elaboración de las tareas de ingeniería que le dan cumplimiento a cada una de las historias de usuario implementadas, culminando cada una con la realización de un conjunto de pruebas de aceptación.

3.2 Iteraciones

A continuación se muestran las iteraciones planificadas para el sistema informático desarrollado durante el proceso de implementación. Donde se describirán las tareas de ingeniería y pruebas de aceptación correspondiente a cada historia de usuario.

3.2.1 Primera iteración

Tabla 37 Planificación de las Tareas de Ingeniería de la primera iteración.

| Historia de usuario | Tareas de ingeniería | Puntos estimados |
|---------------------|--|------------------|
| Autenticar usuario | <ul style="list-style-type: none">• Diseño del prototipo de la interfaz principal de la aplicación para autenticar usuario.• Implementar código para autenticar usuario, específicamente la clase LoginLayout en conjunto con las clases que se involucran para obtener los datos del LDAP. | 2 |
| Asignar permisos | <ul style="list-style-type: none">• Implementación de código para la asignación de permisos a cada usuario según su rol. | 2 |
| Adicionar curso | <ul style="list-style-type: none">• Implementación de la clase FabricaCampos.• Iniciar la implementación de la clase Ventana.• Implementación de la clase UploadTest. | 2 |

| | | |
|-----------------|--|---|
| | <ul style="list-style-type: none"> Implementación del método adicionar de la clase ContenedorVertical. | |
| Modificar curso | <ul style="list-style-type: none"> Culminar la implementación de la clase Ventana. Implementación del método modificar de la clase ContenedorVertical. | 1 |
| Eliminar curso | <ul style="list-style-type: none"> Implementación del método eliminar de la clase ContenedorVertical. | 1 |

3.2.1.1 Descripción de las tareas de ingeniería de la primera iteración

Tabla 38 Diseño del prototipo de la interfaz principal de la aplicación para autenticar usuario.

| Tarea | |
|---|----------------------|
| Número tarea: 1 | Número historia: 1 |
| Nombre tarea: Diseño del prototipo de la interfaz principal de la aplicación para autenticar usuario. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 28/10/2013 | Fecha fin: 2/11/2013 |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: La interfaz principal de la aplicación está diseñada de la siguiente manera: en el centro de la interfaz tiene un modelo de autenticación con un campo para insertar el usuario y otro para insertar la contraseña | |

Tabla 39 Implementar código para autenticar usuario.

| Tarea | |
|---|---------------------|
| Número tarea: 2 | Número historia: 1 |
| Nombre tarea: Implementar código para autenticar usuario. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |

| | |
|---|----------------------------|
| Fecha inicio: 4/11/2013 | Fecha fin:9/11/2013 |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: El usuario accede al sistema a través de la web, entra al sistema mediante un formulario previamente implementado que posee un campo para usuario y otro para la contraseña, si es usuario del sistema pues accede a la página de inicio con las funcionalidades que les corresponden. Aquí se involucra la implementación de la clase LoginLayout en conjunto con las clases que se involucran para obtener los datos del LDAP. | |

Tabla 40 Implementación de código para la asignación de permisos a cada usuario según su rol.

| | |
|---|------------------------------|
| Tarea | |
| Número tarea: 3 | Número historia: 2 |
| Nombre tarea: Implementación de código para la asignación de permisos a cada usuario según su rol. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 2 |
| Fecha inicio: 11/11/2013 | Fecha fin: 22/11/2013 |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: Al acceder a la página principal y autenticarse, si es usuario del sistema este identifica los datos de la persona que accede a la aplicación, por lo que se debe mostrar directamente la página a la cual debe acceder, según su nivel de permisos, permitiéndole ejecutar las acciones necesarias según su rol. La clase donde se implementan y se controlan los permisos es en GexmacApplication que viene siendo la interfaz que se muestra después de autenticado el usuario. | |

Tabla 41 Implementación de la clase FabricaCampos.

| | |
|------------------------|---------------------------|
| Tarea | |
| Número tarea: 4 | Número historia: 3 |

| | |
|---|-----------------------------|
| Nombre tarea: Implementación de la clase FabricaCampos. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 25/11/2013 | Fecha fin:30/11/2013 |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: <p>Es la clase que crea un formulario a partir de cada tipo de campo que es definido por cada tipo de dato que conforma un curso. Este formulario es uno de los componentes que se integra a la clase Ventana.</p> | |

Tabla 42 Iniciar la implementación de la clase Ventana.

| | |
|--|------------------------------|
| Tarea | |
| Número tarea: 5 | Número historia: 3 |
| Nombre tarea: Iniciar la implementación de la clase Ventana. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 25/11/2013 | Fecha fin: 30/11/2013 |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: <p>Es la clase que crea una especie de ventana, que contiene el formulario que se muestra cuando se accede a adicionar un curso, incluyendo en su interfaz los botones guardar y cancelar. Por lo que incluye un método para ejecutar la operación adicionar los datos de un curso.</p> | |

Tabla 43 Implementación de la clase UploadTest.

| | |
|---|-----------------------------|
| Tarea | |
| Número tarea: 6 | Número historia: 3 |
| Nombre tarea: Implementación de la clase UploadTest. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 2/12/2013 | Fecha fin: 7/12/2013 |

| |
|---|
| Programador responsable: Yuleisy Galloso Soler |
| <p>Descripción:</p> <p>Es la clase que se encarga de subir en este caso documentos.pdf, a una determinada dirección del servidor de la aplicación. Se ejecuta a partir de la implementación de los métodos (Upload.Receiver y Upload.SucceededListener).</p> |

Tabla 44 Implementación del método adicional de la clase ContenedorVertical.

| | |
|--|-----------------------------|
| Tarea | |
| Número tarea: 7 | Número historia: 3 |
| Nombre tarea: Implementación del método adicional de la clase ContenedorVertical. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 2/12/2013 | Fecha fin: 7/12/2013 |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción:</p> <p>Si se presiona la opción (Adicionar Curso), automáticamente se abre una ventana con un formulario, conformado por distintos campos donde según el tipo de dato se deberá ir llenando con la información del curso a adicionar.</p> <p>En el caso donde se debe subir el documento de la descripción del curso, se encontrara una opción llamada (Examinar) al presionar en esta opción deberá salir una ventana para buscar el archivo que desea subir, en esta misma ventana al seleccionar el archivo se presiona la opción (Guardar), al ver el archivo en la página se presiona la opción (Upload) y debe salir el mensaje (Archivo Subido).</p> <p>Una vez concluido la inserción de los datos del curso, se presiona en la opción (Guardar) en el caso de guardar los datos, sino se presiona en la opción (Cancelar).</p> | |

Tabla 45 Culminar la implementación de la clase Ventana.

| | |
|------------------------|---------------------------|
| Tarea | |
| Número tarea: 8 | Número historia: 4 |

| | |
|--|------------------------------|
| Nombre tarea: Culminar la implementación de la clase Ventana. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 9/12/2013 | Fecha fin: 14/12/2013 |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: <p>Es la clase que crea una especie de ventana, que contiene el formulario que se muestra cuando se accede a modificar un curso, incluyendo en su interfaz los botones guardar y cancelar. Por lo que incluye un método para ejecutar la operación modificar los datos de un curso.</p> | |

Tabla 46 Implementación del método modificar de la clase ContenedorVertical.

| | |
|---|------------------------------|
| Tarea | |
| Número tarea: 9 | Número historia: 4 |
| Nombre tarea: Implementación del método modificar de la clase ContenedorVertical. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 9/12/2013 | Fecha fin: 14/12/2013 |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: <p>Al mostrarse la interfaz donde se encuentran los cursos registrados, se puede apreciar que en la parte derecha al terminar la información de cada curso, se encuentra la opción modificar, al presionar en esta opción se abre una ventana con todos los campos correspondiente a los datos del curso seleccionado, una vez aquí se modifican los datos que se deseen cambiar y se presiona en la opción (Guardar) en el caso de guardar los datos, sino se presiona en la opción (Cancelar).</p> | |

Tabla 47 Implementación del método eliminar de la clase ContenedorVertical.

| | |
|-------------------------|---------------------------|
| Tarea | |
| Número tarea: 10 | Número historia: 5 |

| | |
|--|------------------------------|
| Nombre tarea: Implementación del método eliminar de la clase ContenedorVertical. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 15/12/2013 | Fecha fin: 19/12/2013 |
| Programador responsable: Yuleisy Galloso Soler | |
| Descripción: <p>Al mostrarse la interfaz donde se encuentran los cursos registrados, se puede apreciar que en la parte derecha al terminar la información de cada curso, se encuentra la opción eliminar, al presionar en esta opción debe mostrarse un mensaje de si (Esta seguro que desea eliminar el curso) en conjunto a este mensaje debe aparecer las opciones (Si) y (No).</p> | |

3.2.1.2 Pruebas de aceptación de la primera iteración

- Historia de usuario 1

Tabla 48 Probar la funcionalidad Autenticar usuario.

| Caso de prueba de aceptación 1 |
|---|
| Historia de usuario: Autenticar usuario |
| Nombre: Probar la funcionalidad Autenticar usuario. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: <p>Probar que el sistema les permita a los usuarios autenticarse correctamente y que tengan acceso a las funcionalidades que le corresponden según su nivel de permisos.</p> |
| Condiciones de ejecución: <p>Debe ser usuario del sistema.</p> |
| Entradas/Pasos de ejecución: <p>Se inserta usuario y contraseña</p> |

| |
|---|
| Resultado esperado: Se autentico el usuario correctamente. |
| Evaluación de la prueba: Satisfactoria. |

Ver en Anexo 2

Alternativa 1 Caso de prueba de aceptación 1

Alternativa 2 Caso de prueba de aceptación 1

- Historia de usuario 2

Tabla 49 Probar la funcionalidad Asignar permisos.

| Caso de prueba de aceptación 2 |
|--|
| Historia de usuario: Asignar permisos |
| Nombre: Probar la funcionalidad Asignar permisos. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que el sistema le muestra a cada usuario las funcionalidades que le corresponden según su rol. |
| Condiciones de ejecución: El usuario debe autenticarse y tener permisos previamente asignados. |
| Entradas/Pasos de ejecución: Se inserta usuario y contraseña |
| Resultado esperado: Se ejecutan los permisos de usuario correctamente y se le muestran las funcionalidades que le corresponden. |
| Evaluación de la prueba: Satisfactoria. |

- Historia de usuario 3

Tabla 50 Probar la funcionalidad Adicionar curso.

| Caso de prueba de aceptación 3 |
|---|
| Historia de usuario: Adicionar curso |

| |
|--|
| Nombre: Probar la funcionalidad Adicionar curso. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que al adicionar los datos de un curso estos se guardan y se muestran correctamente. |
| Condiciones de ejecución: El usuario debe estar autenticado como administrador del sistema para acceder a esta funcionalidad. |
| Entradas/Pasos de ejecución: Una vez autenticado el administrador del sistema se le mostrara una interfaz y adicionara un curso con los siguientes pasos: <ul style="list-style-type: none"> • Presionar opción (Listado de cursos). • Presionar opción (Adicionar curso). • Insertar los datos del curso: nombre del curso, cantidad de estudiantes a matricular, año de los estudiantes que les corresponde el curso, semestre en que se impartirá, disciplina a la que pertenece, curso escolar, tipo de curso: electivo o optativo, descripción del curso, profesor que impartirá el curso y las fechas de la convocatoria de matrícula del curso. • Presionar la opción (Registrar) |
| Resultado esperado: Se adiciona correctamente la información de los cursos. |
| Evaluación de la prueba: Satisfactoria. |

Ver en Anexo 3

Alternativa 1 Caso de prueba de aceptación 3

- **Historia de usuario 4**

Tabla 51 Probar la funcionalidad Modificar curso.

| |
|---|
| Caso de prueba de aceptación 4 |
| Historia de usuario: Modificar curso |

| |
|---|
| Nombre: Probar la funcionalidad Modificar curso. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que al modificar los datos de un curso estos se guardan y se muestran correctamente. |
| Condiciones de ejecución: El usuario debe estar autenticado como administrador del sistema para acceder a esta funcionalidad. |
| Entradas/Pasos de ejecución: Una vez autenticado el administrador del sistema se le mostrara una interfaz y modificara un curso con los siguientes pasos: <ul style="list-style-type: none"> • Presionar opción (Listado de cursos). • Buscar y Presionar en la opción (Modificar). • Modificar los datos que se deseen de los que se mencionan a continuación: nombre del curso, cantidad de estudiantes a matricular, año de los estudiantes que les corresponde el curso, semestre en que se impartirá, disciplina a la que pertenece, curso escolar, tipo de curso: electivo o optativo, descripción del curso, profesor que impartirá el curso y las fechas de la convocatoria de matrícula del curso. • Presionar la opción (Modificar) |
| Resultado esperado: Se modifica correctamente la información de los cursos. |
| Evaluación de la prueba: Satisfactoria. |

Ver en Anexo 4

Alternativa 1 Caso de prueba de aceptación 4

- **Historia de usuario 5**

Tabla 52 Probar la funcionalidad Eliminar curso.

Caso de prueba de aceptación 5

| |
|---|
| Historia de usuario: Eliminar curso |
| Nombre: Probar la funcionalidad Eliminar curso. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que al eliminar un curso se ejecute correctamente esta funcionalidad. |
| Condiciones de ejecución: El usuario debe estar autenticado como administrador del sistema para acceder a esta funcionalidad. |
| Entradas/Pasos de ejecución: Una vez autenticado el administrador del sistema se le mostrara una interfaz y eliminara un curso con los siguientes pasos: <ul style="list-style-type: none"> • Presionar opción (Listado de cursos). • Buscar y Presionar en la opción (Eliminar). • Se mostrara un mensaje: “Esta seguro que desea eliminar el curso“. • Presionar la opción (Si) |
| Resultado esperado: Se elimina el curso correctamente. |
| Evaluación de la prueba: Satisfactoria. |

Ver en Anexo 5

Alternativa 1 Caso de prueba de aceptación 5

3.2.2 Segunda Iteración

Tabla 53 Planificación de las Tareas de Ingeniería de la segunda iteración.

| Historia de usuario | Tareas de ingeniería | Puntos estimados |
|--------------------------------|---|------------------|
| Matricular cursos para tercero | <ul style="list-style-type: none"> • Implementación del método Matricular de la clase ContenedorVerticalE3 | 1 |
| Matricular cursos para cuarto | <ul style="list-style-type: none"> • Implementación del método Matricular de la clase ContenedorVerticalE4 | 1 |

| | | |
|--|--|---|
| Matricular cursos para quinto | <ul style="list-style-type: none"> Implementación del método Matricular de la clase ContenedorVerticalE5. | 1 |
| Mostrar el listado de matrícula de un curso. | <ul style="list-style-type: none"> Implementación del método ListarEstudiantes de la clase ContenedorVerticalP. | 1 |

3.2.2.1 Descripción de las tareas de ingeniería de la segunda iteración

Tabla 54 Implementación del método Matricular de la clase ContenedorVerticalE3.

| Tarea | |
|---|------------------------------|
| Número tarea: 1 | Número historia: 6 |
| Nombre tarea: Implementación del método Matricular de la clase ContenedorVerticalE3. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 13/01/2014 | Fecha fin: 18/01/2014 |

Programador responsable: Yuleisy Galloso Soler

Descripción:

La implementación del método (Matricular) permite matricular en cada uno de los cursos, mostrándose esta opción al final de la información de cada curso que se muestra en la interfaz que representa la clase ContenedorVerticalE3. Al presionar en esta opción automáticamente se ejecutan varias verificaciones y se le muestra a la persona que accedió a matricular uno de estos tres mensajes:

En el caso del mensaje 1 se incluye la implementación de las clases JDBCUtil que es una de las clases que se conecta con la base de datos y la clase MatricularDAO que la contiene varias consultas para poder insertar en la BD los datos de la persona matriculada.

Mensaje 1: Se ha matriculado correctamente

Mensaje 2: No puede matricular en este curso porque ya no quedan capacidades de matrícula.

Mensaje 3: No puede matricular en este curso porque ya usted está matriculado en este curso.

Tabla 55 Implementación del método Matricular de la clase ContenedorVerticalE4.

| Tarea | |
|---|------------------------------|
| Número tarea: 2 | Número historia: 7 |
| Nombre tarea: Implementación del método Matricular de la clase ContenedorVerticalE4. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 24/01/2014 | Fecha fin: 31/01/2014 |

Programador responsable: Yuleisy Galloso Soler

Descripción:

La implementación del método (Matricular) permite matricular en cada uno de los cursos, mostrándose esta opción al final de la información de cada curso que se muestra en la interfaz que representa la clase ContenedorVerticalE4. Al presionar en esta opción automáticamente se ejecutan varias verificaciones y se le muestra a la persona que accedió a matricular uno de estos tres mensajes:

En el caso del mensaje 1 se incluye la implementación de las clases JDBCUtil que es una de las clases que se conecta con la base de datos y la clase MatricularDAO que la contiene varias consultas para poder insertar en la BD los datos de la persona matriculada.

Mensaje 1: Se ha matriculado correctamente

Mensaje 2: No puede matricular en este curso porque ya no quedan capacidades de matrícula.

Mensaje 3: No puede matricular en este curso porque ya usted está matriculado en este curso.

Tabla 56 Implementación del método Matricular de la clase ContenedorVerticalE5.

| Tarea | |
|---|------------------------------|
| Número tarea: 3 | Número historia: 8 |
| Nombre tarea: Implementación del método Matricular de la clase ContenedorVerticalE5. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 8/02/2014 | Fecha fin: 15/02/2014 |

Programador responsable: Yuleisy Galloso Soler

Descripción:

La implementación del método (Matricular) permite matricular en cada uno de los cursos, mostrándose esta opción al final de la información de cada curso que se muestra en la interfaz que representa la clase ContenedorVerticalE5. Al presionar en esta opción automáticamente se ejecutan varias verificaciones y se le muestra a la persona que accedió a matricular uno de estos tres mensajes:

En el caso del mensaje 1 se incluye la implementación de las clases JDBCUtil que es una de las clases que se conecta con la base de datos y la clase MatricularDAO que la contiene varias consultas para poder insertar en la BD los datos de la persona matriculada.

Mensaje 1: Se ha matriculado correctamente

Mensaje 2: No puede matricular en este curso porque ya no quedan capacidades de matrícula.

Mensaje 3: No puede matricular en este curso porque ya usted está matriculado en este curso.

Tabla 57 Implementación del método ListarEstudiantes de la clase ContenedorVerticalP.

| Tarea | |
|---|------------------------------|
| Número tarea: 4 | Número historia: 9 |
| Nombre tarea: Implementación del método ListarEstudiantes de la clase ContenedorVerticalP. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 20/02/2014 | Fecha fin: 25/02/2014 |

Programador responsable: Yuleisy Galloso Soler

Descripción:

La implementación del método (ListarEstudiantes) permite que a partir de la realización de consultas a la base de datos de la aplicación se pueda conformar el listado de matrícula de estudiantes de un curso previamente seleccionado. Mostrándose nombre y apellidos de cada estudiante matriculado.

3.2.2.2 Pruebas de aceptación de la segunda iteración

- Historia de usuario 6

Tabla 58 Probar la funcionalidad Matricular cursos para tercero.

| Caso de prueba de aceptación 6 |
|---|
| Historia de usuario: Matricular cursos para tercero |
| Nombre: Probar la funcionalidad Matricular cursos para tercero. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que un estudiante de tercero se matricule correctamente. |
| Condiciones de ejecución: El usuario debe estar registrado y tener permisos para acceder a esta funcionalidad. En este caso un estudiante. |
| Entradas/Pasos de ejecución: Una vez autenticado un estudiante se le mostrara una interfaz y matriculara un curso con los siguientes pasos: <ul style="list-style-type: none">• Presionar opción (Cursos disponibles).• Seleccionar el año que cursa• Se le muestra una interfaz con todos los cursos de su año y aquí presiona en el matricular del curso que desee. |
| Resultado esperado: Se muestra un mensaje (Se ha matriculado correctamente). |

Evaluación de la prueba: Satisfactoria.

Ver en Anexo 6

Alternativa 1 Caso de prueba de aceptación 6

Alternativa 2 Caso de prueba de aceptación 6

- **Historia de usuario 7**

Tabla 59 Probar la funcionalidad Matricular cursos para cuarto.

| Caso de prueba de aceptación 7 |
|---|
| Historia de usuario: Matricular cursos para cuarto |
| Nombre: Probar la funcionalidad Matricular cursos para cuarto. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que un estudiante de cuarto se matricule correctamente. |
| Condiciones de ejecución: El usuario debe estar registrado y tener permisos para acceder a esta funcionalidad. En este caso un estudiante. |
| Entradas/Pasos de ejecución: Una vez autenticado un estudiante se le mostrara una interfaz y matriculara un curso con los siguientes pasos: <ul style="list-style-type: none">• Presionar opción (Cursos disponibles).• Seleccionar el año que cursa• Se le muestra una interfaz con todos los cursos de su año y aquí presiona en el matricular del curso que desee. |
| Resultado esperado: Se muestra un mensaje (Se ha matriculado correctamente). |
| Evaluación de la prueba: Satisfactoria. |

- **Historia de usuario 8**

Tabla 60 Probar la funcionalidad Matricular cursos para quinto.

| Caso de prueba de aceptación 8 |
|--|
| Historia de usuario: Matricular cursos para quinto |
| Nombre: Probar la funcionalidad Matricular cursos para quinto. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que un estudiante de quinto se matricule correctamente. |
| Condiciones de ejecución: El usuario debe estar registrado y tener permisos para acceder a esta funcionalidad. En este caso un estudiante. |
| Entradas/Pasos de ejecución: Una vez autenticado un estudiante se le mostrara una interfaz y matriculara un curso con los siguientes pasos: <ul style="list-style-type: none"> • Presionar opción (Cursos disponibles). • Seleccionar el año que cursa Se le muestra una interfaz con todos los cursos de su año y aquí presiona en el matricular del curso que desee. |
| Resultado esperado: Se muestra un mensaje (Se ha matriculado correctamente). |
| Evaluación de la prueba: Satisfactoria. |

- **Historia de usuario 9**

Tabla 61 Probar la funcionalidad Mostrar el listado de matrícula de un curso.

| Caso de prueba de aceptación 9 |
|--|
| Historia de usuario: Mostrar el listado de matrícula de un curso. |
| Nombre: Probar la funcionalidad Mostrar el listado de matrícula de un curso. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que al acceder a mostrar un listado de matrícula este se muestre |

| |
|--|
| correctamente |
| <p>Condiciones de ejecución:</p> <p>El usuario debe estar registrado y tener permisos para acceder a esta funcionalidad.</p> <p>En este caso un profesor.</p> |
| <p>Entradas/Pasos de ejecución:</p> <p>Una vez autenticado un profesor se le mostrara una interfaz y podrá ver los listados de matrícula de los cursos que el imparte con los siguientes pasos:</p> <ul style="list-style-type: none"> • Presionar opción (Datos de mi curso). • Seleccionar el curso del cual quiere ver el listado de matrícula. <p>Se le muestra una interfaz con todos los nombres y apellidos de los estudiantes matriculados.</p> |
| <p>Resultado esperado: La operación se efectúa correctamente.</p> |
| <p>Evaluación de la prueba: Satisfactoria.</p> |

3.2.3 Tercera iteración

Tabla 62 Planificación de las Tareas de Ingeniería de la tercera iteración.

| Historia de usuario | Tareas de ingeniería | Puntos estimados |
|---|---|------------------|
| Generar reporte con el listado de los estudiantes matriculados en un curso. | <ul style="list-style-type: none"> • Implementación de la clase Utilidades para reporte de matrícula. | 2 |
| Generar reporte con los nombres de todos los cursos optativos. | <ul style="list-style-type: none"> • Implementación de la clase Utilidades para reporte de cursos optativos. | 1 |
| Generar reporte con los nombres de | <ul style="list-style-type: none"> • Implementación de la clase Utilidades para reporte de cursos electivos. | 1 |

| | | |
|-----------------------------|--|--|
| todos los cursos electivos. | | |
|-----------------------------|--|--|

3.2.3.1 Descripción de las tareas de ingeniería de la tercera iteración

Tabla 63 Implementación de la clase Utilidades para reporte de matrícula.

| Tarea | |
|--|-----------------------|
| Número tarea: 1 | Número historia: 10 |
| Nombre tarea: Implementación de la clase Utilidades para reporte de matrícula. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 2 |
| Fecha inicio: 17/03/2014 | Fecha fin: 28/03/2014 |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción:</p> <p>Es la clase que se encarga de crear los distintos reportes a partir de la información de los cursos. En esta clase se implementa el método (GuardarPDF_PruebasLE) para crear el reporte referente al listado de matrícula de estudiantes correspondiente a cada curso, quedando reflejado nombre, apellidos y brigada de cada estudiante que matriculo en el curso seleccionado.</p> | |

Tabla 64 Implementación de la clase Utilidades para reporte de cursos optativos.

| Tarea | |
|---|----------------------|
| Número tarea: 2 | Número historia: 11 |
| Nombre tarea: Implementación de la clase Utilidades para reporte de cursos optativos. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 2/04/2014 | Fecha fin: 7/04/2014 |

| |
|---|
| Programador responsable: Yuleisy Galloso Soler |
| <p>Descripción: Es la clase que se encarga de crear los distintos reportes a partir de la información de los cursos. En esta clase se implementa el método (GuardarPDF_Pruebas) para crear el reporte, en complemento con la clase ContenedorVertical donde se implementa el método (Reporte_O) el cual se encarga de emitir el reporte a partir de consultas a la base de datos especificando que la única información que se consulte sea la de los cursos optativos.</p> |

Tabla 65 Implementación de la clase Utilidades para reporte de cursos electivos.

| | |
|---|------------------------------|
| Tarea | |
| Número tarea: 3 | Número historia: 12 |
| Nombre tarea: Implementación de la clase Utilidades para reporte de cursos electivos. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 10/04/2014 | Fecha fin: 15/04/2014 |
| Programador responsable: Yuleisy Galloso Soler | |
| <p>Descripción: Es la clase que se encarga de crear los distintos reportes a partir de la información de los cursos. En esta clase se implementa el método (GuardarPDF_Pruebas) para crear el reporte, en complemento con la clase ContenedorVertical donde se implementa el método (Reporte_E) el cual se encarga de emitir el reporte a partir de consultas a la base de datos especificando que la única información que se consulte sea la de los cursos electivos.</p> | |

3.2.3.2 Pruebas de aceptación de la tercera iteración

- **Historia de usuario 10**

Tabla 66 Probar la funcionalidad Generar reporte con el listado de los estudiantes

matriculados en un curso.

| Caso de prueba de aceptación 10 |
|---|
| Historia de usuario: Generar reporte con el listado de los estudiantes matriculados en un curso. |
| Nombre: Probar la funcionalidad Generar reporte con el listado de los estudiantes matriculados en un curso. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que se genere correctamente el reporte del listado de matrícula de estudiantes de un curso determinado. |
| Condiciones de ejecución: El usuario debe estar registrado y tener permisos para acceder a esta funcionalidad. En este caso un profesor. |
| Entradas/Pasos de ejecución: Una vez autenticado un profesor se le mostrara una interfaz y podrá ver los listados de matrícula de los cursos que el imparte y podrá obtener los reportes con los siguientes pasos: <ul style="list-style-type: none">• Presionar opción (Datos de mi curso).• Seleccionar el curso del cual quiere ver el listado de matrícula.• Se le muestra una interfaz con todos los nombres y apellidos de los estudiantes matriculados.• Presionar opción (Reporte de Matricula). |
| Resultado esperado: El reporte se emite correctamente. |
| Evaluación de la prueba: Satisfactoria. |

- **Historia de usuario 11**

Tabla 67 Probar la funcionalidad Generar reporte con los nombres de todos los

cursos optativos.

| Caso de prueba de aceptación 11 |
|---|
| Historia de usuario: Generar reporte con los nombres de todos los cursos optativos. |
| Nombre: Probar la funcionalidad Generar reporte con los nombres de todos los cursos optativos. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que se genere correctamente el reporte referente a la información de los cursos optativos. |
| Condiciones de ejecución: El usuario debe estar registrado como administrador para acceder a esta funcionalidad. |
| Entradas/Pasos de ejecución: Una vez autenticado el administrador del sistema se le mostrara una interfaz y podrá obtener los reportes con los siguientes pasos: <ul style="list-style-type: none">• Presionar opción (Listado de cursos).• Presionar opción (Reporte Optativo). |
| Resultado esperado: El reporte se emite correctamente. |
| Evaluación de la prueba: Satisfactoria. |

- **Historia de usuario 12**

Tabla 68 Probar la funcionalidad Generar reporte con los nombres de todos los cursos electivos.

| Caso de prueba de aceptación 12 |
|---|
| Historia de usuario: Generar reporte con los nombres de todos los cursos |

| |
|--|
| electivos. |
| Nombre: Probar la funcionalidad Generar reporte con los nombres de todos los cursos electivos. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que se genere correctamente el reporte referente a la información de los cursos electivos. |
| Condiciones de ejecución: El usuario debe estar registrado como administrador para acceder a esta funcionalidad. |
| Entradas/Pasos de ejecución: Una vez autenticado el administrador del sistema se le mostrara una interfaz y podrá obtener los reportes con los siguientes pasos: <ul style="list-style-type: none"> • Presionar opción (Listado de cursos). • Presionar opción (Reporte Electivo). |
| Resultado esperado: El reporte se emite correctamente. |
| Evaluación de la prueba: Satisfactoria. |

3.3 Resumen del resultado de las iteraciones

Por cada iteración se le realizaron pruebas de aceptación al sistema, para verificar que las funcionalidades implementadas se correspondan con las historias de usuario diseñadas, donde los resultados obtenidos fueron satisfactorios. De los trece casos de prueba de aceptación diseñados se probaron satisfactoriamente cinco en la primera iteración, nueve en la segunda iteración y trece en la tercera iteración.

El gráfico que se presenta a continuación muestra los resultados obtenidos por cada iteración.

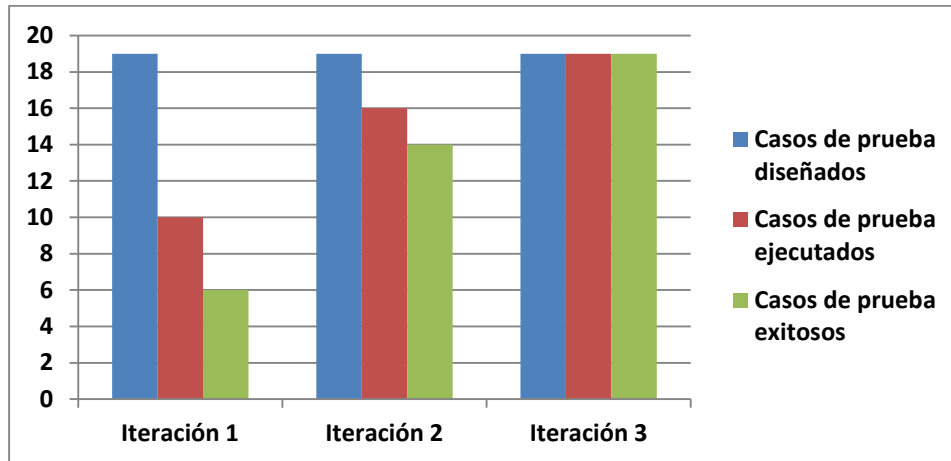


Ilustración 3 Resultado de las pruebas realizadas.

3.4 Conclusiones parciales

En el presente capítulo se llevó a cabo la planificación de las tres iteraciones correspondientes a la implementación de la aplicación, concluyendo en la conformación de 17 tareas de ingeniería que constituyeron los pasos a realizar para darle cumplimiento a las historias de usuario especificadas por el cliente. Por otra parte se realizó el diseño de 19 casos de prueba de aceptación, como mecanismo para asegurar la correcta ejecución de las funcionalidades del sistema. Cumpliendo satisfactoriamente con los requisitos definidos por el cliente.

Conclusiones generales

- Después de haber realizado un estudio bibliográfico de la situación actual y las tendencias de los sistemas de gestión académica existentes, se llegó a la conclusión de desarrollar un sistema que sea capaz de mejorar el proceso de planificación, organización y control de la matrícula de cursos optativos y electivos.
- El desarrollo del sistema fue guiado por la metodología XP. La modelación del negocio, la captura de requerimientos funcionales y no funcionales, así como los artefactos generados, sentaron las bases para la posterior implementación del sistema.
- A partir del diseño del esquema de base de datos y la ejecución de las historias de usuarios en conjunto con las tareas de ingeniería, fue construido el sistema, cumpliéndose los objetivos del presente trabajo.
- Partiendo del diseño y la aplicación de los casos de prueba, se pudo comprobar que las funcionalidades descritas, satisfacen las necesidades del proceso en cuestión garantizando así resultados satisfactorios.

Recomendaciones

Referencias Bibliográficas

1. **REGLAMENTO_DOCENTE_Y_METODOLOGICO**_con_Resolucion_del_Ministro_RES OLUCIÓN No. 210/07_ARTÍCULO 72. La Habana : s.n., 2007.
2. **REGLAMENTO_DOCENTE_Y_METODOLOGICO**_con_Resolucion_del_Ministro_RES OLUCIÓN No. 210/07_ARTÍCULO 70. La Habana : s.n., 2007.
3. **REGLAMENTO_DOCENTE_Y_METODOLOGICO**_con_Resolucion_del_Ministro_RES OLUCIÓN No. 210/07_ARTÍCULO 71. La Habana : s.n., 2007.
4. **Barrios, Norma.** La gestión de información y sus recursos(Parte I). s.l. : Bibliotecas. Edición Especial. No 1-2.[cited 3 de octubre 2013],<http://revistas.mes.edu.cu/greenstone/collect/repo/import/repo/20090606/0006176X010305.pdf>., 2001-2003.
5. **II Jornadas de trabajo del Grupo SIOU, Salamanca:** Gestión de la Información, gestión de contenidos y conocimiento,. **Arévalo, Alonso Julio.** Salamanca : http://eprints.rclis.org/11273/1/Jornadas_GRUPO_SIOU.pdf, [cited 3 de octubre 2013].
6. **Infante, Carballosa Mailin.** Sistema de Gestión de Información para el Centro de Tecnologías para la Formación de la Universidad de las Ciencias Informáticas. La Habana : <http://fec.uh.cu/CUGIO/1%20acciones/ProyectosProtocolos/15/MIC%20Mailin%20Carballosa%20Infante%20UCI.pdf>, [cited 5 de octubre 2013].
7. **Lara, Portela Lourdes.** Los Sistemas de Gestión de Información, piedra angular de la estrategia integral de gerencia. s.l. : <http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH01f2.dir/doc.pdf>, [cited 8 de octubre 2013].
8. **González, Moreiro José Antonio.** Introducción al estudio de la información y la documentación. [cited 8 de octubre 2013] : <http://www.revistas.unam.mx/index.php/ibi/article/view/3926/3478>, 1998.
9. **Universidad Alberto Hurtado.** [cited 15 de octubre 2013] : <http://www.uahurtado.cl/estudiantes/formacion-integral/cursos-optativos-de-formacion-general-ofg>.
10. **Búho.** [cited 18 de octubre 2013] : <http://www.lidesoft.com/productos/software-educativo/12-producto-buho.html>.
11. **SIGENU.** [cited 20 octubre de 2013] : <http://sigenu.mes.edu.cu:8080/dmmes/pages/info/aboutUsWelcome.faces>.

12. **Urbay, Pérez Elisabel y Herrera, Miño Noel.** Análisis y diseño del Módulo Estudiante para el sistema de gestión Académica-Akademios. [cited 20 de octubre 2013] : http://bibliodoc.uci.cu/TD/TD_0325_07.pdf, 2007.

13. **Legrá, Molina Beatriz.** Sistema de gestión académica para la facultad 3. [cited 22 de octubre 2013] : http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_05472_12, 2012.

14. **Introducción a las Aplicaciones Web.Universidad de Sevilla.** [cited 23 de octubre 2013] : <http://www.josedomingo.org/web/mod/resource/view.php?id=2337>, [10-2004] pág.2.

15. **Características de las aplicaciones web.** [cited 28 de octubre 2013] : <http://sites.google.com/site/smr2teresa/ventajas>.

16. **Morales, Cuesta Pedro.** Desarrollo de Aplicaciones Distribuidas basadas en Tecnologías Web.Dpto. Lenguajes y Sistemas Informáticos (Universidad de Vigo). [cited 28 de octubre 2013] : <http://trevinca.ei.uvigo.es/~pcuesta/publicaciones/TecWeb.pdf>.

17. **González, Hurtado Yaisel y Martínez, Ramírez Luis Felipe.** Complementos para evaluar resultados de auditorías a SGBD Oracle y SQL Server. [cited 30 de octubre 2013] : http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_05324_12.

18. **González, Pérez Yosmany.** Integración de Liferay con Openxava para el desarrollo de una plataforma para los proyectos del Sistema de Gestión del Convenio Integral de Cooperación Cuba – Venezuela Fase II. [cited 31 de octubre de 2013] : http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_04277_11.

19. **Reyes, Ochoa Alexeis Joel, y otros, y otros.** SLD070 COMPONENTE WEB PARA EL ANÁLISIS DE INFORMACIÓN CLÍNICA USANDO LA TÉCNICA DE MINERÍA DE DATOS AGRUPAMIENTO. [cited 6 de noviembre 2013] : <http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/view/168/4>.

20. **Jalón, García Javier, y otros, y otros.** Aprenda Java como si estuviera en primero. [cited 8 de noviembre 2013] : http://eva.uci.cu/file.php/106/Bibliografia_Basica/01_Aprenda_Java_como_en_Primer.pdf

21. **Pérez, Ramírez José Felipe, Pérez, Silva Rosel y Valdés, Morejón Maylevis.** Componente informático para gestionar la información en el Programa Nacional de Atención Materno-Infantil. [cited 9 de noviembre 2013] : <http://publicaciones.uci.cu/index.php/SC/article/view/1044/630>.

22. **León, Orellana Mireydis y Beatón, Alonso Yurisleis.** Análisis y Diseño de los Procesos de Gestión de Personal para Akademios v2.0. [cited 11 de noviembre 2013] : http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_1934_09.

23. **Ventajas de Java.** [cited 11 de noviembre 2013] :
<http://www.qualitrain.com.mx/objelIndirecto/javavsvbasic.htm>.
24. **Tomcat.** [cited 13 de noviembre 2013] : <http://tomcat.apache.org/>.
25. **JBoss.** [cited 14 de noviembre 2013] : <https://community.jboss.org/docs/DOC-10226?uniqueTitle=false>.
26. **Ortíz, Vazquez Yudisney, y otros, y otros.** Propuesta del gestor de bases de datos cubano basado en PostgreSQL. [cited 16 de noviembre 2013] :
<http://publicaciones.uci.cu/index.php/SC/article/view/314/284>.
27. **Hernández, Morales Angel Luis.** Sistema de Gestión de la información del conocimiento de los Recursos Humanos para los Polos Productivos de la facultad 9. [cited 18 de noviembre 2013] :
http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_03660_10.
28. **Mas, Lara Elvin y Almaguer, Herrera Dayana.** Sistema de Gestión Académica para la Vice Dirección de Docencia e Investigaciones del Policlínico - Docente Ernesto Che Guevara. [cited 18 de noviembre 2013] :
http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_02835_10.
29. **Reyes, Ochoa Alexeis Joel y García, Orellana Arturo.** Vista de análisis usando técnicas de agrupamiento para el sistema integral para la atención primaria de salud. [cited 19 de noviembre 2013] :
<http://publicaciones.uci.cu/index.php/SC/article/view/1034/582>.
30. **Tutorial_pgAdmin.pdf.** [cited 19 de noviembre 2013] :
http://eva.uci.cu/file.php/180/2._Clases/Tema_2/Materiales_complementarios/Herramientas/Tutorial_pgAdmin.pdf.
31. **Liferay.** [cited 20 de noviembre 2013] : <http://www.liferay.com/es/products/liferay-portal/overview>.
32. **Vaadin.** [cited 22 de noviembre 2013] : <http://partes.wikispaces.com/Vaadin>.
33. **Definición de metodología.** [cited 21 de noviembre 2013] :
<http://www.slideshare.net/vavp115/metodologia-de-desarrollo-de-software-7378940>.
34. **Ervin, Flores y Luis, Cordero Jorge.** UNIVERSIDAD UNION BOLIVARIANA CARRERA DE INGENIERIA DE SISTEMAS, METODOLOGIAS AGILES PROCESO UNIFICADO AGIL (AUP). [cited 21 de noviembre 2013] :
<http://www.adolfo.mex.tl/images/18149/METODOLOGIAS%20AGILES.pdf>.
35. **Patricio, Letelier y Carmen, Penadés Maria.** Metodologías ágiles para el desarrollo de software:EXtreme Programming (XP). [cited 23 de noviembre 2013] :
http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf.

36. **Wilson, Ortega.** PATRONES ARQUITECTURALES PARA APLICACIONES WEB.Ing. de software III. [cited 5 de diciembre 2013] : <http://eva.uci.cu/mod/resource/view.php?id=8978>.

37. **Rojas, Olivares Juan Carlos .** Patrones de Diseño. [cited 10 de diciembre 2013] : http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_1/Diseno_de_software/Patrones_de_Disen_Art._2.pdf.

38. **Craig, Larman.** UML y patrones: una introducción al análisis diseño orientado a objetos. [cited 12 de diciembre 2013] : http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/UML_y_Patrones/00_Presentacion_y_contenido.pdf.

Bibliografía

1. **Calderón, Amaro Sarah Dámaris y Rebaza, Valverde Jorge Carlos.** Metodologías Ágiles. [cited 25 de octubre de 2013] : <http://www.seccperu.org/files/Metodologias%20Agiles.pdf>.

2. **Garcia, Mato Rosa Maria .** *DISEÑO de BASES DE DATOS*. [cited 23 de octubre 2013] : http://eva.uci.cu/file.php/180/3._Bibliografia/Bibliografia_Basica/Rosa_Maria_Disen_o_de_bases_de_datos.pdf.

3. **Talavera, Barrios Carmen .** GUÍA DE ADMINISTRACIÓN CON LIFERAY. [cited 20 de octubre 2013] : http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CC8QFjAB&url=http%3A%2F%2Fwww.policar.es%2Fc%2Fdocument_library%2Fget_file%3Fp_l_id%3D51306450%26folderId%3D87038053%26name%3DDLFE-8330.pdf&ei=xSF1U-jel4qPqAbl-IK4DA&usq=AFQjCNG2bevSHD.

4. **Marko, Grönroos.** Book of Vaadin Vaadin 6.4. [cited 22 de octubre 2013] : <http://dev.vaadin.com/svn/doc/tags/6.4.1/manual/book-pocket.pdf>.

5. Vaadin Application,Tutorial Building a Simple Address Book. [cited 23 de octubre 2013] : <http://vaadin.com/download/release/6.4/6.4.1/docs/vaadin-tutorial.pdf>.

6. **Suarez, Lasprilla Maria Fernanda y Peñaloza, Niño Sebastian .** Exploración del framework vaadin para el desarrollo de un gestor de tareas para la asignatura Ingeniería de Software I. 2011.

Glosario de Términos

HTML: Lenguaje de Marcas de Hipertexto (HyperText Markup Language), lenguaje de marcado predominante para la construcción de páginas web; usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTP: Protocolo de Transferencia de Hipertextos (HyperText Transfer Protocol), modo de comunicación para solicitar páginas Web.

Herramienta CASE: Ingeniería de Software Asistida por Ordenador (Computer Aided Software Engineering). Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.

Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.

Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

MySQL: Sistema Gestor de Base de Datos Relacional, multihilos y multiusuario.

PostgreSQL: Sistema Gestor de Base de Datos Relacional, orientado a objetos y libre.

PHP: Procesador de Hipertexto (Hypertext Preprocessor). Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas.

AUP: Proceso Unificado Ágil. Metodología para el desarrollo de Software.

XP: Programación Extrema (EXtreme Programming). Metodología para el desarrollo de Software.

SGBD: Sistema de Gestión de Bases de Datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de acceso.

UML: Lenguaje Unificado de Modelado (Unified Modeling Language). Lenguaje gráfico para visualizar, especificar, construir y documentar un sistema informático.

API: Interfaz de Programación de Aplicaciones (Application Programming Interface), conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

CMS: Un sistema de gestión de contenidos (Content Management System) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web.

GUI: Interfaz Gráfica de Usuario (Graphic User Interface), artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

IBM: (International Business Machines), es una empresa multinacional que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

Javascript: Lenguaje interpretado, no requiere compilación, utilizado principalmente en páginas Web.

DB2: Marca comercial, propiedad de IBM, bajo la cual se comercializa un sistema de gestión de base de datos.

JEE: Edición Empresarial de Java (Java Enterprise Edition), plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JBOSS: Es un servidor de aplicaciones J2EE de código abierto implementado en Java puro.

LDAP: Protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

OpenID: Estándar de identificación digital descentralizado, con el que un usuario puede identificarse en una página web a través de una URL.

Plug-in: Complemento, aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

PORTLET: Componentes modulares de las interfaces de usuario gestionadas y visualizadas en un portal web.

Servlet: Objetos que corren dentro del contexto de un contenedor y extienden su funcionalidad.

SSL: Protocolo de Capa de Conexión Segura (Secure Sockets Layer), protocolo criptográfico que proporciona comunicaciones seguras por una red, comúnmente Internet.

WSRP: Protocolo estándar aprobado por OASIS diseñado para la comunicación con portlets remotos.

WebOS: Sistema Operativo Web en constante cambio Imagina un ordenador virtual, en el que sus componentes principales de hardware no están presentes del lado del usuario, sino del otro lado de la red en forma de servidor.

XML: Lenguaje extensible de marcas (Extensible Markup Language), es un lenguaje creado por W3C (World Wide WEB Consortium) para describir la estructura de los datos a almacenar en un documento.

Oracle: Sistema de gestión de base de datos relacional, fabricado por Oracle Corporación. Es considerado uno de los más completos sistemas de Base de Datos

Base de datos: Conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo.

Facultad: Es la forma en que se agrupan estudiantes, profesores y trabajadores para obtener un mejor control de todo lo contendiente a la docencia y a la producción.

Usuario: Humano que interactúa con un sistema.

Navegador Web: Aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web a través de una red.

Clase: Descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semánticas.

Framework: Esquema (un esqueleto, un patrón, un marco) para el desarrollo y/o la implementación de una aplicación.

Servidor: Sistema que proporciona servicios a varios usuarios o estaciones de trabajo en una red.

Sistema Informático: Conjunto de partes interrelacionadas, hardware, software y recurso humano, que permite almacenar y procesar información.

JSON: Notación de Objetos de JavaScript, es una notación Javascript para escribir o codificar objetos, es un formato ligero para almacenar información que deseamos compartir entre distintos componentes o lenguajes de las aplicaciones Web.

HU: Las historias de usuario son la técnica utilizada en la metodología XP para especificar los requisitos del software. Son tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

Internet: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias.

PC: Computadora Personal (Personal Computer), es generalmente de tamaño medio y es usado por un solo usuario. Suele estar equipada para cumplir tareas comunes de la

informática moderna, es decir permite navegar por Internet, escribir textos y realizar otros trabajos de oficina o educativos, como editar textos y bases de datos.

Anexos

Anexo 1 Encuesta a los vicedecanos de formación en cuanto a cómo se realiza el proceso de matrícula de cursos optativos y electivos.

Compañero(a).

Buenas soy una estudiante de 5to año de la Universidad de las Ciencias Informáticas, específicamente de la facultad 5. Actualmente y como parte de mi tesis de diploma, realizo una investigación sobre el proceso de matrícula de cursos optativos y electivos.

En tanto sus respuestas al cuestionario que sigue constituyen una información valiosa a esta investigación.

1- ¿Quién es la persona encargada de realizar el proceso de matrícula de cursos optativos y electivos?

2- ¿Porque vía le llega a los estudiantes la solicitud de que deben presentarse para realizar el proceso de matrícula de cursos optativos y electivos?

3- ¿Porque vía le llega a los estudiantes la información referente a las temáticas de cada curso optativos o electivos?

4- ¿Una vez estando presente los estudiantes, para la realización de la matrícula de los cursos, como es llevado a cabo este proceso?

5- ¿Dónde es que se almacena la información referente a la matrícula de cursos optativos y electivos?

6- ¿Qué nivel de seguridad posee el lugar o dispositivo donde se almacena la información referente a la matrícula de cursos optativos y electivos?

7- ¿Cuáles son las ventajas, dificultades, consecuencias o desventajas que posee este proceso de matrícula de la manera que se lleva a cabo actualmente?

8-¿Existe, poseen o se encuentra en desarrollo algún sistema informático que cumpla con los objetivos y a la vez lleve a cabo el proceso de matrícula de cursos optativos y electivos?

Anexo 2

Alternativa 1 Caso de prueba de aceptación 1

Tabla 69 Probar la funcionalidad Autenticar usuario Alternativa 1.

| Caso de prueba de aceptación 13 |
|--|
| Historia de usuario: Autenticar usuario |
| Nombre: Probar la funcionalidad Autenticar usuario Alternativa 1. |

| |
|--|
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que el sistema verifique si el usuario y la contraseña son correctos. |
| Condiciones de ejecución: Debe ser usuario del sistema. |
| Entradas/Pasos de ejecución: Se inserta usuario y contraseña incorrectamente. |
| Resultado esperado: Se muestra un mensaje (El usuario o la contraseña especificada no son correctos). |
| Evaluación de la prueba: Satisfactoria. |

Alternativa 2 Caso de prueba de aceptación 1

Tabla 70 Probar la funcionalidad Autenticar usuario Alternativa 2.

| Caso de prueba de aceptación 14 |
|--|
| Historia de usuario: Autenticar usuario |
| Nombre: Probar la funcionalidad Autenticar usuario Alternativa 2. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que el sistema verifique si los campos de usuario y contraseña están vacíos. |
| Condiciones de ejecución: Debe ser usuario del sistema. |
| Entradas/Pasos de ejecución: Se inserta usuario y contraseña incorrectamente. |
| Resultado esperado: Se muestra un mensaje (Debe especificar el usuario y la contraseña). |
| Evaluación de la prueba: Satisfactoria. |

Anexo 3

Alternativa 1 Caso de prueba de aceptación 3

Tabla 71 Probar la funcionalidad Adicionar curso Alternativa 1.

| Caso de prueba de aceptación 15 |
|--|
| Historia de usuario: Adicionar curso |
| Nombre: Probar la funcionalidad Adicionar curso Alternativa 1. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que al adicionar los datos de un curso se cancela esta operación correctamente. |
| Condiciones de ejecución: El usuario debe estar autenticado como administrador del sistema para acceder a esta funcionalidad. |
| Entradas/Pasos de ejecución: Una vez autenticado el administrador del sistema se le mostrara una interfaz y adicionara un curso con los siguientes pasos: <ul style="list-style-type: none">• Presionar opción (Listado de cursos).• Presionar opción (Adicionar curso).• Insertar los datos del curso: nombre del curso, cantidad de estudiantes a matricular, año de los estudiantes que les corresponde el curso, semestre en que se impartirá, disciplina a la que pertenece, curso escolar, tipo de curso: electivo o optativo, descripción del curso, profesor que impartirá el curso y las fechas de la convocatoria de matrícula del curso.• Presionar la opción (Cancelar) |
| Resultado esperado: Se cancela la operación correctamente. |
| Evaluación de la prueba: Satisfactoria. |

Anexo 4

Alternativa 1 Caso de prueba de aceptación 4

Tabla 72 Probar la funcionalidad Modificar curso Alternativa 1.

| Caso de prueba de aceptación 16 |
|---|
| Historia de usuario: Modificar curso |
| Nombre: Probar la funcionalidad Modificar curso Alternativa 1. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que al modificar los datos de un curso estos se guardan y se muestran correctamente. |
| Condiciones de ejecución: El usuario debe estar autenticado como administrador del sistema para acceder a esta funcionalidad. |
| Entradas/Pasos de ejecución: Una vez autenticado el administrador del sistema se le mostrara una interfaz y modificara un curso con los siguientes pasos: <ul style="list-style-type: none">• Presionar opción (Listado de cursos).• Buscar y Presionar en la opción (Modificar).• Modificar los datos que se deseen de los que se mencionan a continuación: nombre del curso, cantidad de estudiantes a matricular, año de los estudiantes que les corresponde el curso, semestre en que se impartirá, disciplina a la que pertenece, curso escolar, tipo de curso: electivo o optativo, descripción del curso, profesor que impartirá el curso y las fechas de la convocatoria de matrícula del curso.• Presionar la opción (Cancelar) |
| Resultado esperado: Se cancela la operación correctamente. |
| Evaluación de la prueba: Satisfactoria. |

Anexo 5

Alternativa 1 Caso de prueba de aceptación 5

Tabla 73 Probar la funcionalidad Eliminar curso Alternativa 1.

| Caso de prueba de aceptación 17 |
|--|
| Historia de usuario: Eliminar curso |
| Nombre: Probar la funcionalidad Eliminar curso Alternativa 1. |
| Responsable: Yuleisy Galloso Soler |
| Descripción: Probar que al eliminar un curso se ejecute correctamente esta funcionalidad. |
| Condiciones de ejecución: El usuario debe estar autenticado como administrador del sistema para acceder a esta funcionalidad. |
| Entradas/Pasos de ejecución: Una vez autenticado el administrador del sistema se le mostrara una interfaz y eliminara un curso con los siguientes pasos: <ul style="list-style-type: none">• Presionar opción (Listado de cursos).• Buscar y Presionar en la opción (Eliminar).• Se mostrara un mensaje: “Esta seguro que desea eliminar el curso“.• Presionar la opción (No) |
| Resultado esperado: Se cancela la operación correctamente. |
| Evaluación de la prueba: Satisfactoria. |

Anexo 6

Alternativa 1 Caso de prueba de aceptación 6

Tabla 74 Probar la funcionalidad Matricular cursos para tercero Alternativa 1.

| Caso de prueba de aceptación 18 |
|--|
| Historia de usuario: Matricular cursos para tercero |
| Nombre: Probar la funcionalidad Matricular cursos para tercero Alternativa 1. |
| Responsable: Yuleisy Galloso Soler |

| |
|--|
| <p>Descripción:</p> <p>Probar que un estudiante de tercero se matricule correctamente.</p> |
| <p>Condiciones de ejecución:</p> <p>El usuario debe estar registrado y tener permisos para acceder a esta funcionalidad.</p> |
| <p>Entradas/Pasos de ejecución:</p> <p>Una vez autenticado un estudiante se le mostrara una interfaz y matriculara un curso con los siguientes pasos:</p> <ul style="list-style-type: none"> • Presionar opción (Cursos disponibles). • Seleccionar el año que cursa • Se le muestra una interfaz con todos los cursos de su año y aquí presiona en el matricular del curso que desee. |
| <p>Resultado esperado: Se muestra un mensaje (No puede matricular en este curso porque ya usted está matriculado en este curso).</p> |
| <p>Evaluación de la prueba: Satisfactoria.</p> |

Alternativa 2 Caso de prueba de aceptación 6

Tabla 75 Probar la funcionalidad Matricular cursos para tercero Alternativa 2.

| Caso de prueba de aceptación 19 |
|---|
| <p>Historia de usuario: Matricular cursos para tercero</p> |
| <p>Nombre: Probar la funcionalidad Matricular cursos para tercero Alternativa 2.</p> |
| <p>Responsable: Yuleisy Galloso Soler</p> |
| <p>Descripción:</p> <p>Probar que un estudiante de tercero se matricule correctamente.</p> |
| <p>Condiciones de ejecución:</p> <p>El usuario debe estar registrado y tener permisos para acceder a esta funcionalidad.</p> |
| <p>Entradas/Pasos de ejecución:</p> <p>Una vez autenticado un estudiante se le mostrara una interfaz y matriculara un</p> |

curso con los siguientes pasos:

- Presionar opción (Cursos disponibles).
- Seleccionar el año que cursa
- Se le muestra una interfaz con todos los cursos de su año y aquí presiona en el matricular del curso que desee.

Resultado esperado: Se muestra un mensaje (No puede matricular en este curso porque ya no quedan capacidades de matrícula).

Evaluación de la prueba: Satisfactoria.

La **Alternativa 1 Caso de prueba de aceptación 6** y **Alternativa 2 Caso de prueba de aceptación 6** se aplica a los casos de prueba siguientes: **Caso de prueba de aceptación 7** y **Caso de prueba de aceptación 8**.

Anexo 7 Autenticar Usuario



Nombre de usuario:

Contraseña:

Ilustración 4 Autenticar Usuario

Anexo 8 Adicionar Curso

Registrar el Curso

Nombre del Curso *

Capacidad *

Año *

Semestre *

Disciplina *

Curso Escolar *

Tipo de curso *

Convocatoria *

Profesor *

Activar curso

Upload

Examinar... No se ha seleccionado ningún archivo. Upload

Registrar Cancelar

Ilustración 5 Adicionar Curso

Anexo 9 Modificar Curso

Modificar Datos del Curso

Nombre del Curso *

Capacidad *

Año *

Semestre *

Disciplina *

Curso Escolar *

Tipo de curso *

Convocatoria *

Profesor *

Activar curso

Upload

Examinar... No se ha seleccionado ningún archivo. Upload

Modificar Cancelar

Ilustración 6 Modificar Curso

Anexo 10 Eliminar Curso

Listado de cursos

| Nombre | Matrícula | Semestre | Año | Disciplina | Curso Escolar | Tipo | Convocatoria |
|---------------|-----------|--------------|---------|----------------|---------------|----------|----------------------|
| SOA | 15 | 1er Semestre | 5to Año | SOA | 2013-2014 | Optativo | 5/9/2013-10/9/2013 |
| EDE | 18 | 2do Semestre | 3er Año | EDE | 2013-2014 | Optativo | 5-10/9/2013*5/9/2013 |
| Algoritmia | 30 | 1er Semestre | 4to Año | Algoritmia | 2013-2014 | Optativo | 5/9/2013-10/9/2013 |
| GPComputadora | 15 | 2do Semestre | 4to Año | GPComputadora | 2013-2014 | Optativo | 5/9/2013-10/9/2013 |
| Masaje | 10 | 2do Semestre | 4to Año | Cultura Fisica | 2013-2014 | Electivo | 1/11/2013-15/11/2013 |

Ilustración 7 Eliminar Curso 1

Listado de cursos

| Nombre | Matricula | Semestre | Año | Disciplina | Curso Escolar | Tipo | Convocatoria |
|---------------|-----------|--------------|---------|---------------|---------------|----------|--------------------|
| SOA | 15 | 1er Semestre | 5to Año | SOA | 2013-2014 | Optativo | 5/9/2013-10/9/2013 |
| EDE | 18 | 2do Semestre | 3er Año | EDE | 2013-2014 | Optativo | 5/9/2013-10/9/2013 |
| Algoritmia | 19 | 1er Semestre | 4to Año | Algoritmia | 2013-2014 | Optativo | 5/9/2013-10/9/2013 |
| GPComputadora | 15 | 2do Semestre | 4to Año | GPComputadora | 2013-2014 | Optativo | 5/9/2013-10/9/2013 |

Ilustración 8 Eliminar Curso 2

Anexo 11 Matricular Curso

Sistema para la Gestión de Matrículas de Cursos

YULEISY GALLOSO SOLER
 Salir
 ESTUDIANTE
 Cursos disponibles

Por favor seleccione el año que está cursando:
 4to

Cursos Optativos

| Nombre | Matricula | Semestre | Año | Disciplina | Curso Escolar |
|---------------|-----------|--------------|---------|---------------|---------------|
| Algoritmia | 1 | 1er Semestre | 4to Año | gfbajing | 2013-2014 |
| GPComputadora | 2 | 2do Semestre | 4to Año | GPComputadora | 2013-2014 |

Se ha matriculado correctamente.

Cursos Electivos

| Año | Disciplina | Curso Escolar | Tipo | Convocatoria | Usuario Profesor |
|-----|----------------|---------------|----------|----------------------|------------------|
| | Cultura Fisica | 2013-2014 | Electivo | 1/11/2013-15/11/2013 | ymarcheco |

Ilustración 9 Matricular curso

Anexo 12 Listado de matricula

Seleccione el curso: *

Algoritmia

Reporte de Matricula

Registro de estudiantes matriculados.

| Nombre | 1ro Apellido | 2do Apellido | Brigada |
|-----------|--------------|--------------|---------|
| Alejandro | Ortega | Baez | |

Ilustración 10 Listado de matricula