INSTITUTO SUPERIOR MINERO
METALÚRGICO DE MOA
DR. ANTONIO NÚÑEZ JIMÉNEZ

Ingeniería Informática
Facultad: Geología y Minas

Trabajo de Diploma

Para Optar por el Título de

Ingeniero Informático

Título:

Software educativo para el apoyo a la enseñanza de la asignatura de Sistemas Operativos

Autor: Carlos Ernesto Velázquez Rodríguez

Tutor (es): Dr. C. José Luis Montero 'Farril

Ing. Arletis Cobas Rodríguez

Moa, 2014
Año 56 de la Revolución

Declaración de autoría

Yo, Carlos Ernesto Velázquez Rodríguez, declaro que soy el único autor del trabajo *"Software de apoyo a la enseñanza de la asignatura Sistemas Operativos de la carrera de Ingeniería Informática en el ISMMM"* y certifico su propiedad a favor del Instituto Superior Minero-Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", el cual podrá hacer uso del mismo con la finalidad que estime conveniente.

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

_____

Carlos Ernesto Velázquez Rodríguez

Autor

_____ _____

Dr. C. José Luis Montero O'Farril

Tutor

Ing. Arletis Cobas Rodríguez

Tutora

Dedicatoria

A mis padres Carlos y Dignora,
por traerme al mundo y darme todo lo que he necesitado
para ser quien soy hoy, con mis virtudes y defectos.

A mi hermano Alejandro.

Agradecimientos

A mis padres, a mis hermanos.

A todos mis tíos por la ayuda que me han brindado siempre y sobre todo cuando más los he necesitado.

A todos mis abuelos, los que aún viven y los que ya partieron.

A Rebeca, por el mejor apoyo que ha podido darme, por ayudarme con todas sus fuerzas y ser uno de los motores de la culminación de este trabajo.

A sus padres Débora y Manuel, y a sus hermanas Deborita, Raquel y Amanda por acogerme como uno más de los suyos y por desear lo mejor para mí.

A Lourdes. A los doctores y demás personal médico que me atendió durante mi recuperación en el hospital militar de Holguín.

A RG-4, por ser el punto de inflexión de mi existencia, hace ya 7 años. Gracias a ti hoy soy más fuerte.

A mis ex-compañeros del regimiento de tropas especiales de Holguín, quienes primero me salvaron la vida, donde quiera que estén cada uno de ellos.

A los compañeros de siempre, por las madrugadas compartidas.

A mis ex-compañeros de la UCI, los que quedan en Cuba y los que ahora hacen sus vidas en otras latitudes del planeta. A Ariadna por poner sus granos de arena en este trabajo.

A quienes hicieron mi camino más difícil, porque plantearon los obstáculos para crecerme en cada choque y caída. Ustedes me enseñaron de qué estoy hecho.

A SRD, por descubrirme la esencia de las cosas.

RESUMEN

Las Tecnologías de la Informática y las Comunicaciones han jugado un papel preponderante en la nueva sociedad, implicándose en todos los aspectos del desarrollo humano. Uno de los campos impactados por las nuevas tecnologías es la educación, para el cual han sido creados productos tales como los softwares educativos. La investigación que se presentó en este trabajo tuvo como objetivo el desarrollo de un software educativo con características de ejercitador con el fin de favorecer el proceso de enseñanza-aprendizaje de los estudiantes de la carrera de Ingeniería Informática en el Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", que en el tercer año del plan de estudios se enfrentan a la asignatura de Sistemas Operativos. Para acometer este trabajo se utilizó la metodología de desarrollo de software conocida como SXP, híbrido de las ya establecidas eXtreme Programming y Scrum, creada en la Universidad de las Ciencias Informáticas en el seno del proyecto Unicornios. Fue realizado un estudio de factibilidad a través del análisis de datos recogidos en un cuestionario aplicado a una muestra de los sujetos que componen la población de los usuarios finales del software desarrollado. Este estudio demostró la aceptación que generó en los estudiantes el uso del ejercitador, manifestando la viabilidad de su uso como medio de enseñanza sugerido a los alumnos como parte de su estudio independiente.

Palabras claves: aprendizaje, ejercitador, enseñanza, informática, sistemas operativos, software educativo.

ABSTRACT

The Computer science's Technologies and the Communications have played a preponderant paper in the new society, being implied in all the aspects of the human development. One of the fields impacted by the new technologies is the education, for which many products have been created such as the educational softwares. The investigation that was presented in this work had as objective the development of an educational software with characteristic of a trainer with the purpose of favoring the process of teaching-learning of the students of the career of Computer Engineering in the Mining-Metallurgist Superior Institute of Moa "Dr. Antonio Núñez Jiménez" that face to the subject of Operating Systems in the third year of the plan of studies. To attack this work, the methodology of software development used was the one known as SXP, hybrid of those already established eXtreme Programming and Scrum, created in the University of the Computer Sciences in the breast of the project Unicorns. A study of feasibility was carried out through the analysis of data picked up in a questionnaire applied to a sample of the fellows that you/they compose the population of the final users of the developed software. This study demonstrated the acceptance that generated in the students the use of the trainer, manifesting the viability of its use as a way of teaching suggested to the students as a part of its independent study.

ÍNDICE

Resumen ..... *IV*

Abstract ..... *V*

Índice ..... *VI*

Índice de tablas ..... *VIII*

Índice de figuras ..... *IX*

INTRODUCCIÓN *1*

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA *8*

1.1. Introducción *8*

1.2. Conceptos fundamentales *8*

1.2.1. Implicación de las TIC en la educación 8

1.2.2. Diseño y elaboración de un software educativo 13

1.3. Antecedentes de la investigación *18*

1.4. Tendencias tecnológicas aplicadas a la investigación *24*

1.4.1. Software Libre (SWL) 24

1.4.2. Lenguajes de programación 25

1.5. Herramientas *28*

1.5.1. Entorno de desarrollo integrado NetBeans 28

1.5.2. Visual Paradigm 29

1.6. Metodologías de desarrollo de aplicaciones informáticas *30*

1.6.1. Extreme Programming (XP) 31

1.6.2. Scrum 31

1.6.3. SXP 32

1.7. Conclusiones del capítulo *37*

CAPÍTULO 2: PASOS INICIALES. EXPLORACIÓN Y PLANIFICACIÓN DEL PROYECTO *38*

2.1. Introducción *38*

2.2. Breve descripción de los procesos a informatizar *38*

2.2.1. Planificación de procesos 39

2.2.2. Paginación de memoria virtual 41

2.2.3. Cálculo de direcciones físicas 42

2.2.4. Protección de archivos 42

2.2.5. Programación del brazo del disco 43

2.2.6. Detección de interbloqueos 45

2.3. Fase de Exploración y Planificación *46*

2.3.1. Concepción del sistema 46
2.3.2. Lista de Reserva del Producto 47
2.3.3. Historias de usuario 49
2.3.4. Estimación del esfuerzo por historia de usuario 51
2.3.5. Plan de Iteraciones y duraciones respectivas 51
2.3.6. Plan de Entregas 52

2.4. Conclusiones del capítulo 54

CAPÍTULO 3: DISEÑO, IMPLEMENTACIÓN Y PRUEBA 55

3.1. Introducción 55

3.2. Fase de Diseño 55
3.2.1. Diagrama de paquetes 55

3.3. Arquitectura del sistema 57
3.3.1. Modelo Vista Controlador 58

3.4. Desarrollo de las iteraciones 58
Iteración 1 59
Iteración 2 59
Iteración 3 61
Iteración 4 61
Iteración 5 61
Iteración 6 61

3.5. Pruebas de aceptación 61

3.6. Conclusiones del capítulo 63

CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD 64

4.1. Introducción 64

4.2. Aplicación del cuestionario 65

4.3. Conclusiones del capítulo 73

CONCLUSIONES GENERALES 74

RECOMENDACIONES 75

BIBLIOGRAFÍA 76

ANEXOS 79

ANEXO 1: Programa analítico de la asignatura Sistemas Operativos en el ISMMM 80

ANEXO 2: Cuestionario para los estudiantes de la asignatura Sistemas Operativos 87

ÍNDICE DE TABLAS

Tabla 1.1: Clasificación de los software educativos según su tipo 15
Tabla 1.2: Clasificación de los software educativos según su función 16
Tabla 2.1: Grupos de usuarios y criterios de permisos 43
Tabla 2.2: Lista de Reserva del Producto 47
Tabla 2.3: Ejemplo de historia de usuario elaborada 50
Tabla 2.4: Plan de duración de las iteraciones 52
Tabla 2.5: Plan de entregas 53
Tabla 3.1: Tarea de programación 1 de historia de usuario 1 60
Tabla 3.2: Tarea de programación 1 de historia de usuario 4 60
Tabla 3.3: Caso de prueba de aceptación 1.1 62
Tabla 3.4: Caso de prueba de aceptación 2.1 63
Tabla 4.1: Significado de las siglas 66
Tabla 4.2: Valoración del ítem 1 sobre el desempeño de ENSO 66
Tabla 4.3: Valoración del ítem 2 sobre el desempeño de ENSO 67
Tabla 4.4: Valoración del ítem 3 sobre el desempeño de ENSO 67
Tabla 4.5: Valoración del ítem 4 sobre el desempeño de ENSO 68
Tabla 4.6: Valoración del ítem 5 sobre el desempeño de ENSO 69
Tabla 4.7: Valoración del ítem 6 sobre el desempeño de ENSO 69
Tabla 4.8: Valoración del ítem 7 sobre el desempeño de ENSO 70
Tabla 4.9: Valoración del ítem 8 sobre el desempeño de ENSO 71
Tabla 4.10: Valoración del ítem 9 sobre el desempeño de ENSO 71
Tabla 4.11: Valoración del ítem 10 sobre el desempeño de ENSO 72

ÍNDICE DE FIGURAS

Imagen 3.1: Diagrama de paquetes de ENSO *56*

Imagen 4.1: Gráfico para el ítem 1 sobre el desempeño de ENSO *66*

Imagen 4.2: Gráfico para el ítem 2 sobre el desempeño de ENSO *67*

Imagen 4.3: Gráfico para el ítem 3 sobre el desempeño de ENSO *68*

Imagen 4.4: Gráfico para el ítem 4 sobre el desempeño de ENSO *68*

Imagen 4.5: Gráfico para el ítem 5 sobre el desempeño de ENSO *69*

Imagen 4.6: Gráfico para el ítem 6 sobre el desempeño de ENSO *70*

Imagen 4.7: Gráfico para el ítem 7 sobre el desempeño de ENSO *70*

Imagen 4.8: Gráfico para el ítem 8 sobre el desempeño de ENSO *71*

Imagen 4.9: Gráfico para el ítem 9 sobre el desempeño de ENSO *72*

Imagen 4.10: Gráfico para el ítem 10 sobre el desempeño de ENSO *72*

INTRODUCCIÓN

El Instituto Superior Minero-Metalúrgico de Moa "Dr. Antonio Núñez Jiménez" (ISMMM) fue fundado en este municipio de la provincia de Holguín perteneciente a la República de Cuba, en el año 1976. Es la principal institución académica del país en la rama Geólogo-Minera y Metalúrgica, desde su creación se ha convertido en un importante eslabón en el desarrollo de la Industria Cubana del Níquel.[1] Cuenta con tres facultades docentes en las que se imparten las carreras de Ingeniería en Geología, Minería, Metalurgia, Eléctrica, Mecánica e Informática, además de las Licenciaturas en Contabilidad y Finanzas, Estudios Socioculturales y Ciencias de la Información.

La relevancia que confiere el estado al progreso de la Informática y la industria del software se explicitó en la Resolución Económica del V Congreso del Partido Comunista de Cuba (PCC) en octubre de 1997:

- En particular las industrias (…), incluida la del software, deberán trabajar con calidad, por incrementar la sustitución de importaciones, y desarrollar la exportación de líneas de elevada eficiencia.
- El país debe encaminarse resueltamente a la modernización informática mediante un programa integral que involucre a las organizaciones que deben proveer los recursos materiales, financieros e intelectuales y a las entidades económicas, políticas y sociales que deben traducirlos en más y mejores productos y servicios.

En consecuencia el perfil de la carrera de Ingeniería Informática a estado encaminada en el progreso y perfeccionamiento de la economía del país, con

[1] Sitio web del ISMM [Consultado en línea 26/05/2014] http://www.ismm.edu.cu/universidad/historia

el único objetivo de hacer las instituciones lugares donde premie la eficiencia y la eficacia, a través de la vinculación indiscutible de las nuevas tecnologías. El Ingeniero Informático está llamado a desempeñar un rol en el que facilite la dirección de las empresas mediante la automatización de los sistemas de información, la incorporación de diferentes programas informáticos que sirvan de guía y ayuda en el trabajo y en la toma de decisiones para la gestión de procesos tecnológicos. Además de proveer una serie de aplicaciones que contribuya en mejores servicios y productos factibles al mejoramiento continuo del país.

Hoy se exigen profesionales capaces de enfrentar los retos que impone esta sociedad de la información donde las tecnologías ocupan un papel relevante en todas las esferas del trabajo humano, por lo que el desarrollo de habilidades, el uso de técnicas para el procesamiento y uso de la información junto a la creación de mejores herramientas que favorezcan el trabajo permitiendo transmitir experiencias y conocimientos mediante el aprendizaje, se hace necesario.

Tal es el caso de las diferentes disciplinas impartidas en la especialidad de las Ciencias Informáticas, que recogen toda una historia de cambios que tributan a esa necesidad de formar profesionales altamente competitivos y capaces de satisfacer las demandas y exigencias actuales.

Entre las disciplinas que manifiesta el plan de estudios de la carrera de Informática se encuentra la de Infraestructura de Sistemas Informáticos, la cual existió en los planes de estudio anteriores con el nombre de Sistemas Digitales y Aseguramiento Básico de Programas, y con los mismos elementos estuvo presente en el Plan C. Esta disciplina juega un papel significativo en la formación del profesional informático pues ofrece los conocimientos y habilidades que le permitirán determinar los recursos más adecuados que necesita un software específico y la explotación satisfactoria del mismo.

Como resultado del desarrollo dinámico en el campo de la informática en general, incluyendo el hardware, la cada vez mayor flexibilidad y versatilidad de los sistemas operativos e igualmente la creciente interconexión de servicios y computadoras hace necesario que los objetivos y contenidos de la misma estén en correspondencia con los requerimientos del ámbito de actuación del ingeniero informático, sin soslayar la necesidad de dotar a los estudiantes de conocimientos básicos que les permitan adaptarse a los constantes cambios que ocurren en el entorno.[2]

Por consiguiente, entre los objetivos instructivos de la carrera en el tercer año de la misma se encuentra el de seleccionar sistemas operativos a partir de la evaluación de parámetros de funcionalidad, configuración, mantenimiento y administración de seguridad. Con este fin se diseñó la asignatura de Sistemas Operativos (SO), que se imparte dentro de la mencionada disciplina de Infraestructuras de Sistemas Informáticos.

Las Tecnologías de la Informática y las Comunicaciones (TIC) son la integración y convergencia de la computación, las telecomunicaciones y la técnica para el procesamiento de datos, donde sus principales componentes son el factor humano, los contenidos de la información, el equipamiento, la infraestructura, el software y los mecanismos de intercambio de información, los elementos de política y regulaciones, además de los recursos financieros.[3]

Las TIC han impactado en todas las áreas del desarrollo humano, incluyendo obviamente la educación. Hoy en las escuelas los profesores se valen de los avances tecnológicos para acrecentar la calidad de sus clases, los centros educativos asimilan e implantan los aportes de las TIC en el campo educativo y los ministerios y gobiernos apuestan claramente por la inclusión

---

[2] MES: "Ciencias Técnicas. Plan D. Ingeniería Informática Presencial". República de Cuba, Ministerio de Educación, Julio de 2007

[3] Cobas Rodríguez, Arletis: "Sistema para la gestión de viajes y combustibles en el CEDINIQ", Trabajo de Diploma. ISMMM, 2012.

cada vez más omnipresente de las bondades de la tecnología en la educación de sus ciudadanos de todas las edades y niveles educacionales.

En la actualidad los estudiantes de la carrera de Informática que estudian la asignatura de SO en el ISMMM no cuentan con una manera de comprobar sus soluciones de manera autónoma. Dependen de la disponibilidad del profesor de la asignatura para la revisión de las respuestas que tributan a los ejercicios que desarrollan como parte del proceso de estudio independiente. No existe tampoco para ellos un elemento centralizador de referencia donde tengan acceso a ejercicios y sus soluciones, que es lo más importante, dado que estas son la medida del aprendizaje y asimilación de los conceptos y contenidos de los temas tratados en la materia.

La investigación que se presenta en este documento aborda el proceso de enseñanza-aprendizaje de la asignatura de SO, que se imparte en el tercer año de la carrera de Ingeniería Informática en el ISMMM.

Como consecuencia de lo analizado se plantea el siguiente ***problema social****:* Los estudiantes de la asignatura de SO en la carrera de Ingeniería Informática en el ISMMM se enfrentan a algunas dificultades en el proceso de enseñanza-aprendizaje de esta materia. La falta de ejercicios propuestos y resueltos en la bibliografía básica y complementaria de la asignatura dificulta la asimilación de los contenidos expuestos en el programa de la misma. Así mismo, los estudiantes no cuentan con una vía para comprobar las soluciones que tributan a los ejercicios que realizan como parte del estudio independiente.

A partir de la situación antes comentada es presentado el siguiente ***problema científico***: ¿Cómo favorecer el proceso de enseñanza-aprendizaje de la asignatura de SO en la carrera de Ingeniería Informática?

Como ***objeto de estudio*** se presenta la informatización del proceso de enseñanza-aprendizaje de la asignatura de SO enmarcado en el ***campo de acción*** los software educativos.

Se propone como ***objetivo general*** desarrollar una aplicación informática que sirva de guía para comprobar soluciones de ejercicios, además de ofrecer estos a los estudiantes que se enfrentan a esta materia en el tercer año de la carrera de Ingeniería Informática en el ISMMM.

Es así que se presenta la ***idea a defender*** expuesta a continuación: Si se elabora un software educativo con características de un ejercitador será favorecido el proceso de enseñanza-aprendizaje de esta materia en la carrera de Ingeniería Informática en el ISMMM.

Para alcanzar el objetivo general se trazan las siguientes ***tareas de investigación***:

- Establecer el estado del arte
- Determinar las tecnologías, herramientas y metodología de desarrollo de software
- Aplicar la metodología de desarrollo de software seleccionada
- Realizar el estudio de factibilidad

Entre los ***métodos de investigación*** utilizados se encuentran los *teóricos* y los *empíricos*. Ambos arquetipos se enumeran a continuación:

Métodos Empíricos

- *Entrevista:* la aplicación de entrevistas funcionó dentro de la metodología para conocer aspectos importantes sobre los procesos a informatizar, así como características funcionales del sistema.
- *Cuestionario:* la aplicación de cuestionarios posibilitó evaluar la factibilidad del software desarrollado y presentado a los estudiantes.
- *Métodos estadísticos-matemáticos:* Posibilitó la evaluación de los resultados obtenidos a partir del cuestionario realizado.

Métodos Teóricos

- *Histórico-lógico:* permitió conocer los antecedentes del tema a investigar, si existen otras aplicaciones informáticas para apoyar el proceso de enseñanza-aprendizaje de la asignatura de SO y las características que poseen.
- *Analítico-sintético:* facilitó el análisis y procesamiento de la información recopilada a partir de los métodos empíricos para darle correcta culminación a la investigación.
- *Hipotético-deductivo:* a través de este método se realizó el correcto análisis para el desarrollo de la herramienta que sustentará la idea a defender y guiará el proceso hasta darle solución al problema planteado.
- *La modelación*: Fue empleada la construcción de modelos como parte de la metodología de desarrollo de software y además como instrumento de apoyo a la investigación basándose en el lenguaje UML.

Este documento consta de cuatro capítulos, de los cuales se ofrece un resumen a continuación:

Capítulo 1 "Fundamentación Teórica": Parte de un marco teórico donde se recogen los preceptos doctrinarios que sustentan el estudio. Se comenta la implicación de las TIC en la educación y aspectos relativos al diseño y elaboración de los software educativos. Además se estudian las diferentes tendencias tecnológicas actuales que se usarán en el desarrollo del producto de la investigación. También se hace una breve introducción de las herramientas que serán utilizadas para implementar la aplicación, así como la presentación de la metodología de desarrollo de software seleccionada para la elaboración del ejercitador.

Capítulo 2 "Pasos Iniciales. Exploración y Planificación del Proyecto": Son presentados los procesos en forma de una breve descripción, e introducidos los elementos referentes a la concepción inicial del sistema, además de definir los aspectos en los que sustenta la planificación del proyecto, es decir, las historias de usuario y las estimaciones de los esfuerzos necesarios para desarrollar la aplicación, además de los planes de entrega de cada iteración.

Capítulo 3 "Desarrollo de la Solución Propuesta y Prueba del Ejercitador": Abarca los atributos relacionados con el diseño y la implementación del sistema, así como el patrón arquitectónico escogido para su implementación. Igualmente se presentan las tareas de programación definidas para cada historia de usuario definida en la fase de planificación. Por último se presentan las pruebas de aceptación llevadas a cabo para comprobar el correcto desempeño del sistema y validar que cada funcionalidad procesa los datos de la manera que se espera que lo haga.

Capítulo 4 "Estudio de Factibilidad": Se realiza un estudio para comprobar que el producto obtenido cumple los objetivos propuestos, es decir, favorece el proceso de enseñanza-aprendizaje de la asignatura de SO en el ISMMM. El método utilizado ha sido basado en entrevistas y cuestionarios a los estudiantes de la materia, los usuarios principales del software.

Finalizando, se presentan las **conclusiones** alcanzadas, las **recomendaciones** y la **bibliografía** consultada además de los **anexos**.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se presentan los conceptos fundamentales en el campo de la investigación que se introduce en este trabajo como son informatización, aprendizaje, enseñanza, proceso de enseñanza-aprendizaje, tecnología educativa, software educativo y sistema operativo. Además se examinan el estado del arte, las tendencias tecnológicas actuales que intervendrán en el resultado de esta investigación y se presenta la metodología de software que será usada para el desarrollo de la aplicación informática.

1.2. Conceptos fundamentales

1.2.1. Implicación de las TIC en la educación

Las Tecnologías de la Información y las Comunicaciones (TIC) son medios para transmitir y gestionar datos, información y conocimiento[4], y los impactos que han generado denotan la importancia de incluirlas en los procesos formativos universitarios, permitiendo la concepción de profesionales con una formación científica y técnica plausible.

El uso cada vez creciente de las TIC en la universidad es parte de la informatización de la sociedad, este es el proceso continuo de utilización masiva y ordenada de las TIC de manera que satisfagan las insuficiencias de información y

[4] Montero O'Farril, José Luis: "Las TIC en la Nueva Universidad Cubana", CEP, ISMMM, Moa, Holguín, 2010.

conocimientos en todos los niveles del país, acarreando consigo el acrecentamiento de la calidad de vida al extenderse la cantidad y calidad de los servicios que pueden recibirse. Cuenta con una estrategia que consiste en una composición y alineación de métodos y materiales seleccionados para tutelar este proceso. El mismo busca lograr más eficacia y eficiencia permitiendo así una mayor generación de riquezas que hagan sustentable el aumento sistemático del bienestar espiritual y material de los ciudadanos.

Una sociedad que emplee la informatización en la totalidad de sus esferas y procesos será más eficaz, eficiente y competitiva. Resulta indudable que para los países subdesarrollados significa un desafío alcanzar este propósito, debido a que su dificultad esencial está en conseguir la satisfacción de las necesidades de sus ciudadanos, incluyendo las de información.

Nuestro país, Cuba, ha reconocido precozmente el beneficio y la obligación de destacar e incluir en la estrategia de desarrollo nacional las TIC, y lograr una cultura digital como una de las características indispensables del hombre nuevo, lo que propiciaría a nuestra sociedad acercarse más hacia el objetivo de un desarrollo sostenible.

En el mundo actual las TIC se propagan cada vez más por los sistemas educacionales en todos los niveles, incluyendo la educación superior. Casi la totalidad de las universidades que desarrollan programas de estudio presenciales se valen de la tecnología para aprovechar las facilidades que brinda en beneficio de una enseñanza más efectiva.

La enseñanza es el proceso sistémico de transmisión de la cultura en la institución escolar en función del encargo social, que se organiza a partir de los niveles de desarrollo actual y potencial de los estudiantes, y conduce el tránsito continuo hacia niveles superiores de desarrollo, con la finalidad de formar una

personalidad integral y auto determinada, capaz de transformarse y de transformar su realidad en un contexto histórico concreto.[5]

Otros investigadores[6] aseguran que la enseñanza se concibe como la acción pedagógica mediante la cual se crean, organizan y brindan al alumno una serie de situaciones de aprendizaje que tienden a incidir sobre el proceso que él desarrolla para aprender. Se acentúa que estas situaciones de aprendizaje deben facilitarle participar activamente, interactuar con el medio para confrontar sus propias hipótesis con los hechos de la realidad y vivir experiencias útiles y significativas.

El propósito de la enseñanza es conseguir el aprendizaje, según estos autores. Todas las actitudes, estrategias o métodos que pueden implementarse para la búsqueda de aprendizaje, constituyen la enseñanza. Plantean que cuando se articulan estos métodos y estrategias de manera que el aprendizaje logrado resulta lo más significativo posible, se está en presencia de la didáctica.

Por su parte, Oscar López Regalado[7] citando a Orellana[8], plantea que la enseñanza escolar es una actividad especializada de la educación que consiste en la actuación de un profesional que recibe el encargo de una institución educativa, contando con un reconocimiento público para poder actuar sobre un sujeto de aprendizaje, en este caso el alumno, al que le transmite una información vigente, actualizada por la cultura y la ciencia, con el objetivo de desarrollar sus capacidades personales, especializar determinadas habilidades y promover la reconstrucción del conocimiento a través de la conciencia.

---

[5] González, A.; Recarey, S. y Addine, F. 2008; "El proceso de enseñanza aprendizaje: un reto para el cambio educativo". En: Ginoris, O. "Fundamentos didácticos de la Educación Superior en Cuba". Félix Varela, La Habana, 38-60

[6] Macias, Nelson, &Torres, Ytaliar. 2009. "Software educativo como apoyo en el proceso de enseñanza aprendizaje del método de reducción en la resolución de sistemas de ecuaciones lineales". Departamento de Física y Matemática, Universidad de los Andes.

[7] López Regalado, O. 2010. "Trabajos de investigación como catalizadores de la innovación educativa". Juan Carlos Tojar Hurtado (Coord.), varios autores, Editorial Infornet, Sevilla

[8] Orellana, O. 2003. "Enseñanza aprendizaje: La mediación Constructivista". Lima, San Marcos.

No obstante, no necesariamente la enseñanza ha de ser entendida como lo esbozado en el párrafo anterior. En las últimas décadas los avances científico-técnicos, que han sido protagonistas en tantas esferas del desarrollo humano, han impactado también en la enseñanza, como ha quedado establecido anteriormente, insertándose entre los métodos para ejercerla.

Hoy día se puede aprender sin la participación directa de un docente, los estudiantes pueden ser más autónomos. Es así que la enseñanza se ha visto favorecida, para bien, por las nuevas tecnologías en muchas dimensiones.

Por otra parte, el aprendizaje es el proceso dialéctico de apropiación de los contenidos y las formas de conocer, hacer, convivir y ser construidos en la experiencia sociohistórica, en el cual se producen, como resultado de la actividad del individuo y de la interacción con otras personas, cambios relativamente duraderos y generalizables, que le permiten adaptarse a la realidad, transformarla y crecer como personalidad.[9]

Según Macias y Torres, el aprendizaje es un cambio relativamente permanente de la conducta en términos de experiencia o práctica[10]. Tiene lugar en el cerebro del sujeto y después se manifiesta en conductas observables. Es un proceso inherente a la vida del hombre, es absolutamente necesario para que el mismo pueda transformar y adaptarse al medio en el que habita. El sujeto recibe constantemente un conjunto de ideas y contenidos, los cuales transforma e interioriza de acuerdo a sus necesidades y paralelamente los manifiesta a través de los cambios que sufre el ambiente que lo rodea.

Oscar López Regalado[11], citando a Fandos[12] plantea que el aprendizaje es un proceso en el que se inicia o se modifica una conducta debido a la experiencia.

---

[9] Castellanos Simón, Doris y otros. "Aprender y enseñar en la escuela". Editorial Pueblo y Educación. La Habana. Página 24, 2002.

[10] Macias, Nelson; Torres, Ytaliar. (2009). "Software educativo como apoyo en el proceso de enseñanza aprendizaje del método de reducción en la resolución de sistemas de ecuaciones lineales". Departamento de Física y Matemática, Universidad de los Andes.

[11] Ídem.[7]

Más adelante explica que existen características comunes en todo aprendizaje, enunciadas a continuación:

- Se produce por discernimiento, una comprensión de las relaciones entre los elementos esenciales para su solución.
- El discernimiento depende de la situación experimental. Más fácilmente sucederá cuanto más organizada se encuentre, de manera que se puedan percibir las relaciones.
- Obtenido el discernimiento, este se repite inmediatamente cuando aparece de nuevo la situación problemática.
- El discernimiento se transfiere a otras situaciones.

El proceso de enseñanza-aprendizaje es de tipo pedagógico escolar que posee las características esenciales de este, pero se distingue por ser mucho más sistemático, planificado, dirigido y específico por cuanto la interrelación maestro-alumno, deviene en un accionar didáctico mucho más directo, cuyo único fin es el desarrollo integral de la personalidad de los educandos.[13]

Algunos autores[14] señalan que este es un par que conserva una relación de causalidad manifestada en el hecho de que la enseñanza es una actividad intencional que está diseñada para dar lugar al aprendizaje de los educandos. Visto así, el aprendizaje es un producto del proceso de enseñanza.

Este es un proceso más que causal, ontológico, porque la enseñanza se realiza para provocar un aprendizaje que al final se convierte en una interactividad en la que coadyuvan tres elementos básicos: el profesor (que enseña), el alumno (que aprende) y el contexto (donde se desenvuelve y concierta todo el proceso).

---

[12] Fandos, M. (2003). "Formación basada en las Tecnologías de la Información y Comunicación: Análisis didáctico del proceso de enseñanza-aprendizaje". Tarragona, España: Universitat Rovira I Virgili.

[13] Ídem [12]

[14] Ídem [10]

Ramón Collazo y José Zilberstein[15] citando a Fernández[16] aseguran que el proceso de enseñanza-aprendizaje es esencialmente interactivo y comunicativo, de intercambio de información, compartiendo experiencias, conocimientos, vivencias que logran una influencia mutua en las relaciones personales.

1.2.2. Diseño y elaboración de un software educativo

Se entiende como tecnología educativa la teoría y la práctica del diseño y desarrollo, selección y utilización, evaluación y gestión de los recursos tecnológicos aplicados a los entornos educativos.[17]

La tecnología educativa se centra en elaborar una tecnología de la producción material y es por esto que su atención se dirige hacia los medios y métodos más que hacia los contenidos. Su creación, en 1954, se debe al psicólogo, filósofo social y autor estadounidense Burrhus Frederic Skinner (1904-1990), quien fue profesor de la Universidad de Harvard.

Ciertos autores[18] plantean que la tecnología educativa ha apoyado el aumento de la efectividad del trabajo del profesor, aunque sin llegar a sustituir su función educativa y humana. Asimismo ha logrado organizar la carga de trabajo de los educandos y el tiempo necesario para su formación científica, elevando la motivación hacia la enseñanza y el aprendizaje y garantizando la asimilación de lo esencial de los contenidos.

Estos escritores especifican que la tecnología educativa es el resultado de la aplicación de diferentes concepciones y teorías educativas para la resolución de

---

[15] Collazo Delgado, Ramón; Zilberstein Toruncha, José (2006). "Preparación pedagógica integral para profesores integrales". Editorial Félix Varela, Colectivo de autores, ISBN: 959-258-535-0

[16] Fernández, A. (2000). "El formador de formación profesional y ocupacional". Ediciones Octaedro, Barcelona, 2000.

[17] Marques, P.: "La Tecnología Educativa: conceptualización, Líneas de investigación". 2005. [online] http://dewey.uab.es/pmarques/tec.htm

[18] Pozo Cruz, Carlos Raúl del; Vidal Ledo, María. "Tecnología educativa, medios y recursos de enseñanza-aprendizaje". http://bvs.sld.cu/revistas/ems/vol22_4_08/ems10408.htm.Biblioteca Virtual de Salud. Consultado 16/04/2014.

una extensa franja de problemas referidos a la enseñanza y el aprendizaje. Surgió como disciplina en los años 50 del siglo XX en Estados Unidos y ha recorrido diferentes orientaciones y enfoques. Ejemplo de ello son la enseñanza audiovisual, la enseñanza programada, la tecnología instruccional y el diseño curricular, entre otros. La tecnología educativa utiliza los medios y recursos de la enseñanza como elementos dinámicos en cualquier proceso encaminado al progreso de la instrucción.

Los software educativos son resultado directo de la tecnología educativa. Un software educativo es una aplicación informática destinada a facilitar la enseñanza y el aprendizaje independientes que permite el desarrollo de ciertas habilidades cognitivas. Briceño y Guillén[19], citando a Ojeda y Piña[20], señalan que los software educativos son un conjunto de recursos informáticos diseñados con la intención de ser utilizados en el contexto de un proceso de enseñanza-aprendizaje.

Las características de los software educativos, como proponen María Vidal Ledo y otros citando a trabajos anteriores, son de manera general[21].

- Finalidad: son orientados a la enseñanza y el aprendizaje en todas sus formas.
- Utilización de ordenadores: el medio usado como soporte es la computadora.
- Facilidad de uso: son intuitivos y aplican reglas generales de uso y de fácil comprensión.

---

[19] Briceño, Jorge Luis; Guillén, José Gregorio. (2011). "Software educativo como apoyo en el proceso de enseñanza-aprendizaje de las variaciones y permutaciones". Departamento de Física y Matemática, Universidad de los Andes.

[20] Ojeda, J. y Piña, C. 2010. "Software en el contexto del proceso de enseñanza-aprendizaje". Odiseo. [En línea] diciembre de 2010. http://www.odiseo.com.mx

[21] Vidal Ledo, María; Gómez Martínez, Freddy; Ruiz Piedra, Alina M. "Software educativos". Educación Médica Superior [En línea]. 2010, vol.24, n.1 [citado 2014-04-22], pp. 97-110. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21412010000100012&lng=es&nrm=iso. ISSN 0864-2141.

- Interactividad: permiten un intercambio efectivo de información con el educando.

Estas herramientas, como plantean los investigadores antes mencionados, pueden catalogarse según su tipo como puede verse en la tabla 1.1.

Tabla 1.1: Clasificación de los software educativos según su tipo

Tipo	Propósito
Tutoriales	Presentan de manera secuencial el desarrollo de contenidos específicos.
Hipertextos o hipermedias	Proveen un entorno de aprendizaje no lineal
Micromundo	Proporcionan un entorno cerrado, desarrollado a partir de la solución de problemas.
Simulador	Facilitan entornos de aprendizaje basados en situaciones reales.
Práctica y ejercitación	Sirven un conjunto de ejercicios para la adquisición de destreza mediante su realización.

Entre las funciones que ostentan los software educativos están las siguientes[22]:

- Son un medio de comunicación entre el profesor y el estudiante, con el fin de hacer más eficaz la enseñanza y el aprendizaje.
- Facilitan la comprensión de contenidos y la práctica de conceptos y soluciones de ejercicios.
- Presentan información estructurada de la realidad y la ordenan (función informativa).
- Orientan y regulan el aprendizaje explícita o implícitamente (función instructiva).

[22] Ídem [19]

- Tienen connotación lúdica porque su uso franquea irremediablemente por el uso de las computadoras (función lúdica).

Aunque no siempre sus planteamientos pedagógicos resulten innovadores, las aplicaciones informáticas educativas pueden ser consideradas como materiales didácticos con esta función, dado que utilizan una tecnología recientemente incorporada a los centros educativos (función innovadora).

Estas funciones hacen atractivo el uso de los programas informáticos educativos, dado que no tributan a un único fin, sino que poseen varias aptitudes que desembocan en una macro función que es en general que la enseñanza sea significativa.

Los software educativos se clasifican, como plantean Zilberstein y Collazo[23] citando a Fernández[24], según su función en el proceso de enseñanza-aprendizaje. De esta categorización se muestra un fragmento en la tabla 1.2.

Tabla 1.2: Clasificación de los software educativos según su función

Clasificación	Medios
Para la transmisión y apropiación de la información	Películas, vídeos, ***software educativos***, libros de texto, manuales, medios planos y tridimensionales.
Para la experimentación del estudiante	Instrumentos de laboratorio, computadoras, cámaras fotográficas, videos, grabadoras de sonido.
Para el entrenamiento	Equipos y ***programas computarizados***, simuladores que requieren habilidades manipulativas.
Para el control del enseñar y el aprender	Incluye variantes desde pruebas impresas hasta el uso de ***programas de computación*** para evaluar.

[23] Ídem 15

[24] Ídem 16

Al hacer uso del software educativo como medio de enseñanza en el proceso de enseñanza-aprendizaje, implícitamente se interactúa con el ordenador. Es posible asegurar, comprobándolo en la tabla, que la utilización de las aplicaciones informáticas destinadas al apoyo a la enseñanza están contenidas en las cuatro categorías dadas por los autores referidos. Así puede afirmarse que estos son medios significativos en el proceso de enseñanza-aprendizaje porque su uso conlleva invariablemente a que el alumno desarrolle habilidades y capacidades en varios niveles de su psiquis.

Como quedó establecido anteriormente, muchas universidades que poseen programas presenciales han implantado el uso de las TIC en sus más diversas maneras dentro de los métodos pedagógicos para desarrollar sus programas de estudio para los estudiantes.

La asignatura de SO fue recomendada por la ACM (*Association for Computing Machinery*) y la IEEE (*Institute for Electrical and Electronic Engineers*) en 1991 como parte de las materias imprescindibles que deben impartirse en los planes de estudio de las carreras afines a las ciencias de la informática. La mayor parte de las universidades del mundo la han incluido en los programas de estudio de esas ingenierías.[25]

El programa analítico de la asignatura en el ISMMM plantea que el objetivo general de la misma es caracterizar los SO a partir del análisis de sus conceptos fundamentales, identificando el más eficiente en un ambiente de cómputo determinado e interpretar sus técnicas de programación para elaborar otros programas, con una adecuada cultura informática.

El sistema de conocimientos de la asignatura incluye la definición de sistema operativo, conocer su clasificación y las generaciones que han existido, así como

[25] Velázquez Rodríguez, Carlos Ernesto: Consultado en línea 26/06/2014. "SMRP: Una aplicación informática para simular el reemplazo de páginas de memoria en los Sistemas Operativos"; Revista "Ciencia & Futuro", Vol. 3, No. 2, Año 2013, ISSN 2306-823X, http://revista.ismm.edu.cu/index.php/revista_estudiantil/article/view/826/406#

la estructura básica del mismo, los servicios brindados y las principales funciones y programas. Además se circunscribe el concepto de proceso, la comunicación entre ellos y su planificación y se trata el tema de los procesos concurrentes. Así mismo este sistema contiene aspectos relativos a los dispositivos de entrada y salida de información, el aseguramiento de los programas que se encargan de ello. La asignatura también trata el tema de los interbloqueos y los procedimientos para prevenir, detectar, recuperar y evitar los bloqueos mutuos. El programa enuncia además los conceptos referentes a la administración de memoria, los sistemas mono y multiprogramados, igualmente enfoca la técnica de memoria virtual y sus beneficios, incluyendo su gestión por la técnica de paginación. Finalmente se inserta en el plan de la asignatura el tema de los sistemas de archivos, la administración del espacio en los medios externos de almacenamiento, la estructura de los directorios y la seguridad y protección de los archivos.

1.3. Antecedentes de la investigación

Durante la búsqueda de antecedentes relativos a esta investigación se evidenció que no existen aplicaciones informáticas que apoyen la enseñanza de la asignatura en cuestión de la manera abordada en el presente trabajo. Sin embargo, existen varios precedentes de los software educativos en general.

En el mundo se han dado pasos importantes con relación al uso de la informática para la enseñanza. Existen sistemas operativos completos diseñados con este fin, principalmente distribuciones basadas en GNU/Linux, como es el caso de Edubuntu, derivación oficial de la distribución Ubuntu, destinada para uso en ambientes escolares. Incluye aplicaciones educativas como GCompris y la KDE Edutainment Suite.

GCompris es un software educativo con diferentes actividades para niños de 2 a 10 años de edad, que presenta algunas tareas como videojuegos, pero siempre

en sentido educativo. Esta aplicación permite aprender y practicar cálculos y textos, así como abordar el uso del ordenador.

Kdeedu es un paquete de software educativos desarrollados como parte del proyecto KDE. Contiene una larga lista de aplicaciones agrupadas en las categorías de Lenguajes, Matemáticas y Ciencias. En el conjunto de programas para el aprendizaje del lenguaje se destacan:

- Kanagram: es un juego de anagramas.
- Kiten: es un juego de aprendizaje y referencia para el idioma japonés.
- KLatin: es un juego para la enseñanza del latín.
- KVerbos: es una aplicación diseñada para estudiar conjugaciones verbales del español.

Para las matemáticas se distinguen:

- KBruch: aplicación para aprender cálculo con fracciones.
- Kig: es un programa para explorar construcciones geométricas.
- KmPlot: software para dibujar funciones matemáticas.
- KPercentage: aplicación que ayuda a mejorar las habilidades en el cálculo de porcientos.

En el conjunto de los software para las ciencias se encuentran:

- Kalzium: muestra información sobre la tabla periódica de los elementos.
- KStars: aplicación para aprender astronomía.

Así mismo, existen otras aplicaciones con propósitos diversos, como son:

- blinKen: es una versión informática del juego *"Simón dice"*.

- KGeography: es un programa para aprender geografía.
- KTouch: aplicación para aprender a escribir a máquina.
- KTurtle: entorno para la enseñanza de técnicas de programación que usa el lenguaje Logo.
- KWordQuiz: es una herramienta para aprender nuevo vocabulario.

En Cuba la educación ha sido uno de los objetivos fundamentales de la estrategia de informatización de la sociedad cubana, concebida para su desarrollo y modernización. Desde el curso 2001-2002 con la introducción de las computadoras como medios de enseñanza en el sistema nacional comenzaron a concretarse acciones para transitar progresivamente hacia un uso masivo de las aplicaciones educativas y abrir nuevas posibilidades para el desarrollo de un entorno educativo realmente efectivo.[26]

La experiencia nacional en software educativos se manifiesta sobre todo en los paquetes de aplicaciones dedicadas a los diferentes niveles educacionales del sistema nacional de enseñanza, concebidos desde la categoría preescolar, transitando por la primaria y especial, la secundaria, hasta el nivel medio superior.

CubaEduca es el portal educativo del Ministerio de Educación de la República de Cuba, es coordinado por la Dirección Nacional de Informática Educativa y constituye un canal de comunicación interactivo, sistémico, dinámico, comprometido con el currículum de la escuela cubana, dirigido y confeccionado por docentes y técnicos. Es a todas luces una opción de búsqueda, de facilitación del proceso de enseñanza-aprendizaje. Entre sus objetivos fundamentales se

[26] Periódico Granma [Consultado en línea 03/06/2014]
http://www.granma.cu/granmad/secciones/cienciaytec/informatica/informatica19.htm

encuentra el de brindar a los docentes, profesores en formación, estudiantes y familia, recursos y herramientas para el aprovechamiento pedagógico de las TIC.[27]

El portal CubaEduca ofrece entre sus recursos el catálogo de los software educativos desarrollados, agrupados en cuatro colecciones: *A Jugar* (preescolar), *Multisaber* (primaria y especial), *El Navegante* (secundaria básica) y *Futuro* (media superior). A continuación se presenta una breve lista de los software educativos disponibles por cada colección:

- Colección *A Jugar* (Enseñanza Preescolar):
  - *Jugar y Aprender*: Es el software principal de la colección. Está dirigido al desarrollo de las habilidades intelectuales generales.
  - *Ser Como Ellos*: es un software para el conocimiento del mundo social y la formación de valores. Muestra pequeñas anécdotas ilustradas donde se destaca la amistad, la honestidad y la honradez entre otras.
  - *Un Mundo Para Contar*: propuesto para el conocimiento del mundo de los objetos y sus relaciones, además del mundo de las cantidades. Pretende ejercitar el conocimiento de los niños sobre el mundo sensorial.
- Colección *Multisaber* (Enseñanza Primaria y Especial):
  - *El Ratón y la Ventana*: Proporciona información sobre el desarrollo y evolución de los equipos de cómputo, características del hardware y software, entre otros. Aborda además el sistema operativo Windows.
  - *Así es mi País*: Contiene información acerca de las características físico-económico-geográficas de la República de Cuba. Se

[27] Portal web CubaEduca [Consultado en línea 03/06/2014] http://www.cubaeduca.cu/index.php?option=com_content&view=article&id=16467023&Itemid=253

describen y explican los símbolos patrios. Incluye videos, mapas y referencias bibliográficas de la isla de Cuba que se puede acceder a ellos en cualquier momento que el alumno lo necesite para resolver con éxito las tareas que se proponen, entre otras funcionalidades.

- *El País de los Números*: Dispone de información acerca del trabajo con números naturales y fraccionarios (numeración, el trabajo con magnitudes, cálculo, mínimo común múltiplo y tanto por ciento). Cuenta con ejercicios variados y graduados en diferentes niveles de complejidad que permiten reforzar los conocimientos a través de un entorno educativo, atractivo y dinámico, además de otras posibilidades.

❖ Colección *El Navegante* (Enseñanza Secundaria Básica):

- *GeoClío*: En este software aparecen los contenidos que se abordan en el programa de 8vo grado relacionados con la "Historia Moderna y Contemporánea". Se caracteriza por la integración de diversos conocimientos de Geografía, Artes, Literatura e Historia. Desde los textos se propicia un enfoque ideológico de personalidades históricas.
- *EducArte*: Corresponde a la asignatura Educación Artística que se imparte en los tres grados de la Educación Secundaria Básica.
- *Aprende Construyendo*: El contenido se corresponde con el plan de estudios de la asignatura Educación Laboral 9no grado, además se incluyen los temas de: Construye con papel, con plástico y mecanismos, los que se consideran necesarios pues amplían la concepción de la asignatura y ofrecen una solución a la carencia de medios técnicos en los talleres docentes.

- Colección *Futuro* (Enseñanza Media Superior):
  - *Eureka*: El software Eureka tiene como objetivo formar en los estudiantes procedimientos generales y particulares que propicien en los mismos, independencia en la solución de ejercicios y problemas, para lo cual se dan sugerencias que se ejemplifican en cada caso. Se incluyen facilidades como un sistema para acceder a las efemérides, a un forum de discusión, un visualizador/editor de noticias y un sistema de navegación web.
  - *ADN*: Este programa de Biología incluye los siguientes contenidos: el nivel celular; los virus biológicos y sus características. En el nivel de organismo, se profundiza en el enfoque de unidad y diversidad, dada su complejidad estructural, el proceso de regulación de las funciones biológicas y sus resultados, así como las funciones vegetativas y la reproducción como atributos de todos los seres vivientes.
  - *Sophía*: Este software educativo está dirigido a la preparación de los estudiantes potencialmente talentosos para su participación en los concursos de conocimientos de la Educación Preuniversitaria en las asignaturas: Matemática, Física, Química e Informática. En el mismo se abordan los temas que los estudiantes deben conocer para participar en estos eventos, se han elegido estas asignaturas debido a que las mismas reciben una atención especial y son en las que Cuba participa en Olimpiadas Internacionales.

1.4. Tendencias tecnológicas aplicadas a la investigación

1.4.1. *Software Libre (SWL)*

La Free Software Foundation (FSF) refiere que el software libre significa la libertad de los usuarios para ejecutar, estudiar, distribuir, cambiar, copiar o mejorar los programas.

Según Richard Stallman, el SWL es un asunto de libertad, no de precio. El término SWL, en inglés *free software,* tiende a confundir, pues *free* significa tanto *libre* como *gratuito* en esa lengua, y la FSF recalca que la palabra *free* se usa en su acepción de *libertad* y no en términos de *gratuidad*.

Tampoco el SWL debe ser entendido como *software de dominio público*, categoría esta que se refiere a las aplicaciones que no requieren licencia dado que sus derechos de explotación son para todo el que desee usarlas y pertenece a todos por igual.

Libertades del SWL

- La libertad de ejecutar el programa para cualquier propósito (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de mejorar el programa y publicar sus mejoras, y versiones modificadas en general para que se beneficie toda la comunidad (libertad 3). El acceso al código fuente es una condición necesaria.

En varios países se favorece la aplicación y el desarrollo del SWL en las administraciones públicas, ya sea migrando total o parcialmente sus servidores y sistemas de escritorio o subvencionándolos. Entre los ejemplos más conocidos

están Cuba, Venezuela, Alemania, Brasil, Chile, Argentina, China, Ecuador, Colombia, España, Francia, México y República Dominicana.

Entre las instituciones que usualmente prefieren el SWL se encuentran las universidades porque esta tecnología satisface más sus necesidades debido a la naturaleza de la apertura de su código y la libertad de compartir los resultados obtenidos, además de que acarrea gastos por concepto de licencias de uso del software.

Las ventajas económicas que ofrece el SWL hacen atractiva y necesaria para nuestro país la migración hacia esta tecnología. A tono con esta realidad se decide el uso de herramientas y tecnologías libres para desarrollar la aplicación que se introduce en el presente trabajo. Además, es deseable que el software se ejecute también en ambientes Linux.

1.4.2. Lenguajes de programación

Un lenguaje de programación es un lenguaje artificial que puede ser utilizado para controlar el comportamiento físico y lógico de una máquina computadora, para articular algoritmos con precisión o como forma de comunicación humana. Está conformado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Se le llama programar al proceso en el que se escribe, se depura, se compila y se mantiene el código fuente de un programa informático.

Los lenguajes de programación contienen varios elementos con los que puede construirse un software. Estos elementos son *variables y vectores*, *condicionantes*, *bucles*, *funciones*, *sintaxis*, *semántica estática* y *sistema de tipos*.

Los lenguajes de programación pueden clasificarse por el paradigma que suscriben. Los paradigmas principales son:

- Imperativo
- Declarativo
- Orientado a objetos

- Orientado a aspectos (relativamente reciente)

Java

Es un lenguaje de programación orientado a objetos que fue desarrollado por Sun Microsystems en los 90's. Su sintaxis es parecida a C y C++, pero posee un patrón de objetos más uniforme y elimina totalmente las herramientas de operación a bajo nivel, las que normalmente inducen a errores. La gestión de la memoria es realizada por el mismo lenguaje, evitándole así al programador ocuparse de estos menesteres. Las principales características de Java son[28]:

- *Simple*: Java ofrece todas las funcionalidades de un lenguaje potente. Java se diseñó a partir de C y C++ para facilitar su aprendizaje debido a que los últimos, en la época del surgimiento de Java, eran los lenguajes de programación más difundidos. Java se ocupa por sí mismo de liberar la memoria que no se usa mediante un mecanismo llamado *garbage collector.*
- *Orientado a objetos*: Java trabaja con sus datos como objetos, los cuales tienen interfaces de entrada y salida de datos. Asimila perfectamente las características de la filosofía orientada a objetos: encapsulación, herencia y polimorfismo. Una definición general de un objeto es una *clase*, y un objeto específico de estas plantillas se llama *instancia*.
- *Distribuido*: Java ha sido construido con capacidades de interacción mediante el protocolo TCP/IP. Maneja librerías de rutinas para usar las potencialidades de los protocolos HTTP y FTP. Esto facilita que los programadores accedan a informaciones a través de la red. Proporciona las librerías y herramientas para que los programas puedan ser distribuidos.

[28] Zukowiski, J. "Programación Java2 J2SE 1.4". La Habana: Editorial Félix Varela, 2006. Volumen 1.

- *Robusto*: Java realiza verificaciones en busca de problemas en tiempo de compilación y en tiempo de ejecución. Detecta errores rápidamente gracias a la comprobación de tipos en el ciclo de desarrollo. Obliga a la declaración explícita de métodos, reduciendo de esta manera las posibilidades de error.
- *Arquitectura neutral*: Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (run-time) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. Actualmente existen sistemas run-time para Solaris, SunOs, Windows, Linux, Irix, Aix, Mac y Apple. Esto permite que Java pueda ser usado en todas estas plataformas y que no dependa de los sistemas operativos.
- *Seguro*: Como lenguaje, Java tiene características para prevenir el acceso ilegal a la memoria. El código pasa por varias pruebas antes de ejecutarse en los ordenadores. El código se pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal.
- *Multihebra*: Java soporta la sincronización de varios hilos de ejecución, esta característica es llamada multithreading a nivel de lenguaje es especialmente útil para el desarrollo de aplicaciones distribuidas.

Varias características de Java (entre ellas su diseño, su robustez, el respaldo de la industria y su portabilidad) le han valido para convertirse en uno de los lenguajes de programación de mayor evolución y de uso crecidamente versátil en distintas esferas de la industria de la computación. Actualmente Java está presente en dispositivos móviles y sistemas empotrados, en navegadores web, en sistemas de servidor y en sistemas de escritorio, entre otras.

Debido a las características de este lenguaje anteriormente descritas y agregando que el programador tiene experiencia en el desarrollo de aplicaciones con este lenguaje se decide el uso del mismo para ejecutar del proyecto.

1.5. Herramientas

Una herramienta de programación es todo software del que hace uso un programador para crear, depurar, gestionar y/o mantener un programa informático.

1.5.1. Entorno de desarrollo integrado NetBeans

NetBeans es un proyecto de código abierto muy exitoso, con muchos usuarios y una comunidad en constante crecimiento. *Sun Mycrosistems* fundó este proyecto en el 2000 y continúa siendo el patrocinador principal de los proyectos. Esencialmente se usa para el desarrollo de aplicaciones en el lenguaje de programación Java, creado también por *Sun Mycrosistems*.

NetBeans comenzó siendo un proyecto estudiantil en la República Checa, y originalmente se llamó Xelfi, porque el colofón era escribir un IDE (*Integrated Development Environment*) para Java similar a Delphi. Xelfi fue el primer IDE escrito en Java, su lanzamiento se hizo en 1997.[29]

Ocurrió algo curioso durante el desarrollo de *NetBeans*, y es que muchos usuarios empezaron a construir aplicaciones usando la herramienta con sus propios *plug-ins*, lo cual generó un mercado de fuerza considerable. La plataforma brinda servicios habituales en las aplicaciones de escritorio, permitiéndole al programador concentrarse solo en la lógica de su software.

El IDE *NetBeans* es una herramienta para programadores, pensada para escribir, compilar, depurar y ejecutar programas en muchos lenguajes de programación, sin restricciones de uso. Soporta el desarrollo de todos los tipos de

[29] Breve historia de NetBeans. [Consultado en línea 11/06/2014] http://www.netbeans.org/about/history.html

aplicación Java. Entre sus principales características se encuentran el control de versiones y el *refactoring*. *NetBeans* es multiplataforma y necesita como mínimo tener previamente instalada la máquina virtual de Java.

Además de las bondades antes mencionadas se decide el uso de esta herramienta para el desarrollo del proyecto porque el desarrollador posee experiencia en su uso.

1.5.2. Visual Paradigm

Visual Paradigm es una herramienta CASE (siglas de Ingeniería de Software Asistida por Computador, en inglés) que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y diseño, hasta la generación de código fuente de las aplicaciones y la documentación. Ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a menor costo. Provee funcionalidades para la representación de todos los diagramas necesarios en el ciclo de desarrollo de aplicaciones. Fue diseñada para una extensa escala de usuarios motivados por la construcción de sistemas de software de forma fiable a través del uso de una perspectiva orientada a objetos. Aunque es una herramienta privada, existe una versión libre y gratuita.

Visual Paradigm es un producto de calidad, soporta el diseño de aplicaciones web y varios idiomas, es fácil de instalar, así como de actualizar y la licencia para su uso es gratuita. Algunas características principales de Visual Paradigm son[30]:

- Está disponible en las plataformas Linux y Windows.
- El diseño está centrado en los casos de uso y enfocado al negocio, lo que genera un software de mayor calidad.
- Se usa un lenguaje estándar que es común a todo el equipo de desarrollo que facilita la comunicación.

[30] Sitio de descargas de software: [Consultado en línea 11/06/2014] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/

- Posee capacidades de ingeniería directa e inversa.
- El modelo y el código permanecen sincronizados en todo el ciclo de desarrollo.
- Soporta aplicaciones web.
- Es fácil de instalar y actualizar.
- Existe compatibilidad entre ediciones.
- Soporta ORM.
- Genera informes.

Además de las características planteadas con anterioridad, Visual Paradigm se escoge como herramienta CASE para el proyecto porque el desarrollador posee pericia en su empleo.

1.6. Metodologías de desarrollo de aplicaciones informáticas

Una metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. La finalización en buen término de una aplicación informática está determinada necesariamente por varias actividades y etapas, las cuales se ven influenciadas dramáticamente por la metodología escogida para su desarrollo. Crear un software no es una tarea fácil y existen distintas estrategias para obtener un producto informático de calidad. Por una parte hay propuestas que se concentran en la gestión y otras se esfuerzan más en el desarrollo.

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de programas. A menudo se comparan con un plan de contingencias en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando también quiénes deben participar en el desarrollo de las actividades y

qué papel deben tener, especificando además la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

1.6.1. Extreme Programming (XP)

La Programación Extrema es una metodología ligera de desarrollo de aplicaciones informáticas que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Es la más notoria de las técnicas ágiles de desarrollo de software y como todas ellas, agrega más importancia a la adaptabilidad que a la previsibilidad.

Los favorecedores de XP advierten que las variaciones de los requisitos sobre el curso del ciclo de vida de la construcción son una propiedad normal, necesaria e incluso preferible del desarrollo de proyectos. Creen que ser capaz de ajustarse a los cambios de requisitos en cualquier punto de la vida del plan es un acercamiento superior y más objetivo que pretender precisar todas las necesidades al inicio del proyecto y emplear energías posteriormente en vigilar los cambios en los requisitos.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Cuanto más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.[31]

1.6.2. Scrum

Scrum es un marco de trabajo para el desarrollo de aplicaciones basado en un proceso iterativo e incremental, que es manejado usualmente en ambientes instituidos en el desarrollo ligero de software. Es un patrón de referencia que precisa un cúmulo de prácticas y roles, y que puede tomarse como partida para precisar el transcurso de desarrollo que se ejecutará durante un proyecto.

---

[31] Joskowicz. J.: "Reglas y prácticas en XP" [Consultado en línea 13/06/2014] http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf

Los roles primordiales en Scrum son:[32]

- *ScrumMaster*, que mantiene los procesos y trabaja de manera análoga al director de proyecto.
- *ProductOwner*, que representa a los *stakeholders* (interesados externos o internos)
- *Team* que contiene a los desarrolladores.

Scrum permite la creación de equipos auto-organizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Scrum define métodos de gestión y control para complementar la aplicación de otros métodos ágiles como XP que, centrados en prácticas de tipo técnico, carecen de ellas.

1.6.3. SXP

Según autores cubanos[33] *SXP* es un híbrido cubano de metodologías ligeras que tiene como fundamentos las metodologías *Scrum* y *eXtreme Programming*. Está compuesta por cuatro fases principales, y entre todas se despliegan un conjunto de siete flujos de trabajo:

- *Planificación-Definición*: Se establece la visión, se fijan las expectativas y se asegura el financiamiento del proyecto
  - Concepción inicial.
  - *Captura de requisitos.*
  - *Diseño de metáforas.*

[32] Schwaber, Ken: "Agile Project Management with Scrum", Microsoft Press, ISBN 0-7356-1993-X

[33] García de la Puente, S. J. et. al.: "SXP, metodología de desarrollo de software". Serie científica de la Universidad de las Ciencias Informáticas UCI No. 11, Vol. 4, Año: 2011

- *Desarrollo*: Se realiza la implementación del sistema hasta que esté listo para entregarse.
    - *Implementación.*
    - *Pruebas.*
- *Entrega*: Es la puesta en marcha del producto.
    - *Entrega de la documentación.*
- *Mantenimiento*: Se realiza el soporte para el cliente.
    - *Soporte e investigación.*

De estos flujos se ejecutan muchas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, descripción de las Historias de Usuario, diseño, implementación, organización de las iteraciones y las actividades que se efectúan para conseguir el producto, pruebas; conjuntamente con los trabajos ineludibles para cometer las pesquisas para documentar todo el proceso.

Estos flujos y actividades conciben artefactos para la documentación del expediente del proyecto. *SXP* robustece la labor en equipo y lo orienta en una misma trayectoria, con un propósito manifiesto y además admite acompañar transparentemente el progreso de las tareas que se consuman.

Roles

Para la realización de las actividades relacionadas anteriormente se definen los siguientes roles[34]:

- *Líder del Proyecto (Scrum Master)*: Es un rol de administración que debe asegurar que el proyecto se esté llevando a cabo de acuerdo con las prácticas y que todo funcione según lo planeado. Su principal trabajo es remover impedimentos y reducir riesgos del producto, así como coordinar

[34] Íbdem [33]

y facilitar las reuniones, asegurar que se consigan los objetivos de la reunión de planificación de la iteración y determinar cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

- *Gerente (Manager)*: Es el responsable de tomar las decisiones finales acerca de estándares y convenciones a seguir durante el proyecto. Participa en la selección de objetivos y requerimientos y en la selección del Usuario Interno. Tiene la responsabilidad de controlar el progreso y trabaja junto con el Jefe de Proyecto en la reducción de la Lista de Reserva del Producto. Realiza el seguimiento del progreso de cada iteración, verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presente. Puede ocupar el rol de programador master (líder de desarrollo) durante la etapa de desarrollo.
- *Especialistas*: Son los responsables del proceso global. Es necesario que conozcan a fondo el proceso, ya sea de la metodología utilizada o cualquier otro proceso o elementos de gran importancia para el desarrollo de software. Particularmente es una especialización que está activa, los miembros del grupo de trabajo que la desempeñan siempre están ejecutándola y alcanzando un grado mayor de conocimientos en el tema. Ejemplos: Ingenieros de Software, Especialistas de Servicios (migración a software libre, jefe del consejo editorial), Especialistas en diseño gráfico, etc.
- *Consultor*: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan sugerir problemas, además aportan ideas y experiencias para el beneficio del sistema en desarrollo. Esta es una especialización menos activa, quien la ejecuta funciona en este rol por un corto período de tiempo. Está más orientada a roles de desarrollo, y sus ejecutores pueden trabajar en otros roles (en otro equipo) durante el desarrollo del

software. Ejemplos: Consultor de desarrollo web, consultor de desarrollo de escritorio, diseñador de base de datos, etc.

- *Cliente (Customer)*: El cliente participa en las tareas que involucran la lista de reserva del producto.

    1. *Presentar la Reserva del producto al equipo, enfatizando el valor y prioridades del mismo.*
    2. *Definir la meta de la iteración.*
    3. *Aprobar las modificaciones en la Reserva del producto y en el alcance de la iteración.*

- *Miembros del Proyecto (Scrum Team)*: Es el equipo del proyecto que tiene la autoridad para decidir cómo organizarse para cumplir con los objetivos de un Sprint. Sus tareas son: Estimar esfuerzo, crear la reserva del Sprint, revisar la Lista de Reserva del Producto y sugerir obstáculos que deban ser removidos para cumplir con los ítems que aparecen. Típicamente es un equipo de entre 5 y 10 personas cada una especializada en algún elemento que conforma los objetivos a cumplir, por ejemplo: Programadores, Diseñadores de Interfaz de usuario, etc. La dedicación de los miembros del equipo deberá ser a tiempo completo con algunas excepciones. La membresía solo puede cambiar entre sprints (no durante). Los roles dentro del equipo de miembros del proyecto son:

    - *Programadores (Programmers)*: Es el encargado de producir el código y escribir las pruebas unitarias. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
    - *Analista (Analyst)*: Es el encargado de escribir las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide

cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio, todo esto lo realiza junto con el cliente.

- *Diseñadores (Designers)*: Encargados del diseño del sistema; así como el de los prototipos de interfaces, máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción.
- *Encargado de Pruebas (Tester)*: Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- *Arquitecto (Architect)*: Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño en grande del sistema. Ayuda en el diseño de las metáforas.
- *Gestor de Investigaciones*: Persona encargada de gestionar todas las tareas investigativas que se desarrollan. Tiene como responsabilidad planificar el desarrollo de las investigaciones, verificar el cumplimiento y calidad de las mismas. Es el responsable de la elaboración del Plan de Desarrollo de las Investigaciones (PDI). No necesariamente debe tener conocimientos informáticos, pero si algún dominio de metodología de Investigación.

Debido a las características de SXP se decide que esta será la metodología de desarrollo de software que se empleará para la creación de la aplicación informática que será el producto de esta investigación.

1.7. Conclusiones del capítulo

De manera general se han introducido en este capítulo los conceptos fundamentales que aborda la investigación, se han definido que las tendencias tecnológicas existentes en el mundo en la actualidad que serán usadas en el proyecto serán el software libre y el lenguaje de programación Java y han sido precisadas las herramientas *NetBeans* y *Visual Paradigm* para el desarrollo del software, además de la metodología SXP para la elaboración del mismo.

CAPÍTULO 2: PASOS INICIALES. EXPLORACIÓN Y PLANIFICACIÓN DEL PROYECTO

2.1. Introducción

En esta etapa se desarrollan las fases de Exploración y Planificación de la metodología que se usa para la construcción del software (SXP), además se generan documentos como la concepción general del sistema, que presenta información relativa al proyecto que se llevará a cabo, se realiza una breve descripción de los procesos que se informatizarán en la aplicación y se incluyen documentos como la lista de reserva del producto, las historias de usuario, la estimación del esfuerzo necesario por cada historia de usuario y la lista de riesgos. Adicionalmente se introduce el modelo del diseño y los planes de las iteraciones, de la duración de las mismas y de entrega de cada funcionalidad.

2.2. Breve descripción de los procesos a informatizar

La asignatura de Sistemas Operativos que se imparte en el tercer año de la carrera de Ingeniería Informática en el ISMMM consta de seis temas, enunciados a continuación:

- Tema I: Conceptos fundamentales de los sistemas operativos. Es un tema netamente teórico, en el que se introducen los conceptos que son imprescindibles para adentrarse en el estudio de los sistemas operativos.
- Tema II: Gestión de procesos. Los objetivos de este tema comprenden la interpretación de conceptos, técnicas y algoritmos más generales inherentes a la gestión de procesos en los sistemas operativos.
- Tema III: Administración de Memoria. Las metas de este contenido engloban la interpretación de los conceptos, los algoritmos y las técnicas

más habituales referidas a la administración de la memoria en los sistemas operativos.

- Tema IV: Sistemas de archivos. El tema comprende, entre otros aspectos, la seguridad de los archivos en los sistemas operativos.
- Tema V: Administración de Entrada/Salida (E/S). Los propósitos de este contenido implican la asimilación de las nociones, algoritmos y procedimientos más comunes que se estudian en la administración de los dispositivos de entrada y salida de información en los ordenadores.
- Tema VI: Bloqueos. Este apartado se ocupa de examinar los elementos, algoritmos y técnicas fundamentales presentes en los interbloqueos que ocurren en los sistemas informáticos.

El producto de esta investigación se propone informatizar los procedimientos que se llevan a cabo para resolver los ejercicios que se exponen en la asignatura. En esta sección se presenta una breve descripción de los procesos que se han llevado a código en la aplicación, resumiendo brevemente en qué consisten, qué datos se necesitan y qué se obtiene al finalizar cada uno de ellos.

2.2.1. Planificación de procesos

El procedimiento consiste en obtener la planificación de los procesos cuyos datos (tiempo de llegada, tiempo de CPU1, tiempo de E/S, tiempo de CPU2) se conocen de antemano. Esta planificación se realiza mediante los algoritmos siguientes: First Come, First Served (FCFS), Shortest Job First (SJF), Shortest Remaining Time First (SRTF) y Round Robin (RR).

El algoritmo FCFS es el que se sigue en cualquier cola sin prioridades, el primer proceso en llegar será el primero en ser servido.

En el caso del algoritmo SJF el criterio que se sigue es el de menor tiempo restante. El proceso que cumpla esta condición se ejecutará hasta terminar, y así sucederá con el resto de los procesos.

Para el algoritmo SRTF el razonamiento es el mismo que para el SJF, sin embargo, esta vez se trata de la versión apropiativa del citado algoritmo, por lo que en cada instante se verifica si existe otro proceso cuyo tiempo de ejecución inmediato sea menor que el que está en ejecución.

Igualmente, el algoritmo RR es apropiativo, a diferencia de los dos primeros. RR usa un elemento adicional llamado quantum de tiempo. El criterio de este algoritmo es que cada proceso se puede ejecutar tantos instantes de tiempo como permite el quantum. Al completarse este conteo, se carga en el estado de ejecución otro proceso.

Para la planificación de procesos es importante conocer el comportamiento de algunos parámetros relativos al desempeño de los algoritmos. Estos son:

- *Porciento de uso de CPU*, el cual se obtiene por la fórmula:

$$\%UsoCPU = \left(\frac{TpoUsoCPU}{TpoTotal}\right) * 100$$

  donde *TpoUsoCPU* es la cantidad de instantes de tiempo en los que se usó la CPU y *TpoTotal* es el tiempo sucedido desde que se crea el primer proceso hasta que termina el último.

- *Tiempo de Espera para cada proceso*, que es la cantidad de instantes de tiempo que pasa cada proceso en la cola de Listos.

- *Tiempo de Retorno para cada proceso*, que es el tiempo que pasa desde que el proceso se crea hasta que se termina.

- *Tiempo de Espera Promedio*, es el promedio de los tiempos de espera de cada proceso, denotado por la fórmula siguiente:

$$\frac{\sum_{i=0}^{n} TpoEspera_i}{n}$$

- *Tiempo de Retorno Promedio*, se trata del promedio de los tiempos de retorno de cada proceso, denotado por la fórmula siguiente:

$$\frac{\sum_{i=0}^{n} TpoRetorno_i}{n}$$

2.2.2. Paginación de memoria virtual

La paginación es una de las técnicas que usa el mecanismo de Memoria Virtual (MV) para gestionar las solicitudes de este recurso que hacen los procesos. Esta valiosa técnica consiste en particionar la memoria física en pequeñas unidades llamadas marcos, que tienen el mismo tamaño que sus equivalentes en la memoria virtual, las páginas.

La MV es un mecanismo que permite que los programas excedan el tamaño de la memoria física, y que aplicaciones tan grandes como juegos de 7, 8 o 10 Gb corran en sistemas informáticos de apenas 4 Gb de RAM.

La paginación consiste en asignar las páginas a los marcos para que se ejecuten ahí. Cuando todos los marcos están ocupados y una nueva página hace pedido de espacio para correr, la MMU (Unidad de Administración de Memoria -por sus siglas en inglés-, que es una parte del Sistema Operativo) se encarga de seleccionar una de las páginas que están en memoria para despojarla del marco, con el objetivo de cedérselo a la página que está solicitando memoria. Hay varias políticas para escoger una víctima. Las que se implementarán en el módulo Simulación de Administración de Memoria Virtual por Paginación (SIMEP) del ejercitador son FIFO: *First In, First Out*; LRU: *Least Recently Used* y LFU: *Least Frequently Used*.

FIFO se comporta como el algoritmo que se sigue en cualquier cola sin prioridades. Al hacerse una solicitud de memoria, la MMU comprueba si aún existen marcos libres. De haberlos, se asigna uno a la página que hizo la petición. En caso contrario, se dice que hay un “fallo de página”, y se toma el marco de la primera página que se cargó en memoria.

LRU, en cambio, sigue la política de analizar cuál es la página que ha tenido menor uso recientemente, y cede el marco de esta a la página que genere el fallo.

El caso de LFU es similar, pero define qué página ha tenido menor frecuencia de uso, para otorgar el marco de esta a la página que suscite el fallo de página.

2.2.3. Cálculo de direcciones físicas

En un Sistema Operativo los programas referencian direcciones de memoria para leer o escribir valores de variables y otros datos. Con el uso del mecanismo de memoria virtual, las direcciones generadas no se corresponden con las direcciones en la memoria física, puesto que estas direcciones se refieren a las páginas de la memoria virtual. La MMU necesita convertir estas direcciones lógicas a sus equivalentes físicas.

Esta conversión viene dada por la fórmula:

$$DF = TP * M + D$$

donde *DF* es la dirección física que se obtendrá; *TP* es el tamaño de página, que es el mismo para el marco; *M* es el marco correspondiente a la página de la memoria virtual que hace referencia a la dirección lógica. Esta página se obtiene de la fórmula

$$P = \frac{DL}{TP}$$

en la que *P* es el índice de la tabla de páginas en la MV; el numerador DL es la dirección lógica y TP es el tamaño de página.

El elemento *D* en la fórmula de conversión de la dirección lógica a la física está determinado por la fórmula

$$D = DL \,\%\, TP$$

Para la cual *D* es el resto de la división entera de *DL* (dirección lógica) y *TP* (tamaño de página). En la fórmula inicial D representa el desplazamiento en el marco dentro de la memoria física.

2.2.4. Protección de archivos

Los archivos en un Sistema Operativo deben ser protegidos de usuarios que con malicia premeditada por ignorancia intentan hacer modificaciones en aquellos que no son de su propiedad. También existe el caso en el que sea preferible derogarle al propietario las exclusividades sobre algunos de sus ficheros por

alguna necesidad. En cualquier escenario, la protección de archivos se basa en establecer qué acciones pueden realizar tres conjuntos de usuarios sobre estos. En primer lugar, el usuario propietario del fichero; en segundo orden, los usuarios que pertenecen al grupo del propietario y por último, los usuarios clasificados en otros grupos. Las acciones sobre las que se declaran los permisos para cada grupo de usuarios son las de lectura, escritura y ejecución del archivo.

Existen dos formas significativas de indicar los permisos para cada acción. En primer orden, mediante el sistema binario de numeración, donde el 1 se interpreta como que la acción puede ser realizada por el o los usuarios del grupo que se analiza, y el 0 se entiende como la negación del permiso para realizar la acción.

Tabla 2.1: Grupos de usuarios y criterios de permisos

r	w	x	r	w	x	r	w	x
	A			B			C	

En la figura, A indica los permisos del propietario del archivo; b, los de los usuarios que pertenecen al mismo grupo del propietario y C refiere los de los usuarios de otros grupos. Las letras r, w y x expresan las posiciones de las acciones **leer**, **escribir** y **ejecutar**.

Otra manera de indicar los permisos es mediante el sistema octal de numeración, que puede deducirse a partir del sistema binario simplemente tomando tres grupos de bits y convirtiéndolos al sistema decimal, como indica el ejemplo:

Binario:	111	101	001
Octal:	7	5	1

2.2.5. Programación del brazo del disco

Los programas en un Sistema Operativo necesitan leer y escribir datos que persistan en el tiempo aún cuando dichas aplicaciones hayan finalizado su

ejecución. La manera de acceder a las unidades de almacenamiento masivo como los discos duros es a través de un brazo que consta de un cabezal que realiza la transferencia de información. Los discos se dividen en pistas que son solicitadas al brazo del disco.

Existen varias políticas para atender estas solicitudes. En la asignatura se estudian al menos cuatro, las cuales son FCFS: *First Come, First Served*, SSF: *Shortest Seek First*, LOOK: Algoritmo del Elevador y C-LOOK: Búsqueda circular.

El FCFS es, como ya se conoce, la política que se usa en cualquier cola sin prioridades. La petición a la cabeza de la lista es la primera que se satisface.

Para SSF se tiene en cuenta la distancia desde la posición actual del cabezal hasta las pistas solicitadas y se elige siempre la distancia más corta primero.

En LOOK se trabaja como lo hace un elevador común, se atienden las solicitudes en orden ascendente y luego, si han llegado nuevas peticiones, en orden descendente.

Con C-LOOK el caso es similar a LOOK, pero una vez que se llega al otro extremo, el cabezal se sitúa en el principio de las pistas para atender las nuevas solicitudes generadas.

Existen algunos parámetros para caracterizar el desempeño de estos algoritmos. En general, se tienen en cuenta el número de pistas recorridas (PR) para atender cada solicitud, el tiempo empleado para atender cada petición (TAP), el total de pistas recorridas para atender todas las solicitudes (TPR) y el tiempo general de servicio (TGS), que es la sumatoria de los tiempos de atención a cada petición.

En el caso del número de pistas recorridas se obtiene la diferencia entre la pista de la solicitud o pista final (Pf) y la pista donde se encontraba el cabezal antes de atender la solicitud o pista inicial (Pi):

$$PR = Pf - Pi$$

El tiempo de atención a petición (TAP) es la suma del tiempo promedio de acceso (TPA) que es el tiempo que tarda el cabezal en hacer la transferencia de información al disco, y el producto del PR con el tiempo de desplazamiento entre cilindros (TDC), es decir, el tiempo que tarda el cabezal en moverse de un cilindro a otro:

$$TAP = TPA + (PR * TDC)$$

El total de pistas recorridas (TPR) es la sumatoria de las pistas recorridas (PR):

$$TPR = \sum_{i=0}^{n} PR_i$$

El tiempo general de servicio (TGS) indica el total de tiempo dedicado a atender todas las solicitudes (TAP):

$$TGS = \sum_{i=0}^{n} TAP_i$$

2.2.6. Detección de interbloqueos

Los interbloqueos, deadlocks o abrazos fatales son situaciones que suceden a veces en los sistemas operativos. Este escenario compromete el desempeño de los SO y existen varias aproximaciones para enfrentar estos problemas. Una de ellas se denomina algoritmo del banquero, que fue propuesto por Dijkstra en 1965.

Se modela de la forma en que un banquero trabajaría con un grupo de clientes, cada uno de los cuales tiene asignado cierto número de unidades de crédito. El banquero sabe que no todos los clientes necesitarán su crédito máximo en forma inmediata. Para esta analogía los clientes son procesos, los créditos son recursos y el banquero es el SO.

Los clientes emprenden sus respectivos negocios y solicitan préstamos de vez en vez. Si el banquero puede satisfacer las necesidades de los clientes, se dice que el estado es seguro; en caso contrario se llama a la situación estado inseguro.

Este algoritmo puede ser generalizado para el control de varios recursos. Se necesitarán siempre la matriz de asignación de recursos para cada proceso, la de

necesidad o solicitud de recursos para cada proceso, y el vector de disponibilidad de cada recurso del sistema.

El algoritmo consiste en satisfacer las necesidades de cada proceso con los recursos disponibles en cada instante. Si todos los procesos pueden ser finalizados, se dice que el estado inicial es seguro, en caso contrario es inseguro, como se definió anteriormente.

2.3. Fase de Exploración y Planificación

En esta fase se presentan las ideas iniciales sobre el proyecto, se define su clasificación como software, se realiza un análisis de su surgimiento, en qué consistirá, la misión y visión del mismo, el alcance, la metodología a utilizar así como las herramientas que se usarán y la propuesta de solución. Además se presenta la lista de reserva del producto, donde se especifican las funcionalidades que deberá ofrecer el sistema, organizadas por orden de prioridad y catalogadas con una estimación.

Cada una de estas funcionalidades será convertida en historia de usuario, para a partir de su especificación, desarrollarlas mediante las tareas de ingeniería relativas.

2.3.1. Concepción del sistema

El documento de la concepción del sistema es el primer artefacto que genera la metodología SXP. El mismo queda definido gracias a las entrevistas con el cliente, a partir de las cuales se estableció el tipo de proyecto, los factores que propician su surgimiento, los roles que interactúan en el proceso de construcción del software. Se incluyen también la misión y visión que concibe el proyecto, el alcance del mismo, la metodología de desarrollo de software escogida para cumplimentarlo y las características de la solución propuesta.

El documento completo de la concepción del sistema puede ser analizado en el expediente del proyecto

2.3.2. Lista de Reserva del Producto

Luego de elaborar una concepción general del sistema en la que se han establecido varios aspectos significativos que son influyentes en la plausibilidad del proyecto, se procede a definir qué objetivos persigue la aplicación para darles cumplimiento. En la Lista de Reserva del Producto se consignan las funcionalidades que brindará el software, clasificándolas por su prioridad y asignándoles una estimación del esfuerzo que se necesita para desarrollarlas, además de catalogarlos también en funcionales y no funcionales, y se designa quién es el responsable de programarlos.

La Lista de Reserva del Producto es otro artefacto que se genera en las etapas tratadas en este capítulo. Al iniciarse un proyecto es bastante difícil conocer todos los requisitos que se necesita en el producto, y es por esta razón que XP ofrece la posibilidad de agregar más funcionalidades a medida que se conoce más el producto, observando la limitación de hacer los cambios solo entre iteraciones.

Se persigue la intención de que al finalizar la lista el producto que queda sea el más apropiado, conveniente y profesional.

Tabla 2.2: Lista de Reserva del Producto

Funcionalidades				
Asignado a	**Ítem**	**Descripción**	**Estimación**	**Estimado por**
Prioridad			**Muy alta**	
Carlos E. Velázquez (Programador)	1	Detección de interbloqueos	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	2	Planificación de procesos en modo E/S+CPU2 para algoritmos FCFS	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	3	Planificación de procesos en modo E/S+CPU2 para algoritmos SJF	1	Carlos Ernesto Velázquez (Analista)

Carlos E. Velázquez (Programador)	4	Planificación de procesos en modo E/S+CPU2 para algoritmos SRTF	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	5	Planificación de procesos en modo E/S+CPU2 para algoritmos RR	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	6	Planificación de procesos en modo simple para algoritmos FCFS	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	7	Planificación de procesos en modo simple para algoritmos SJF	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	8	Planificación de procesos en modo simple para algoritmos SRTF	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	9	Planificación de procesos en modo E/S+CPU2 para algoritmos RR	1	Carlos Ernesto Velázquez (Analista)
Prioridad			**Alta**	
Carlos E. Velázquez (Programador)	10	Programación del brazo del disco según algoritmo LOOK	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	11	Programación del brazo del disco según algoritmo C-LOOK	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	12	Programación del brazo del disco según algoritmo SSF	1	Carlos Ernesto Velázquez (Analista)
Prioridad			**Media**	
Carlos E. Velázquez (Programador)	14	Paginación utilizando algoritmo FIFO	2	Carlos Ernesto Velázquez (Analista)

Carlos E. Velázquez (Programador)	15	Paginación utilizando algoritmo LRU	2	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	16	Paginación utilizando algoritmo LFU	2	Carlos Ernesto Velázquez (Analista)
Prioridad			**Baja**	
Carlos E. Velázquez (Programador)	17	Protección de archivos	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	18	Cálculo de direcciones físicas	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	19	Ver ejercicios propuestos	1	Carlos Ernesto Velázquez (Analista)
Características del Sistema				
Carlos E. Velázquez (Programador)	20	Apariencia o interfaz externa	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	21	Software	1	Carlos Ernesto Velázquez (Analista)
Carlos E. Velázquez (Programador)	22	Hardware	1	Carlos Ernesto Velázquez (Analista)

2.3.3. Historias de usuario

En XP se usa la técnica de las historias de usuario (HU) para especificar los requisitos del sistema. Las mismas deben acompañarse de las pruebas de aceptación, y son significativas porque gestionan rápidamente las necesidades de los usuarios sin que esto conlleve la necesidad de generar una gran cantidad de documentación formal, evitando consumir tiempo en administrarlos puesto que un requerimiento de software es descrito completa y concretamente de una forma muy sencilla, usando el lenguaje propio del cliente.

Las HU componen la consecuencia directa de la interacción entre los clientes y desarrolladores a través de concilios donde el flujo de ideas establece no solo los requerimientos del proyecto sino también las posibles soluciones. De forma general se detallan concisamente las características que el sistema debe poseer desde el punto de vista del cliente.

En la fase de exploración se evidenció que el sistema contará con 18 historias de usuario que componen todas las funcionalidades de la aplicación. Como ejemplo se presenta la siguiente:

Tabla 2.3: Ejemplo de historia de usuario elaborada

Historias de Usuario	
Número: 1	**Usuario:** Estudiantes
Nombre de historia: Detección de interbloqueos	
Modificación de Historia de Usuario Número: Ninguna	
Prioridad en negocio: Muy alta	**Riesgo en Desarrollo:** Muy alto
Puntos estimados: 1	**Iteración Asignada:** 1
Programador responsable: Carlos Ernesto Velázquez Rodríguez	
Descripción: Se dictamina la seguridad o inseguridad de un estado hipotético del sistema operativo a partir de un conjunto de datos consistentes en una matriz de asignación de recursos (en la que las filas representan los procesos existentes en el sistema y las columnas figuran como las asignaciones de instancias de los recursos que dispone dicho sistema), una matriz de necesidad (en la que las filas representan igualmente los antes mencionados procesos y las columnas, las necesidades de cada recurso por cada proceso) y un vector de disponibilidad (que explicita la cantidad de instancias por recurso que dispone el sistema).	
Observaciones: Se hace referencia a la funcionalidad 1 enumerada en la Lista de Reservas del Producto.	

La totalidad de las historias de usuario identificadas en este proyecto puede ser estudiada en el documento del expediente del proyecto.

2.3.4. Estimación del esfuerzo por historia de usuario

El cliente establece en este punto la prioridad de cada HU detectada y el programador estima el esfuerzo necesario para cumplimentar cada una de ellas. También se establecen acuerdos sobre el contenido de la primera entrega y se determina un cronograma de acuerdo con el cliente. En esta sección se realiza una estimación del esfuerzo para desarrollar cada una de las historias de usuario definidas anteriormente.

Las estimaciones del esfuerzo son determinadas por los programadores utilizando como medida el punto de estimación, que no es más que una semana ideal de trabajo, en la que los miembros del equipo de desarrollo trabajan el tiempo calculado sin ninguna interrupción.

El esfuerzo estimado para desarrollar las historias de usuario identificadas en el proyecto está consignado en el expediente del proyecto

En el sistema que se desarrolla para la presente investigación fueron realizadas las estimaciones pertinentes y están consignadas en el artefacto Estimación del esfuerzo por historia de usuario, que puede ser consultado en el expediente del proyecto.

2.3.5. Plan de Iteraciones y duraciones respectivas

Luego de definidas las HU y establecido el esfuerzo necesario para realizarlas, es momento de constituir la planificación de la etapa de implementación del sistema. Este plan precisa cuántas iteraciones constituyen el desarrollo del sistema y cuáles historias de usuario se desarrollarán en cada iteración, además de proveer una descripción de cada una de estas.

Este artefacto significa una ayuda para organizar mejor el trabajo a realizar y seccionarlo en etapas, proporcionando una distribución factible. A tenor con lo planteado, en el caso del proyecto en desarrollo se han determinado 6 iteraciones.

En el caso del plan de duración de las iteraciones, como su nombre indica, especifica una estimación de la duración de cada iteración definida en el punto

anterior de la metodología. Este documento tiene como fin mostrar la duración de las iteraciones que se desarrollarán en el proyecto.

Tabla 2.4: Plan de duración de las iteraciones

Iteración	Orden de las historias de usuario a implementar	Duración de la iteración
1	1: Detección de interbloqueo 2: Planificación de procesos en modo E/S+CPU2 para algoritmos FCFS 3: Planificación de procesos en modo E/S+CPU2 para algoritmos SJF	3
2	4: Planificación de procesos en modo E/S+CPU2 para algoritmos SRTF 5: Planificación de procesos en modo E/S+CPU2 para algoritmos RR 6: Planificación de procesos en modo simple para algoritmos FCFS	3
3	7: Planificación de procesos en modo simple para algoritmos SJF 8: Planificación de procesos en modo simple para algoritmos SRTF 9: Planificación de procesos en modo simple para algoritmos RR	3
4	10: Paginación utilizando algoritmo FIFO 11: Paginación utilizando algoritmo LRU 12: Paginación utilizando algoritmo LFU	3
5	13: Programación del brazo del disco según algoritmo FCFS 14: Programación del brazo del disco según algoritmo SSF 15: Programación del brazo del disco según algoritmo LOOK	3
6	16: Programación del brazo del disco según algoritmo C-LOOK 17: Cálculo de direcciones físicas 18: Protección de archivos 19: Agregar ejercicios propuestos	3
Total		**24**

2.3.6. Plan de Entregas

En el Plan de Entregas se fijan las fechas de culminación para cada iteración y de entrega de la implementación de cada funcionalidad descrita en las historias de usuario, lo que hace más fácil la administración del proyecto en todas sus etapas.

A continuación se presenta el plan de entregas para el proyecto de software en desarrollo:

Tabla 2.5: Plan de entregas

	Iteraciones	Fechas de entrega
Iteración 1	1: Detección de interbloqueo	21/02/2014
	2: Planificación de procesos en modo simple para algoritmos FCFS	28/02/2014
	3: Planificación de procesos en modo simple para algoritmos SJF	07/03/2014
Iteración 2	4: Planificación de procesos en modo simple para algoritmos SRTF	14/03/2014
	5: Planificación de procesos en modo simple para algoritmos RR	21/03/2014
	6: Planificación de procesos en modo E/S+CPU2 para algoritmos FCFS	28/03/2014
Iteración 3	7: Planificación de procesos en modo E/S+CPU2 para algoritmos SJF	04/04/2014
	8: Planificación de procesos en modo E/S+CPU2 para algoritmos SRTF	11/04/2014
	9: Planificación de procesos en modo E/S+CPU2 para algoritmos RR	18/04/2014
Iteración 4	10: Paginación utilizando algoritmo FIFO	25/04/2014
	11: Paginación utilizando algoritmo LRU	02/05/2014
	12: Paginación utilizando algoritmo LFU	09/05/2014
Iteración 5	13: Programación del brazo del disco según algoritmo FCFS	16/05/2014
	14: Programación del brazo del disco según algoritmo LOOK	23/05/2014
	15: Programación del brazo del disco según algoritmo C-LOOK	30/05/2014
Iteración 6	16: Programación del brazo del disco según algoritmo SSF	06/05/2014
	17: Cálculo de direcciones físicas	13/06/2014
	18: Protección de archivos	20/06/2014
	19: Ver problemas propuestos	22/06/2014

2.4. Conclusiones del capítulo

En este capítulo se inició el desarrollo del producto informático que se propone como solución. Fue establecida la concepción del sistema y se generaron los artefactos de las dos primeras fases de la metodología que se utiliza para la elaboración del proyecto. Del mismo modo la lista de reserva del producto fue especificada, donde se consignaron todas funcionalidades que ofrecerá el ejercitador. Las historias de usuario fueron determinadas, además de la estimación del esfuerzo necesario para el desarrollo de cada una de ellas. Fueron planificadas las iteraciones, la duración de cada una de ellas y las fechas límites para sus entregas.

CAPÍTULO 3: DISEÑO, IMPLEMENTACIÓN Y PRUEBA

3.1. Introducción

Este capítulo desarrolla las etapas de diseño, implementación y prueba de la metodología SXP de la que se hace uso para el desarrollo de la aplicación. En este punto se muestra la arquitectura del sistema, se desarrollan las iteraciones, se describen las interfaces gráficas de usuario y se realizan las pruebas de aceptación del producto informático que se propone en este trabajo.

3.2. Fase de Diseño

La metodología sugiere que en el desarrollo de un proyecto se deben obtener diseños simples y sencillos. Para conseguir un diseño fácilmente entendible y fácil de implementar es vital realizarlo lo menos complejo posible.

En esta fase se especifican las técnicas y principios con el propósito de concretar el sistema, a tal detalle que sea posible su interpretación y realización física. Esta fase convierte componentes estructurales de la arquitectura del software y se define la calidad del proyecto.

El diseño debe efectuar todos los requisitos comprendidos en la Lista de Reserva del producto.

3.2.1. Diagrama de paquetes

En UML o Lenguaje Unificado de Modelado se usa el diagrama de paquetes para mostrar cómo un sistema se divide en agrupaciones lógicas y las dependencias entre ellas. Comúnmente un paquete está concebido como un

directorio, y es por ello que este tipo de diagramas proveen información sobre la jerarquía lógica de un sistema.[35]

Un paquete está normalmente organizado para maximizar la coherencia interna dentro de cada uno de ellos y minimizar el acoplamiento externo entre los mismos. Siguiendo estas directivas, los paquetes son excelentes elementos de gestión. Las relaciones entre ellos pueden indicar el orden de desarrollo demandado.[36]

El diagrama de paquetes del producto que se obtiene en esta investigación es el siguiente:

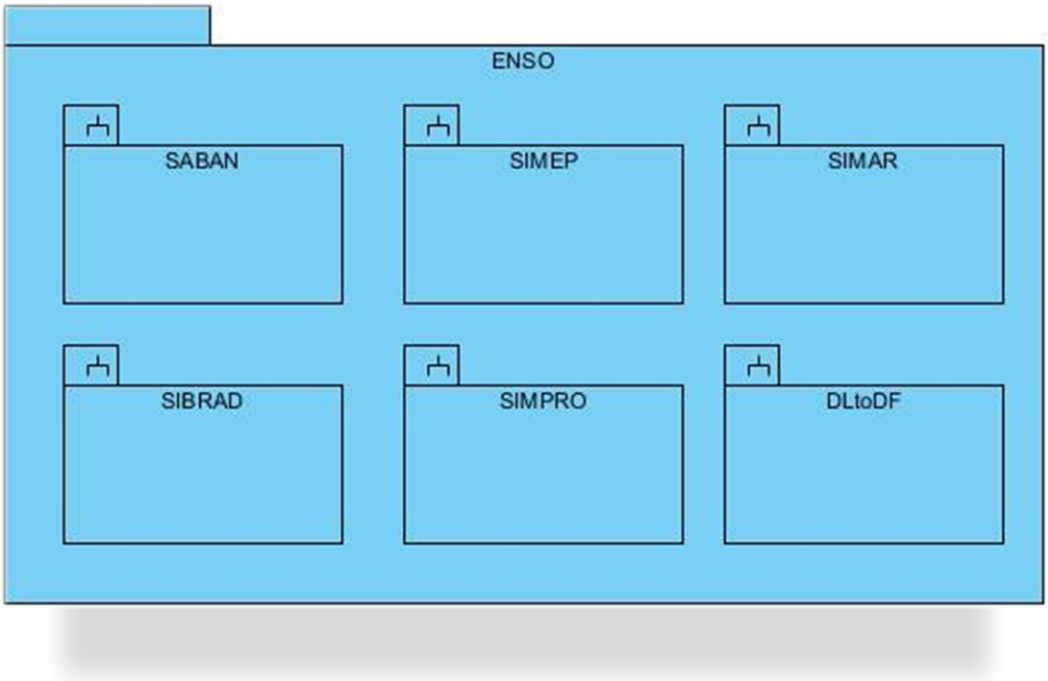

Imagen 3.1: Diagrama de paquetes de ENSO

[35] Introduction to UML 2 Package Diagrams [Consultado en línea 11/06/2014] http://www.agilemodeling.com/artifacts/packageDiagram.htm

[36] Ídem [35]

3.3. Arquitectura del sistema

A las formas y guías generales para desarrollar el proceso de programar se le llama *arquitectura de software*. Similar a los planos de un edificio, la arquitectura de software indica la estructura, funcionamiento e interacción entre las partes del software. La arquitectura es un nivel de diseño enfocado en aspectos que rebasan los algoritmos y estructuras de datos de la programación que consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco, define de manera abstracta los componentes que participan en una aplicación, sus interfaces y la comunicación entre ellos.

Generalmente, no es necesario inventar una nueva arquitectura de software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Las arquitecturas más universales son:

- *Monolítica:* Donde el software se estructura en grupos funcionales muy acoplados.
- *Cliente-servidor:* Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.
    - *Arquitectura de tres niveles:* Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

Otras arquitecturas menos conocidas son:

- En pipeline.
- Entre pares.
- En pizarra.

- Orientada a servicios.
- Dirigida por eventos.
- Máquinas virtuales.

3.3.1. Modelo Vista Controlador

El estilo Modelo Vista Controlador (MVC) fue descrito por primera vez en 1979 por Trygve Reenskaug, quien entonces trabajaba en los laboratorios de investigación de Xerox[37]. Es un patrón de arquitectura de software que separa los datos de la interfaz y la lógica del negocio en tres componentes distintos.

La descripción de los módulos de este patrón es la siguiente:[38]

- *Modelo*: Es la representación específica de la información con la cual opera el sistema.
- *Vista*: Presenta el modelo en un formato adecuado para interactuar, normalmente una GUI (*graphic user interface*, o interfaz gráfica de usuario).
- *Controlador*: Responde a eventos, usualmente acciones del usuario, invoca peticiones al modelo y probablemente a la vista.

3.4. Desarrollo de las iteraciones

Los clientes crean las historias de usuario bajo la supervisión de los desarrolladores del proyecto, es por eso que se convierten en integrantes del equipo. Cuando el cliente puede reconocer con precisión la funcionalidad pretendida, estas historias de usuario son creadas por él mismo. El cliente debe estar presente también cuando se realicen las pruebas de aceptación, pues es

[37] Páginas de Trygve M. H. Reenskaug [Consultado en línea 12/06/2014] http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html

[38] Páginas de Trygve Reenskaug: [Consultado en línea 11/06/2014] http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html

quien determina si la funcionalidad se comporta como desea o como se supone que debe ser.

En las fases de Exploración y Planificación fueron identificadas las historias de usuario que necesita el proyecto en cada una de las iteraciones especificadas, asumiendo las primacías y limitaciones de tiempo que el cliente previno. Las historias de usuario son descompuestas en tareas de ingeniería (TI) o tareas de programación, las que son asignadas a un equipo de desarrolladores que serán quienes tengan la responsabilidad de implementarla. Así se desarrollan las funcionalidades básicas que después concertarán las funcionalidades generales de cada historia. No es necesario que el cliente entienda las tareas, pues sus redacciones serán usadas únicamente por los desarrolladores, por esta razón pueden ser descritas con lenguaje técnico.

Una vez descrito en qué consiste el proceso de desarrollo de las iteraciones, e introducido el concepto de tarea de programación, se muestran a continuación algunas de las que componen las iteraciones previstas para el proyecto:

Iteración 1

Para esta iteración se implementaron 5 historias de usuario, correspondientes a los módulos de Simulación del Algoritmo del Banquero (Detección de Interbloqueos) y Simulación de Planificación de Procesos en el modo E/S+CPU2 para los algoritmos FCFS, SJF, SRTF y RR. Un ejemplo de las tareas de ingeniería desarrolladas para esta iteración es la representada en la tabla 3.1.

Iteración 2

En el transcurso de esta iteración se implementaron 4 historias de usuario (las número 6, 7, 8 y 9) correspondientes al módulo de Simulación de Planificación de Procesos, del modo Simple de los algoritmos FCFS, SJF, SRTF y RR. La tabla 3.2 representa un ejemplo de las tareas de ingeniería desarrolladas para esta iteración.

Tabla 3.1: Tarea de programación 1 de historia de usuario 1

Tareas de Historia de Usuario	
Número tarea: 1	**Número historia: 1 : Detección de interbloqueos**
Nombre tarea: Insertar cantidad de procesos	
Tipo de tarea: Tarea de programación	**Puntos estimados:** 0.2
Fecha inicio: 07/02/2014	**Fecha fin:** 09/02/2014
Responsable: Carlos Ernesto Velázquez Rodríguez (Programador)	
Descripción: El usuario especifica cuántos procesos participarán en la simulación del algoritmo del banquero. La cantidad de filas de la matriz de asignación y de necesidad debe modificarse de acuerdo al valor dado por el usuario.	

Tabla 3.2: Tarea de programación 1 de historia de usuario 4

Tareas de Historia de Usuario	
Número tarea: 1	**Número historia: 4 : Planificación de procesos en modo E/S+CPU2 para algoritmos SRTF**
Nombre tarea: Calcular porciento de uso de CPU	
Tipo de tarea: Tarea de programación	**Puntos estimados:** 0.2
Fecha inicio: 21/04/2014	**Fecha fin:** 23/04/2014
Responsable: Carlos Ernesto Velázquez Rodríguez (Programador)	
Descripción: El porciento de uso de CPU es un parámetro que es necesario calcular para mostrarle al usuario, luego de realizada la simulación de la planificación de los procesos	

Iteración 3

Durante esta iteración se desarrollaron 3 historias de usuario (números 10, 11 y 12), pertenecientes todas al módulo de Simulación de Administración de Memoria por Paginación.

Iteración 4

En esta iteración fueron desarrolladas 4 historias de usuario (identificadas en orden por los números 13, 14, 15 y 16) correspondientes al módulo Simulación de Programación del Brazo del Disco.

Iteración 5

En esta iteración se desarrollaron las tres primeras funcionalidades del módulo de Simulador de Programación del Brazo del Disco, desarrollándose las historias de usuario números 13, 14 y 15.

Iteración 6

Con esta iteración se completaron las funcionalidades del software, agregando el algoritmo restante del módulo anterior, y los módulos de Simulador de Conversión de Direcciones Lógicas a Físicas y el Simulador de Protección de Archivos, desarrollándose las historias de usuario números 16, 17, 18 y 19. Esta última significó agregar los ejercicios propuestos para cada módulo.

Todas las tareas de programación desarrolladas para el proyecto están disponibles para su consulta en el expediente del proyecto.

3.5. Pruebas de aceptación

La planificación y ejecución de pruebas es un paso fundamental en el proceso de desarrollo de software, pues permite detectar errores o fallas en el comportamiento del código implementado. El tratamiento aplicado de pruebas permite que se desarrolle un producto con mayor calidad, lo que ofrece a los programadores una mayor seguridad en la labor que despliegan. Las pruebas de aceptación se ejecutan al consumar cada iteración para comprobar que las funcionalidades requeridas por el cliente fueron efectuadas cabalmente. Gracias a

esta regla se obtiene generalmente un código simple y funcional de manera ágil, lo que implica que sea necesario que las pruebas se lleven a cabo a un 100 %.

Las pruebas de aceptación son equiparables con las pruebas de caja negra que se realizan en algunas metodologías. Se crean a partir de las historias de usuario y no por un listado de requisitos. En el desarrollo de las iteraciones se escogen historias de usuarios que serán convertidas en pruebas de aceptación. Estas han de declarar los escenarios para probar las historias de usuario y que sean correctas a la manera de entender del cliente. Las historias de usuario pueden y deben tener asociadas todas las pruebas de aceptación que sean necesarias para confirmar que el código desarrollado funciona perfectamente y hace justamente todo lo que debe hacer. De hecho, las historias de usuario no se consideran completadas hasta que hayan pasado satisfactoriamente todas las pruebas de aceptación.

Es de esta manera que a continuación se presentan algunas de las pruebas de aceptación llevadas a cabo para confirmar el correcto funcionamiento de la aplicación:

Tabla 3.3: Caso de prueba de aceptación 1.1

Caso de prueba de aceptación	
Codigo: H1_P1	**Historia de usuario #:** 1
Nombre de la persona que realiza la prueba: Carlos Ernesto Velázquez Rodríguez	
Descripción: Prueba para comprobar la validación del módulo SABAN	
Condiciones de ejecución: El usuario debe completar los datos solicitados en la interfaz, los valores de la matriz de asignación, de necesidad y el vector de disponibilidad, en correspondencia con la cantidad de procesos y recursos que ha definido para la simulación.	
Entrada/Pasos de ejecución: El cliente inserta los datos solicitados por el módulo, procede a ejecutar el algoritmo y verifica la salida del procesamiento.	
Resultado Esperado: El módulo devuelve los resultados esperados, para los datos entrados, especificando si el estado es seguro o inseguro y la descripción paso a paso del algoritmo para los datos entrados al módulo.	

Tabla 3.4: Caso de prueba de aceptación 2.1

Caso de prueba de aceptación	
Código: H2_P1	**Historia de usuario # :** 2
Nombre de la persona que realiza la prueba: Carlos Ernesto Velázquez Rodríguez	
Descripción: Prueba para comprobar el funcionamiento del algoritmo FCFS Simple en el módulo SIMPRO	
Condiciones de ejecución: El usuario completa los datos solicitados en la interfaz, es decir, los procesos con sus instantes de llegada, de CPU 1 y de acuerdo al modo (Simple o E/S + CPU 2), los instantes de E/S y CPU 2, además de especificado el algoritmo por el que realizará la planificación.	
Entrada/Pasos de ejecución: El cliente inserta los datos solicitados por el algoritmo del módulo, procede a ejecutar el algoritmo y verifica la salida del procesamiento.	
Resultado Esperado: El módulo devuelve los resultados esperados, la tabla de planificación con los parámetros de interés calculados.	

Así mismo, el conjunto de pruebas llevadas a cabo para comprobar la estabilidad y calidad del sistema pueden ser aquilatadas en el expediente del proyecto.

3.6. Conclusiones del capítulo

En el capítulo que concluye se desarrollaron los detalles sobre el diseño, la implementación y las pruebas llevadas a cabo sobre la aplicación, usando la metodología definida para su desarrollo. Se especificó el patrón MVC como arquitectura del sistema y fueron ampliadas las iteraciones del proyecto. Las tareas de programación fueron concebidas y se realizaron las pruebas de aceptación.

CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD

4.1. Introducción

En este capítulo se abordan los elementos principales sobre el estudio de factibilidad realizado para el uso del software educativo propuesto en este trabajo, que se plantea apoyar la enseñanza de la asignatura de SO en la carrera de Informática en el ISMMM. El estudio consiste en la aplicación de un cuestionario para observar el impacto que causa el uso del programa en los alumnos.

El cuestionario se diseñó adaptando el usado por los profesores venezolanos José Gregorio Guillén y Jorge Luis Briceño, del departamento de Física y Matemática del núcleo universitario “Rafael Rangel”, de la Universidad de Los Andes, en la República Bolivariana de Venezuela. Estos catedráticos publicaron en el año 2011 un documento presentando un software educativo para apoyar la enseñanza del tema de las variaciones y permutaciones en Matemáticas, para los docentes y estudiantes del quinto año de bachillerato del Liceo Bolivariano “Butaque”.

Se decide la adaptación de este método para la evaluación de la factibilidad del software debido a que el producto de la investigación introducida en este documento es un software educativo destinado a un grupo de estudiantes, por lo que la factibilidad del mismo viene dada más por el impacto que causa en los alumnos que por costos por concepto de desarrollo. La investigación llevada a cabo para el desarrollo de este software tiene puntos en común con la de los profesores venezolanos y debido a que en la suya los catedráticos persiguen los mismos intereses que los que dieron inicio a este trabajo, se decide adaptar su fase de evaluación para considerar la factibilidad del producto que se introduce en este documento.

A través del análisis estadístico que se aplicó a cada uno de los ítems se estableció la descripción correspondiente dependiendo de lo expresado en la muestra, que es mayor que la población actual de sujetos. Fueron además establecidas las frecuencias porcentuales y expuestas en tablas y gráficas, las que igualmente se analizaron cualitativamente en función de las respuestas emitidas por los alumnos.

4.2. Aplicación del cuestionario

El cuestionario se diseñó para conocer el impacto que causa la aplicación informática propuesta en los alumnos que se enfrentan a la asignatura de SO en la carrera de Ingeniería Informática en el ISMMM.

Entre los aspectos de la aplicación que se evalúan se circunscriben los atributos sobre la facilidad de uso, el formato de entrada de datos y salida de resultados y si es comprensible para el estudiante los datos que pide y los resultados que muestra. Además de estos, también se relacionan la rapidez con que se desempeña el software, el conjunto de funcionalidades que posee, si el manejo del rango de datos es adecuado o no, si abarca ejercicios de todos los temas y si son suficientes los ejercicios presentados. El modelo del cuestionario puede ser consultado en el anexo 2 de este documento.

Este cuestionario fue aplicado con carácter anónimo y confidencial a una muestra de 18 sujetos. Es importante recalcar que el tamaño de la población varía cada año, pues la cantidad de estudiantes se altera cada curso. En este caso específico, la cantidad de estudiantes de tercer año es de 8, por lo que es notable que la muestra tomada para efectuar el estudio de factibilidad sea mayor que la población presente, pero se ha considerado que el valor actual de la población es muy pequeño como para ser representativo de la cantidad de estudiantes de un aula. La muestra se completó con estudiantes que recibieron la asignatura en años precedentes al que actualmente cursan. A continuación se aprecian los resultados que salieron a la luz al finalizar la recopilación de información.

Observando el significado de las siglas que se presenta en la tabla 4.1, en el caso del aspecto de la interfaz gráfica se obtuvieron los siguientes resultados en la aplicación del cuestionario:

Tabla 4.1: Significado de las siglas

SIGNIFICADO DE LAS SIGLAS				
M	S	R	P	N
Mucho	*Suficiente*	*Regular*	*Poco*	*Nada*

El primer ítem que se tuvo en cuenta para el aspecto de la interfaz gráfica fue el criterio de facilidad de uso. Estos fueron los resultados:

Tabla 4.2: Valoración del ítem 1 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Fácil de usar	12	4	2			18
	M	**S**	**R**	**P**	**N**	**TOTAL**
Fácil de usar (%)	66,67	22,22	11,11			100

La gráfica correspondiente muestra más cómodamente la correlación entre las opiniones de los estudiantes:

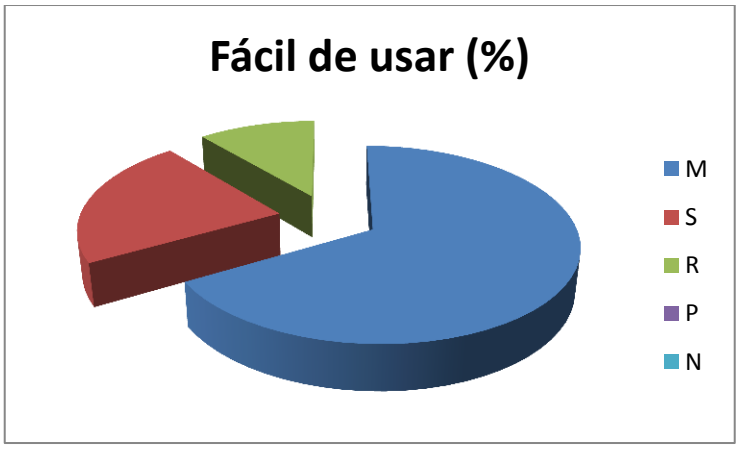

Imagen 4.1: Gráfico para el ítem 1 sobre el desempeño de ENSO

El siguiente ítem a tener en cuenta fue si el formato de los datos de entrada es o no intuitivo:

Tabla 4.3: Valoración del ítem 2 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Formato de entrada de datos es intuitivo	14	3	1			18
	M	S	R	P	N	TOTAL
Formato de entrada de datos es intuitivo (%)	77,78	16,67	5,56			100

Estos números generan la gráfica mostrada a continuación:

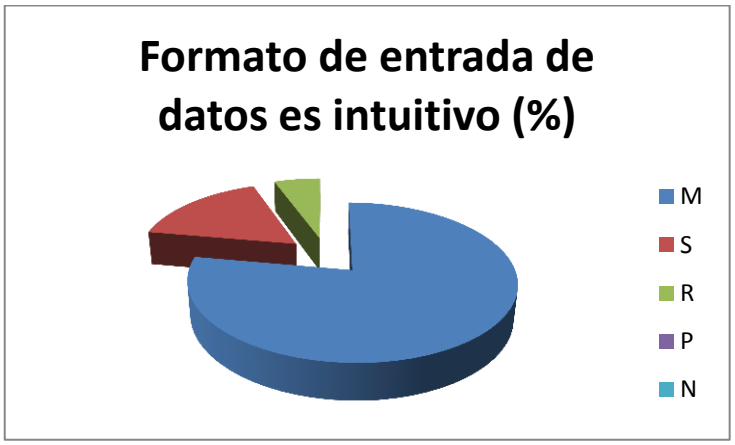

Imagen 4.2: Gráfico para el ítem 2 sobre el desempeño de ENSO

Igualmente se evaluó si el formato de salida de los resultados es o no intuitivo:

Tabla 4.4: Valoración del ítem 3 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Formato de salida de resultados es intuitivo	10	6	2			18
	M	S	R	P	N	TOTAL
Formato de salida de resultados es intuitivo (%)	55,56	33,33	11,11			100

La imagen gráfica que simboliza la relación entre estas respuestas es la próxima:

Imagen 4.3: Gráfico para el ítem 3 sobre el desempeño de ENSO

Luego se valoró si los estudiantes entienden qué datos pide cada módulo, lo que arrojó las siguientes conclusiones:

Tabla 4.5: Valoración del ítem 4 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Entiendo qué datos pide	13	4	1			18
	M	S	R	P	N	TOTAL
Entiendo qué datos pide (%)	72,22	22,22	5,56			100

A su vez estos valores manifestados gráficamente lucen de esta forma:

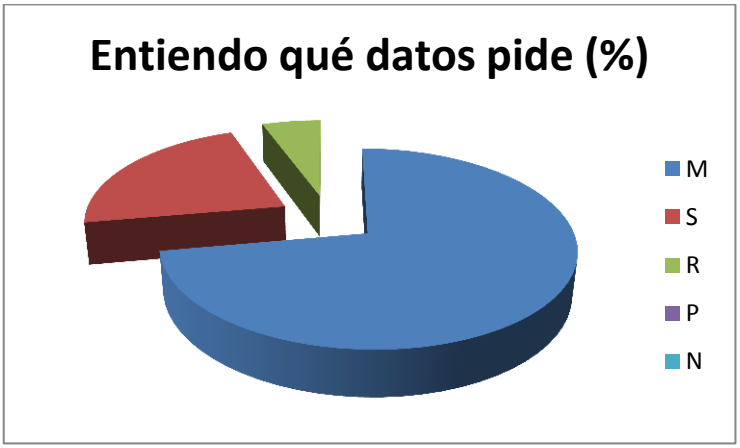

Imagen 4.4: Gráfico para el ítem 4 sobre el desempeño de ENSO

Igualmente se apreció la opinión de los educandos acerca del nivel de comprensión de los resultados que ofrece la aplicación:

Tabla 4.6: Valoración del ítem 5 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Entiendo qué resultados ofrece	12	4	2			18
	M	S	R	P	N	TOTAL
Entiendo qué resultados ofrece (%)	66,67	22,22	11,11			100

Representados en gráficos de pastel estos valores se ven como sigue:

Imagen 4.5: Gráfico para el ítem 5 sobre el desempeño de ENSO

También se evaluó la rapidez con la que el sistema devuelve la información:

Tabla 4.7: Valoración del ítem 6 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Rápida	17	1				18
	M	S	R	P	N	TOTAL
Rápida (%)	94,44	5,56				100

Para estos resultados el gráfico generado es el siguiente:

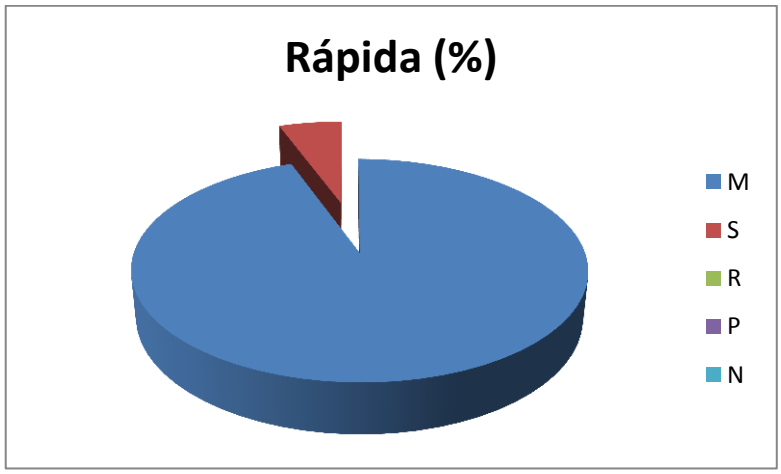

Imagen 4.6: Gráfico para el ítem 6 sobre el desempeño de ENSO

Así mismo, se apreció la opinión de los estudiantes sobre las funcionalidades que ofrece la aplicación teniendo en cuenta los temas de la asignatura:

Tabla 4.8: Valoración del ítem 7 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Funcionalidades	10	7	1			18
	M	S	R	P	N	TOTAL
Funcionalidades (%)	55,56	38,89	5,56			100

Estos valores son representados por el siguiente gráfico:

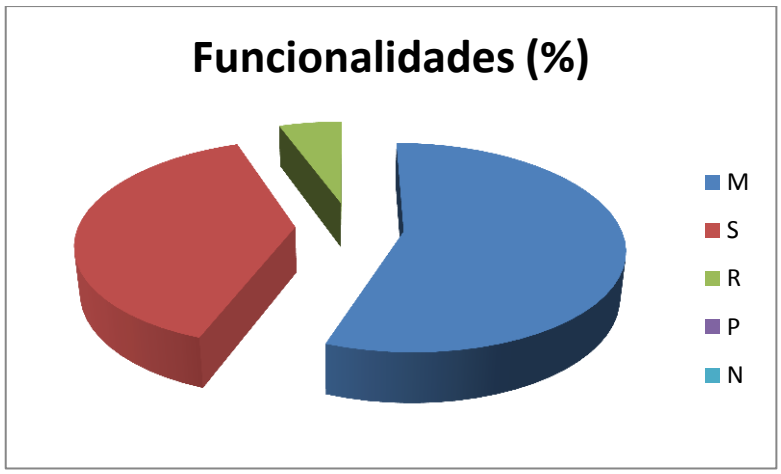

Imagen 4.7: Gráfico para el ítem 7 sobre el desempeño de ENSO

De la misma manera se averiguó el concepto de los usuarios finales sobre el rango de datos que maneja la aplicación:

Tabla 4.9: Valoración del ítem 8 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Rango de datos adecuado	12	4	2			18
	M	S	R	P	N	TOTAL
Rango de datos adecuado (%)	66,67	22,22	11,11			100

Las opiniones de los estudiantes se grafican de esta manera:

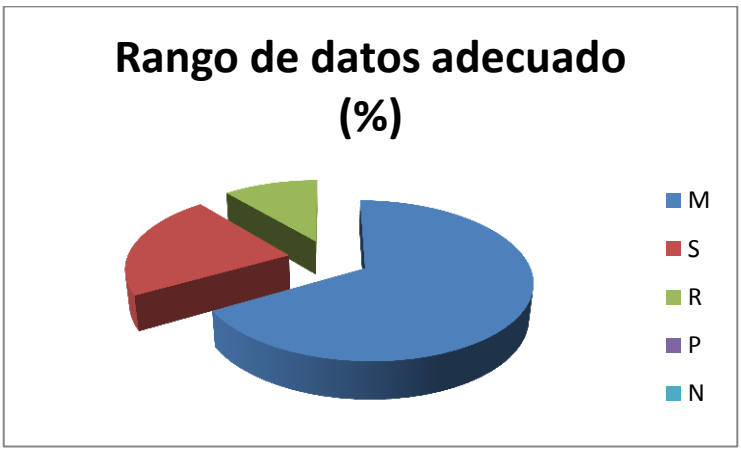

Imagen 4.8: Gráfico para el ítem 8 sobre el desempeño de ENSO

Además se conoció si los estudiantes creen que los ejercicios presentados abarcan todos los temas:

Tabla 4.10: Valoración del ítem 9 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Abarca ejercicios de todos los temas	9	7	2			18
	M	S	R	P	N	TOTAL
Abarca ejercicios de todos los temas (%)	50,00	38,89	11,11			100

A continuación se muestra el gráfico para estos resultados:

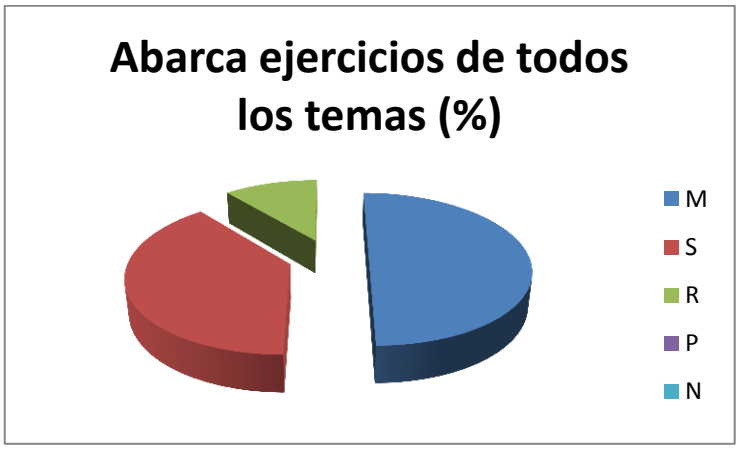

Imagen 4.9: Gráfico para el ítem 9 sobre el desempeño de ENSO

Finalmente se observó si los escolares creen que los ejercicios presentados son suficientes:

Tabla 4.11: Valoración del ítem 10 sobre el desempeño de ENSO

	M	S	R	P	N	TOTAL
Los ejercicios presentados son suficientes	10	7	1			18
	M	S	R	P	N	TOTAL
Los ejercicios presentados son suficientes (%)	55,56	38,89	5,56			100

Estos datos son reflejados en el siguiente gráfico:

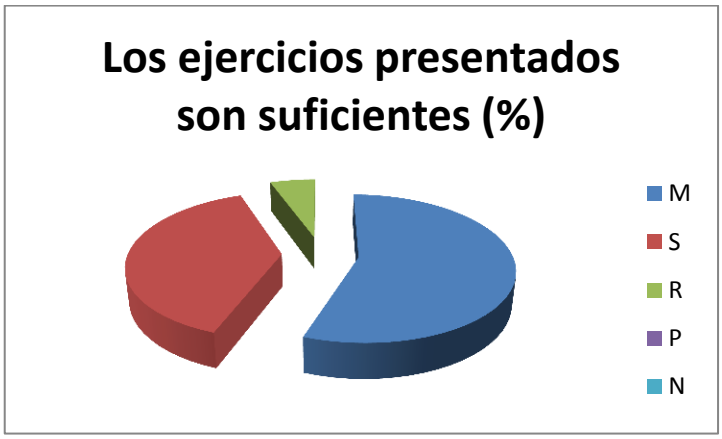

Imagen 4.10: Gráfico para el ítem 10 sobre el desempeño de ENSO

4.3. Conclusiones del capítulo

El estudio de factibilidad realizado puso de manifiesto que el software educativo desarrollado es un producto que puede ser utilizado como un medio más para apoyar el proceso de enseñanza-aprendizaje de la asignatura de SO, dado el grado de aceptación que obtuvo en la aplicación del cuestionario a los estudiantes sobre los aspectos de la aplicación.

Además es válido resaltar que la creación de medios de este tipo agrega nuevas oportunidades y estrategias de estudio al alumno que forma parte de la cantera de profesionales que se forman en el Instituto.

En general cualquier medio que le otorgue independencia y autonomía al estudiantado en su preparación es siempre deseable e incluso necesario dentro del contexto del paradigma de educación al que el mundo está entrando hace algunas décadas, en el que se enseña a aprender por sí mismo, y va quedando obsoleto poco a poco el modelo tradicional de enseñanza, donde el profesor establece los contenidos e imparte los conocimientos, y los estudiantes asimilan sin descubrir por sí mismos los axiomas fundamentales de las disciplinas y las verdades esenciales de los conocimientos de su profesión.

CONCLUSIONES GENERALES

Finalizada la investigación presentada en este documento puede asegurarse que:

- Se desarrolló una aplicación de escritorio que soluciona y muestra ejercicios de los contenidos de la asignatura de SO.
- Fue consumado un análisis sobre los conceptos fundamentales afiliados al objeto de estudio y al campo de acción, lo que mostró la relevancia de la inclusión de las TIC en la educación, como en tantas otras esferas del desarrollo de la sociedad humana.
- Se cumplió el análisis y diseño del software informatizando los procedimientos que son realizados para resolver los ejercicios de la asignatura, partiendo de una lista de características funcionales del sistema y otras no funcionales que se tuvieron en cuenta para la elaboración del programa, e igualmente las fases de exploración y planificación, como parte de la metodología de desarrollo de software utilizada.
- Fueron llevadas a cabo las pruebas de aceptación del programa, develándose la conformidad del cliente.

De manera general se concluye que ha sido alcanzado el objetivo inicialmente propuesto. Los estudiantes de la asignatura de SO cuentan con un medio más para su autopreparación, que puede ser usado para comprobar soluciones de los ejercicios que desarrollan en su estudio independiente, viéndose así favorecido el proceso de enseñanza-aprendizaje de esta materia en la carrera de Ingeniería Informática en el Instituto Superior Minero-Metalúrgico de Moa.

RECOMENDACIONES

Persiguiendo el fin de desarrollar aún más en el futuro el producto obtenido es recomendable enriquecerlo con otras características y funcionalidades que le otorguen integralidad y usabilidad como software educativo.

En primer lugar, es deseable que para cada módulo se incluyan funcionalidades que permitan explicar la teoría tratada en el tema del programa de la asignatura del que se ocupa el módulo. Este aspecto le agregará mayor valor a la aplicación y su impacto será más notable en el proceso de enseñanza-aprendizaje de la materia.

Es también conveniente extender esta experiencia a otras asignaturas que desarrollan habilidades técnicas, debido a la escasez de material educativo de esta categoría en las carreras de ingeniería en el país y porque con estos medios el proceso de enseñanza-aprendizaje se ve favorecido.

Finalmente, y tomando en cuenta los resultados del estudio de factibilidad aplicado, es también ideal realizar mejoras en la manera en que se captan los datos y se muestran los resultados, e incluso aplicar un cambio en el enfoque del uso del software para mejorar la experiencia de su uso. El objetivo es hacer más ameno, rico e instructivo el proceso de aprender y estudiar la asignatura de SO.

BIBLIOGRAFÍA

Bouza, O. (1999). La informatización, una disciplina necesaria en la educación superior cubana. Pedagogía Universitaria. Editorial Universitaria. 4: 11.

CARNOTA, O. y. V., PEDRO (1989). "Proyección de Sistemas Automatizados de Dirección" La Habana, Editorial Pueblo y Educación.

FERNÁDEZ, G. Introducción a Extreme Programming. 2002. Disponible en:

[http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo- XP.pdf].

GARCÍA, F. y. G., J (2001). "Los espacios virtuales educativos en el ámbito de Internet: Un refuerzo a la formación tradicional, Teoría de la Educación. Educación y Cultura en la Sociedad de la Información, 3. ."

GRADY, B. (2004). El proceso Unificado de Desarrollo de Software. La Habana, Editorial Félix Varela.

Hernández, A. S. (2013). Desarrollo del módulo Peso Específico de los Suelos para el Sistema de Gestión de Informes de Ensayos. Ingeniería Informática. Moa, Instituto Superior Minero Metalúrgico Moa.

Hernández, S. A. y. A. (2000). Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML. La Habana.

JACOBSON, I., RUMBAUGH, J. y BOOCH, G (2004). El proceso unificado de desarrollo de software. La Habana, Félix Varela.

Joskowicz, J. Reglas y Prácticas en eXtreme Programming.

Larman, C. (2004). UML y patrones.Introducción al análisis y diseño orientado a objeto. La Habana, Editorial Félix Varela.

LETELIER, P., EMILIO, A. S. L. Metodologías Ágiles en el Desarrollo de Software. Alicante, España. 2003. Disponible en: *[http://issi.dsic.upv.es/tallerma/actas.pdf].*

LEYVA, L. “Trabajo de diploma: Flujo de Investigación para la Metodología Ágil SXP”. Ciudad de La Habana, Universidad de las Ciencias Informáticas. 2009.

Mendoza, M. Metodologías de Desarrollo de Software.

MONTERO O’FARRILL, J.L.: “Las TIC en la Nueva Universidad Cubana”, CEP, ISMMM, Moa, Holguín, 2010.

PALACIOS, J. Flexibilidad con Scrum. 2007. Disponible en: *[http://www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf].*

PANADÉS M., PATRICIO L. Metodologías ágiles para el desarrollo de software: eXtreme Programming

(XP). Universidad Politécnica de Valencia. 2006. Disponible en: *[http://www.journaldatabase.org/articles/112000/Maetodologias_agiles_para.html]*.

PEÑALVER, G; ABEL, M. A.; SERGIO, J.P.G. SXP en el modelo “Universidad - Empresa” cubano de desarrollo de software. Ciudad de La Habana, Universidad de las Ciencias Informáticas. 2010.

PEÑALVER, G. “Trabajo de diploma: Metodología ágil para proyectos de software libre”. Ciudad de La

Habana, Universidad de las Ciencias Informáticas. Disponible en:

*[*http://10.33.2.195:3389/svn/Documentacion_General/Expediente_SXP/*].*

Peralta, M. Estimación del esfuerzo basada en casos de uso.

Pressman, R. (2002). Ingeniería del Software.Un enfoque Práctico. La Habana.CANÓS, J.; PATRICIO L.; MA CARMEN P. Metodologías Ágiles de Software en el Desarrollo. 2005. Disponible en: *[http://www.willydev.net/descargas/prev/TodoAgil.Pdf].*

ANEXOS

ANEXO 1: Programa analítico de la asignatura Sistemas Operativos en el ISMMM

Introducción general

Fundamentación

Sin el software una computadora es en esencia una masa metálica sin utilidad[1]. Con el software, una computadora puede almacenar, procesar y recuperar información, encontrar errores de ortografía e intervenir en muchas otras valiosas actividades dirigida por el hombre que lo ayuda a resolver muchos problemas de su profesión.

El software para computadoras puede clasificarse en general, en 2 clases:

- los **programas de sistema**, que controlan la operación de la computadora en sí y;
- los **programas de aplicación**, los cuales resuelven problemas para sus usuarios.

El programa fundamental de todos los programas de sistema, es el Sistema Operativo, que controla todos los recursos de la computadora y proporciona la base sobre la cual pueden escribirse los programas de aplicación.

El sistema operativo es una parte esencial dentro de un sistema informático. Administra los recursos del sistema y ofrece un entorno de programación y operación a los usuarios, permitiendo trabajar con la máquina de forma cómoda y eficaz. Es imprescindible que un ingeniero en Informática conozca los conceptos fundamentales vinculados a los sistemas operativos, sus funciones, su estructura y su implementación.

Problema

¿Cómo favorecer los modos de actuación del Ingeniero Informático relacionados con el desarrollo y explotación del software de un Sistema Informático?

Objeto

Los programas del Sistema Informático que controlan la operación de la computadora: Sistema Operativo.

Objetivo General

Caracterizar los sistemas operativos a partir del análisis de sus conceptos fundamentales, identificando el más eficiente en un ambiente de cómputo determinado e interpretar sus técnicas de programación para elaborar otros programas, con una adecuada cultura informática, atendiendo a las necesidades del país y las orientaciones principales del Partido.

Objetivos Específicos

1. Caracterizar los sistemas operativos a partir de la interpretación de sus conceptos fundamentales, identificando el más eficiente en un ambiente de cómputo, con modestia y ética profesional, reconociendo la importancia del trabajo colectivo y teniendo en cuenta las necesidades e intereses sociales.
2. Montar y explotar un nuevo sistema operativo a partir del estudio de sus manuales técnico y de usuario (en idioma inglés), utilizando con eficiencia las facilidades brindadas por los mismos y atendiendo a las necesidades del país y las orientaciones principales del Partido.
3. Interpretar técnicas avanzadas de programación inherentes a los sistemas operativos para aplicarlas a la programación de sistemas informáticos propiciando la eficiencia y la calidad estética de los resultados.

Contenido

Sistema de conocimientos:

Definición de sistema operativo. Diferentes tipos y generaciones. Estructura de un sistema operativo. Servicios brindados por un sistema operativo. Principales funciones y programas.

Concepto de proceso. Procesos concurrentes. Comunicación entre procesos. Planificación de procesos. Entrada y salida. Dispositivos de entrada\salida. Aseguramiento de programas para la entrada\salida. Concepto de bloqueo. Formas de detectar, recuperar, prevenir y evitar los bloqueos.

Administración de memoria. Sistemas de monoprogramación y de multiprogramación. Sistemas de memoria virtual.

El sistema de ficheros. La administración del espacio en los medios de almacenamiento externos. Estructura de los directorios. Seguridad y protección de los ficheros.

Sistema de habilidades

- Valorar las características de un sistema operativo.
- Manejar desde un punto de vista teórico y utilizar con eficiencia las facilidades brindadas por un sistema de operación específico.
- Dominar las características de un sistema operativo a partir del estudio de sus manuales técnicos y de usuarios.
- Configurar sistemas operativos.
- Interpretar los diferentes tipos de estructura de datos que se usan en la instrumentación de los sistemas operativos.
- Modificar algoritmos pertenecientes a sistemas operativos para adaptarlos a otros comportamientos.
- Programar algoritmos de procesos concurrentes.

Valores

- Formar en los egresados los hábitos de responsabilidad que se requerirá en la explotación e interacción de equipos de cómputo a través de los sistemas operativos.
- Estimular el hábito de la autosuperación que requiere la evolución de los sistemas operativos en su alcance.
- Estimular el desarrollo de la creatividad y un enfoque independiente en la solución de diferentes tareas.
- Contribuir a formar profesionales en los que se conjuguen la alta calificación en el campo con cualidades personales entre las que destaque la modestia y una actuación ética.

Conocimientos previos

- Programación.
- Análisis Matemático.
- Matemáticas Discretas.
- Arquitectura de Computadoras.

El curso se estructura en 6 temas y trece unidades didácticas.

Indicaciones metodológicas.

Para apropiarse de esta lógica le recomendamos comenzar por el tema 1, que aborda conceptos generales sobre los sistemas operativos. A continuación se debe estudiar el tema 2 que guarda una estrecha relación con el primero y sirve de base a los demás temas del curso.

El tema 2 trata los conceptos básicos sobre los procesos, punto neural en todo sistema operativo.

Los demás temas pueden estudiarse en cualquier orden, no obstante recomendamos el que presentamos en la estructura didáctica de la asignatura que coincide con la de su libro de texto.

Cada unidad didáctica propone un conjunto de actividades, cuya realización permitirá avanzar en la comprensión y apropiación de las habilidades previstas.

Aunque el curso se realiza en la modalidad presencial, y esta guía es un medio de apoyo, puede servir para desarrollar el curso en la modalidad semipresencial, ya que se adecua al modelo mixto de formación en Cuba.

El curso combina los métodos problémicos, trabajo colaborativo y trabajo independiente, y utiliza formas que van desde el encuentro de orientación y control, a la consulta y la tutoría, potenciando la actividad independiente del participante donde se concentra el proceso de autoaprendizaje.

El sistema de medios está compuesto por:

- Guía de estudio.
- Software.
- Texto básico: Sistemas Operativos Modernos, Tanenbaum, A.

- Vídeos ponentes.
- Otros materiales complementarios en formato digital

Se recomienda tener en biblioteca varios ejemplares de los textos básicos.

Textos Básicos

1. Tanenbaum, A. S., Sistemas Operativos Modernos, Ed. Prentice-Hall Hispanoamericana, 1996 (825).
2. Silberachts, A., Peterson, J.L. y Galvin, P. B., Sistemas Operativos. Conceptos Fundamentales, Ed. Addison-Wesley, 1994 (727).

Complementarios

1. Stallins, William, Sistemas Operativos, 2Ed. Prentice-Hall Hispanoamericana, 1997.
2. Silberachts, A., Galvin, P. B. y Gagne, G., Fundamentos de Sistemas Operativos, 7Ed. Mc Graw Hill, 2006.
3. Carretero, J., García, F., Anasagasti, P. M. y Pérez, F., Sistemas Operativos. Una visión aplicada, McGraw Hill Interamericana, 2001

Tema I Conceptos fundamentales de Sistemas Operativos

Objetivos Parciales:

Comprender el concepto de sistemas operativos atendiendo a sus funciones fundamentales, sus componentes y su estructura, para obtener una visión general que permita operar con eficiencia sus facilidades en dependencia de las necesidades e intereses sociales.

Conocimientos:

Sistema operativo funciones. Diferentes tipos y generaciones. Estructura de un sistema operativo. Servicios brindados por un sistema operativo. Principales funciones y programas. Instalar y operar un sistema operativo.

Habilidades

- Comprender los conceptos fundamentales de los sistemas operativos.
- Describir las características y la estructura de los sistemas operativos.
- Operar con las facilidades brindadas por un sistema operativo.

Tema II Gestión de procesos.

Objetivos Parciales:

Interpretar conceptos, técnicas y algoritmos más generales inherentes a la gestión de procesos en los sistemas operativos, para identificar el más eficiente en un ambiente de cómputo determinado y elaborar otros programas, con una adecuada cultura informática.

Conocimientos:

Concepto de proceso. Procesos concurrentes. Comunicación entre procesos. Planificación de procesos.

Habilidades:

- Familiarizarse con los conceptos fundamentales sobre la gestión de procesos de los sistemas operativos.
- Programación concurrente de procesos.
- Interpretar las formas de comunicación de los procesos.
- Interpretar los algoritmos de planificación de procesos como funciones fundamentales de los sistemas operativos para la administración de recursos.

Tema III Administración de memoria

Objetivos Parciales:

Interpretar conceptos, técnicas y algoritmos más generales inherentes a la gestión de memoria en los sistemas operativos, para identificar el más eficiente en un ambiente de cómputo determinado y elaborar otros programas, con una adecuada cultura informática.

Conocimientos:

Aspectos introductorios. Multiprogramación con particiones fijas. Fragmentación interna. Multiprogramación con particiones variables. Fragmentación externa. Técnicas para el registro del uso de la memoria. Memoria Virtual: Paginado, segmentado, sistemas combinados. Algoritmos de reemplazamiento de páginas.

Habilidades:

- Familiarizarse con los conceptos fundamentales de la administración de memoria de los sistemas operativos.
- Analizar las diversas técnicas de gestión de memoria incluyendo la paginación, la segmentación y la memoria virtual.
- Valorar la utilización de un sistema operativo en dependencia de sus algoritmos de gestión de memoria.

Tema IV Sistemas de Archivos.

Objetivos Parciales:

Interpretar conceptos, técnicas y algoritmos más generales inherentes a la gestión de archivos y del espacio en disco en los sistemas operativos, para identificar el más

eficiente en un ambiente de cómputo determinado y elaborar otros programas, con una adecuada cultura informática.

Conocimientos:

El sistema de ficheros. La administración del espacio en los medios de almacenamiento externos. Estructura de los directorios. Seguridad y protección de los ficheros.

Habilidades:

- Familiarizarse con las funciones principales de los sistemas operativos en la administración de archivos.
- **Interpretar los conceptos y algoritmos para la gestión de ficheros y el espacio en disco de los sistemas operativos.**

Tema V Administración de E/S

Objetivos Parciales:

Interpretar conceptos, técnicas y algoritmos más generales inherentes a la gestión de E/S en los sistemas operativos, para identificar el más eficiente en un ambiente de cómputo determinado y elaborar otros programas, con una adecuada cultura informática.

Conocimientos:

Entrada y salida. Dispositivos de entrada\salida. Aseguramiento de programas para la entrada\salida.

Habilidades:

- Familiarizarse con las funciones principales de los sistemas operativos en la administración de E/S.

Tema VI Bloqueos.

Objetivos Parciales:

Interpretar conceptos, técnicas y algoritmos más generales inherentes a los interbloqueos en los sistemas operativos para identificar el más eficiente en un ambiente de cómputo determinado y utilizarlos en la elaboración de otros programas, con una adecuada cultura informática.

Conocimientos:

Recursos. Bloqueos. Ignorar el problema (Algoritmo del Avestruz). Detección y recuperación de bloqueos. Evasión de bloqueos. Prevención de bloqueos. Bloqueos sin recursos.

Habilidades:

Interpretar los algoritmos para la prevención, detección y recuperación de los bloqueos.

Sistema de Evaluación

El sistema de evaluación del curso se concibe de manera procesal, en la que los estudiantes, en consenso con su profesor, seleccionarán las actividades que decidan realizar de las propuestas en cada unidad didáctica. Todas ellas pueden aportar elementos para conformar la evaluación final.

El examen final escrito complementa el sistema de evaluación, que junto con las evaluaciones parciales y el desempeño integral de los estudiantes en el curso conformarán la evaluación final.

Tema	Actividad frecuente	Actividad Parcial	Actividad Final
Tema 1	Laboratorio 1; Seminario 1	TCC #1	Examen
Tema II	Clase práctica 1-6; Laboratorio 2		
Tema III	Clase práctica 7 y 8; Laboratorio 3	TCC #2	
Tema IV	Clase práctica 9		
Tema V	Clase práctica 10		
Tema IV	Clase Práctica 11		

1. 1Tanenbaum, A. S., Sistemas Operativos Modernos, Ed. Prentice-Hall Hispanoamericana, 1996 (825).

ANEXO 2: Cuestionario para los estudiantes de la asignatura Sistemas Operativos

Instituto Superior Minero-Metalúrgico de Moa

"Dr. Antonio Núñez Jiménez"

Cuestionario para investigación sobre el proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos en la carrera de informática

Estimado alumno:

La finalidad de este cuestionario es recabar información para evaluar la factibilidad del Software Educativo para el apoyo a la enseñanza de la asignatura de Sistemas Operativos.

Tu ayuda es necesaria para alcanzar los objetivos de la investigación, por ello solicitamos tu colaboración.

Tus respuestas tendrán carácter anónimo y confidencial, y serán usadas únicamente para analizar los aspectos significativos del estudio. Te exhortamos a ofrecer respuestas sinceras.

Instrucciones:

1. Lee cuidadosamente cada enunciado antes de responder cada planteamiento.
2. Marca con una X la casilla que crees que se ajusta mejor a su respuesta.
3. Llena todas las tablas presentadas.
4. Consulta con el encuestador en caso de dudas.

Significado de las siglas:

M	S	R	P	N
Mucho	Suficiente	Regular	Poco	Nada

Gracias por tu colaboración.

Atentamente:

Carlos Ernesto Velázquez Rodríguez

Ficha de evaluación del Software Educativo				
M	S	R	P	N
Mucho	Suficiente	Regular	Poco	Nada

ASPECTOS DE LA APLICACIÓN					
	M	S	R	P	N
Fácil de usar					
Formato de entrada de datos es intuitiva					
Formato de salida de resultados es intuitiva					
Entiendo qué datos pide					
Entiendo qué resultados ofrece					
Rápida					
Funcionalidades					
Rango de datos adecuado					
Abarca ejercicios de todos los temas					
Los ejercicios presentados son suficientes					

Si deseas expresar alguna otra idea, puedes hacerlo aquí: